Technische Universität Kaiserslautern

Fachbereich Informatik
Computergrafik und HCI

Scalable Human-Centered Decision Making Processes in Virtual Environments

Vom Fachbereich Informatik der
Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Ingeneurswissenschaften (Dr.-Ing.)
genehmigte Dissertation

von

Franca-Alexandra Rupprecht

Datum der wissenschaftlichen Aussprache:
22. März 2018

Prüfungskommission/Berichterstatter:
Prof. Dr. Jens Schmitt, TU Kaiserslautern (Vorsitz)
Prof. Dr. Achim Ebert, TU Kaiserslautern
Prof. Dr. Bernd Hamann, UC Davis
Prof. Dr. Jan Aurich, TU Kaiserslautern

Dekan:
Prof. Dr. Stefan Deßloch

D386

"The product should suit the user, rather than making the user suit the product."

Kathy Baxter & Catherine Courage

Franca-Alexandra Rupprecht
Scalable Human-Centered Decision
Making Processes in Virtual Environments

Dissertation, 22 March, 2018
University of Kaiserslautern
Computer Graphics and HCI
Department of Computer Science
Postfach 3049
67653 - Kaiserslautern, Germany

Abstract

Collaboration aims to increase the efficiency of problem solving and decision making by bringing diverse areas of expertise together, i.e., teams of experts from various disciplines, all necessary to come up with acceptable concepts. This dissertation is concerned with the design of highly efficient computer-supported collaborative work involving active participation of user groups with diverse expertise. Three main contributions can be highlighted: (1) the definition and design of a framework facilitating collaborative decision making; (2) the deployment and evaluation of more natural and intuitive interaction and visualization techniques in order to support multiple decision makers in virtual reality environments; and (3) the integration of novel techniques into a single proof-of-concept system.

Decision making processes are time-consuming, typically involving several iterations of different options before a generally acceptable solution is obtained. Although, collaboration is an often-applied method, the execution of collaborative sessions is often inefficient, does not involve all participants, and decisions are often finalized without the agreement of all participants. An increasing number of computer-supported cooperative work systems (CSCW) facilitate collaborative work by providing shared viewpoints and tools to solve joint tasks. However, most of these software systems are designed from a feature-oriented perspective, rather than a human-centered perspective and without the consideration of user groups with diverse experience and joint goals instead of joint tasks. The aim of this dissertation is to bring insights to the following research question: How can computer-supported cooperative work be designed to be more efficient? This question opens up more specific questions like: How can collaborative work be designed to be more efficient? How can all participants be involved in the collaboration process? And how can interaction interfaces that support collaborative work be designed to be more efficient? As such, this dissertation makes contributions in:

1. **Definition and design of a framework facilitating decision making and collaborative work.**
   Based on examinations of collaborative work and decision making processes requirements of a collaboration framework are assorted and formulated. Following, an approach to define and rate software/frameworks is introduced. This approach is used to translate the assorted requirements into a software's architecture design. Next, an approach to evaluate alternatives based on Multi Criteria Decision Making (MCDM) and Multi Attribute Utility Theory (MAUT) is presented. Two case studies demonstrate the usability of this approach for (1) benchmarking between systems and evaluates the value of the desired collaboration framework, and (2) ranking a set of alternatives resulting from a decision-making process incorporating the points of view of multiple stakeholders.

2. **Deployment and evaluation of natural and intuitive interaction and visualization techniques in order to support multiple diverse decision makers.**

A user taxonomy of industrial corporations serves to create a petri network of users in order to identify dependencies and information flows between each other. An explicit characterization and design of task models was developed to define interfaces and further components of the collaboration framework. In order to involve and support user groups with diverse experiences, smart devices and virtual reality are used within the presented collaboration framework. Natural and intuitive interaction techniques as well as advanced visualizations of user centered views of the collaboratively processed data are developed in order to support and increase the efficiency of decision making processes. The smartwatch as one of the latest technologies of smart devices, offers new possibilities of interaction techniques. A multi-modal interaction interface is provided, realized with smartwatch and smartphone in full immersive environments, including touch-input, in-air gestures, and speech.

3. **Integration of novel techniques into a single proof-of-concept system.** Finally, all findings and designed components are combined into the new collaboration framework called IN^{2}CO, for distributed or co-located participants to efficiently collaborate using diverse mobile devices. In a prototypical implementation, all described components are integrated and evaluated.

Examples where next-generation network-enabled collaborative environments, connected by visual and mobile interaction devices, can have significant impact are: design and simulation of automobiles and aircrafts; urban planning and simulation of urban infrastructure; or the design of complex and large buildings, including efficiency- and cost-optimized manufacturing buildings as task in factory planning. To demonstrate the functionality and usability of the framework, case studies referring to factory planning are demonstrated. Considering that factory planning is a process that involves the interaction of multiple aspects as well as the participation of experts from different domains (i.e., mechanical engineering, electrical engineering, computer engineering, ergonomics, material science, and even more), this application is suitable to demonstrate the utilization and usability of the collaboration framework.

The various software modules and the integrated system resulting from the research will all be subjected to evaluations. Thus, collaborative decision making for co-located and distributed participants is enhanced by the use of natural and intuitive multi-modal interaction interfaces and techniques.

Zusammenfassung

Eine effiziente Lösung von Problemen setzt heutzutage nicht nur die Expertise eines Einzelnen voraus, sondern die Kollaboration von Experten über mehrere Disziplinen hinweg. Zur Erreichung eines Konsenses in solch heterogenen Teams wird dem Prozess der Entscheidungsfindung eine zentrale Rolle zuteil. Diese Dissertation befasst sich daher mit dem Design effizienter rechnergestützter Kollaborationstechniken, die eine aktive Teilnahme von Benutzern verschiedenster Disziplinen ermöglicht. Dabei sind drei Beiträge besonders hervorzuheben: (1) die Definition und das Design einer Umgebung, die kooperative Entscheidungen erst ermöglicht; (2) die Anwendung und Evaluation von immersiven und intuitiven Interaktions- und Visualisierungstechniken zur Entscheidungsunterstützung in virtuellen Umgebungen; und (3) die Integration der neuen Techniken in ein übergreifendes Proof-of-Concept System.

Entscheidungsprozesse sind typischerweise sehr zeitintensiv und erfordern mehrere Iterationen mit unterschiedlichen Annahmen, bevor sich auf eine Lösung geeinigt werden kann. Kollaborationstechniken sollen diese Probleme lösen und werden daher bereits frequentiert angewandt. Allerdings ist die Durchführung nur allzu oft ineffizient, Teilnehmer werden nicht ausreichend integriert und Entscheidungen werden ohne vollständige Übereinstimmung aller Teilnehmer getroffen. Eine Vielzahl rechnergestützter kooperativer Arbeitssysteme (CSCW) ermöglichen den Teilnehmern bereits ihre Perspektive zu teilen und stellen Werkzeuge zur gemeinsamen Lösung von Aufgaben bereit. Die heterogene Zusammenstellung der Teilnehmer legt einen benutzerorientierten Entwurf der Systeme nahe. Allerdings standen bei der Entwicklung solcher Systeme bisher häufiger die Probleme und deren gemeinsame Lösung und nicht der Mensch und sein Kommunikationsbedarf im Fokus, wodurch sich die zentrale Forschungsfrage dieser Dissertation ergibt: Wie kann man effizienterer CSCW entwickeln? Antworten auf diese Frage liefern die folgenden Forschungsbeiträge dieser Dissertation:

1. **Definition und Gestaltung eines Frameworks zur Unterstützung von Team-basierter Entscheidungsfindung.**
   Basierend auf Untersuchungen von Kollaborationen und Team-basierten Entscheidungsprozessen werden Anforderungen eines Frameworks zur Unterstützung dieser beiden Arbeitsmethoden systematisiert und formuliert. Anschließend wird eine Methode basierend auf der Erweiterung des House-of-Quality Ansatzes vorgestellt, die zweierlei Problemstellungen löst. Zum einen, wird einen Ansatz zur Definition und Bewertung von Software Komponenten vorgestellt. Dieses erweiterte House-of-Quality Verfahren ermöglicht die verschiedenen Anforderungen an ein solches System in dessen Architekturdesign zu überführen. Ein weiteres Verfahren innerhalb der Erweiterung des House-of-Quality Ansatzes wird zur Bewertung von Lösungs-Alternativen basierend

auf multikriterielle Entscheidungsanalyse und multiattributrieller Nutzentheorie eingesetzt. Zwei Fallstudien demonstrieren die Anwendbarkeit und Einsatzmöglichkeiten dieses Ansatzes für (1) das Benchmarking von Systemen unter Einbeziehung des entstehenden Nutzens und (2) die Aufstellung einer Rangfolge von Systemalternativen, unter der Berücksichtigung verschiedener Interessensgruppen.

2. **Bereitstellung und Evaluierung von natürlich und intuitiv gestalteten Interaktions- und Visualisierungstechniken zur Unterstützung diverser Kollaborateure in virtuellen Umgebungen.**
   Eine Taxonomie von Personengruppen dient dazu, ein Personen-Netzwerk zu erstellen. Abhängigkeiten und Informationsflüsse untereinander werden dadurch identifizierbar. Basierend auf diesen Erkenntnissen dienen explizite Definition und Ausgestaltung von Aufgabenmodellen (task models) dazu, Schnittstellen und Systemeigenschaften des Kollaborations-Frameworks zu erstellen. Durch den Einsatz von Smart Devices und virtueller Realität wird die aktive Mitarbeit von Benutzergruppen unterschiedlichster Erfahrungen und Fachbereiche unterstützt und gefördert. Immersive und intuitive Interaktionstechniken sowie innovative Visualisierungen wurden in der der Dissertation entwickelt, um die Effizienz von Entscheid-ungsprozessen zu erhöhen. Die Smartwatch als eine der neuesten Klasse von Smart Devices bietet innovative Möglichkeiten zur Interaktion. Ein multimodales Interaktions-Interface wird vorgestellt, welches die Interaktion mit Hilfe von Smartwatches und Smartphones in immersiven Umgebung erlaubt. Verschiedenste Eingabemöglichkeiten, einschließlich Touch-Eingaben, Armgesten und Sprachsteuerung, werden unterstützt.

3. **Integration der neuen Techniken in ein übergreifendes Proof-of-Concept-System.**
   Alle Erkenntnisse und Ergebnisse aus den zuvor beschriebenen Untersuchungen werden in das neue Kollaborationsframework mit dem Namen IN²CO integriert. Dieses wird für die Realisierung effizienter Kollaborationen von räumlich verteilten als auch räumlich konzentrierten Kollaborateuren eingesetzt. In einer prototypischen Implementierung werden alle beschriebenen Komponenten integriert und evaluiert.

Beispiele, bei denen innovative netzwerkfähige kollaborative Umgebungen erhebliche Auswirkungen haben können, sind unter anderem: Entwurf und Simulation von Automobilen und Flugzeugen, Stadtplanung und Simulation der städtischen Infrastruktur sowie das Design von komplexen und großen Gebäuden, einschließlich effizienz- und kosten optimierten Fertigungsgebäuden als Gestaltungsziel in der Fabrikplanung. Um den Nutzen und die Benutzerfreundlichkeit des Frameworks zu demonstrieren, werden Fallstudien zur Fabrikplanung demonstriert. Fabrikplanung zeichnet sich durch das Zusammenspiel vielfältiger Disziplinen und Teilnehmern aus, die häufig auch räumlich getrennt sind. Folglich, eignet sich dieses Szenario um den Nutzen und die Vorteile des Kollaborations-Frameworks zu evaluieren.

Die aus dieser Forschung entstandenen Softwaremodule sowie das integrierte Gesamtsystem wurden Nutzerstudien unterzogen. Es zeigte sich, dass durch die Verwendung von immersiven und intuitiven multimodalen Interaktionsschnittstellen und -techniken sowie individuell angepassten Visualisierungen die Effizienz von kollaborativen Entscheidungsfindungen für räumlich konzentrierte sowie verteilte Kollaborateure gesteigert werden kann.

Acknowledgements

This Dissertation would not have been completed without the constant encouragement and support of my supervisors, Prof. Dr. Achim Ebert, Prof. Dr. Bernd Hamann, and Prof. Dr. Jan Aurich. Their guidance during my doctoral studies has been invaluable for me. They constantly stimulated my creativity and kept up my motivation.

I am thankful for having the opportunity to be part of the international research training group IRTG2057. My supervisors provided me with an excellent working atmosphere at the University of Kaiserslautern as well as the University of California Davis. I appreciate the possibility to visit international conferences and enhancing my inter-cultural competences due to several research stays in Davis, California.

I would like to thank Prof. Dr. Heike Leitte for giving me the opportunity to work in her group and for being so patient with me during the time of writing this Dissertation.

I would also like to express my deep appreciation to Prof. Dr. Hans Hagen, who made me an important and responsible-minded member of his group. He has given me guidance and advice on a personal and professional level that has been instrumental in my growth as a research student preparing to be a professional. Prof. Hagen supported me to surpass myself and increased my strengths by giving me new opportunities in leadership and constantly meeting new challenges.

I will give a special thanks to Dr. Jens Bauer, who was the first person I was able to work with in the group of computer graphics and HCI. During my studies in industrial engineering, he introduced me into the human computer interaction community and supported me in the decision to start my doctoral studies in computer science. The final decision for starting my doctoral studies in the group of computer graphics and HCI was made due to the encouraging and open words from Prof. Christoph Garth who stilled my fears of changing the field of studies - Thank You!

I would also like to express my appreciation to my collaborators Prof. Dr. Gerrit van der Veer and Prof. Dr. Bahram Ravani, who detected my strengths early on and stimulated me to higher achievements by steering me into the right directions.

I would like to thank all my colleagues at the Computer Graphics and HCI group as well my IRTG 2057 colleges for the rewarding discussions, intellectual exchanges, and heartily encouragements concerning this dissertation as well as personal belongings, especially Diana Fernandez, Kathrin Häb, Francois Torner, Georg Kassakow, Johannes Schwank, and Andreas Schneider. Special thanks to Dr. Taimur Khan and Sebastian Schöffel for the editorial support of this Dissertation. Additionally, I would like to express my gratitude to Mady Gruys and Roger Daneker for their assistance in the official aspects of my studies as well as all their kind words.

The completion of this dissertation would have been much more difficult without all of the friends that stood by my side and offered their unconditional support. Most importantly, I would like to wholeheartedly thank my parents, Gerhard and Carmen Rupprecht, my sister Jennifer and her daughter Selin, as well as my loved Grandma Ingrid Wolf, and certainly Falco for their trust, understanding, and endless support. None of this would have been possible without the strength that they have given me through their confidence and love.

Contents

Chapter 1

Introduction

In developed economies, changes in the labour markets have been established from agrarian over industrial production to information economies. Also, the requirements of skills for many jobs have been changed [105]. On the one hand work environments are technology-rich and on the other hand problems are frequently ill-defined, hence, often people in multidisciplinary teams have to come together to deal with the new kind of problems [116]. Thus, the ability to learn, collaborate, and solve problems in a digital information environment is described as 21st century skills and has become crucial for the society in present-day research and labour markets. According to Griffin et al. [116] there exists a shift in the way humans learn, the way humans think, and the way humans work, encouraged due to increased emphasis on omnipresent technology that accelerates the need for these new skills. Furthermore, they stated that capabilities to analyze credibility and utility of information, evaluate information appropriateness and intelligent application of learned information is aroused due to the accelerating need to access and process information in the workplace. Rapid advances in computer networks and visualization-based human- computer interaction technologies are promising to impact a large spectrum of graphics-based design-simulations. For instance, conceptual design and critical assessment of complex systems generally requires large teams of scientists, engineers and planners to work together. The conceptual design process is extremely time-consuming, typically involving several iterations of different options before a generally acceptable solution is obtained.

Collaboration, performed by a large group of experts from diverse fields and competences, is a time-demanding and complex process. Collaboration among stakeholders is an increasingly used practice for design, evaluation, and concept balancing. This practice is used to discuss information and problems from different perspectives, to re-construct and co-construct knowledge or to solve problems [66] enabling decision-making processes with a rich environment allowing different points of view and a variety of competencies. Reflecting and integrating all ideas and expertise is crucial. Large superordinate goals become clearer when people have worked together in collaboration [182]. Although, collaboration is an often-applied method, the execution of collaborative sessions is often inefficient, does not involve all participants, and decisions are often finalized without the agreement of all participants.

According to Becker [28], there is only little value of the advances of omnipresent new technologies in economies with few skilled workers who are supposed to use those advanced technologies. Thus, the economic growth depends on a synergy between new knowledge and human capital. The rapid advances in technology however, must consider the evolution and adaption of needed skills of humans is not evolved at the same pace. Thus, centering the human into the focus of the design of those new technologies is crucial. Most collaboration systems focus on single tasks and are designed from a feature oriented perspective rather than a human-centered perspective. The quality and acceptance of a product is endangered, if product designers focus merely on the feature-oriented approach rather than incorporating the end-user during the complete design and development process. A big difference between these approaches is that user-centered design focus on the optimization of a product around the end users' aspects (how users can, want, or need to use the product), instead of trying to "suit" users to the product by changing their behavior to adapt to the product [85]. Users have individual needs implying different ways of utilization and benefiting from the technology [235].

User-centered design (UCD) follows the philosophy, that the product should suit the user, rather than making the user suit the product [77]. It is a design methodology and philosophy in which the needs, goals, and success of the end user are considered by matching the user's mental model with the system usability design [174]. Therefore, analysis of the user's typical tasks and identification of different groups of users based on their needs is a key aspect and a prerequisite for designing intuitive and highly usable technology. The first UCD principle stated by Gould and Lewis [114] fastened their attention on the systematic and structured collection of users' requirements by setting the focus in the earliest stage on users and their tasks. In order to cover this principle, Courage and Baxter [77] suggested the following structured actions:

1. **Learn about the product.** Before requirements of the users can be gathered it is important to understand the domain in order to link the often roughly formulated user needs into practical system requirements. Knowledge about the domain background facilitates the designer to sort and structure the collected requirements efficiently and helps to discard unnecessary wishes in an early stage.

2. **Learn about the users.** The quality and the acceptance of the product is compromised if the customers and potential users are not well understood. In order to develop a quality product, the most critical activity is to understand who are the users and what do they need in order to perform their tasks.

3. **Putting it all together.** The knowledge base of the product, the users, and the tasks provides a solid foundation for designing intuitive systems with high usability. The learned and gathered information can be finally linked together concerning the actual desired tasks, usage, and implementation.

Ideally, in a distributed and collaborative networked environment, experts can work independently or jointly on sub-systems of an overall design to be achieved. However, the adoption of existing and available device and network technologies that support this type of work is still in its early stages. We introduce a framework enabling a team, in a distributed setting, to collaborate via computer networks using various mobile interface and visualization devices. The framework makes possible the effective and synergistic combination of team members' complementary competencies and expertise.

1.1 Aims of the Thesis

Common collaboration technologies are mostly addressing work of distributed teams. There exist a wide range of tools undertaking mind mapping, file sharing, messaging, and so on. Those tools are mainly developed for single desktop applications. Co-located collaboration is often performed by one presenter and several spectators, whereby active participation is strongly limited. Our research focus is on an environmental setup for co-located and distributed collaborative work. Due to the large size and high resolution, large display devices (LDDs) enable the reproduction of large datasets in one view. However, most LDD's interaction capabilities are designed for single users, hence powerful and intuitive visualization and interaction capabilities are needed to support a larger number of users. Complementary, smart devices offer a wide range of interaction metaphors, leading to natural and intuitive interaction. In addition, they come with a display that can be used as secondary output device. Complex data often comprises several levels on which different activity emphases exist (e.g., machine energy consumption or production rate). Those emphases can have interdependencies that must be identified and collaboratively solved. Changing attributes in one level might have an unaware or undesirable impact in another level of the same data. With the number of participants, the requirements for the visualization tool and techniques accumulate. Combining different core-competences and supporting intuitive data exploration for, and between different activity emphases, is still a challenging task. The desire for a common framework to support decision-making of complex tasks is the main motivation of this dissertation. It is crucial to provide tools to facilitate the identification and manipulation of interdependencies through diverse interests, as well as the active collaboration process.

The presented collaborative framework is aimed at pointing out the efficiency gained when bringing diverse areas of expertise together, i.e., teams of experts from various disciplines, all necessary to come up with acceptable concepts. Examples where next-generation network-enabled collaborative environments, connected by visual and mobile interaction devices, can have significant impact are: design and simulation of automobiles and aircrafts; urban planning and simulation of urban infrastructure (e.g., transportation, electricity, water and communication grids); or design of complex and large buildings, including efficiency- and cost-optimized manufacturing buildings as discipline in factory planning, which is used as main use-cases in this work. Factory planning is characterized by the parallel consideration of multiple aspects such as production resources, production process and technology, and products while anticipating uncertainty and future developments over the factory life-cycle. These aspects usually result in different partial-models with specific information content (e.g., layout model, material flow model) and components of the factory (e.g., building, machinery, foundation, media), which need to be analyzed in combination. The conceptual design and simulation-based evaluation of next generation manufacturing disciplines requires to incorporate experts from differing fields. For example, factory planning requires to bring together experts from mechanical engineering, electrical engineering, computer engineering, ergonomics, material science, and even more fields.

In order to involve all participants in an active and intuitive way, mobile smart devices are used within the presented collaboration framework. Smart devices are almost ubiquitous today and feature a large collection of input and output capabilities like touch screens, cameras, accelerometer, microphones, speakers, near- field communication, Wi-Fi, etc. The usage of smart-devices is easy and intuitive, and they offer a wide range of interaction metaphors, which can lead to a more natural and intuitive interaction as well as a broad array of control elements. Based on smart devices, we design and develop intuitive and natural interaction capabilities in a collaborative framework. The interaction with large VR systems is observed as well as the collaboration is more efficient.

1.2 Research Questions and Goals

This dissertation makes contributions in 1. the design and architecture of a collaboration framework based on multi criteria decision making; 2. the deployment and evaluation of more natural and intuitive interaction and visualization techniques in order to support multiple decision makers; and 3. the integration of novel techniques into a single proof-of-concept system.

The topics that are being investigated in this dissertation focus on supporting the collaboration process of multidisciplinary teams through a user-centered approach. One of the major problems of collaborative systems is grounded on the complexity of interpersonal interaction and the absence of joint collaboration by active participation. While most collaboration systems fasten their attention on the functionality for multiple users, less attention is payed on the development or improvement of interaction and visualization techniques to satisfy the requirements of a truly collaborative system.

The main goal of this dissertation is the exploration and development of novel natural and intuitive interaction- and visualization techniques through a user-centered approach, combined in a collaboration framework which enhances collaborative decision making for co-located and distributed participants.

The problems and proposed improvements tackled in this work are stated as below, and they follow the suggested activity structure by Courage and Baxter [77]:

A **Learn about the Product**

(a) **Criteria to support active collaboration**: The complexity of collaborative work and interpersonal interaction needs to observed, and requirements to support this kind of working style are to be identified, sorted, and structured. The objective is to establish and provide a set of qualitative criteria that apply to generic collaboration environments that can substantially and holistically advance the productivity of collaborations.

(b) **System requirements for computer supported collaborative work**: Defining software modules of a software project is a multi-layered process of close co-operation between customer, designer, and developer. Unfortunately, there is a gab between the requirements the customer defines and requirements the developer needs. Traditional techniques to translate customer needs into technical requirements are the use of system requirement documents and system specification documents. An alternative approach to define software modules and their value deriving from user needs, has been using the House of Quality method originally used in end-customer product development is presented.

(c) **Evaluating, benchmarking, and ranking collaboration solutions**: Quantitative ranking between alternatives and the selection of the most promising one for a given task is highly challenging. Typically, multiple criteria need to be granted and weighed based on the task and the user requirements. Many of the existing techniques for evaluating software are performed for experimental use and ranking based on usability aspects. The problem becomes more complicated with an increasing number of requirements and stakeholders, which is common in collaborative software. Hence, there is a need for a systematic method of dealing with this Multi-Criteria Decision Making (MCDM) problem. A systematic approach based on integration of the House of Quality (HoQ) method and the Multi-Attribute Utility Theory (MAUT) is developed. It is shown that combining HoQ and MAUT allows for the integration of task and user requirements. This method is used to perform comparisons at the subcategory level, which results neither in an overestimation nor an underestimation and consequently provides more accurate and predictable results.

B **Learn about the User**

(a) **User synergies and inferdependencies**: Dependency graphs of the participants can help to improve processes, to plan tasks, and to identify potential for more efficient cooperation. Such a dependency graph comprises clear defined entities, which are linked with each other based on defined relationships [40] in order to describe their inferdependency. Whereby, inferdependency refers to the combination of influence and dependence between two elements, one- or bi-directional, which will be defined in the second chapter. In the course of this dissertation, a taxonomy of users in industrial corporations will be introduced, which is needed to define the entities and relationships of the dependency graph and is easily adaptable to specific corporations. Such a taxonomy cannot be found in the literature, but is important for the design and development of software products under the principles of user centered design [174]. However, there is still the big challenge to display a meaningful relation between those entities and to give an easy understandable overview of the whole relationship with the goal to solve complex tasks and to improve a groups' performance. Therefore, a set of parameters will be introduced, which help to find out how good tasks and work packages are distributed within the network.

(b) **Providing opportunity of actions by inducing awareness of inferdependencies:** A framework for distributed or co-located teams to collaborate highly efficiently using diverse mobile devices for design and assessment of complex systems is introduced. The framework enhances the efficiency of collaborations arising in design, simulation, or data analysis, including visualization. The devices provide three views of data to be processed collaboratively: (1) a simulation view; (2) a status report view; and (3) a status update view. These views serve the purpose of providing overview, detail, and performance views. A smart watch view shows at-a-glance summary information and the environment or the process being inspected (possibly influenced by a user). Users can use their mobile devices as control interfaces. The framework is especially effective for combining the synergistic, complementary competencies of a team.

(c) **Enhancing intuitive and more natural interaction:** Despite the fact of their increasing popularity, virtual environments still lack useful and natural interaction techniques. The presented research was driven by the goal to enable users to invoke actions with their body movements, causing the correct action of the VR environment. The work introduces a system that tracks a user's movements that are recognized as specific gestures. Smartwatches are promising devices enabling new modes of interaction. They can support natural, hands-free interaction. We present a multi-modal interaction interface, designed for smartwatches and smartphones for fully immersive environments. Our approach enhances the efficiency of interaction in virtual worlds in a natural and intuitive way. Methods for handling seven gestures are designed and implemented, which are furthermore compared with common VR input technology, namely body tracking using a 3D camera.

(d) **Transforming signal processing data into arm gestures:** The presented effort is concerned with the replacement of common touch input gestures with body movement gestures. Missing or insufficiently precise sensor data poses a challenge in data processing, e.g., gyroscope and magnetometer data. This data is needed, together with acceleration data, to compute orientation and motion of the device. A transformation of recorded smartwatch data to arm movement gestures is introduced, involving data smoothing and gesture state machines.

(e) **Enhancing multi-modal interaction with speech:** Traditionally, the most common way to interact with large display devices has been through the keyboard/mouse interaction model. However, more recently, there has been an increasing adoption of natural communication means such as speech, touch, or gesture (captured using sensors) for interacting with computers. Large display devices usually do not come with the needed sensors for tracking natural interaction; hence, users have to purchase dedicated devices (embedded with these sensors), in order to make use of these modern interaction means, which are more often than not, useless outside the reason for which they were purchased. Therefore, a common device, specifically a smartwatch is extended and integrated into a multi-modal interaction interface.

C **Combine all of this**
First, theoretical foundations of the product and methods to design a collaborative framework are established. Second, practical examinations have been performed enhancing and improving the natural and intuitive interaction in such a framework independently. Finally, all this investigations are combined in a proof-of-concept implementation called IN^{2}CO, a human-centric visualization framework for intuitive and collaborative data exploration and manipulation. Specifically, it's contribution is the integration of ubiquitous technologies and existing techniques to explore data and dependencies in collaborative decision-making for co-located and distributed participants. A challenging task in designing such a collaborative framework is to support the active participation of each user as well as the design of the underlying architecture and infrastructure.

1.3 Scientific Outcomes of this Thesis

This dissertation has the following scientific outcomes:

- A thorough study of collaborative work, decision making, and computer supported cooperative work. This leads to the definition of a comprehensive and structured requirement and criteria catalog.
- The enhanced House of Quality is a comprehensive approach that facilitates software developers to design and evaluate collaboration software and setups. The integrated task taxonomy enables to define software attributes based on user defined requirements in a well-structured way. Incorporated dependencies and provided user needs are used to calculate the importance of each attribute, which are needed to plan existing resources and capacities in an efficient way.
- The combination of the House of Quality methodology with multi attribute utility theory overcomes well-known problems of multi criteria decision making. The approach results in a more accurate evaluation of alternative collaboration solutions and transforms qualitative ratings into quantitative measurable and comparable results. As such, the design and development process of collaborative software/ frameworks is performed in a well-structured way, while incorporating all requirements and dependencies of known and unknown user requirements.
- Following, novel interaction- and visualization techniques are developed to fulfill these requirements. A collaboration framework that enhances the efficiency of collaborative work by encouraging active participation and thorough decision making has been developed. It combines large virtual reality environments with several smart devices that serve as input and output capabilities. The setup provides three different views of the data in order to supply users with overview, detail, and performance interviews.
- Several natural and intuitive interaction techniques, that enrich the set of input capabilities and encourage the active participation of all users in the collaborative session have been integrated. Highly intuitive and effective gesture-based interaction is provided through non-touch gestures, namely body movement and speech recognition. The framework covers nearly all stated requirements of an efficient collaborative framework that supports thorough decision making of diverse participants.
- Comprehensive decision making in Virtual Reality is improved by encouraging active participation and incorporating dependencies and influences in a highly scalable and flexible framework.

1.4 Overview and Structure

The remaining parts of this dissertation are structured as follows. In Chapter 2, the background to this dissertation is given. The main topics of this dissertation: (1) collaboration software, (2) virtual reality, and (3) smart devices, are introduced. After giving this technical overview, the field of collaborative work and decision making is examined. A comprehensive study of collaborative work and decision making is provided. Based on the findings, the conceptual framework of collaborative work is presented. The chapter is completed with deduced requirements for efficient collaboration and decision making.

Chapter 3 discuss the adaption of the stated requirements of collaborative computer-aided decision making. Design objectives for collaborative computer-aided decision making are given. A quality insurance method, called eHoQ is introduced that is applicable for designing, benchmarking, and evaluating such frameworks/software. The introduced method is applied and a framework architecture ontology is derived as result.

The aim of Chapter 4 is the deployment and evaluation of novel natural and intuitive interaction and visualization techniques in order to support multiple diverse decision makers in collaborative design and assessment in virtual reality. A network of user synergies and inferdependencies is employed in order to identify potential to improve processes, to plan tasks, and to increase the efficiency of the cooperation. As a result, dedicated user and task models, serving for the design of a collaborative design and assessment system, are created according to the perceived insights. Following, the prototypical realization of such a system is deployed, integrating an interface supporting multiple diverse decision makers. This prototype is stepwise enhanced with multi-modal interaction capabilities centering the user into the investigations.

Chapter 5 combines all findings and results by examining the resulting proof-of-concept implementation into one single system that makes collaborative computer aided decision making and design in virtual reality possible. First, the framework's concept will be introduced. Followed by detailed description of technical realization and integration, as well as supporting functionalities and roles. An evaluation of the complete prototype based on multi attribute decision making and user studies concludes this chapter.

In Chapter 6, a brief summary of all findings and results is presented. The Dissertation is completed with a short conclusion motivating the adaptability and applicability of this work in the industrial context.

Chapter 2

Background

In this chapter, the background to this dissertation is given. The technical background of this dissertation, namely (1) collaboration software, (2) virtual reality, and (3) smart devices, is introduced. After giving this technical overview, the field of collaborative work and decision making is examined. A comprehensive study of collaborative work and decision making is provided. Based on the findings, the conceptual framework of collaborative work is presented. The chapter is completed with deduced requirements for efficient collaboration and decision making.

2.1 Technical Background

The collaborative framework, developed during this work, is aimed at pointing out the efficiency gained when bringing diverse areas of expertise together. Our approach base on the idea of bringing together: (1) high immersive technology, providing comprehensive insights into the data, (2) ubiquitous technology like smart devices, enabling a wide range of interaction methods that are scalable with the number of users, and (3) humans, actively participating in collaboration processes. The technical foundations of this Dissertation are composed of the fields of collaboration software, virtual reality and smart devices, as described in the following.

2.1.1 Collaboration Software

Most existing software systems only support single-user interaction, which is not suited for the exchange of ideas and competencies between people. Many software systems provide add-ons in order to share, combine, and track changes on artifacts, like the one in Microsoft Office Word [200]. However, collaborative software or groupware allows asynchronous group activities to be carried out. Asynchronous operations allow to simultaneously work on and modify the same objects and see each other's modifications in real time. This means that a group of people can work on a common task, no matter if they are in a shared environment or physically apart. Groupware can be used in many contexts and different domain fields. The scientific Computer Supported Cooperative Work (CSCW) community has explored use cases such as outline and graph editing [97], mind mapping [204], in the medical field [35], orthography systems [67], software engineering [181], landscape planning [315] and in industrial contexts [229, 219].

Real-time groupware systems (equated with collaboration systems) have the following characteristics according to Ellis et al. [96]: highly interactive, distributed (users may be in different physical locations), volatile (participants can join and leave), ad hoc (participant actions do not follow a planned script), and focused (participants work on the same data, causing a high number of access conflicts). As a result, groupware poses unique challenges such as maintaining consistency of the shared documents between all collaborators, while also offering a short response time when propagating the actions of a user to all other users, and concurrent editing [274]. Groupware is different compared to other multi-user software such as database systems that try to give the impression that there is only one user by means of locks and transactions. Groupware wants to achieve the opposite impression and requires concurrency control to resolve conflicts between users.

According to Ellis et al. [97] two dimensions are used to describe systems: (1) common task dimension and (2) shared environment dimension. The first dimension describes the degree of how closely related the tasks of individual users are. The second dimension describes how close the participants are physically (co-located or distributed). For groupware systems, the first dimension is high as all users work towards a common goal. The shared environment dimension can alternate between a high value for co-located scenarios or a low value for distributed group work depending on the usage scenario. Probably, the most popular and well-known groupware systems these days are office suites like Google Docs[112] or collaborative text editors like Etherpad [279] or Overleaf [310], all designed for distributed collaboration. Co-located collaboration still lacks appropriate support systems.

Co-located collaborative problem solving on shared workspaces was already elucidated in 1992 by Elrod et al. [98] but also in Xerox PARC's Colab [224] and DynaWall in i-Land [272]. This research provided general observations on interaction techniques with large displays and the corresponding effectiveness of team activities in such environments. Common factors that influence the adoption and use of large display groupware is identified by Huang et al. [136]. Critical factors are the ability for casual ad-hoc use, informal communication and awareness, as well as synchronous use. Most co-located collaboration systems lack the synchronous control capacity of the users hindering the active participation of all users. For instance, in field studies of the system BlueBoard, at most one person was active manipulating the scene, whereby others often stepped back to form an audience and had to take turns quite often in order to bring in their own ideas [242]. Co-located synchronous collaboration around shared displays as examined in [143] suggests that multi-touch wall displays can support different collaboration styles and fluid transitions in group work while enabling control capabilities for each user.

2.1.2 Virtual Reality

Virtual Reality (VR) is a comprehensive information technology that provides a realistic interactive 3D simulated environment [160]. It is used to simulate imaginary or realistic worlds, displayed on immersive 3D output devices [282] while delivering stimulative information through various sensations such as vision, hearing, touch, and smell [306]. Depending on the designers' wishes, imaginations, or needs, 3D virtual worlds can be created and manipulated without any qualification and are not limited to any boundaries [282]. Virtual Reality visual interaction environments make possible the sensation of being physically present in a non-physical world [226]. The value of this experience is a better perception and comprehension of complex data based on simulation and visualization from a near-real-world perspective [52].

Users experience and observe the scene "from inside" are able to concentrate their attention on the task exclusively [169]. A user's sense of immersion, the perception of being physically present in a non-physical world, increases when the used devices are efficient, intuitive, and as "natural" as possible. Direct and natural interaction with special purpose equipment is provided to obtain the maximum freedom for controlling and managing the whole environment by the users [306].

Virtual worlds naturally enable more complex interaction, encourage the learning experience, and facilitate users' empowerments [69, 81]. Through increased interactivity, more constructive understandings are attained [89]. With the use of virtual reality technology better cohesion and cooperation among users could be reported [117]. Virtual reality technologies are increasingly applied and practiced in situations that assumes a need of experience beyond the realistic feasible [133, 261] or to enhance productivity, quality, or safety aspects. Exemplary domains are teaching and assistance [231, 117], stress relief and meditation [207, 282], gaming and serious gaming [64, 276], medicine [306, 313], and increasingly in the industrial context [79, 316, 291, 144, 263].

Motivation for the innovative use of VR technology is founded in the fact that disputes of practical realization are easily overcome and potential benefits are premature identifiable without the common challenges: high costs, implementation risks (including risk of injury or death), inflexibility to adapt alternate scenarios, and difficulty to replicate [121]. Especially in industrial contexts, those challenges can be immense and can lead to the failure of the corporation. A commonly used concept in the industrial context is Virtual Commissioning (VC), which is described as the visualization, programming, and validation of a production system in a virtual environment. Those tasks are elaborated as planning steps before the actual construction or reconstruction of new manufacturing plants [126, 176]. Virtual reality technology applied for validation and prototyping enables a user to step into the virtual production system and experience it as if it already exists. Thus, users can observe the production system from inside and also actively interact with it [79]. Errors or possible problems in the planned system (from facility layout over material flow to programmable logic controller code) are discovered early in the development stage [176, 126, 263] and a safe way of testing the integration of new technology and software is enabled with an unlimited number of prototype alternatives while excluding the risk of physical destruction [23, 88]. Thanks to virtual prototyping and corresponding simulation results, the need for costly mockups is eliminated and engineering analysis becomes more efficient [263].

2.1.3 Smart Devices

Smart devices (such as smartphones, tablet PCs, or data-glasses) are wireless, mobile (hand held) electronic devices, equipped with a range of in-built sensors, for the personal use. In comparison with cellphones, which stated the beginning of mobile phones, smartphones provide much more purpose than merely sending and receiving voice and short message communications. Smart devices are connected to other devices or networks with the use of network technologies and communication protocols like Bluetooth, NFC, WiFi, 3G, etc. [302].

The most obvious characteristics of smart devices, as used in this dissertation, are the relatively big screens covering the front side of the devices. These touch displays serve as output device giving visual feedback to the user and simultaneously serve as input capacity, with which the device is controlled. The touch input technology revolutionized the interaction capabilities and enhanced interaction techniques with multi touch, force touch, and touch gestures (also single and multi-touch). Smart devices nowadays are equipped with customized software and Internet access making them extremely powerful and ubiquitous. They enhance people's everyday life in nearly each situation while superseding a range of other common electronic devices, like digital cameras, calculators, portable music players, e-book reader, and even non electronic tools like levels. The application fields of smart devices are innumerable as they are used as little personal computers in nearly each and every situation.

Smart devices are provided with a great variety of sensors that can provide useful information that is used to increase the usability of the devices, and for customization of the installed applications. According to Alepsis et al. [4] the following sensors can be in-built in smart devices nowadays:

- Wifi
- Bluetooth
- GPS
- Proximity Sensor
- Orientation Sensor
- Magnetic Field Sensor
- Ambient Light Sensor
- Tilt Sensor
- Accelerometer Sensor
- Gesture Sensor
- Gyroscope
- Geomagnetic Sensor
- Temperature/Humidity Sensor
- Barometer
- Hall Sensor
- RGB Light Sensor
- Fingerprint Sensor
- Moisture Sensor
- Pressure Sensor
- Health Tracking Sensors
- Thermometer Sensor
- Hygrometer Sensor

Due to the high amount of input and output capabilities, smartphones are more versatile than most other standardized input devices. General input metaphors and techniques are already familiar to the user, making it easy and intuitive to use them as a remote-control device. The high versatility of smart devices allows new interaction metaphors and applicability in new and diverse contexts. Smart devices are almost ubiquitous today. In a world population of 7.47 billion people, 4.93 billion smartphones are registered in year 2018 [269]. It can be assumed, that nearly everyone has a smart device. Thus, the personal device can be used to connect with other systems according to the Bring-Your-Own-Device (BYOD) policy. Assuming that the system, that is controlled by smart devices allows the connection with multiple devices, the BYOD policy enables high scalability of the number of users and participants. Furthermore, such devices can be used as mobile data storage, the identification of users is unambiguous and personalized. As such, customized interaction experience for each user is achievable.

Smart devices can have different appearance and sizes. The smart devices considered in this dissertation, mainly smartphones, tablets, and smartwatches ranging from really small displays (272 px x 340 px) up to relatively big displays (9,7"). It is implied, that the visualization and interaction techniques need to be designed in a fashion that they provide a uniform way of interaction for the range of different applications and sizes. On the one hand, the applications must be responsive by adapting sizes and positions of User Interface (UI) elements with regard to the display size. On the other hand, it has to be considered, which kinds of interaction techniques and metaphors fit best for the underlying system. As such, not each interaction- or visualization technique fits for all kinds of smart devices. But, the exploration of new techniques and application fields makes smart devices extraordinarily powerful and versatile devices. Thus, smart devices are as cutting edge technology even more indispensable.

2.2 Collaborative Work and Decision Making

According to the first design principle of User Centered Design, this section discusses the concept of collaborative work and decision making in order to examine the domain and to *learn about the product*. It is necessary to understand the domain from early on in order to link the often roughly formulated user needs into practical system requirements. Knowledge about the domain background facilitates the designer to sort and structure the collected requirements efficiently and helps to discard unnecessary wishes in an early stage.

First, the task of collaborative work will be analyzed before deducing a conceptual framework integrating activities and structuring of collaborative tasks. Afterwards, an overview and classification of collaboration styles will be given and the concept of inferdependencies concerning task dependencies in complex activities is introduced. While observing the domain and background, requirements will be identified and pointed out. In conclusion, general requirements for efficient collaboration and decision making are deployed, elucidated, and transferred into six capacities of collaborative work.

2.2.1 Task Analysis

Collaboration is the combination and exchange of different core competencies and expertise with the goal of creating a joint outcome in agreement, considering ideas and objectives of all participants. In order to examine the task of collaboration work we follow a structured ontology that is used to describe the task world rather than just the task.

According to van der Veer and Van Welie [297], it is necessary to include descriptions of many more aspects of the task world rather than just the task themselves in order to design groupware systems. Therefore, they presented a framework intended to structure task models. Task models for complex situations are composed of three different aspects: agent, work, and situation. These three aspects are further decomposed into the five main focal points as described in [295]:

- **Agents:** personified instances that perform tasks;
- **Roles:** an agent acts in a certain role to perform role activity;
- **Activities:** sub tasks to reach the overall goal;
- **Objects:** artifacts that will be shared between agents;
- **Events:** trigger relevant changes in the state of the task.

Accordingly, each focal point describes the task world from a different viewpoint and relates to each other via relationships. For the design of the task supporting tools, designers can read and design from different angles assuring consistency and completeness. An overview of the focal points and their relations is sketched in Figure 2.1:

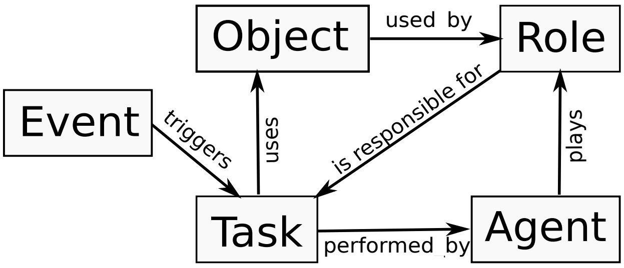


FIGURE 2.1: Ontology of task world models in accordance with [295].

With respect to the availability of information and communication technology, the collaboration between people changed strongly into the direction of computer supported cooperative work. This implies that computers and other technologies as additional actors in collaborative processes. Hereby, roles get exchanged more easily between actors, and activities get more easily delegated to systems [296].

To cover the design of collaborative task models and a supporting system, the following questions have to be answered and defined:

- Which activities have to be performed?
- Who is performing those activities?
- What objects are required?
- How can the information be represented to the user groups?
- How can the interaction with the systems be enabled?
- What kind of dependencies exist between activities/roles/agents?

In the course of this dissertation, those questions will be answered and and application examples will be presented.

2.2.2 Conceptual Framework of Collaborative Work

Based on a thorough literature review, the main task phases involved in collaborative processes are identified as: (1) assigning tasks and roles, (2) drafting, (3) discussion, (4) task execution, (5) reviewing, and (6) task establishment.

The main task phases and incorporating feedback-loops are depicted in Figure 2.2. Collaborative work starts with the assignment of tasks and roles to each actor, which is facilitated by a software tool or performed in a group meeting setup beforehand (phase 1). On the one hand, assignment of tasks and roles is necessary to coordinate the work and team member in order to manage dependencies between activities [190]. Members without a task or a role can simply not participate in the group work. Passive or even no participation of actors lowers the individual's satisfaction that is linked with motivation [41], engagement [51], and self-perceptions [132].

On the other hand, task assignment might lead to workload equality, which increases individual's satisfaction as stated in [55]. This individual satisfaction may influence the performance of the complete team, which has been investigated in [180] but also the individuals' willingness to continue the cooperation, what could be observed in [275]. Therefore, the first requirement on a collaboration framework is that members can coordinate their activities.

Requirement 1

Participants can coordinate activities.

In the second phase, users create drafts of the desired goals and the approach for reaching these goals (phase 2). This task can be performed individually or in joint sessions. In this phase, the frame of the joint work is created and the assigned tasks are refined.

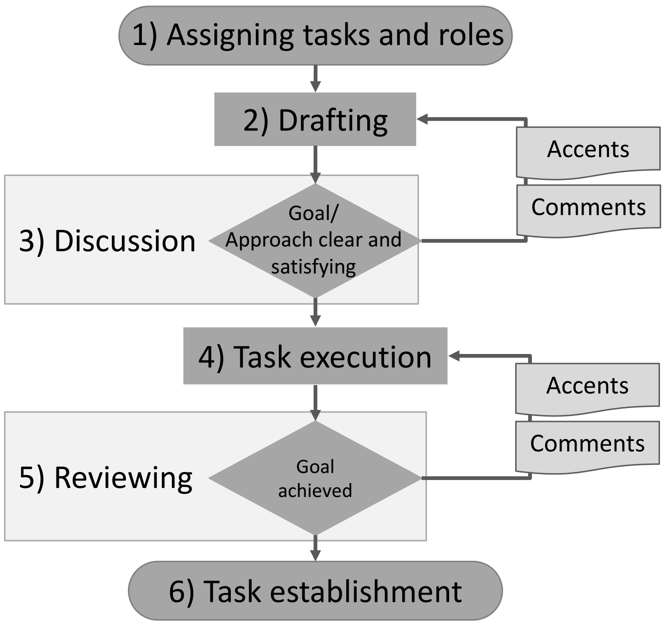


Figure 2.2: Working phases in generalized task model of collaborative working.

The second phase is accompanied by continuous comments and feedback loops, which lead into group discussions, in phase 3. The outcome of this phase is a draft in which ideas and expertise from all participants are considered and integrated. Communication between members across phases is crucial in order to synchronize the approaches to guarantee the progress towards the agreed joint goal. Thus, communication between members is required in order to support collaborative work.

Requirement 2

Participants can communicate with each other.

Subsequently, the working style changes from group task to single task performance where the actual execution of the assigned tasks is performed (phase 4). The actual execution of the assigned tasks requires a setup providing the ability to a member to actively perform an activity..

Requirement 3

Participants can perform activities.

Phase 4 is closely linked with the iterative reviewing and revision task (phase 5) in which all ideas, comments, and suggestions are discussed and incorporated. The exchange between members is necessary to communicate individual ideas and the satisfaction with the performance state. The decision about accepting the output is the outcome of phase 5 that leads to the final stage of the process, the task establishment (phase 6).

2.2.3 Collaboration Styles

The collaboration process description from above indicates that collaborative work entails different working styles and phases. Among working styles, a distinction is found based on the attribute *division of labor* into: (1) group task performance and (2) individual's task performance.

Group tasks require the participation of all team members. Individuals' tasks, however, describe that team members individually taking on responsibilities for focusing their own task goals to establish the joint overall team goal. Individuals' tasks are performed simultaneous and independent to other members but the integration and collection of the results needs to meet the requirements of all participants. Thus, integration and collection is performed in close-coupled cooperation. Members most likely switch between the different working styles, which is necessary to make one's contribution towards the individual's progress but also to control and adjust the group task performance.

Requirement 4

Participants can switch between collaboration styles.

Considering the wide flexibilities due to new technology and the tighter working schedules in the working environments, nowadays, an increasing number of encounters and discussions are performed with the use of networked systems. In regards to computer support of teamwork, additional attributes like the participants' physical location need to be considered. Following, collaboration can be performed (1) co-located; indicating all participants in the same physical location or (2) distributed; indicating the spatial separation of the participants.

Thus, collaborative task performance can be defined upon the attributes *participants physical location* and *division of labor of tasks*. Figure 2.3 gives an overview of the four distinguishable collaboration styles.

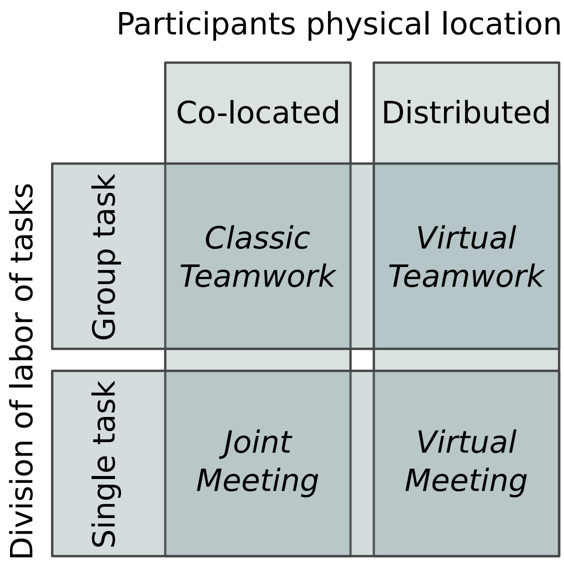


FIGURE 2.3: Collaboration styles classified upon the attributes participants physical location and division of labor of tasks.

Thus, it can be distinguished among four types of collaborative working styles between which actors switch most likely during:

1. **Classic Teamwork:** Co-located group task performance. Solving tasks in a team at the same physical location.
2. **Virtual Teamwork:** Distributed group task performance. Solving tasks in a team at spatial separated locations connected via virtual elements.
3. **Joint Meeting:** Co-located single task performance. Solving tasks individually, meet physically to integrate and collect requirements.
4. **Virtual Meeting:** Distributed single task performance. Solving tasks individually, meet virtually to integrate and collect requirements

Interpersonal interaction has a big influence on the overall results of the collaborative work. A high individuals' satisfaction in the group leads to a higher motivation individually and in the group, which furthermore leads to better group performance [107].

The individuals' satisfaction is improved by preventing frustration in interpersonal interaction as well as interaction with technology. Technology should be designed to be intuitive, simple, and should improve the transferability of facilitation skills [50]. Rules and guidelines facilitates a good group climate and supports the work within the group. In a nutshell, participants should have fun using the technology.

Requirement 5

Participants are satisfied with/in group work.

Concerning interpersonal interaction, the individuals' expectations should be accomplished in order to raise the satisfaction [153].

Requirement 6

Participants expectations are accomplished.

Rules and guidelines facilitates a good group climate and supports the work within the group.

2.2.4 Inferdependencies of Task Dependencies

Changes made by the independently and simultaneously operating team-members have impact on the overall system and tasks being performed. Johnson et al. defined interdependence in the context of joint activity as follows: *"Interdependence" describes the set of complementary relationships that two or more parties rely on to manage required (hard) or opportunistic (soft) dependencies in joint activity* [150]. The view of interdependence, developed during the work of this dissertation, generalizes their definition. The existence of one task is not necessarily dependent on the existence or completeness of another task; there is no necessary "relies-on" relationship, but there may be a "can-be-positively-or-negatively-influenced-by" relationship. The performance of one task can influence another task. Tasks can be interdependent through dependencies, but not as a consequence of merely existing. Dependencies and influences in only one direction can exist, which is not equivalent to interdependence describing a bi-directional dependence. Therefore, we use the term **inferdependency**.

Definition

Inferdependency: The combination of influence and dependence between two elements, one- or bi-directional.

The notion of inferdependency is explained via a simple example: Considering the scenario of biocenosis, where organisms coexist in the same habitat and interactions are evident in food or feeding relationships. In this scenario four actors are identified: (1) a flower, (2) a butterfly, (3) a bee, and (4) a bear. A butterfly depends on a flower for nectar (food); the flower depends on the butterfly to pollinate and make seeds for reproduction. A direct interdependence between the butterfly and the flower exists. In addition to the butterfly, a bee coexists in the same habitat. The bee depends on the flower to produce honey; the flower depends on the bee to cross-pollinate. A direct interdependence between the bee and the flower exists. Bee and butterfly coexist without influencing each other. However, in reality both species influence each other by cross-pollinating flowers, thereby accelerating reproduction. The task performance of both species makes their jobs (pollinating flowers) easier and leads to an overall improved outcome (higher reproduction of flowers), which is also the precondition (food) for task performance.

Consider a third participant, e.g., a bear eating honey (produced by the bees). We find a direct relation between bees and bears and indirect relation between butterflies and bears. Thus, influences and (indirect) dependencies, implying inferdependencies, between all participants exist, see Figure 2.4.

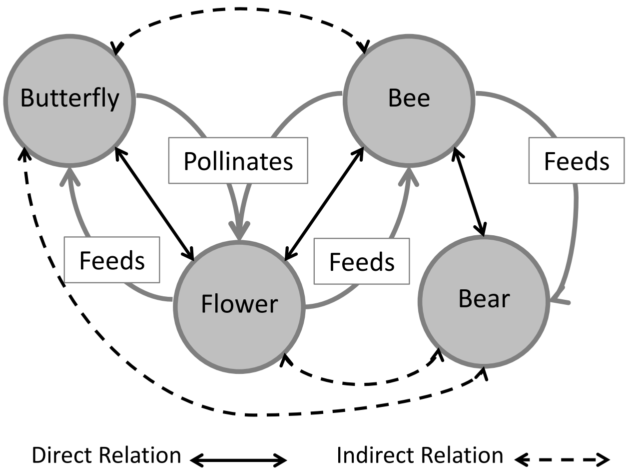


FIGURE 2.4: Inferdependencies showing direct and indirect relations between participants of a system.

Inferdependent activities imply the presence of conflicting interests, which have to be coordinated to capture discrepancies before they become serious in order to achieve common goals with the help of common grounds [166]. Common ground is supported by continually informing others about changes that have occurred outside their views [166]. The determination of other's activities is therefore crucial.

Requirement 7

Participants can determine others' activities.

Johnson et al. [149] stated that not all team members must be fully aware of the entire scope of an activity; but all must be aware of the interdependence in-between their activities. Awareness of tasks and activities influences the coordination and task performance in a positive manner. Due to the establishment of shared knowledge and impact awareness, team members can work together effectively and adjust their activities as necessary [225].

Requirement 8

Participants can understand the impact of changes made.

Requirement 9

Participants can make adjustments based on impact.

In interactive teamwork, data and information exchange between experts has to be performed and inferdependencies need to be connected.

Requirement 10

Participants can exchange information.

2.2.5 Requirements for Efficient Collaboration and Decision Making

An intermediate objective of this Dissertation is to establish the determining factors for a work environment that allows participants of a collaborative session to perform individual tasks as well as group work tasks in a natural manner including the consideration of inferdependencies. Based on the assumptions of collaborative work state above, the requirements for efficient collaboration and decision making are summarized as follows:

R1 Participants can coordinate activities.

R2 Participants can communicate with each other.

R3 Participants can perform activities.

R4 Participants can switch between collaboration styles.

R5 Participants are satisfied with/in group work.

R6 Participants expectations are accomplished.

R7 Participants can determine others' activities.

R8 Participants can understand the impact of changes made.

R9 Participants can make adjustments based on impact.

R10 Participants can exchange information.

Based on this requirements, six capacities of collaborative work are identified as shown in Figure 2.5 and explained below.

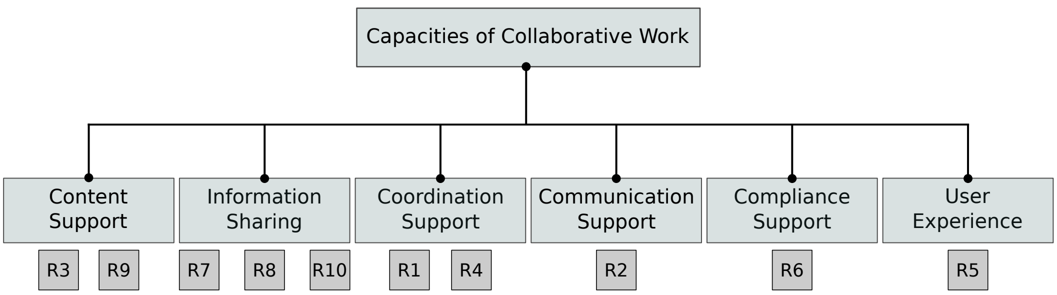


FIGURE 2.5: The six capacities of efficient collaboration and decision making.

Content Support refers to the active interaction and integration of the content by the actors, which highly relies on the underlying task. More detailed, this capacity refers to any kind of functionality realizing creating, editing, or removing content into the shared working space. The requirements R3 and R9 are covered in this category.

Information Sharing involves functionalities and technical modules, which are used to share information. This capacity facilitates the establishment of a common knowledge basis for all actors. The requirements R7, R8, and R10 find consideration in this capacity.

Coordination Support refers to the spectrum of tasks that are used to support coordination. The coordination of work packages as well as the coordination between actors is considered. The requirements R1 and R4 find application in this category.

Communication Support is described by the support of advanced and unimpeded communication among actors. Especially, within distributed collaboration, functionality within this category is used to bridge spatial gaps. Requirement R2 is covered within this capacity.

Compliance Support relates to rules or guidelines that should be fulfilled to conduct a thorough decision making process. The good group climate should be sustained and individuals' notions and opinions should be incorporated. Requirement R6 is applicable in this category.

User Experience describes the actor's emotions and attitudes about using the technology. Frustration or distraction through the technology should be strongly avoided, so that the participant is able to focus on the task and not the interaction with the technology. High satisfaction using the technology, however, can facilitate the collaboration process. Requirement R5 finds consideration in this capacity.

Summarizing, in this chapter the main topics that frame the context of this dissertation have been introduced at a glance. The domain of collaborative work, as underlying concept of computer supported collaborative work, is represented and examined in detail. Based on the exploration of this task, assumptions and requirements for efficient collaboration and decision making are deployed. In the course of this dissertation, the requirements will be stepwise complied.

Parts of this chapter have been previously published in:

F. Rupprecht, T. Khan, G. van der Veer, A. Ebert, "Criteria Catalogue for Collaborative Environments", *31st British Human Computer Interaction: Digital Make Believe 2017*, Sunderland, UK, 3-7 July, 2017.

F. Rupprecht, G. Kasakow, J. Aurich, B. Hamann, A. Ebert, "Improving Collaboration Efficiency via Diverse Networked Mobile Devices", *Journal of Multimodal User Interfaces*, Springer, 2017.

Chapter 3

Collaborative Computer-Aided Decision Making

In Chapter 2 the domain of collaborative work was inspected and assumptions as well as requirements for efficient collaboration and decision making were compiled. Thus, Chapter 2 frames the foundation for the further investigations. Knowledge about *the product*, a system that enables and supports collaborative computer-aided decision making, will be enhanced and refined in this chapter.

Collaboration among stakeholders is an increasingly used practice for design, evaluation, and concept balancing. This practice enables decision-making processes within a rich environment combining different points of view and a variety of competencies. According to [203], most collaboration tools are developed for the mass market and do not suffice the specific needs of a collaborating group. Hence, these needs must be satisfied by the use of customized tools that apply user-centered design methodologies as mentioned by Schümmer et al. [253]. Evaluating and classifying collaboration tools is an important aspect in CSCW and groupware research [147, 118, 318] but also for designing services, where the interaction between service provider and customer is in focus. The definition of collaboration frameworks and service design supporting tools, subserves to support communication, collaboration, and coordination [97]. The increasing integration of functionality in those frameworks is challenging considering that asynchronous and synchronous collaboration needs to be supported.

The objective of this chapter is the provision of a quality assurance tool, called *the enhanced House of Quality* (eHoQ), applied with the purpose of establishing quality into a software-product as well as to evaluate final products considering existing user needs comprehensively. In this chapter *the enhanced House of Quality* is presented in order to illuminate collaborative computer-aided decision making. Based on collected user needs, this new approach defines a quality assurance tool used for two cases of application: (1) definition and rating of software components and (2) benchmarking and ranking of software/frameworks.

Implementation of Computer Integrated Design and Production (CIDP) requires development of software modules to simulate the product and the process functions. It also requires integration of computers in all aspects of design, planning and project delivery. The result is a multi-layered process requiring close co-operation between all stakeholders and communication between all product and process modules. Unfortunately, there exist gaps between the requirements defined by different stakeholders making collaborative work challenging, which describes the first case of application feasible with eHoQ. The second application case concerns quantitative ranking between alternatives and the selection of the most promising one for a given task, which is highly challenging and formulates another research question. Typically, multiple criteria need to be incorporated and weighted based on the task and the user requirements. Many of the existing techniques for evaluating systems and applications are performed for experimental use and ranking based on usability aspects. The problem becomes more complex with the increasing number of requirements and stakeholders, which are now common in feature-rich control systems. Hence, there is a need for a systematic method of dealing with this Multi-Criteria Decision Making (MCDM) problem. We develop a systematic approach based on the integration of the House of Quality (HoQ) method and the Multi-Attribute Utility Theory (MAUT). It is shown that combining HoQ and MAUT allows for the integration of task and different user requirements. The new method is used to perform comparisons at the subcategory level, which results neither in an overestimation nor an underestimation of the criterion's influence and consequently provides more accurate and predictable results. The methods are applied to the evaluation of three different collaborative software environments in order to show their suitability for a typical collaboration task.

The reminder of this chapter is as follows: first, the background of software quality assurance and multi criteria decision making is provided. Afterwards, the methodology for collaborative computer aided decision making with the use of *the enhanced house of quality* is presented. Based on the predefined requirements for efficient collaboration and decision making a comprehensive set of qualitative criteria is established. This criteria catalog is applicable to generic collaboration environments that can substantially and holistically advance the productivity of collaborations. The criteria catalog will be used within the eHoQ methodology to identify software architecture modules of new systems. The catalog will serve as a guideline for evaluating existing collaboration systems. Following, two cases in which the eHoQ method can be applied are described in detail. As a result of applying the eHoQ in these cases, the design scheme for a collaborative computer-aided decision making system and the design evaluation based on multi criteria decision making is presented.

3.1 Related Aspects and Work for Collaborative Computer-Aided Decision Making

In this section, the related aspects of this chapter are introduced. The following subjects are explored: (1) Software quality assurance, (2) Quality Function Deployment, (3) House of Quality as a Function Deployment tool, (4) Multi Criteria Decision Making, and (5) Multi Criteria Decision Making methods.

3.1.1 Software Quality Assurance

Software quality can be defined as the degree to which a system, component, or process meets the expectations and needs of the user [320]. The most significant area in software requirements engineering is customer responsiveness. If the software does not meet the needs of a user, it has no value to them. Implying reliability, safety, security, and maintainability of the software are irrelevant to the users, if the software does not do what they asked for. Negotiating on software requirements between customer and developer bear the challenge to agree on the same terminology and to come to terms between user needs and technical requirements. Often misinterpretations of the actual requirements arise and the outcome can be something totally different than expected.

The traditional method to document the expectations of the users and the corresponding system requirements is system requirement documents and system specification documents, also called performance specification and master software development plan according to [76], which are defined under the IEEE Standard 830-1998 [139]. Functional requirements of a software system are typically collected with the use of interview techniques with users or experts. The System Requirement Specification (SRS) should help software customers to accurately describe what they wish to obtain; and software suppliers to understand exactly what the customer wants.

The software requirements document comprises of all the requirements formulated by the customer specifying terms and conditions. It describes the specification of the end user's viewpoint including boundary conditions. The software requirement document is formulated by the customer, or on whose behalf, and serves as contractual base.

To approve end-user requirements, prototyping is often used as a common method [154]. But this method requires the development of partially working components or functions of the software, which requires time and money and leads to an incremental refinement of the specifications. Traditional methods to improve the quality of a product or process are Statistical Process Control (SPC) and the Define, Measure, Analyze, Improve, Control (DMAIC) problem-solving approach of Six Sigma [223]. These common used techniques efficiently reduce "negative quality" such as defects, problems, and variability. However, similar to prototyping, the requirement to use these methods is that a product or process is already produced or released [320].

Zultner et. al criticized that traditional approaches and life-cycle methods lack a rigorous formal way to obtain customer needs that must be translated into system requirements [321]. He furthermore stated that most available analysis approaches are not appropriate enough as they concentrate on requirement completeness rather than requirement sufficiency. The latter is necessary to concentrate the available efforts and resources on the most important requirements implying a higher chance of satisfying the users. It is intended to deliver a sufficient level of performance on a sufficient number of high-value requirements. Therefore, it is necessary to weight and rank the users' needs. Users and developers have to agree on some kind of ranking or prioritization of requirements/software modules to define the scope of the software project or rate values of the modules for bench marking. Relationships between customer requirements are also considerable but often ignored. The degree of importance or priority of any particular requirement has to be determined and impacts on each other addressed [21]. Rating methods like sensitivity analysis, gap analysis, analysis of the cost of repair [75], and the weighted scoring method could be applied like in [141, 175, 73, 228], however, it would still lack the required granularity of the task analysis and support only one aspect of quality assurance.

Similar statements of the problem are investigated by Liu [185], who introduced a Fisheries Library in R (FLR) model defining a systematic approach to identify the functional relationships between the customer requirements and engineering characteristics. A decision support system facilitating the user to analyze and specify collaboration tools is represented by the Wheel of Collaboration Tools [308]. Collaboration interface, Collaboration Functions, Content Management, and Process Integration are considered within a typology based analysis. However, weightings are not applied and precise modules are not defined.

3.1.2 Quality Function Deployment

A technique commonly used to "listen to the voice of the customer" in order to understand customers' definition of value is the Quality Function Deployment (QFD). This technique, originally designed for industrial design, is used in this research.

Similar investigations to the one presented in the following sections were performed at the QFD Institute (QFDI) [232], where the Blitz QFD approach, a QFD method, was designed to be a better, faster, and cheaper than the traditional QFD approach. The Blitz QFD process has nine steps and no matrices which overtakes the task of gathering, defining, and prioritizing user needs which ends in a house of quality in the last step. Whereas, our work is more concentrated on the translation process of user needs into design attributes. The QFD approach is actually a quite used technique in software engineering and has been applied for over two decades [323]. Software QFD has been pioneered by Yoshizawa [314] and other leading software quality experts in Japan in 1982 and since then, they are widely used in Japan, e.g. in [211, 78, 314]. Since 1988 Software QFD is applied in the united states [322], by well-known firms such as AT&T Bell Laboratories [281], DEC [72, 278], Hewlett-Packard [34, 257], and IBM [259]. QFD is applied in different fields in computer science like for example, of information systems [120], expert systems [215], human-machine interface [216], and more. But there is a lack of a practical method that guide developers to specify, evaluate, and acquire weighted modules to support the design of software or collaboration frameworks in particular.

Similar to the investigation in this chapter, Sarkis and Liles [244] combined the QFD approach with the IDEF0 functional-modeling technique that is designed to model the decisions, actions, and activities of an organization or system using graphical notations, demonstrated on a decision support system. The IDEF0 approach starts with the identification of main functionalities, which will be decomposed and converted into separate parts-deployment matrices, one for each major activity leading to a high overload of actions.

3.1.3 House of Quality as Quality Function Deployment Tool

The House of Quality method is one tool of QFD and has its origin in industrial design to define the requirements of a specific product. It is a product development technique widely used and developed in Japan by Mitsubishi in 1972 [277] and adapted in the United States in 1988 by Hauser et al [122]. Quality has to be designed from the beginning rather than designing or producing first and correcting afterwards. Therefore, it is necessary to consider quality from the perspective of the user rather than only focusing on the technical perspective of the designer or developer [2]. Originally proposed QFD's aim is to collect and analyze the voice of the customer in order to develop products with higher quality to meet or surpass customers' needs while focusing on understanding what value means to the users that are tend to satisfied. The function was primary needed in product development, quality management, and customer needs analysis [62].

Traditional quality-improvement and problem-solving methods, as mentioned earlier, focus on finding the root cause of defects or problems and removing them correspondingly. Therefore, a product that already has been developed can later be tested and enhanced. These techniques optimize a released product and correct/improve it. QFD furthermore drives design and development by incorporating the users' statements and actions as needs in the first pass through development to ensure user satisfaction from the beginning [324]. Satisfaction of a user requires the presence of value, as determined by the customer rather than the absence of defects. Next to discovering the user's voice and gathering corresponding requirements, QFD is incorporating priorities of those requirements from the view of the user.

In the software development process, there are often not enough resources or time to realize all requirements of the users in an appropriate quality. To be more efficient, knowing what are the requirements with the highest values is necessary. In the realization, more effort can be put on those high-value requirements rather than wasting the resources for less but demanding low-value requirements. Based on limited resources and time it is essential to concentrate on the requirements that actually matter for the users [320].

The HoQ method describes a matrix as part of Quality Function Deployment (QFD), which is described as an overall concept to generate a meaningful translation of customer requirements into appropriate technical requirements expanded for each stage of product development and production (i.e., marketing strategies, planning, product design, engineering, prototype evaluation, production process development, production, and sales) [273].

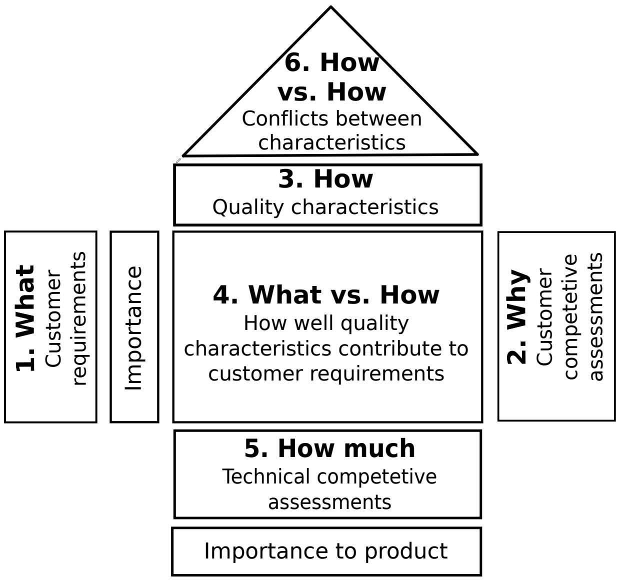


Figure 3.1: Traditional House of Quality matrix

In QFD, the HoQ is used to stepwise translate customer attributes into a production plan. The four-phase model as introduced by Hauser [122] comprises of the following transformations by rotating the matrix in each step in a way that the translated "How's" will be the "What's" in the next iteration.

1. User attributes ⇒ Quality characteristics
2. Quality characteristics ⇒ Part characteristics
3. Part characteristics ⇒ Process parameters
4. Process parameters ⇒ Production plan and inputs

Therefore, user attributes are stepwise refined into a production plan. Thus, the HoQ Matrix supports the translation of the "voice of the customers" into the "voice of the engineer" [122], in which user needs are linked with importance weights and translated in technical design attributes. In the following, the single "rooms" of the HoQ, as seen in Figure 3.1, will be described briefly.

1. What: All needs of the user or end-user, here denoted with "What", are gathered with the use of questionnaires and interviews, sorted, and listed in the left wing of the HoQ matrix. In a scale from 1 to 10 the needs are weighted by designers/ developers and users in cooperation. The weighting and collection of criteria can be performed by the usage of many different common methods.

2. Why: The right wing of the HoQ comprises customer perceptions, which represents the performance of the user needs of different products. Here, end-users are asked to assess the performance of each user need of an alternative product. This method is used to perform benchmarking between these products and to find out where and if potential for improvements is existent.

3. How: Subsequently, the user needs are translated into technical requirements (Quality characteristics), depicted here as design attributes. For the moment, the design attributes are just collected and listed without order in the first floor of the HOQ matrix.

4. What vs. How: Between the user needs and the design attributes the relationship of both parameters is defined in the relationship matrix. The designer records up to three points in the cell indicating that the requirement completely covered the need. The points are summed afterwards to guarantee that all needs find consideration in the technical transformation.

3. How much: Optionally, the deduced technical requirements can be valued with fixed ordinal or interval scale weights to perform a technical competitive assessment. Internal projects mostly do not have to consider technical competitive concerns, thus making this step unnecessary.

6. How vs. How: Design attributes can have a positive, neutral, or negative correlation to each other. Such correlations are indicated with symbols in the roof of the HoQ matrix, which will be used further on.

3.1.4 Multi Criteria Decision Making

Multi Criteria Decision Making (MCDM) concerns comprehensive decision making based on multiple input criteria. Although most decisions for choosing software are still performed in a subjective manner, several authors have tried to tackle the problem of developing a method for comparison. Well-known multi-criteria evaluation methods are: Multi-Attribute Utility Theory (MAUT) [158], the Analytic Hierarchy Process (AHP) [243], the Fuzzy Set Theory [312], and Bayesian Analysis [213]. Different methods require different types of value information and follow various optimization algorithms. Some techniques rank operations, some identify a single optimal alternative, some provide an incomplete ranking, and others differentiate between acceptable and unacceptable alternatives. The major goal of Multi Criteria Decision Analysis (MCDA) is to provide a set of criteria and methodologies that enable the development of decision support models considering the decision-makers' preferential system and judgment policy.

As used, MAUT analysis ([91, 163, 233]) incorporates consideration of attributes, which cannot be directly converted to a common metric for evaluation. The AHP Method is widely used for evaluation of software packages and has been applied in many research studies ([74, 162, 80, 214, 245]). A fuzzy based approach is used to model the uncertainty of human judgments in case of imprecise performance rating and weights ([46, 177, 183]).

Other methods also used for MCDM problems are sensitivity analysis, gap analysis, and analysis of the cost of repair [75] as well as the weighted scoring method in which importance-reflecting weights of a criterion are multiplied with rating scales. This would indicate the degree of meeting the requirement (see, for example, [141, 228]). A Decision Analysis Spreadsheet [20] facilitates the user with the selection of the criterion and uses the weighted scoring method for the analysis. However, the criteria, which are maximized, cannot be combined with the criteria, which are minimized. Therefore, an additional spreadsheet has to be generated, which can lead to inconclusive results.

3.1.5 Alternative MCDM Methods for Comparison

MCDM methodology is used to obtain a meaningful index from multidimensional data to evaluate competing alternatives. Since most MCDM methods require a homogeneous data type, data transformation techniques become necessary. Here, we will depict the methods applied to evaluate software tools designed to perform the activity of collaborative authoring in the subsequent sections.

Qualitative attributes of a system with the requirements to perform an activity within the system are subdivided in more detailed sub-attributes and aggregated into categories. The contemplation of requirements in the sub-attribute level avoids underestimation as well as overestimation of attributes and leads to a better reproducibility of evaluation results. For further examination, we refer to the categories as attributes and refinements as sub-attributes. For instance, one evaluation attribute would be the appearance of the vehicle and the sub-attributes would be color, form, and size. In the following, we describe two widely used MCDM methods with which we will compare the eHoQ method later on.

3.1.5.1 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP) is essentially the formalization of our intuitive understanding of a complex problem using a hierarchical structure. The objective of AHP is to enable a decision-maker to structure a MCDM problem visually in the form of an attribute hierarchy. For example, in the case of a vehicle, a hierarchical structure of the decision-making could be constructed as shown in Figure 3.2:

The basis for calculation of the relative weight of each factor is a pairwise comparison of each attribute. In order to help a decision-maker assess the pairwise comparisons, Saaty [243] created a nine-point intensity scale of importance between two elements.

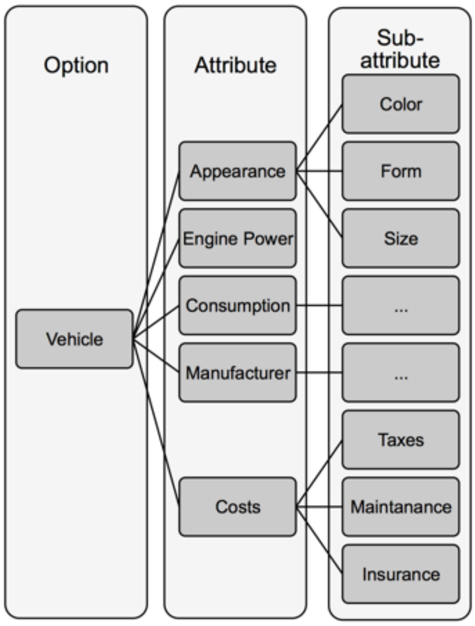


FIGURE 3.2: AHP hierarchy for a vehicle example.

The suggested numbers to express degrees of preference between the two elements A and B are shown in Table 3.1. Intermediate values (2, 4, 6, and 8) can be used to represent compromises between the preferences as well as for a greater number of alternatives.

TABLE 3.1: Preference numbers according to Saaty [243]

If A is ... as/than B	the preference number to assign is
... equally important	1
... moderately more important	3
... strongly more important	5
... very strongly more important	7
... extremely more important	9

With the use of a comparison matrix, the dominant eigenvector is calculated, which is used to extract the importance of weights on the attribute level. Finally, the relative score of each alternative is computed with respect to the decision-making goal.

3.1.5.2 *Entropy Method*

In using the MAUT method weights to indicate the relative importance of the attributes in the evaluation can be used. The entropy method is a systematic and mathematical approach for determining the weights [186]. Taking into account the experience level of the decision-maker is crucial. An inexperienced decision-maker might underestimate or overestimate attributes. This misjudgment can lead to incorrect and unrepeatable results. The entropy method adequately considers the rating of all the attributes provided to balance the relationship among numerous evaluating options [319]. The importance relative to an attribute, measured by weight, is a direct function of the information conveyed by the attribute relative to the whole set of options. This means that the greater the dispersion in the evaluations of the alternatives, the more important is the attribute. In other words, the most important attributes are those which have the greatest discriminating power between options. Using the entropy method to get a set of weights for all attributes, means only one calculation has to be performed[319]. In case of N attributes, the workload is narrowed down to one calculation instead of one calculation per attribute (N calculations), which makes the approach particularly interesting for a dataset with a large number of attributes.

3.1.5.3 *Enhanced House of Quality Implemented as MCDM Method*

As a reminder, eHoQ facilitates the decision maker to perform weighting and rating on sub-attribute level combined with a multi-attribute utility function as described in section 3.2.2. The detailed comparison between the computation and corresponding results of the three MCDM methods is performed in section 3.5.3.

Other evaluation approaches considering criteria on a lower level have been performed in the field of collaborative work with the use of Basili's Goal/ Question/-Metric [22] definition method as for example used in [95] and [250]. Steves et al. [271] used the Goal/Question/Metric to design an evaluation approach consisting of five hierarchical levels in order to refine the measurement goals stepwise resulting in a overall goal of the system. A direct comparison between alternatives is performed by comparing the overall goal of each alternative, whereby the metrics and questions might be different for each alternative. A direct influence of the measures between the alternatives and each other is not possible. Chebil et al. [63] follow a similar approach by refining the goals with a bottom-up process whereby the focus lies on a human reliability analysis in order to detect problems in e-collaboration scenarios at a first step, and an explanation of its causes at a second step. However, computing the overall value of a system and a ranking between alternative systems is not considered.

Jadhav and Sonar [141] list several tools and systems that are designed to support decision-makers in software selection. Grau [115], Bandini et al. [19], and Mohamed et al. [206] developed software tools that help the decision-maker select Commercial Of The Shelf (COTS) software components. Kathuria et al. [156] and Hlupic and Mann [125] present knowledge-based systems that can assist managers and experts in selecting Information Technology (IT) applications based on software information databases. The ESSE tool presented by Vlahavas et al. [299] supports expert assistance by guiding users in feeding values into the multi-criteria decision model. In the existing work evaluated, the observation object or field is not changeable. Thus, the systems are not applicable to other domains. But, also the applications are designed for a highly experienced user, implying that these are not suitable for less or inexperienced users.

3.2 Design Objectives for Collaborative Computer-Aided Decision Making

In this section, the methodology and components of the enhanced House of Quality are presented in order to illuminate collaborative computer-aided decision making. Based on collected user needs, this new approach defines a quality assurance tool used for two cases of application: (1) definition and rating of software components and (2) benchmarking and ranking of software/frameworks.

3.2.1 Methodology of the Enhanced House of Quality

As stated above, there is a lack of practical methods that can guide proper decision making in a collaborating system for specifying, evaluating, and benchmarking alternatives in operational and system components in a computer integrated environment. In this research, we enhance and apply the HoQ for a collaborative framework in implementation of CIDP. We enhance the HoQ in two ways. First, we integrate into the HoQ approach a taxonomy of system tasks to provide a structured approach to translate user needs to system attributes. Second, we augment the HoQ with Multi Criteria Decision Making (MCDM) to allow for a systematic way of rating those system attributes and benchmarking alternatives in a data driven fashion. These enhancements of HoQ would make the method presented in this paper effective as a collaborative framework since it allows for a data driven method of sorting among different viewpoints and variety of expert opinions in a decisive way.

The design and production of a complex system such as an aircraft or a civil infrastructure facility require information exchange among several designers and production and/or construction experts. Using a computer integrated virtual planning and monitoring tool provides a rich environment for the decision-making process allowing integration of different viewpoints and a variety of expertise. It would also facilitate quality control and asset as well as progress documentation and management. In this chapter, we present a framework for implementation of CIDP that can be used as a Computer Integrated System for Virtual Design and Manufacturing (VDM) in product development as well as for Virtual Design and Construction (VDC) in project design and delivery for the civil and transportation infrastructures. In this framework, we develop an enhanced version of the House of Quality.

This approach represents an alternative method to define software modules and their value deriving from user needs using the Quality function deployment methodology originally used in end-customer product development. The House of Quality is one of the QFD tools that is widely used to translate customer's own words into a set of detailed design specifications that can be used to guide all phases of the production process. This objective of the HoQ method is the same which is intended in software development life-cycles [21]. Based on the methodology of viewpoint-oriented requirements definition [170] and the task world ontology presented by Van Welie et al. [295], we present an ontology of system tasks which is integrated into the enhanced HoQ approach in order to understand and translate the user's needs into design and system attributes. A practical method that guides developers to specify, evaluate, and acquire weighted modules to support the design of software or collaboration frameworks and service design in particular by enhancing the traditional House of Quality is provided.

Furthermore, the multi criteria specifications can address the specific needs of the collaborating group. According to Mittleman et al. [203], most of the existing collaboration tools are developed for the mass market and do not suffice the specific needs of a collaborating group. Hence, these needs must be satisfied by the use of customized tools that apply user-centered design methodologies as mentioned by Schümmer et al. [253].

The usual approach in software evaluation is based on the experimental use of the software in which users are asked to conduct several tasks and compare their experience in using the alternative software. The underlying measurement factors in such an evaluation are often computational speed, throughput, accuracy, and usability. In the ISO standard for ergonomics of human computer interactions [234], usability is defined as effectiveness, efficiency, and satisfaction. Following this definition, features of software environments can be evaluated against each other. However, software evaluation holds interesting challenges due to the number of features that may need to be evaluated and the number and complexity of the parameters and weights that are to be considered due to multiple users who may often have divergent views. An exemplifying field, in which such an approach is used, is computer supported cooperative work (CSCW). Collaborative work is the joint activity of different team-members coordinating their tasks in the service of progress towards a group goal. Changes performed by other team members affect the total system but also individual tasks. Team members have to be aware of their impacts to adjust their performance if necessary. Considering all requirements to evaluate, such a system is a challenge. This is more evident for users with sparse expertise who have to select the most promising tool. As a consequence, the methods used to balance and choose a solution might be highly subjective and not reproducible. Multi-Criteria Decision Making (MCDM) is used to close this gap and it deals with choosing the best option from various potential candidates. One important challenge is to develop a method that can lead to reproducible results based on a given set of inputs.

The method presented in this chapter introduces a quantitative evaluation approach following seven steps that facilitates the decision-making process based on integration of Multi-Attribute-Utility-Theory (MAUT) [158] and House of Quality (HoQ) [122]. Using HoQ, we develop a method that would allow use of comparison matrices at subcategory levels, which would therefore enhance the repeatability of the decision-making process. In addition, the utilization of HoQ allows integration of products as well as user defined parameters for evaluation.

None of the existing tools support all stages of our evaluation approach or meet the requirements of facilitating the user through a step-by-step process. The proposed approach allows replacing the data set used in evaluation to allow for a spread-sheet type operation. The approach also considers the experience level of the decision-maker, resulting in a selection recommendation. The approach is easy to apply and can be used for any software evaluation. It remains the evaluation designer's responsibility to determine the evaluation criteria and the software/ system alternatives resulting in an effective reproducibility and higher accuracy of the results.

3.2.2 Components of the Enhanced House of Quality

The traditionally HoQ matrix is enhanced with four additional components: (1) a task ontology that serves to translate user needs into the design attributes in a structured way, (2) system attributes' importance rate based on multi-attribute utility theory considering correlations, (3) a utility function for precise benchmarking; and (4) a clear ranking between alternative solutions. The new components are highlighted in gray color in Figure 3.3.

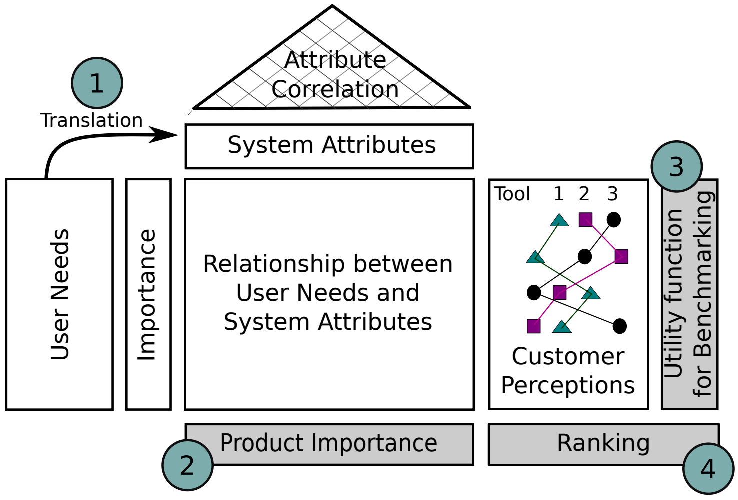


Figure 3.3: Enhanced House of Quality matrix

3.2.2.1 *Appendix 1 – Task ontology: Translating User Needs into Design Attributes*

The first appended component in the enhanced House of Quality is a task ontology applied to translate user needs into design attributes. Many other approaches of software requirement engineering assume that users understand their needs or might be able to formalize them in a proper way. The QFD process, however, provides tools to define a sufficient set of requirements from users. Once knowing what the user needs, it is still challenging to translate those vague needs in actual technical requirements. The requirements engineering process involves a clear understanding of the requirements of the intended system. Services, user, environment, and associated constraints need to be defined and connected. While the traditional HoQ uses four phases to refine the needs stepwise, we propose an alternative approach based on viewpoint-oriented requirement definition (VORD). VORD is used to structure the requirements engineering process using viewpoints associated with sources of requirements [161]. In analogy to the ontology for task world models, a viewpoint oriented approach presented by [295], each need can be formulated as a task that is performed by an actor who plays a specific role. Those tasks can use objects and trigger or are triggered by events (see Figure 2.1). While users and roles might be directly deduced from the user needs, there is a need to find a structure to define tasks and dependencies in a sufficient way. Thus, the following ontology of tasks is defined:

A task describes a relationship between user(s) and the system incorporating some kind of actions either with the underlying dataset or the information resulting from processing the dataset. A task is annotated as

$$\textbf{\textit{task}}\ (type, sender, receiver, via*).$$

Sender and receiver are both high-level actor types in the task world and describe a one directional relation of the triggering event between either the actors system and users, user and system, or user and user. The 7 action types are distinguished as following:

1. **Data acquisition.** The underlying data has to be accessed. This action type comprises the questions of how and what kind of data gets accessible to the user. There is no direct connection from and to the user but an indirect connection from the system to the user.

2. **Data manipulation.** Data is being manipulated due to many kinds of tasks. This action type comprises all kinds concerning the question of how data is manipulated. The triggering event is either initiated by user due to direct interaction with the data or indirect by the system itself.

3. **Data transfer.** Any kind of transferring data to another instance is covered in this action type. Data transfer might imply information processing and data manipulation, however, the actions which are needed to perform relocate data between actors, tasks, or different tools is covered with this type. Data transfer is initiated by the user and allocated to the user via internal steps in the system.

4. **Information acquisition.** Data themselves can be meaningless without the right processing to generate meaningful information to the user. Information acquisition covers the questions how and which information inside the data will be perceived by the user. This type can either be triggered by the same user due to, e.g., filters on the data, implying a connection between system to user, or it can be triggered due to actions of other users, implying a connection from user to user via internal steps in the system.

5. **Information processing.** This action type comprises all kinds of actions triggered by the user in order to receive the sufficient information, which does not necessarily imply the manipulation of the data. It covers the questions of how user can process or trigger the processing of information and how information can be processed by users. This process either defines a connection between user and system, or the indirect connection between users via internal steps in the system. The actions describing information processing between users and system are specifically important where any kind of cooperation between users exist, like in computer supported cooperative work (CSCW) or service design.

6. **Information transfer.** The transfer of information covers all types of exchange of knowledge and notifications between users, which needs to be supported by the system. Information transfer does not necessary include the manipulation of data nor any kind of data processing and information acquisition. Furthermore, it covers the question of how information, knowledge, and ideas that are not grounded on the underlying datasets are transferred. Two directions of information transfer are considered: the exchange from the system to users and from users to the system including information processing.

7. **External condition.** Tasks can cover actions or conditions, that are not grounded on the system. But, those conditions are required in order to perform the tasks successfully. In an analogy to information transfer this type covers also all types of exchange of knowledge and notifications between users, which do not need to be supported by the system.

The corresponding task definitions are listed as following:

(1) *task (data acquisition, system, user)*

(2) *task (data manipulation, user, system)*

(3) *task (data manipulation, system, user)*

(4) *task (data transfer, user, user, system)*

(5) *task (information acquisition, system, user)*

(6) *task (information acquisition, user, user, system)*

(7) *task (information processing, user, system)*

(8) *task (information processing, user, user, system)*

(9) *task (information transfer, user, system)*

(10) *task (information transfer, system, user)*

(11) *task (external condition, user, user*)*

Based on this ontology, design attributes, dependencies, and interfaces can be described and refined in more detail in the next step. These concepts will be illustrated in an example presented in the following sections.

3.2.2.2 *Appendix 2 – Importance Rating of System Attributes*

The correlation matrix which is located in Room 6 of eHoQ, the roof, describes the relation or rather the influence between single system attributes. Elements can have (strong) positive, (strong) negative, or no correlations between each other. A positive correlation is defined if one attribute's characteristic increases, the other attribute's characteristic also increases, but also if one attribute's characteristic decreases, the other decreases too. A typical example for this correlation is the higher the demand of a good, the higher is the good's price. Negative correlations show opposite behavior of attribute's characteristics. Here, if one variable increases, the other decreases and vice versa, e.g., the higher the temperature, the shorter the lengths of skirts. No or zero correlation is present if there is no relationship between the two attribute's characteristics such that the value of one variable changes and the other variable remains constant. In each crossing of two attributes the correlation symbol is noted.

Not only the coverage of the needs influence the importance of a systems attribute but also the correlation between system attributes themselves. While within the traditional HoQ, this matrix is optionally filled by the user in order to get the awareness of those influences, the enhanced House of Quality incorporates those influences in order to calculate the importance of the system attributes.

Based on the identified correlations, each system attribute is assigned with an diligence factor, describing that designers/developers should take reasonable diligence on the attribute as it might have a high positive or negative influence on other attributes. Therefore, each correlation characteristic is assigned to a value. For each attribute the correlation values cv_n are summed up and divided by the number of attribute pairs $(N-1)$ multiplied with the highest assignable value ($|4|$, which represents the maximum boundary).

The computation can be expressed as follows, resulting in a normalized value between 0 and 1:

$$diligence - factor \quad d = 1 + \frac{\sum cv_n}{(N-1) * |4|} \tag{3.1}$$

The possible characteristic forms of correlation and the assigned values are listed in Table 3.2:

TABLE 3.2: Correlation characteristics

Correlation	Symbol	Assigned value
Strong negative	- -	-4
Negative	-	-1
No	0	0
Positive	+	+1
Strong positive	+ +	+4

As reminder, to be more efficient in the development process and in order to enhance the product quality, knowing what are the requirements with the highest values is necessary. Thus, more effort can be spent on those high-value requirements rather than wasting the resources for less but demanding low-value requirements. It can be ensured, that the high value user needs are considered in the first row. End-user forgive poor realization of secondary functionality, but will not use the end product with poor realization or even the lack of primary functionality/needs.

The product importance of the system attributes composites the relative importance weightings of the user needs ($relative\_w_i$), the relationships between user need and system attribute (rv_i), as well as the correlations between single system attributes (d_j). The computation of the system attributes weight w_j can be expressed as following:

$$w_j = \sum (relative\_w_i * rv_i) * d_j \tag{3.2}$$

3.2.2.3 *Appendix 3 – Utility Function for Bench Marking*

The third appended component in the enhanced House of Quality is a utility function for bench marking. Multi Criteria Decision Making (MCDM) methodology is used to obtain a meaningful index from multidimensional data to evaluate competing alternatives. In our approach, HoQ is used to weight criteria (applied as user needs) and evaluate the performance of the tool in relation to other tools. Therefore, the traditional HoQ is enhanced with a multi criteria utility function for more precise benchmarking to rank several tools against each other (see gray components in Figure 3.3). Zultner [320] also stated the need of a more accurate method to perform the benchmarking. By using the HoQ matrix, rating, and weighting is performed on the sub-attribute level. The ratings are normalized, and multiplied with the relative importance weight prescribed by the decision-maker. Computing the normalization on the sub-attribute level leads to a higher value accuracy and reproducible results. The performance of each criteria is evaluated in a range from 1 to 4, while the importance of those is scaled between 0 and 10. To evaluate the single attributes per option, each sub-attribute's (criteria's) importance weight w_j is multiplied with the performance measurement sa_{ij} and normalized to one value a_{ij} per attribute. These attribute values are summed up and a clear ranking between the options is indicated.

In the evaluation approach described in section 3.5 rating and weighting is performed on the sub-attribute level. The ratings are normalized over all options, and multiplied with the relative importance weight prescribed by the decision-maker. Computing the normalization on the sub-attribute level leads to a higher value accuracy and reproducible results. The computation is expressed by the Formulas 3.3 - 3.6, whereby $\mathbf{u}_k$ is the aggregated utility of option o_k:

$$rel\_w_n = \frac{w_n}{\sum_{n=1}^{N-1} w_n} \tag{3.3}$$

$$norm\_p_{n,k} = \frac{p_{n,k}}{\sum_{k=0}^{K-1} p_{n,k}} \tag{3.4}$$

$$\mathbf{u}_k = \sum_{n=0}^{N-1} (relative\_weight \cdot norm\_performancerate) \tag{3.5}$$

$$\mathbf{u}_k = \sum_{n=0}^{N-1} \left(\frac{\mathbf{w}_n}{\sum_{n=0}^{N-1} \mathbf{w}_n} \cdot \frac{\mathbf{p}_{n,k}}{\sum_{k=0}^{K-1} \mathbf{p}_{n,k}} \right) \tag{3.6}$$

To compare the results computed with use of the eHoQ approach with other methods on attribute level (as described in section 3.5), the utility, $u_{m,k}$, of an attribute, a_m, can be formalized as follows:

$$w_m = \sum_{n=0}^{N-1} relative\_w_{m,n} \tag{3.7}$$

$$\mathbf{p}_{m,k} = \frac{\sum_{n=0}^{N-1} p_{m,n,k}}{N} \tag{3.8}$$

3.2.2.4 *Appendix 4 – Clear Ranking Between Alternative Solutions*

The fourth appended component is closely coupled with the utility function for benchmarking as the single utilities are aggregated to one single utility value, which can be compared to each other and brought into a clear relation. Compared to other design verification methods like design rationale, for example applied in [129], weighting and rational calculation can be performed, as there is no final decision between options. All requirements are used to deduce the architecture modules and get weightings of the modules. There are no alternatives which are mutually exclusive. Thus, computing the utility of every solution based on subcategory level of the user needs provides a clear ranking of the technical alternatives to deal with the

MCDM problem. Additionally, a fulfillment parameter hr_k in Formula 3.11 indicates to which degree each option corresponds to the requirements.

$$\mathbf{u}_{m,k} = \sum_{n=0}^{N-1} (relative\_weight_{m,n} \cdot mean(norm\_performancerate_{m,n})) \tag{3.9}$$

$$\mathbf{u}_{m,k} = \sum_{n=0}^{N-1} (relative\_weight_{m,n} \cdot \frac{\sum_{n=0}^{N-1} \mathbf{p}_{m,n,k}}{N}) \tag{3.10}$$

$$hr_k = \frac{\sum relative\_w_n \cdot p_{n,k}}{\sum relative\_w_n \cdot p_{n,max}} \text{ with } p_{n,max} = 4 \tag{3.11}$$

After defining the requirements for collaborative computer-aided decision making, the eHoQ approach is applied and improvements through the enhanced components is demonstrated in the following sections.

3.3 Requirements for Collaborative Computer-Aided Decision Making

Both the traditional and the enhanced House of Quality are based on a structured list of user needs. Existing literature provides evaluation criteria regarding specific tasks ([293, 196]) or strategies for evaluation [265], but lack in rating techniques and do not give further suggestions on how to measure those criteria or provide information of the related measurement. However, such criteria catalogs and strategies can be used as input parameters in our evaluation application.

Thus, an extensive literature review was performed covering the fields of Human Computer Interaction (HCI), Computer-supported collaborative work (CSCW), cognitive science and social science. From this wide range of collected publications, we identified criteria for collaborative work dedicated to specific technology [43] like table tops [255] or mobile devices [84]. We extracted criteria of collaboration support systems in information visualization [285], visual analytics [47], business processes [208], virtual reality [184], and design and engineering [106, 149, 50, 49]. Also, existing work about single aspects of successful cooperation like awareness indication [27, 86], as well as satisfaction and team effectiveness as investigated by [107, 153, 240] have been considered. The criteria catalog below provides an overview of the most important support features for collaboration systems and contains design recommendations to achieve the desired facilitation support. Explanations about the selection process we performed is provided.

3.3.1 Identification Process of Comprehensive Criteria Set

To identify an all-embracing criteria of collaboration support systems, existing theoretical and practical literature that intend to support collaborative work are observed and criteria are identified. In the first step, the criteria have been collected, sorted, and grouped in conformity with the proposed instructions (e.g. highlighting, screen sharing, etc.). Then, analogous criteria have been detected and integrated into single criterion definitions while additional criteria are deduced from existing ones. Additionally, the results of this classification are validated through extensive interviews and questionnaires, and by monitoring and observing collaborative design sessions. Afterwards a categorization based on the impact of the instructions has been performed (e.g. instruction: accentuating leads to impact: information sharing). The resulting catalog of collaboration support criteria is presented below:

1. Content support refers to the active interaction and integration of the content by the actors, which is highly reliant on the underlying task. Drafting and task execution phases as depicted in Figure 2.2 are strongly related to the underlying task model, and can be, for example, a writing process, designing or creation. These criteria are required in adapted form, which are derived from the task models. The following criteria are comprised in this category:

a. Content integration: Add, Associate, Modify, Delete.

b. Move: Change structures and appearances.

c. and Judge: Render an opinion to the made contribution.

2. Information sharing involves functionalities and technical modules, which are used to share information in order to establish the knowledge base for all actors. The following criteria are comprised in this category:

a. Quickly retrieve context-relevant information: Ability to detect changes.

b. Access to shared objects: Ability to access and edit shared artefacts

c. Accentuating: Pointing, marking, annotate.

d. Track others approach: Can improve coordination and skill transferability.

e. Screen sharing: Ability to accentuate own or draw attention to others' viewpoint.

f. Individual and/or shared workspaces: Allows performing single tasks and keep track of overall goal.

3. Coordination support refers to the spectrum of tasks that are used to support the coordination of work packages and the coordination between actors. The following criteria are comprised in this category:

a. Jurisdiction: Assignment of tasks, roles, responsibilities, rights.

b. Transformations: Transitions between personal and group work, between activities, and between tools and external work.

c. Alert mechanisms: Notification of changes or of required user input.

d. Awareness Support: Amplifies coordination and communication.

e. Community Support: Online documentation and strength of the community.

f. Team structure and size: Ability to create team structures and optimal team sizes.

g. Changing work styles: Ability to change between single and group work.

4. Communication support describes the support of advanced and unimpeded communication among actors to bridge spatial gaps [43]. The following criteria are comprised in this category:

a. Communication in group and or individual: Ability to perform public and private conversations.

b. Discussion tool: Rich and powerful communication channels.

c. Encrypted Communication: Increases actor's reliance in the technology.

5. Compliance support relates to rules or guidelines that should be fulfilled to conduct an ideal decision making process. The following criteria are comprised in this category:

a. Avoid team debates: If actors have to explain themselves with low accordance the individual's satisfaction is reduced.

b. Group process training: Supports the sense of cohesiveness and shared goal.

c. Reflecting all individuals' notions/opinions: The feeling of being left out arises user's frustration.

d. Use guidelines and defined restrictions instead of strict rules: Guidelines for task performance, discussion, and decision making.

e. Involving all actors: Actors without a task or role are not part of the work team.

f. Team self-managing behaviors: The ability of actors to collaboratively assume responsibilities for directing their task accomplishment toward the achievement of the established team goals.

6. Content management refers to the action execution of dynamical content manipulation by actors and granting valid and reliable database entries. The following criteria are comprised in this category:

a. Action parameter: Synchronicity of action and identifiability of actors.

b. Access Control: Allocation of access rights.

c. Session Persistence: Degree to which contributions are ephemeral or permanent.

d. Consistency and interactivity: Causality, concurrency, simultaneity, instantaneity.

7. Usability involves next to user satisfaction to the degree of efficiency and effectiveness of the technology, including the acceptability by users. The following criteria are comprised in this category:

a. Reliability: Same results are achievable with different actors.

b. Reusability: Session/ results/ configurations can be recorded and reused.

c. Transferability of skills: Degree of apprenticeship of a novel actor in order to gain insights and being able to conduct work practices on their own.

d. Flexible actor arrangements: Configurable collaborative components scaling over different types and sizes of input devices.

e. Guidance: Provision of facilitation support.

f. Generalizable and ease of maintenance: Generic for different projects.

8. User experience describes the actor's emotions and attitudes about using the technology. The following criteria are comprised in this category:

a. Natural interpersonal interaction: Urges collaborative interaction/communication.

b. High user satisfaction and motivation: Amplifies active participation.

c. Intuitive and simple technology: The tool is easy to understand and use.

d. Reduced cognitive load of actors: Actors should be able to focus on the task.

e. No all-embracing knowledge/expertise needed: actors only learn the techniques needed to conduct own domain related actions.

3.3.2 Applied Criteria Catalog onto Collaborative Authoring Task

We demonstrate the use in practice of the described catalog by illustrating the impacts of the interdisciplinary collaboration requirements using a dedicated task model of collaborative authoring.

3.3.2.1 Collaborative Authoring

The aim of collaborative authoring is to combine input from multiple users in order to create a written document. We could identify several different feasible approaches to accomplish this task, which makes this case study particularly interesting. In the following, we will describe the task model for the activity of collaborative authoring under the definition of task model elements as described in section 2.2.1:

Agents. Human agents as well as non-human agents are considered in collaborative authoring. Human agents can have different levels of expertise but may also come from different domains. The human actors in the task model are author teams and "support staff", which can provide input like images or tables, but are not actively involved in the writing process. Non-human agents are support tools and systems.

Roles. Certain roles are assumed by the agents: author, content editor, literature searcher, and spelling editor.

Activities. Collaborative authoring starts with high-level activities, like conducting a planning meeting to define the concept of the outcome. A document outline may be defined and writing tasks are distributed among the authors. This assignment of work can be done by a single individual or through mutual agreement of the authors. Medium-level activities, like writing single chapters, creating graphs, or collecting references, follow and are mainly performed by single agents. Low-level activities describe sub-tasks of medium level tasks: combining document fragments, note-taking, and storing documents and references electronically. During the writing process, the following medium level activities are repeatedly performed:

- Writing
- Formatting
- Commenting
- Reviewing and revising

Activity order. After the concept of the document is decided, there are multiple possible orders in which tasks can occur:

- *Iteratively:* The document is written and formatted in whole. Thereafter, it is reviewed, possible changes are discussed and the document is revised.
- *Simultaneously:* One group of agents generates text while another group is in charge of the design and formatting.
- *No particular order:* During the writing process the tasks occur if needed. For example: An agent decides that the font size is too large, therefore the document is reformatted.

Although, single agents mainly perform medium level activities (affected by commenting and marking), continuous feedback loops between the activities and roles exist. The activities and their relations are shown in Figure 3.4.

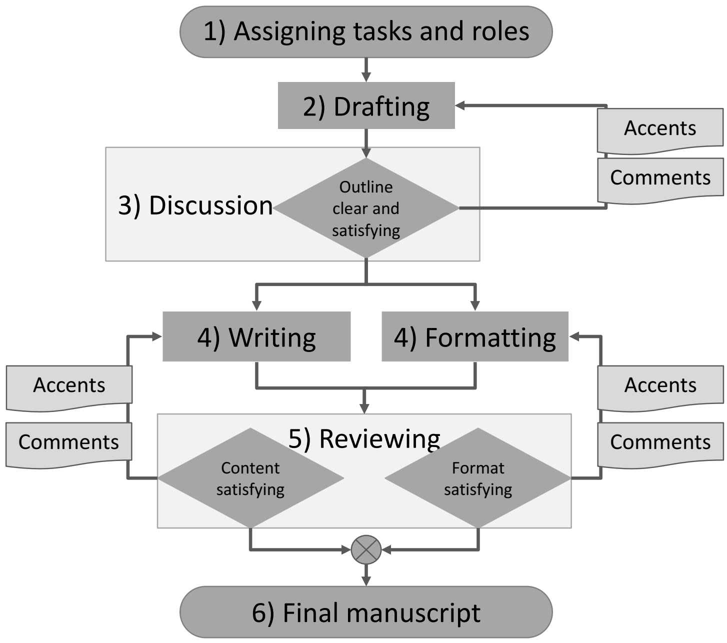


FIGURE 3.4: Activity model of collaborative authoring.

Objects. Identified shared artifacts in the task model are: documents, containers that store artifacts (physically: shelf; electronically: USB), and combiners, which are intended to be permanent (staple, printed document, etc.).

Events. Certain events concern relevant changes in the state of the task that result in starting, changing, or stopping an activity. Thus, the following events can be formulated: start event, deadline event, and stop event.

3.3.2.2 *Criteria Identification for Collaborative Authoring*

During the process of collaborative authoring, the following medium level activities are identified: writing, formatting, commenting, accentuating, reviewing, and revising. These activities can be directly adopted as features that have to be supported by the system. By using the catalog, these features are refined by choosing the corresponding criteria. Finally, the selected criteria for supporting the task of collaborative authoring can be listed. The enumeration value corresponds to the enumeration value within the complete criteria catalog.

1. Content support
   a. Content integration
   b. Move
   c. Judge
2. Information sharing
   a. Quickly retrieving context-relevant information
   b. Access and edit shared artifacts
   c. Accentuating
   e. Shared and individual workspaces
3. Coordination support
   a. Jurisdiction
   b. Transformation: personal and group work; tool and external tool
   g. Changing work styles
4. Communication support
   a. Group discussion tool
5. Compliance support
   e. Involving all actors
6. Content management
   a. Synchronization of actions
   c. Session persistence
   d. Consistency and interactivity
7. Usability
   b. Re-usability of the results and configurations
8. User experience
   b. High user satisfaction and motivation
   c. Intuitive and simple technology

To highlight the identification process, we discuss the criteria selection of the category **Coordination support** from the generic catalog adopted to this example. Jurisdiction (3.a) is crucial to assign tasks and roles to guarantee that every actor can actively participate. Transitions between personal and group work (3.b), including the ability to change between working styles (3.g), is important to enable the authors to make own drafts individually and merge them with the group work when satisfied with the content. Also transitions between the collaborative authoring tool and external tools should be provided to utilize the resulting document.

Excluded from the general list have been the criteria 3.c, 3.d, 3.e and 3.f. Alert mechanism (3.c) are used to notify authors about when and where did changes occur. Awareness support (3.d) enables to recognize who is currently working on the document. These criteria help to coordinate the process and probably lead to a higher user satisfaction, but they are not crucial to successfully perform the overall task, as tested in the collaborative authoring tool Overleaf [310]. Community support (3.e) of the document preparation system (LaTex vs. WYSIWYG word processors like Microsoft Word), might be necessary but it is not considered as requirement of the collaboration environment. As the team structure in the collaborative authoring process might not be multi-layered, it is not crucial to create additional hierarchies in the team structure (3.f). Hence, these criteria are not selected in this case.

The presented criteria catalog is in conformity with the user needs that are further used within the eHoQ methodology to identify software architecture modules of new systems. Furthermore, the catalog also serves as a guideline for evaluating existing collaboration systems.

3.4 Design of a Collaborative Computer-Aided Decision Making System

In this section the deployed task ontology, as first appendix in the enhanced House of Quality, and the importance rating, as second appendix, is applied onto collaborative complex system design of an aircraft vehicle system. The domain of aircraft vehicle system design is explored, before the performed approach is explained and the adaptability is demonstrated. As a result, the definition of a feasible software architecture supporting the collaboration within the underlying use case is developed.

3.4.1 Applied Task Ontology onto Collaborative Complex System Design of an Aircraft Vehicle System

A typical example for complex systems that serves as exemplary scenario are aircraft vehicle systems. As depicted in Figure 3.5, aircraft vehicle systems are composed of many sub-systems, incorporating a set of thousands of equipment linked with interdependent interactions. Only considering the evident sub-systems of an aircraft vehicle, 17 subsystems can be seen: engine control unit, auxiliary power unit, fuel system, air conditioning system, power supply system, access doors system, fire-fighting and smoke-alarming systems, wheel breaking system, oxygen system, hydraulic system, water supply system, anti-icing system, lighting facilities system, cabin pressure control system, landing gear system, taxiway device control system, aircraft and wing-flap control system. Each of these sub-systems is itself composed of sub-systems or equipment and integrated in a complex network of interactivity.

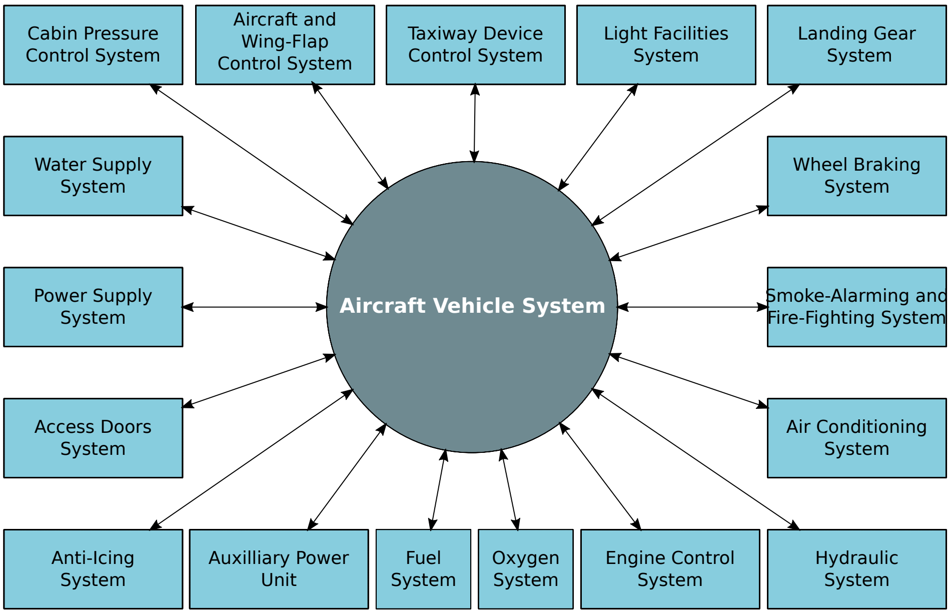


FIGURE 3.5: System architecture of an aircraft vehicle

Figure 3.5 will not be further described as it should merely provide an overview of the complexity of such systems rather than going deeper into the dependencies between each subsystem. A complete overview of all dependencies and interconnections of an aircraft vehicle system can be found in [268].

3.4.1.1 *Activities*

The main activities involved in aircraft vehicle systems are:

1. System design and integration of the vehicle sub-systems into the aircraft system.
2. Follow-up services, technical expertise, and technical facts processing for the aircraft in service.

The activities of category "follow-up services" are not limited to design systems only. These mainly focus on the tasks maintenance and improvements of the aircraft systems during the complete life-cycle of the vehicle as well as during decommissioning. In the further examination, we will focus on the design process as elementary task.

3.4.1.2 *System Design Process*

Single sub-systems are first designed separately and integrated into the complete system successively. Each sub-system constitutes an optimization problem that has to be solved. Here, computer simulations are used frequently in order to replace expensive physical experiments and improve the quality and performance of engineered products. Simulations empower designers with a higher flexibility while studying diverse phenomena under controlled conditions. However, computer simulations require a substantial investment of computation time which can take many minutes, hours, days, or even weeks ([103, 266]). A surrogate model is a simpler approximation model to predict the system performance and to develop a relationship between the system inputs and outputs. When properly constructed, these approximation models mimic the behavior of the simulation accurately while being computationally cheaper to evaluate [113].

The system design process of each subsystem is a collaborative process, which is integrated into the network of the aircraft system, incorporating interdependencies and interaction between the single subsystems. Thus, the collaborative design process of subsystems is mutually contained within the collaborative process of the aircraft system design process. For example, we will examine the collaborative design process of one sub system in order to illustrate the iterative work-flow and the included user-models.

Actors. In the system design process, it can be differentiated between five different actors. The roles and main responsibilities are listed as following:

- *Project Lead:* Responsible of planning the engineering activities within the project
- *Aircraft Architect:* Specify targets for subsystem and arbitrates trade-offs between subsystems
- *Thermal Architect:* Creates architecture of the air-conditioning ECS subsystem
- *CAE(CAD) Analyst:* Generates CAE based models
- *Method Engineer:* Provides sets of predefined simulation services

Tasks. The design process can be characterized by 13 steps, depicted in Figure 3.6. Precondition for each design process is the creation of a work-plan (1). Based on the work-plan and identified requirements, targets for the environment control system (ECS) are defined (2) and refined into the design space and cost functions (3). With that information, the frame and preconditions for creating a surrogate model are given. Subsequent, a surrogate model is requested (4). In order to generate such a surrogate model (7), a computer-aided engineering (CAE)/ computer-aided design (CAD) model is defined (5) and computed (6). Afterwards, the surrogate model is integrated into the ECS (8) and the resulting design space and sensitivity is explored (9). The design space and the surrogate model are iteratively refined until a robust optimization is found (10), which is analyzed and reviewed collaboratively (11). Finally, the ECS is validated and published (12), which leads to the end of project (13).

Work flow. The work flow and the interaction between all actors is depicted in Figure 3.6. While most tasks are solved sequentially in that model, iterations within tasks are most probably between the steps 11 (collaborative review) to 2 (define ECS target) in order to modify requirements in the ECS, 9 (explore design space and sensitivity) to 4 (request surrogate model) in order to refine the surrogate model, and 9 to 8 (integrate surrogate model) in order to refine the design space. Predefined simulation services and other methods are provided in each step and can be requested by method engineers acting in the role of rectification and process optimization support. Simultaneously, project leader is monitoring and controlling the process.

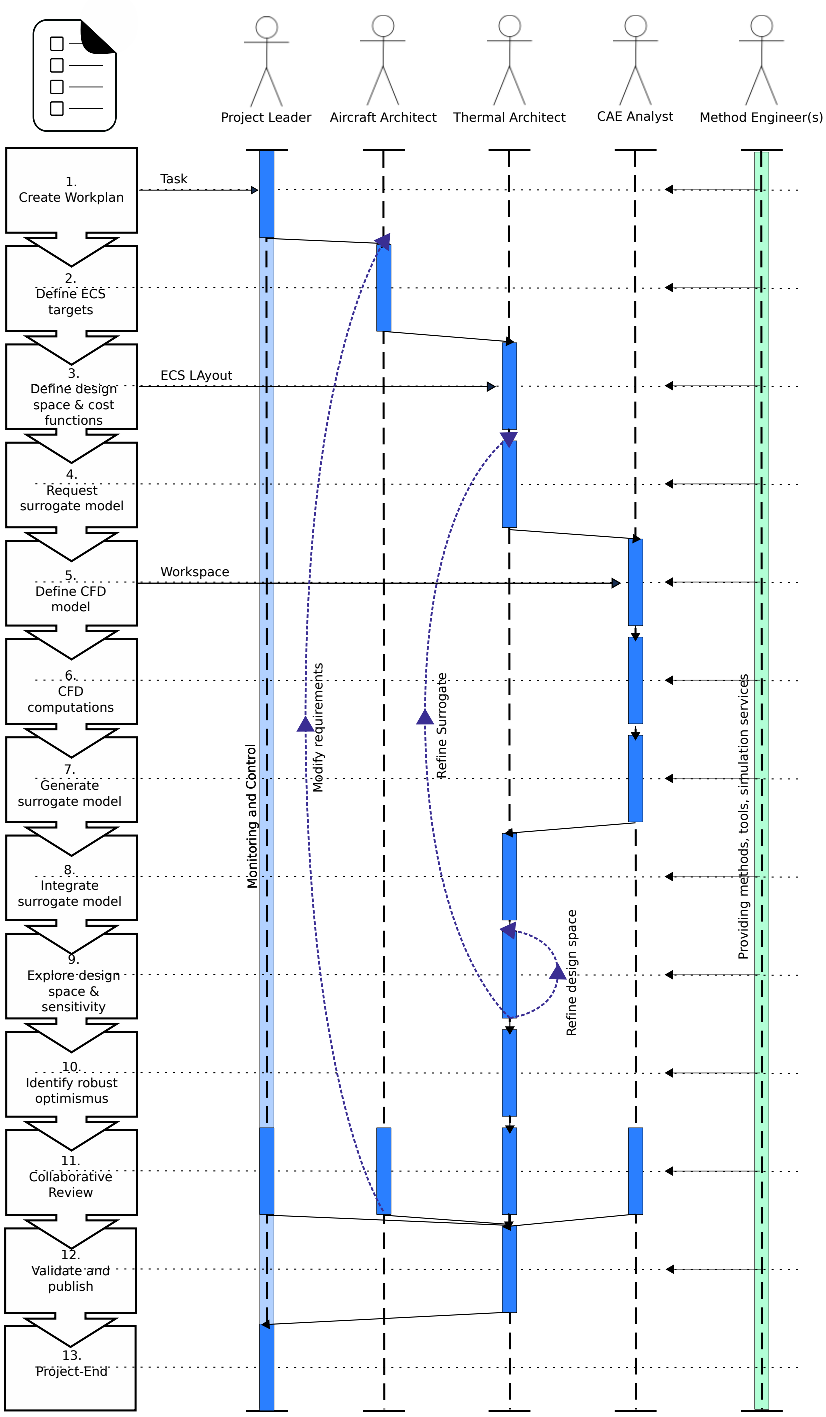


FIGURE 3.6: Tasks and workflow of complex aircraft design

3.4.1.3 *Requirements*

Thomas et al. [280] stated a list of requirements to support the complex design of an aircraft. These requirements are sorted and categorized as presented in chapter 3.3. The general requirements on such a system as stated in the original source are:

- Compatibility with the tool managing.
- Project management during the entire life- time of an aircraft.

- Collaborative work between all stakeholders of the aircraft system design.
- Facilitating System engineering process: requirements, functional, and architecture management.
- Several architecture analyses, in particular behavioral simulations, based on 3D, and system representations.

It is quite easy to see that the software architecture for such a system needs several different components describing individual independent systems to perform (1) performance analysis, (2) modeling, and (3) simulation. Those systems need to be combined and exchanges between them enabled. Additional to the categories stated in section 3.3, requirements that refer not specifically to the supporting system but rather to the included dataset are identified. Thus, the following categories of requirements are in interest:

- Dataset
- Content support
- Information sharing
- Coordination support
- Content management
- Usability
- Communication support

The requirements on a more detailed level will be examined in the next section.

3.4.2 Applied HOQ Design Approach on Collaborative Complex System Design of an Aircraft Vehicle System

In this section, we perform the complete enhanced House of Quality methodology as introduced in Chapter 3.2 in order to deduce system attributes and software modules for collaboration frameworks on the example of complex system design of an aircraft vehicle system. This investigation is intended to be used by software designers in order to support decision making processes and the architecture design of collaborative software.

3.4.2.1 Step 1 – User Needs and Importance Weightings

We start filling out the HoQ matrix with the identified user needs and corresponding weightings as sketched in Figure 3.7.

The user needs are defined based on the identified requirements. The corresponding weightings are evaluated together with end-users in a collaborative discussion. After all weightings are assigned, the relative weightings are calculated by dividing the assigned weight with the sum of all weightings.

3.4.2.2 Step 2 – System Attributes

As described in Chapter 3.2, a need is defined as a task describing a relationship between user(s) and the system incorporating some kind of actions either with the

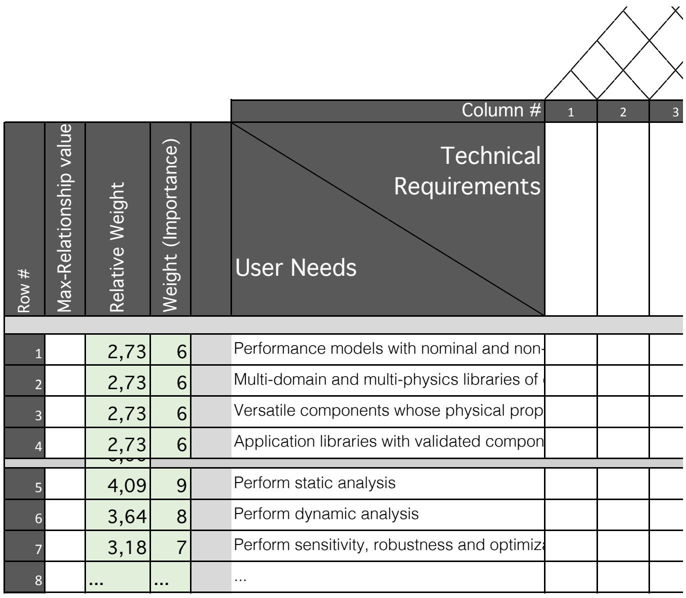


FIGURE 3.7: House of Quality matrix – user needs and importance weighings

underlying dataset or the information resulting from processing the data. Thus, each identified need is translated into a task definition. All needs with corresponding weightings and translated task definition are listed in Table 3.3. The requirements #1-#4 are directly corresponding to the dataset that are translated into external conditions, as these do not have a direct influence on the system architecture. Independent system components are integrated to do a performance analysis, create functional and logical models, and design and perform simulations. Those describe the main functionality of the desired system. The functional requirements of these components (#5-#17) correspond to the category *content support*. The independent components need to be combined and exchanges between them enabled. The requirements #18-#33 assure that the core functionality can be performed simultaneously in collaboration with different stakeholders.

Thus, the following modules of the system architecture can be derived:

- Data acquisition
- Data manipulation
- Data transfer
- Information acquisition
- Information processing
- Information transfer
- External conditions

Room 3 of the HoQ matrix can be filled out accordingly. Figure 3.8 shows the completion of the system attributes.

3.4.2.3 Step 3 – Relationships, Correlations, and Product Importance

Till this step single components of the systems architecture and their interfaces are already defined. In the next step the absolute and relative importance of each component is calculated. Therefore, in the next step of the HoQ method, the relationship

TABLE 3.3: User needs of collaborative complex design of aircraft vehicles according to [280] translated into system tasks as described in section 3.2.2

Requirement				Task
#	w_i	$w_i^{rel.}$	**Dataset**	
1	6	2.68	Performance models with (non-)nominal behaviors	External Condition
2	6	2.68	Multi-domain/multi-physics libraries of components	External Condition
3	6	2.68	Versatile components with physical properties	External Condition
4	6	2.68	Application libraries incl. validated components	External Condition
#	w_i	$w_i^{rel.}$	**Content support**	
5	9	4.02	Perform static analysis	task (data manipulation, user, system)
6	8	3.57	Perform dynamic analysis	task (data manipulation, user, system)
7	7	3.13	Perform sensitivity, robustness, optimization analysis	task (data manipulation, user, system)
8	4	1.79	Ability to enable a model validity checker	task (data manipulation, user, system)
9	8	3.57	Ability to define physical hypotheses	task (data manipulation, user, system)
10	8	3.57	Ability to add stochastic data to models afterwards	task (data manipulation, user, system)
11	5	2.23	Model debugging for developer and end user	task (data manipulation, user, system)
12	7	3.13	Intelligence to find robust and optimized designs	task (data manipulation, user, system)
13	4	1.79	Ability to increase granularity of models	task (data manipulation, user, system)
14	8	3.57	Explore alternative architecture designs	task (info. acquisition, system, user)
15	8	3.57	Ability to compare design points visually	task (info. acquisition, system, user)
16	9	4.02	Perform structural analysis	task (info. acquisition, system, user)
17	9	4.02	Ability to connect and integrate models	task (data manipulation, user, system)
#	w_i	$w_i^{rel.}$	**Coordination Support**	
18	6	2.68	Ability to incorporate new observers	task (info. processing, user, system)
19	7	3.13	Ability to exchange model between partners	task (data transfer, user, user, system) task (info. transfer, user, system)
20	7	3.13	Transformation between sub-systems	task (data transfer, user, user, system) task (info. transfer, user, system)
#	w_i	$w_i^{rel.}$	**Content management**	
21	8	3.57	Action parameter	task (data manipulation, system, user)
22	7	3.13	Access Control	task (data acquisition, system, user)
23	7	3.13	Session Persistence	task (data acquisition, system, user)
24	8	3.57	Consistency and interactivity	task (data acquisition, system, user)
#	w_i	$w_i^{rel.}$	**Usability**	
25	8	3.57	Reliability	task (data manipulation, system, user)
26	8	3.57	Reusability	task (data manipulation, system, user)
27	4	1.79	Flexible actor arrangements	task (info. processing, user, system)
28	5	2.23	Guidance	task (info. acquisition, system, user)
#	w_i	$w_i^{rel.}$	**Communication Support**	
29	4	1.79	Discussion tool	task (info. acquis., user, user, system)
30	4	1.79	Communication in group and or individual	task (info. transfer, user, system)
#	w_i	$w_i^{rel.}$	**Info. sharing**	
31	8	3.57	Access to shared objects	task (data acquisition, system, user)
32	6	2.68	Help users to localize the cause of problem	task (info. acquisition, system, user)
33	9	4.02	Visual features to quickly locate important information	task (info. acquisition, system, user)

matrix of user needs and system attributes has to be completed. Based on the translation of the user needs into task definitions, those relations can be directly transcribed into the HoQ matrix. The relation between a user need and a system attribute is described by the coverage of the need by realizing the system attribute. The characteristics are assigned with the values rv= 1, 3, and 9, which are used further to calculate the importance weight of system attributes.

The possible relations are shown in Table 3.4:

TABLE 3.4: Relationship characteristics

Relation	Symbol	Assigned value
Need is poorly covered	+	1
Need is moderately covered	+ +	3
Need is strongly covered	+ + +	9

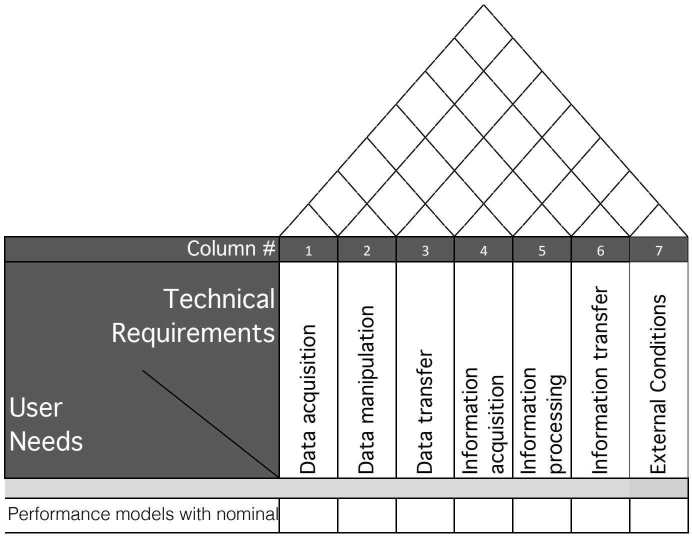


FIGURE 3.8: House of Quality – system attributes

The correlation matrix which is located in Room 6, the roof, describes the relation or rather the influence between single system attributes. Elements can have (strong) positive, (strong) negative, or no correlations between each other. Those correlations are identified and calculated into system attributes importance rates. An exemplary execution of the relationship identification and correlation assimilation is depicted in Figure 3.9.

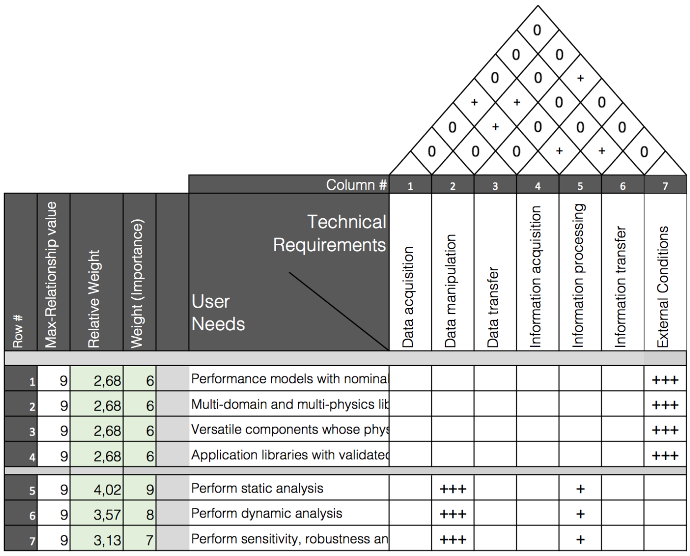


FIGURE 3.9: Relationship matrix

Having identified all relationships and correlations between user needs and system attributes, we can compute the importance rate of the system attributes as stated in section 3.2.2. For example, the weight of *Data acquisition* will be computed. In the following Table 3.5 the user needs covered by the system attribute *Data acquisition* are listed with respective relative weights and coverage symbol.

TABLE 3.5: Covered needs by *Data acquisition*

#	$w_i^{rel.}$	User need	Coverage
22	3.13	Access Control	+++
23	3.13	Session Persistence	+++
24	3.57	Consistency and interactivity	+++
31	3.57	Access to shared objects	+++

First, the sum of relative weights and coverage value is calculated. For the example, the following values are computed:

$$\begin{aligned}\sum(w_i^{rel.} * rv_i) &= 3.13*9. + 3.13*9 + 3.57*9 + 3.57*9 \\ &= 120.60\end{aligned}$$

TABLE 3.6: Correlations with *Data acquisition*

Correlation to...	Correlation Symbol
Data manipulation	o
Data transfer	o
Information acquisition	++
Information processing	+
Information transfer	o
External conditions	o

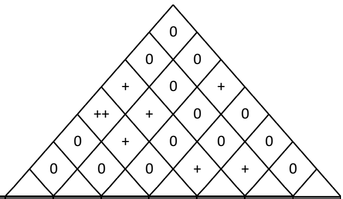
Following, the diligence factor d is calculated, as multiplier describing the degree of elaborateness designers/developers have to apply.

$$\begin{aligned}d_j &= 1 + ((4+1)/(7-1)*4) \\ &= 1.21\end{aligned}$$

Finally, the absolute weighting of the system attribute w_j is computed:

$$\begin{aligned}w_j &= 120.60 * 1.21 \\ &= 145.93\end{aligned}$$

The complete filled House of Quality matrix can be seen in Figure 3.10. For better readability, the values for w_j and d are rounded to integral numbers.



Row #	Max-Relationship value	Relative Weight	Weight (Importance)		User Needs \ Technical Requirements (Column #)	1 Data acquisition	2 Data manipulation	3 Data transfer	4 Information acquisition	5 Information processing	6 Information transfer	7 External Conditions
1	9	2,68	6		Performance models with nominal and non-nominal behaviors							+++
2	9	2,68	6		Multi-domain and multi-physics libraries of components							+++
3	9	2,68	6		Versatile components whose physical properties can be parameterized							+++
4	9	2,68	6		Application libraries with validated components should be valuable							+++
5	9	4,02	9		Perform static analysis		+++			+		
6	9	3,57	8		Perform dynamic analysis		+++			+		
7	9	3,13	7		Perform sensitivity, robustness and optimizations analysis		+++			+		
8	9	1,79	4		Ability to enable a model validity checker		+++			+		
9	9	3,57	8		Ability to define physical hypotheses		+++			+		
10	9	3,57	8		Ability to add stochastic data to models afterwards.		+++			+		
11	9	2,23	5		Model debugging for developer and end user		+++			+		
12	9	3,13	7		Intelligence to find robust and optimized designs		+++			+		
13	9	1,79	4		Ability to increase granularity of models		+++			+		
14	9	3,57	8		Explore alternative architecture designs				+++			
15	9	3,57	8		Ability to compare design points visually				+++			
16	9	4,02	9		Perform structural analysis				+++	+		
17	9	4,02	9		Ability to connect and integrate models		+++			+		
18	9	2,68	6		Ability to incorporate new observer					+++		
19	9	3,13	7		Ability to exchange model between partners			+++			++	
20	9	3,13	7		Transformation between sub-systems			+++			++	
21	9	3,57	8		Action paramete		+++					
22	9	3,13	7		Access Control	+++						
23	9	3,13	7		Session Persistence	+++						
24	9	3,57	8		Consistency and interactivity	+++						
25	9	3,57	8		Reliability		+++					
26	9	3,57	8		Reusability		+++					
27	9	1,79	4		Flexible actor arrangements					+++		
28	9	2,23	5		Guidance				+++			
29	9	1,79	4		Discussion tool				+++			
30	9	1,79	4		Communication in group and or individual						+++	
31	9	3,57	8		Access to shared object	+++						
32	9	2,68	6		Help users to localize the cause of problem				+++			
33	9	4,02	9		Visual features to quickly locate important information				+++			
					Max. Relationship value in column (check #)	9	9	9	9	9	9	9
					Weight (Importance)	121	374	56	197	75	35	96
					Diligence Factor	1,21	1,08	1,04	1,25	1,17	1,04	1,04
					Absolute Weight	146	405	59	246	88	36	100
					Rank	**3**	**1**	**6**	**2**	**5**	**7**	**4**

FIGURE 3.10: House of Quality matrix for collaborative aircraft vehicle design

3.4.3 Feasible Software Architecture Including Ranked System Attributes

The applied approach based on the enhanced House of Quality including the stated system task ontology leads in a first step to the translation from unstructured and unspecific user needs to a clear list of tasks and system attributes. With the use of the multi attribute utility theory corresponding weights and importance of the system attributes can be calculated and deduced considering correlations between each other. As a consequence, the designer/programmer know from early on where to put forth effort and resources in order to enhance the efficiency of the design process and to enhance the quality of the overall product.

With the demonstrated procedure in section 3.4.2 the absolute weights of all system attributes for the underlying use case of collaborative complex design of an aircraft vehicle are calculated. The final results are listed in Table 3.7.

Table 3.7: Absolute weighting of system attributes

System attribute j	w_j
Data acquisition	145.93
Data manipulation	404.79
Data transfer	58.59
Information acquisition	246.09
Information processing	87.5
Information transfer	36.27
External conditions	100.44

As a result, a clear identification of the absolute weightings and rankings of the importance values of the systems attributes is easily detectable. The system attributes sorted according to their importance are listed in Table 3.8. Accordingly, designers/developers should take the most attention on the attribute *Data manipulation*, while *Information transfer* needs less attention.

Table 3.8: Ranking of system attributes importance

Rank	System attribute j
1	Data manipulation
2	Information acquisition
3	Data acquisition
4	External conditions
5	Information processing
6	Data transfer
7	Information transfer

Finally, the task definitions (1) - (11) from section 3.2.2 are connected in an ontology that describes the layer of the system model and required interfaces between the components. Figure 3.11 depicts the system layer, components, and relations between the system and user(s).

Subsequently, the system attributes are further refined into interfaces and functionalities, which will be further demonstrated in Chapter 5. The system attributes are colored according to their importance rating in regards to the presented use case of

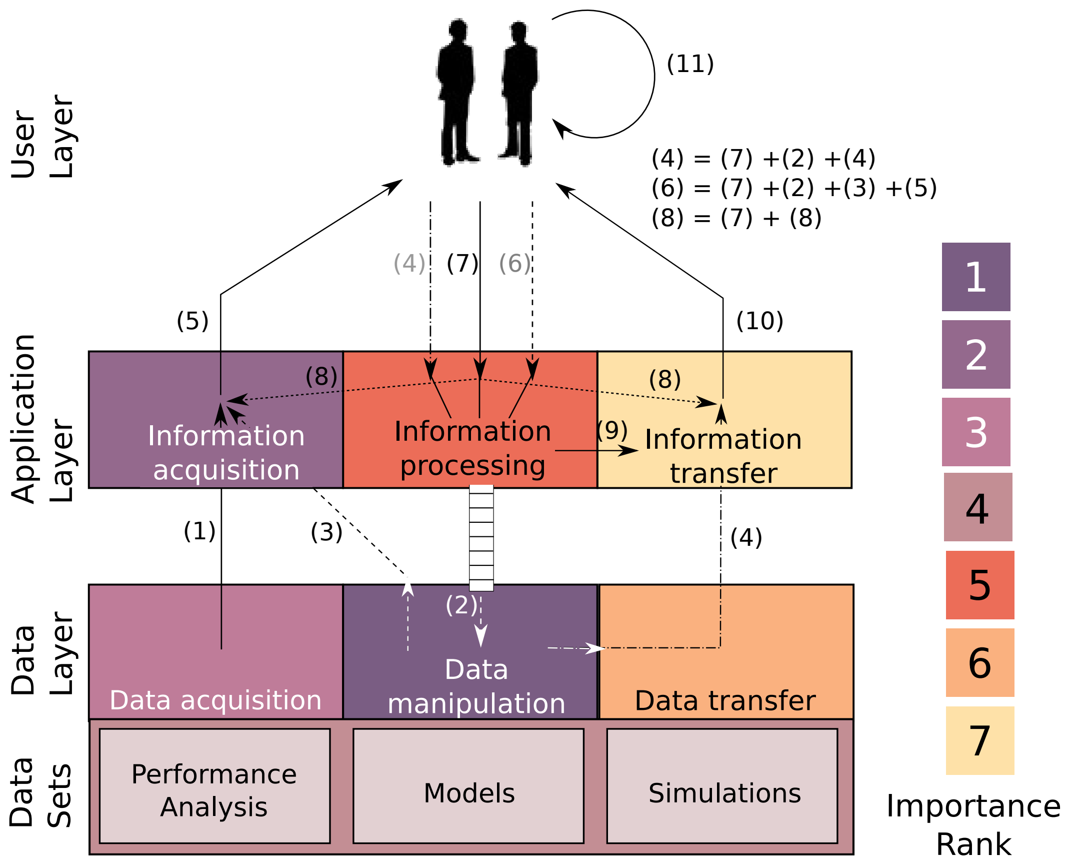


FIGURE 3.11: Ontology of tasks in system design, comprising relations between data sets, data layer, application layer, and user layer.

collaborative complex design of an aircraft vehicle, in Figure 3.11. The colors are inspired by the matter color map provided by Thyng et al. [283]. The original matter color-map is sequential with whitish-yellow for low values and increasing in pink with increasing value.

The stated ontology is generally feasible and can find application in manifold usage scenarios. The importance ranking and elaboration of the single system attributes and interfaces are to be calculated and identified individually in regards to the specific domain and utilization.

3.5 Design Evaluation based on Multi-Criteria Decision Making

In this section appendix (3) a utility function for precise benchmarking; and appendix (4) a clear ranking between alternative solutions of the enhanced House of Quality are applied within an evaluation approach based on multi attribute utility theory (MAUT) and multi-criteria decision making (MCDM). The underlying use case is specified by collaborative authoring. First, the deployed evaluation approach, using eHoQ as MCDM method, is stated and elaborately discussed. Following, the evaluation approach is applied onto collaborative authoring, comparing the three MCDM methods, namely the Analytic Hierarchy Process (AHP) [243], the Entropy method [186], and eHoQ. Three different collaborative software environments are evaluated and compared: Google Docs, ONLYOFFICE Online Editor, and Overleaf. The comparative results presented are not intended as an endorsement or lack thereof any of these software packages. The ranking of these software packages can change if one changes the criteria used in the evaluation indicating that each can be better than other in certain aspects. The comparative results are only presented to show the effectiveness of the developed method. The results show the effective reproducibility of the eHoQ method, developed in Chapter 3.1. Thus, the utilization of eHoQ and the implied improvements are demonstrated, resulting in a clear benchmark and ranking between alternative software solutions.

The evaluation approach is embedded in a web-based application and guides the user through each step. The application can be used in different domains and for different examination objects, as the underlying data set is interchangeable. Considering that decision-makers could have different degrees of experience; the data set is expendable with task models which contain descriptions of the main work packages as well as a prequalification and weighting of the requirements that are necessary for performing the described task. The application contains all MCDM methods and appraises an overall result, easily recognizable for the user.

3.5.1 Evaluation Approach for Benchmarking Software Solutions

One of the most readily understandable approaches for decision analysis is that of Multi-Attribute Utility Theory (MAUT) by Keeney and Raiffa [158]. We use this approach as a guideline for our evaluation process. We follow the 7 steps, depicted in Figure 3.12 and explained underneath.

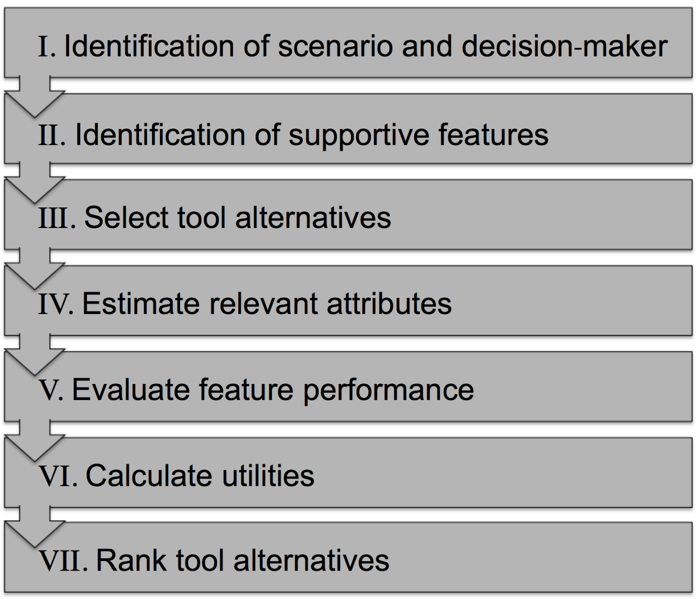


FIGURE 3.12: Seven steps of the evaluation approach.

3.5.1.1 *Step I – Identify the Scenario and Decision-maker*

The starting point of every investigation is to define the underlying conditions. In this case, the scenario respective to the underlying activity/task model and the experience level of the decision-maker has to be defined. The task model describes requirements and linked information of the activity which are used as evaluation criteria in the decision-making approach.

3.5.1.2 Step II – Identification of Supportive Features

Little experience in the design or usage of the desired system can easily lead to the dismissal of important requirements. Having detailed information of the task,it is possible to identify which features are crucial and need to be supported by the system. Analogous to the "needs" section in the HoQ matrix, subdividing requirements into sub-attributes makes the decision-maker aware of the requirements' scope. It is expedient to import predefined task models, including requirements, to assist decision-makers to identify supportive features.

3.5.1.3 Step III – Select Tool Alternatives

To evaluate and choose the most promising option, alternative frameworks have to be detected. The identification of the features directs the decision-maker in finding tools intended to support the tasks and features. This initial selection is mainly performed with the use of literature reviews, product specifications, or recommendations of friends and colleagues.

3.5.1.4 Step IV – Estimate Relevant Attributes

Not all criteria that are imported within a task model are equally important in the specific scenario. The eHoQ method is used to classify attributes and sub-attributes. The visual representation of the sub-attributes assists the decision-maker in performing a weighting, on a scale from 0 to 10, whether the sub-attribute is needed or not in the activity (being 0 not required and 10 strongly required). The remaining sub-attributes are identified as features of the system. AHP and the Entropy method do not consider the importance of the weight of the sub-attributes. Yet, AHP uses a pairwise comparison on a higher level, the level of attributes.

3.5.1.5 Step V – Evaluate Feature Performance

All features (sub-attributes) are rated based on their performance on a scale from 0 to 4 in the actual execution. The sub-attributes rating is normalized to generate one performance value in the attribute level, which enables structuring of the problem to achieve insights of the value distribution on the attribute level. The rating scale is defined as:

0. The feature is not implemented.
1. The feature does not perform as per required expectations.
2. The feature marginally performs as per required expectations.
3. The feature works well. Sometimes exceed expectations.
4. The feature always performs beyond expectations.

3.5.1.6 Step VI – Calculate Utilities

MCDM methodology is used to obtain a meaningful index from multidimensional data to evaluate competing alternatives. The traditional HoQ is a visualization form which helps the user to translate user needs in technical requirements, including a benchmarking "wing" which visualizes a rough benchmark. In the presented approach, eHoQ is used to weight criteria (applied as user needs) and evaluate the performance of the tool in relation to other tools. Therefore, the traditional HoQ is enhanced with a multi criteria utility function for more precise benchmarking to rank several tools. By using the eHoQ matrix, rating and weighting is performed on the sub-attribute level. The ratings are normalized over all options, and multiplied with the relative importance weight prescribed by the decision-maker, as described in section 3.2.2. Computing the normalization on the sub-attribute level leads to a higher value accuracy and reproducible results. The detailed calculation is exemplified in the following sections.

3.5.1.7 Step VII – Rank Tool Alternatives

Computing the utility of every solution provides a clear ranking of the technical alternatives to deal with the MCDM problem. A fulfillment parameter indicates to which degree each option satisfies the requirements. Both parameters are presented and discussed in section 3.2.2.

3.5.2 Applied Evaluation Approach onto Collaborative Authoring

The proposed evaluation approach is applied for use in practice onto a collaboration task specified by collaborative authoring. The task model and the corresponding activity model of collaborative authoring has been already described in section 3.3.2. The approach is performed with two decision makers, with the aim to find the best software solution in order to create a joint written document in a distributed setup. Three different collaboration solutions have been chosen. Each particular functionality is weighted, executed, and joint performance ratings are committed. The evaluation process is described below.

3.5.2.1 Step I – Identification of Scenario and Decision-Maker

The underlying scenario of this investigation is collaborative work and the chosen activity is collaborative authoring. Both decision-makers are experienced in the collaboration task and have knowledge about necessary functionality.

3.5.2.2 Step II – Identification of Supportive Features

A set of evaluation sub-attributes $a_{m,n}$, as stated in section 3.3 and indicated with $n \in 0,\dots,N-1$, where N is quantity of sub-attributes, is categorized in the attributes $\mathbf{a}_m$, indicated with $m \in 0,\dots,M-1$, where M is the quantity of attributes:

1. Content Support
2. Information Sharing
3. Coordination Support

4. Communication Support
5. Content management
6. Usability and User Experience

As described in section 3.3, each of the attributes *communication support* and *compliance support* only contain one sub-attribute for the task model of collaborative authoring. Thus, both attributes are merged into the attribute *communication support*. Also *Usability* and *User Experience* are merged for better readability.

3.5.2.3 Step III – Select Tool Alternatives

Alternate software tools dedicated to the activity of collaborative authoring were selected. For the initial selection, we used literature reviews and chose the following tools: (1) Google Docs, (2) ONLYOFFICE, and (3) Overleaf. Google Docs, which is a word processor and part of a free, web-based software office suite is offered by Google Inc. Originally, Google docs was the combination of two separate products, Writely and Google Spreadsheets. The software suite allows users to create and edit files online while collaborating with other users in real-time. Users are able to access, create, and edit documents "wherever you go" online, but it also offers offline based solutions scaling to different sizes and types of devices. Changes to the document are automatically saved and a versioning service is integrated. Google Docs documents can be easily converted into Microsoft Word files and vice versa. Therefore, an easy transition between tasks and tools can take place. The famous Google search engine can be integrated into the application, which combines the tasks of research and writing. Google Docs is open source and available for free [111].

ONLYOFFICE Online Editor is "the most complete and feature-rich office and productivity suite" made by Ascensio System SIA. The online document editor uses a HTML5 Canvas element, which makes it a complete and feature-rich online office suite, and therefore highly compatible with Microsoft Office and OpenDocument file formats among others. ONLYOFFICE offers a complete productivity suite with document management, project management, CRM, calendar, mail, and corporate network. A high transferability and easy transition between tasks, tools, and users is enabled. Collaboration styles can be adjusted by showing changes instantly or after saving. Thus, "some degree of confidentiality" is offered and it leaves more creativity for the users. Commenting and built-in chat, reviewing and tracking changes are available. ONLYOFFICE is available online and as a desktop application. It is open source and available for free, whereby additional functionality is available for supply by obtaining a chargeable pro-license [16].

Overleaf (OL) is an online collaborative LaTeX editor with integrated real-time preview developed by Writelatex Ltd. The Latex project is compiled next to the code widget to track changes instantly. No additional software apart from a browser has to be installed. A secret link of a created project is used to invite other participants to review, comment, and edit documents. Every user has the latest version as Overleaf synchronizes changes made by all authors transparently. Functionality for commenting, highlighting, and marking important aspects, color coded by users, are supported. Next to the latex editor widget, a rich text mode editor is provided, as not everyone is familiar with Latex. This editor renders headings, formatting, and equations directly into the editor, to appear more familiar to WYSIWYG users.

Writelatex Ltd. uses 256-bit SSL encryption and Amazon S3 storage for highly secure documents. Overleaf is open source and available for free [310].

3.5.2.4 *Step IV – Estimate Relevant Attributes*

The visual representation of the sub-attributes in the HoQ matrix assists the decision-maker to jointly perform a weighting of each identified sub-attribute $a_{m,n}$ on a scale from 0 to 10. For example, the importance of the weighting w_n, with $w_n \in 0,\ldots,10$ of the sub-attributes within *Information Sharing* is estimated in Table 3.9:

Table 3.9: Sub-attribute weighting of *Information Sharing* performed by the decision-makers.

Sub-attribute	Importance
Quickly retrieve context-relevant information	10
Access to shared objects	7
Accentuating	5
Annotate	10
Track others approach	8
Screen sharing	10
Individual and/or shared workspaces	8

3.5.2.5 *Step V – Evaluate Feature Performance*

The decision-maker fills out one questionnaire per tool (o_k, where $k \in 0,\ldots,K-1$ and K is quantity of alternative options) and rates the performance, $p_{n,k}$, (performance rating of $a_{m,n}$ of option o_k and $\in 1,\ldots,4$) of each sub-attribute on a scale from 1 to 4 during an expert discussion. Each sub-attribute is checked for a variety of datasets. If, during the discussion a sub-attribute is identified as not necessary, the importance weight is degraded to 0. For evaluation of technical features, the number of steps, mouse clicks, and speed/time of execution were also taken into consideration. Table 3.10 shows the results of the aggregated performance rating $\mathbf{p}_{m,k}$ of $\mathbf{a}_m$ with option o_k:

Table 3.10: Average performance score $\mathbf{p}_{m,k}$ of attribute per option Google Docs (GD), OPENOFFICE (OO), and Overleaf(OL).

Attribute / Tool	GD	OO	OL
Content Support (CT)	2.33	3.83	2.33
Information Sharing (IS)	1.71	3.00	1.71
Coordination Support (CO)	2.75	3.00	2.75
Communication Support (CS)	3.00	2.85	2.35
Content Management (CM)	2.50	3.00	3.00
Usability & UX (UUX)	2.71	2.28	2.61

Normalization of the ratings enables the preservation of proportionality. Rating one sub-attribute for all options with the performance value 1 leads to a proportional rating of 0.33 for each value, which means that all tools are equally good or bad, respectively. Rating the sub-attribute with a performance value of 2 for all three options leads to the same proportional rating of 0.33 for all options. Although, the absolute rating values in both examples differ, the proportional rating is the same, which means that both attributes in the examples have the same influence on the overall result. The normalization expresses an ex-ante comparison of the options that enables a comparison of results resting upon different bases. The aggregated sub-attribute ratings from Table 4 are then normalized as shown in Table 3.11:

TABLE 3.11: Performance normalization of attributes.

Attribute / Tool	GD	OO	OL
CT (max)	0.274	0.451	0.274
IS (max)	0.266	0.367	0.266
CO (max)	0.324	0.353	0.324
CS (max)	0.366	0.348	0.287
CM (max)	0.294	0.353	0.353
UUX (max)	0.357	0.300	0.343

3.5.2.6 *Step VI – Calculate Utilities*

In order to compare the different MCDM methods as introduced in Chapter 3.1, the application of the Analytic Hierarchy Process (AHP), Entropy method, and eHoQ is presented in detail.

i) Analytic Hierarchy Process.
In order to apply AHP, we first asked the decision-makers to fill out the matrix of pairwise comparisons seen in Table 3.12.

TABLE 3.12: Comparison Matrix of AHP method.

	CT	IS	CO	CS	CM	UUX
CT	1	1/2	1	1	1	1
IS	2	1	1	2	2	2
CO	1	1	1	2	2	2
CS	1	1/2	1/2	1	1	1
CM	1	1/2	1/2	1	1	1
UUX	1	1/2	1/2	1	1	1

The dominant eigenvector q and an inconsistency ratio (IR) are computed. If IR < 10% we accept q, otherwise we ask the decision-maker to review his/her comparisons. The dominant eigenvector of the comparison matrix before normalization is:

$$q = (0.2357 \;\; 0.4102 \;\; 0.3694 \;\; 0.2051 \;\; 0.2051 \;\; 0.2051),$$

In normalization we find:

$$qnorm. = (0.1445 \;\; 0.2515 \;\; 0.2265 \;\; 0.1257 \;\; 0.1257 \;\; 0.1257).$$

The associated dominant eigenvalue $\delta_{max} = 6.0545$ leads to an $IR = 0.879\%$, which is acceptable. The normalized eigenvector is now used for weighting and the utility of each attribute can be calculated as shown in Table 3.13 with:

$$\mathbf{u}_{m,k} = w_m \cdot \mathbf{p}_{m,k} \tag{3.12}$$

TABLE 3.13: Computation of utility, $\mathbf{u}_{m,k}$, with AHP.

Attribute / Tool	GD	OO	OL
CT (max)	0.04	0.0652	0.040
IS (max)	0.07	0.0117	0.067
CO (max)	0.07	0.0800	0.073
CS (max)	0.05	0.0437	0.036
CM (max)	0.04	0.0444	0.044
UUX (max)	0.04	0.0377	0.043

ii) Entropy Method. The Entropy method determines weights without any consideration of the decision-maker's preferences. The entropy variables are calculated:

$$E_m = -s \cdot \sum_{m=0}^{M-1} \mathbf{p}_{m,k} \cdot log(\mathbf{p}_{m,k}) \tag{3.13}$$

$$s = \frac{1}{log(K)} \tag{3.14}$$

$$D_m = 1 - E_m \tag{3.15}$$

$$w_m = \frac{D_m}{\sum_{m=0}^{M-1} D_m} \tag{3.16}$$

Table 3.14 shows the computation of the entropy variables that are used to calculate the utilities of each attribute with Equation 3.12, as shown in Table 3.15:

TABLE 3.14: Calculation of the entropy variables.

Attribute / Tool	E_m	D_m	w_m
CT (max)	0.973	0.027	0.370
IS (max)	0.965	0.035	0.476
CO (max)	0.999	0.001	0.011
CS (max)	0.995	0.005	0.066
CM (max)	0.997	0.003	0.044
UUX (max)	0.997	0.003	0.034

iii) Enhanced House of Quality implemented as MCDM Method. The calculation performed within eHoQ is described in section 3.2.2 in the Equations 3.3 - 3.6. Table 3.16 shows the computation of w_m for the attribute *Information Sharing*:

TABLE 3.15: Computation of utility, $\mathbf{u}_{m,k}$, with Entropy method.

Attribute / Tool	GD	OO	OL
CT (max)	0.101	0.167	0.101
IS (max)	0.127	0.222	0.127
CO (max)	0.003	0.004	0.003
CS (max)	0.024	0.023	0.019
CM (max)	0.013	0.015	0.015
UUX (max)	0.012	0.010	0.012

TABLE 3.16: Relative sub-attribute weighting of *Information Sharing* performed by decision-maker with $\sum w_m = 346$.

Sub-attribute	$w_{m,n}$	$w_{m,n}^{rel.}$
Quickly retrieve context-relevant information	10	0.029
Access to shared objects	7	0.020
Accentuating	5	0.014
Annotate	10	0.029
Track others approach	8	0.023
Screen sharing	10	0.029
Individual and/or shared workspaces	8	0.023
w_m	0.15318	

Table 3.17 shows the normalization of the sub-attributes of Meeting Support and computation of the attributes performance, $\mathbf{p}_{m,k}$, while Table 3.18 summarizes the computed weights, $\mathbf{w}_m$, and the performance, $\mathbf{p}_{m,k}$, of all attributes.

TABLE 3.17: Sub-attribute normalization of attribute *Information Sharing* and computation of the attributes performance $\mathbf{p}_{m,k}$

Sub-attribute	GD		OO		OL	
	$p_{m,n}$	$p_{m,n}^{norm.}$	$p_{m,n}$	$p_{m,n}^{norm.}$	$p_{m,n}$	$p_{m,n}^{norm.}$
Quickly retrieve information	1.00	0.22	2.50	0.56	1.00	0.22
Access to shared objects	1.00	0.20	3.00	0.60	1.00	0.20
Accentuating	1.00	0.20	3.00	0.60	1.00	0.20
Annotate	3.00	0.33	3.00	0.33	3.00	0.33
Track others approach	3.00	0.30	4.00	0.40	3.00	0.30
Screen sharing	1.00	0.17	4.00	0.67	1.00	0.17
Individual/shared workspaces	2.00	0.36	1.50	0.27	2.00	0.36
$p_{m,k}$	0.255		0.489		0.255	

TABLE 3.18: Performance rating of all attributes with HoQ

Attribute	w_m	GD	OO	OL
CT (max)	0.168	0.261	0.478	0.261
IS (max)	0.153	0.255	0.498	0.255
CO (max)	0.214	0.323	0.354	0.323
CS (max)	0.214	0.370	0.349	0.281
CM (max)	0.049	0.294	0.353	0.353
UUX (max)	0.225	0.360	0.298	0.342

3.5.2.7 *Step VII – Rank Tool Alternatives*

A clear ranking, r_k, of the technical alternatives to deal with the problem is indicated and made by measuring the utility, $\mathbf{u}_k$, for the AHP and Entropy method with Equation 3.17 and for HoQ with Equation 3.6.

$$\mathbf{u}_k = \sum_{m=0}^{M-1} u_{m,k} \tag{3.17}$$

Table 3.19 summarizes the results of all MCDM methods and the derived rankings.

Table 3.19: Ranking of the options calculated with all MCDM methods

Method		GD	OO	OL
AHP	u_k	0.3079	0.3885	0.3035
	r_k	2	1	3
Entropy	u_k	0.2808	0.4415	0.2776
	r_k	2	1	3
HoQ	u_k	0.3173	0.3820	0.3006
	r_k	2	1	3
Fulfillment	%	63.62	73.74	60.80

Next to the ranking of the tools, which compares the alternatives against each other, we provide a fulfillment parameter, describing the percentage of the fulfilled requirements of the options compared to maximum achievable values (as discussed within Section 3.2.2 with Equation 3.11). These parameters are calculated based on the normalized performance ratings of sub-attributes per option and the relative weights (Table 3.16) derived from the HoQ methodology.

3.5.3 Comparison of the MCDM Methods and Recommendation for Utilization

As demonstrated, all MCDM methods are applicable in our evaluation approach. However, these methods do not necessarily yield the same results. A strong importance weight of attributes or sub-attributes in AHP or HoQ will yield a different result than the Entropy method, which doesn't consider a weighting performed by the decision-maker at all. Especially if one attribute dominates all other attributes, the result ranking is not corresponding. The Entropy method is suited better if the decision-maker is inexperienced and does not know how to weight the importance of the attributes. While the AHP method is considering a weighting of the higher-level attributes, the eHoQ method requires an importance weighting on the sub-attribute level. The comparison weighting does not have to be performed explicitly by the user using the eHoQ method.

With the AHP method, the only way to emphasize the importance (weight) of one sub-attribute is to make a higher relative ranking of the comprehensive attribute in the comparison matrix, as illustrated in Table 3.20. This implies that all sub-attributes within this attribute acquire a higher weight.

Table 3.20: Impact of users' priority with AHP method.

Strategy	Resulting eigenvector	GD u_k	OO u_k	OL u_k
No weighting; no dominant attribute	$\vec{q} = \begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix}$	0.327	0.345	0.327
Ranking		2	1	2
Third attribute is dominant above all other	$\vec{q} = \begin{pmatrix}1\\1\\2\\1\\1\\1\end{pmatrix}$	0.326	0.343	0.332
Ranking		3	1	2
Fourth attribute is dominant above all other	$\vec{q} = \begin{pmatrix}1\\1\\1\\2\\1\\1\end{pmatrix}$	0.330	0.342	0.328
Ranking		2	1	3

However, changing the weight of one sub-attribute should not lead to a higher value of the embracing attribute and all its sub-attributes. We will demonstrate the impacts for the AHP method and compare the results.

Changing one sub-attribute's weight should not lead to the same result as changing all sub-attribute weights dedicated to one attribute, as shown in Table 3.21.

Table 3.21: Impact of users' priority with HoQ method.

Strategy	Hoq		
Tool	A	B	C
$w_n = w = 1$	0.298	0.307	0.289
Ranking	2	1	3
Use predefined weightings	0.328	0.343	0.329
Ranking	3	1	2
Set one sub-attribute to $w_i = 10$	0.328	0.343	0.329
Ranking	3	1	2
Set one attribute to $w_i = 10$	0.331	0.344	0.325
Ranking	2	1	3

Furthermore, eHoQ as MCDM method considers, that changing one sub-attribute has an impact on, and can of course lead to, a change of the complete ranking, as demonstrated in Table 3.22.

TABLE 3.22: Sub-attributes impact on the overall result

Sub-attribute	w_n	$w_n^{rel.}$	u_{GD}	u_{OO}	u_{OL}
sa_5	2	0.00585	2.5	3.0	4.0
Ranking			2	1	3
sa_5	10	0.02857	2.5	3.0	4.0
Ranking			3	1	2

A higher absolute weight of one sub-attribute leads to a higher relative weight. While in the first example the relatively poor rating of 2.5 didn't have a big influence on the total rating, it does have a big impact in the second example as it leads to a rank loss of the option Google Docs.

These examples show the importance of weighting on the sub-attribute level. On the one hand, weighting on attribute level does lead to an automatic acquirement of higher weights for all sub-attributes leading to misjudgment and miscalculation (overestimation). On the other hand, the other methods do not consider the weight of a sub-attribute at all. Although it can have a big impact on the total calculation, which also leads to misjudgment and miscalculation (underestimation).

The biggest shortfall of our approach is that evaluation criteria have to be defined and weighted by the user. That concludes, that the user is still the biggest influence in the approach and subjective ratings might be incorporated. The import of predefined and weighted criteria lists decreases the users influence. Although our approach needs more evaluation and performance effort, especially for inexperienced decision-makers, it is an adequate technique, which includes the user's knowledge and experience.

3.5.4 Web-based User Interface Facilitating Software Benchmarking

A web based user interface was developed to guide the user through the ranking process. The tool calculates the results based on the eHoQ ranking method. The results are displayed in an easy-to-understand fashion. A recommended selection based on the users' experience level, the computed ranking, and a fulfillment parameter are displayed at the center of the screen. The purpose of the web based user interface is to facilitate performing the evaluation approach presented earlier for the decision-maker. Therefore, the evaluation of software frameworks regarding their usability and applicability, respectively, for a specific task model to postulate a ranking between different observed alternatives. The web-based user interface can also be used to perform interviews, questionnaire-like surveys, or comparative evaluations in a broader manner with the aim to create a ranking between alternatives by replacing the dataset and formulating weights on the fly. To test the web application, we use a dataset of collaborative work, including a catalog of supportive criteria stated in section 3.3 and task models for collaborative authoring, collaborative reviewing, collaborative design, and project management.

The implementation of all described steps in one single view may result in visual clutter, while creating seven distinct views, one for each step, may lead to a tedious experience for users. Therefore, the tool is divided into three sections: (1) Initial Selection; (2) Questionnaire; and (3) Result Display, with semantically complementary steps being treated in the same section.

3.5.4.1 *Section 1 – Initial Selections*

The first view assists the user in selecting the dataset as well as the use case (task model), and provides the functionality to enable expert settings. A short informative description of the task is displayed. Names and quantity of the tool alternatives can be entered (see Figure 3.13).

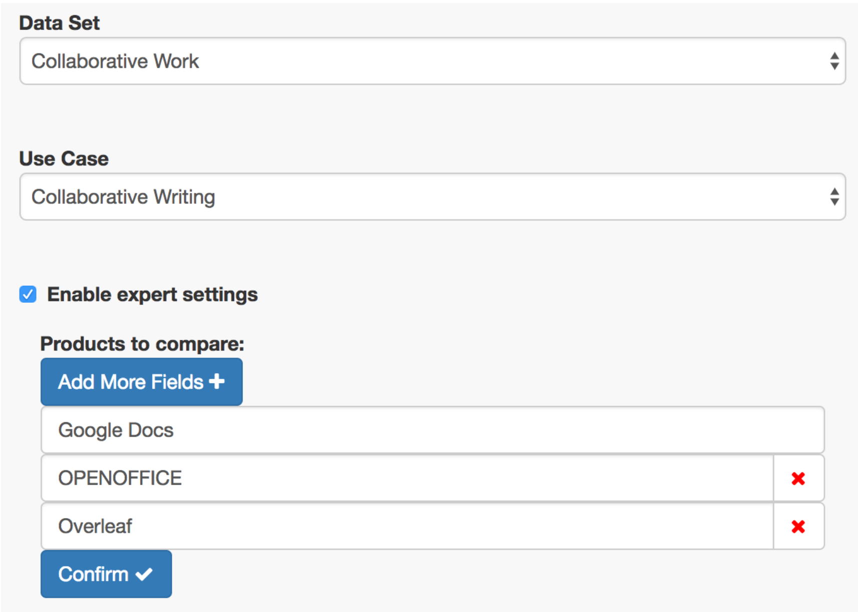


FIGURE 3.13: Widget: initial selection of data set, scenario, and products.

3.5.4.2 *Section 2 – Questionnaire*

In regards to the following steps of the evaluation approach, tools can be classified according their attributes and subordinated features. However, users may not be aware of all of these features, which is why they need to be displayed in an ordered, easy-to-understand structure.

Each feature requires a corresponding importance weighting on a scale from 0 to 10, that dictates how much influence that feature will have on the final score (Step IV). However, inexperienced users may not know the exact importance of features in a specific use case, which is why an array of predefined importance weightings needs to be included. This data needs to be hidden from inexperienced users to not overwhelm them. Assuming users have deemed themselves experienced (expert settings) in the first section, they are able to modify importance weightings with the use of sliders, as seen in Figure 3.14. Importance weightings need to be displayed in a similar fashion as the features, in order to clarify their correlation to each other.

The evaluation of feature performance is realized with the use of star based input UI elements and a clear visual distinction between a bad rating (one star), a good rating (all stars), and the indication, that a feature was not existent (no stars, Figure

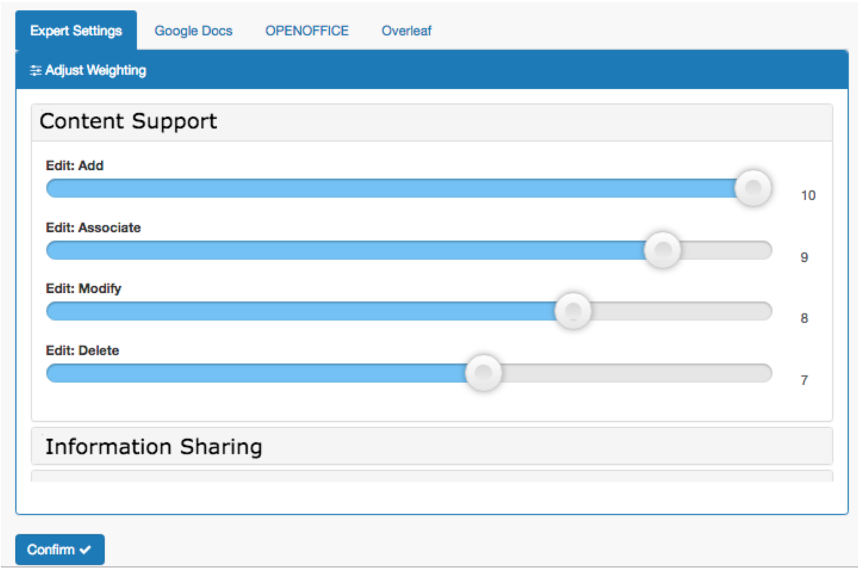


FIGURE 3.14: Weighting questionnaire: view assists to perform weighting and identification of supportive features.

3.15). Tool alternatives are organized in tabs next to each other in that way users only see the rating of one product at a time, which counters comparison bias between the feature rating of the alternatives.

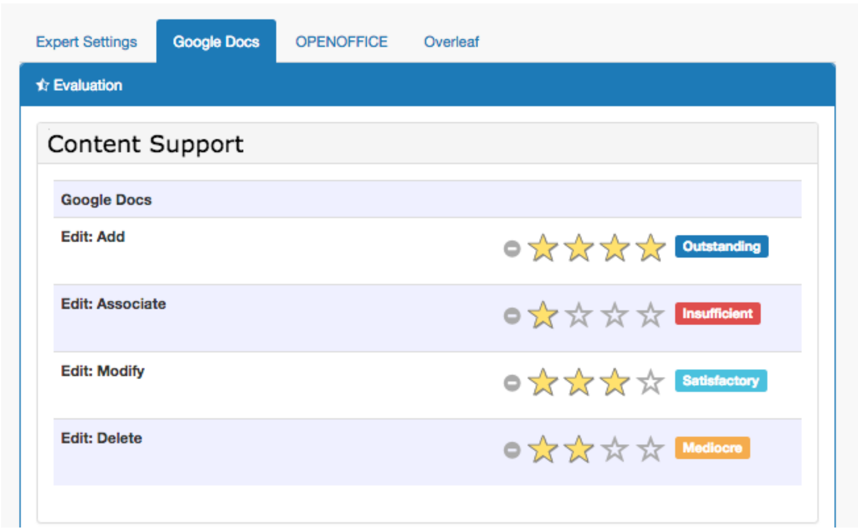


FIGURE 3.15: Rating questionnaire: view assists to perform performance evaluation of the features.

3.5.4.3 *Section 3 – Result Display*

Calculating utility scores manually can be a tedious procedure. Fortunately, this can easily be automated, which is why the tool assumes this task, thereby disburdening users. Users do not need to see the actual calculation process, though, this is automatically performed by the tool. The results are displayed in an easy-to-understand fashion, as seen in Figure 3.16.

The display makes a clear distinction between the alternatives rated and portrays scores in a way that makes them easily comparable, illustrated in a bar chart. An additional total fulfillment percentage represents how many of the features met the users' requirements is depicted in a doughnut chart in the same color as the bar chart.

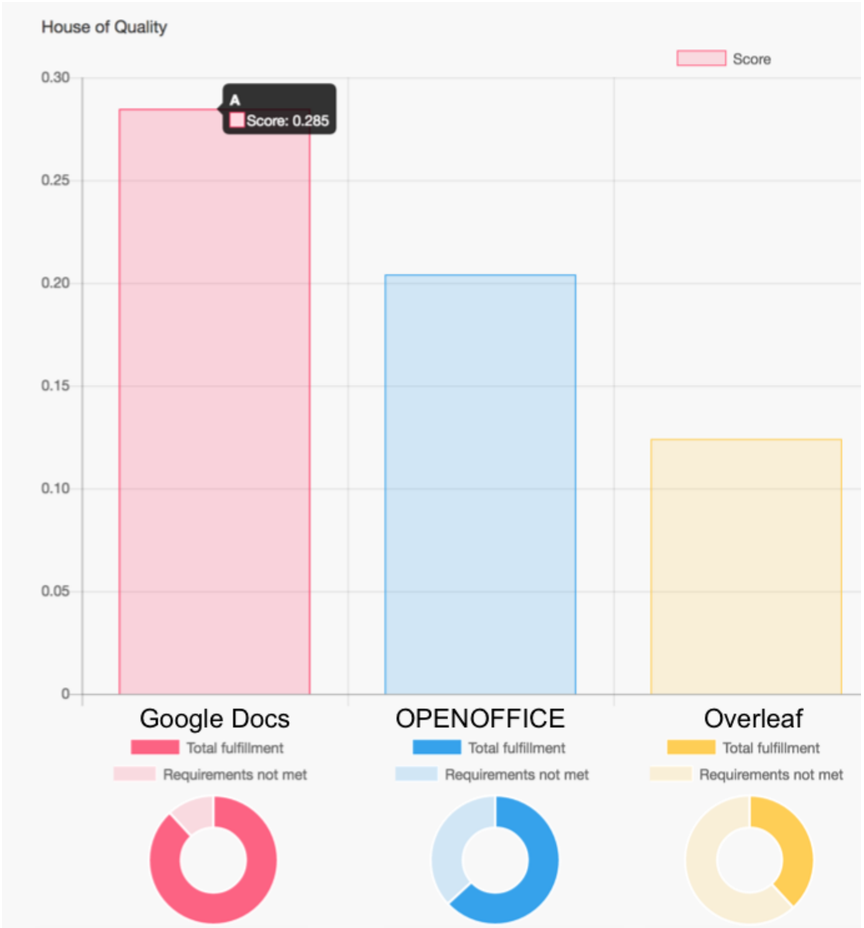


FIGURE 3.16: Resulting rankings.

The web-based user interface consists of three pages that communicate via the HTTP POST request method. PHP is used to process the data conveyed via POST. PHP is also used for dynamic generation of HTML DOM elements. Javascript is used for animation. Bootstrap [222] is used as a framework for CSS and Javascript. Several third-party libraries are used for various design features. Bootstrap Star Rating is used and slightly modified as the star rating feature in the questionnaire section. Rangeslider.js [241] is used as the slider input feature in the expert options section. For the diagrams in the evaluation section, Chart.js [284] is used. UI-Elements are assessed with regards to good usability and high user satisfaction and design principles, where concepts, such as the laws of simplicity [188], are considered.

3.6 Discussion and Summary

In this chapter further knowledge about *the product*, according to the first User-Centered Design principle, is enhanced and refined by examining requirements, components, and providing a framework for collaborative computer-aided decision making. The enhanced House of Quality is presented as a methodology to illuminate collaborative computer-aided decision making. Based on collected user needs, this new approach defines a quality assurance tool used for two cases of application: (1) definition and rating of software components and (2) benchmarking and ranking of software/frameworks. The traditionally HoQ matrix is enhanced with four additional components: (1) a task ontology that serves to translate user needs into the design attributes in a structured way, (2) system attributes' importance rate based on multi-attribute utility theory considering correlations in between, (3) a utility function for precise benchmarking, and (4) a clear ranking between alternative solutions.

First, a generic criteria catalog for collaborative environments was presented and it's use in practice demonstrated on the task model of collaborative authoring. The presented criteria catalog was further used within the eHoQ methodology to identify software architecture modules of new systems. The catalog served as a guideline for evaluating existing collaboration systems.

Second, the first two appendices of eHoQ were presented in order to define system attributes of a software architecture and their value derived from user needs. Thus, we provide a practical method that guide developers to specify, evaluate, and acquire weighted modules to support the design of software or collaboration frameworks and service design. In particular, we enhance the traditional House of Quality with (1) an task ontology based on the methodology of viewpoint-oriented requirements definition combined with task world ontology, that serves to translate user needs into the design attributes in a structured way, and (2) the computation of system attributes' importance rate based on multi-attribute utility theory considering correlations. With the presented ontology translating user needs and defining system attributes is simplified and follows a clear scheme. As demonstrated, software architecture design is easily detectable and importance of system attributes are computable. Thus, designers or developers can easily identify which components/-modules/system attributes have to be developed and with which priorities, in order to enhance user satisfaction, quality of the product, and improvements of the design process. The method was successfully applied to a use case of collaborative complex system designof an aircraft vehicle system proving the usability of the methodology in practice.

Third, the second two appendices of eHoQ were utilized in a systematic evaluation approach based on integration of eHoQ as multi criteria decision making (MCDM) method with multi-attribute utility theory (MAUT). It is shown that combining eHoQ and MAUT incorporates tasks as well as different user requirements, which provides comparisons at the subcategory level, resulting in a more accurate evaluation of alternative collaboration solutions. The presented evaluation approach was used in practice by benchmarking three different collaborative software environments, namely Google Docs, ONLYOFFICE Online Editor, and Overleaf. It is observed that the MCDM methods AHP, Entropy, and eHoQ provide attributes' weights, which are further used in an additive form of utility function to compute priorities leading to a clear ranking of the collaboration solutions. A demonstrative execution of the evaluation approach that supports users in selecting the most promising tool for collaborative authoring was demonstrated along with the use of the earlier deployed criteria catalog of the task model collaborative authoring. Using the presented eHoQ method solves the stated MCDM problem and the performance is effectively assessed. The eHoQ method observes users' knowledge and experience, which leads to a facilitated decision-making process and reproducible results. We implemented the presented evaluation approach in a widget-based application that allows for the performance of sophisticated computations and narrows the complexity of the evaluation approach to less extensive steps. Thus, providing the utilization of the eHoq methodology in a user facilitating framework for collaborative computer-aided decision making. Summarizing, those investigations answer the question of how to design and evaluate a *product* that enables and supports collaborative computer-aided decision making.

Contents of this chapter have lead to the following publications:

F. Rupprecht, V. Soni, C. Schmidt, B. Ravani, A. Ebert, G. van der Veer, "An Approach for Evaluating Collaborative Software Environments Based on Integration of House of Quality with Multi-Attribute Utility Theory", *The 9th International Congress on Ultra Modern Telecommunications and Control Systems - ICUMT*, 2017.

F. Rupprecht, T. Khan, G. van der Veer, A. Ebert, "Criteria Catalogue for Collaborative Environments", *31st British Human Computer Interaction: Digital Make Believe 2017*, Sunderland, UK, 3-7 July, 2017.

Recently Submitted:

F. Rupprecht, B. Ravani, A. Ebert, "Design of a Collaboration Framework Based on an Enhanced House of Quality Methodology", at *Journal of Research in Engineering Design*, Springer Journals, status: in review.

Chapter 4

Collaborative Design and Assessment in Virtual Reality

In Chapter 2 the domain of collaborative work and decision making was inspected, while in Chapter 3 collaborative computer-aided design was examined. Following the User Centered Design principles, the knowledge of *the product* is compiled in a structured way and serves as foundation for further investigations. Hence, both chapters were needed to define the requirements and scope of such a system. As such, the theoretical background of the technology is provided.

Considering that the role of humans inside the system is still not investigated as well as their scope of influence within the intended framework. It is essential to identify the activities that need to be supported and how the interaction capabilities need to be designed. In order to develop a product of high quality, the single most critical activity is to know and to understand the (potential) users [77]. Starting point is the detailed description of the users' characteristics, attributes, and synergies. Scenarios describe the users' tasks and the daily (aimed) utilization of the technology. Hence, scenarios are used to test the system and to develop functionality into the system that users will actually want to use.

This chapter focuses on *the user* by centering the human into the design process for improving and realizing collaborative design and assessment in virtual reality. The aim of this chapter is the deployment and evaluation of novel, natural, and intuitive interaction and visualization techniques in order to support multiple diverse decision makers in collaborative design and assessment in virtual reality.

First, a brief review of existing systems for collaborative design and decision making is provided including background of related aspects of this investigation. Following, a review of design objectives for collaborative design and assessment in virtual reality. Afterwards, a network of user synergies and inferdependencies is employed in order to identify potential for improving the efficiency of the cooperation.

According to the identified attributes of such a network, two dedicated user groups and associated activities with a high level of detail will be modeled and explored. Those user-models serve as scenarios for the development of novel and intuitive interaction and visualization techniques of a collaborative design and assessment system.

Following, a prototype of such a system is deployed, integrating a multi-modal interface and ubiquitous technology that supports multiple diverse decision makers. As such, a setup is designed that increases the efficiency of collaborative design and assessment in virtual reality. Novel adequate visualization techniques will be applied and proven in a collaborative task. Subsequently, we will focus on the aspect of active interaction and participation. Novel, natural, and intuitive interaction techniques are developed. Thus, multi-modal interaction capabilities, centering the user in the investigations, are deployed and evaluated.

4.1 Related Aspects and Work for Collaborative Design and Assessment in Virtual Reality

In this section, the related aspects of this chapter are introduced. Thus, the following subjects are explored: (1) supporting awareness of user and task dependencies, (2) smart devices as interaction devices in VR environments, (3) gesture recognizer based on smart device motion sensors, and (4) smart devices used as multi-modal interfaces.

4.1.1 Supporting Awareness of User and Task Dependencies

Joint activity is exemplified by Johnson et al. [148]: "In joint activity, individual participants share an obligation to coordinate, to a degree sacrificing their individual autonomy in the service of progress toward group goals." In contrast, coordination is defined as "managing dependencies between activities" [190]. Within interdependent activities, conflicting interests are present. In order to achieve common goals, conflicting interests have to be coordinated to capture discrepancies before they become problematic with the help of common grounds [166]. Those are supported by continuously informing others about changes that have occurred outside their views [148]. Furthermore, interesting findings have been explored by Johnson et al. who stated, that not all team members must be fully aware of the entire scope of an activity; but every participant needs to be aware of the interdependence in-between their activities. The awareness of tasks and performed activities within a collaborative work session influences the coordination and performance of tasks in a positive manner. Due to established common knowledge and impact awareness, team members can work together effectively and adjust their activities as necessary [225]. Van der Veer et al. [296] determined, that a high level of detail in task modeling is needed for collaborative work in order to design activity assignments and optimal support by the system. Similar research is being pursued in the field of robotics, where monitoring processes only performed on an overview level is crucial. One well-known example is supervisory control, where a user allocates tasks to machines and monitors execution performance [262, 150]. One solution providing the desired insight is via an additional display monitoring the current status of a process [54].

The "visual information-seeking mantra" of Ben Shneiderman [264] states: "Overview first, zoom and filter, then details on demand". In the spirit of this mantra, overview-and-detail-view techniques are widely used and supported today by mobile devices [239, 56, 58]. These techniques also imply challenges for system design. An "O+D interface" (overview-plus-detail interface) is related to coordinated views, implemented with one small overview provided on top of a larger detail view. An implication pointed out by Burigat [57] is the fact that O+D interfaces on mobile devices do not provide advantages in terms of navigation performance, compared to traditional presentation techniques. Considering small displays, users perceive O+D interfaces as being detrimental. Chittaro [68] determined that O+D techniques tend to fail on mobile devices, as it becomes more difficult to relate two different views due to limited screen space. He suggested to use visual references pointing to interesting parts outside the visualization area, or intuitive methods supporting the switching between parts of the visualization. As an alternative to the O+D technique, Pelurson and Nigay [227] introduced a "bifocal view" as a focus-and-context technique for mobile devices. Myers [209] introduced semantic snarfing, where a region of interest is tracked via pointing devices and copied to a secondary hand-held device. Baumgärtner et al. [26] presented a hybrid 2D+3D interface for visual data exploration that combines visual design techniques with mixed-mode interaction capabilities, demonstrated for document management.

4.1.2 Smart Devices as Interaction Device in VR Environments

The Pittsburgh Pebbles PDA Project [59] was one of the first projects using mobile devices as remote controllers for PCs. Borchers et al. [42] demonstrated a software framework that integrates ubiquitous technologies to support collaborative work on large-scale devices. Lee et al. [178] detected that collaborative virtual environments have the potential to improve collaborative work but still lack sufficient communication capability for distributed teams. *SourceVis* is a collaborative visualization system for co-located environments based on multi-touch tables developed by Anslow et al. [7]. Though, the number of active users is strongly limited. Myers [209] introduced semantic snarfing. Here, latest smartphone technologies can lead to a more natural and intuitive effect of semantic snarfing.

Other framework approaches such as *Munin* [17] focus on solving the data exchange problem of application and communication data between ubiquitous devices. Even when they provide a software framework used for ubiquitous analytics and visualization, the domain oriented multilevel perspective is not fully addressed. Marquardt et al. [193] demonstrated that information exchange between multiple users via mobile devices as input and output devices can be facilitated in support of collaborative work. With the use of a public screen and several mobile devices, the transfer of artifacts between differently scaled devices is supported. Awareness of participants and accessible content are emphasized. However, the performed task in the discussed setup is the same for all participants. Versatility and design space with cross-device interaction using hand-held devices was investigated by Marquardt et al. [194]. Based on micro-mobility and F-formations, natural conversation in collaborations is facilitated in addition to content exchange and cross-device interaction between hand-held devices. However, there clearly exists a need to consider additional aspects of collaborative settings.

Mendes et al. [199] introduced CEDAR, a design review tool supporting collaborative tasks using a Cave Automatic Virtual Environment (CAVE) and hand-held devices. The hand-held devices are acting as independent clients that are applied to the same scene as shown in the CAVE system. The CAVE-system is controlled by gesture tracking (Microsoft Kinect [201]). The Apple iPad acts as independent application, and the device's display shows a first-person view of the scene that is synchronized with the large screen. While the presented tracking setup only supports one active user, the iPad configuration is scalable to a multi-user setting. However, executing tasks cooperatively with the CEDAR system is not supported. Simultaneous work performed by several users is not considered. Hühn et al. [138] pointed out the lack of evaluation tools for pervasive applications. Their CAVE-Smartphone setup was used to evaluate the User Experience (UX) of a location-based advertising application, where a smart phone is used as an alert mechanism.

Smart devices offer a broader range of interaction capabilities, which are not fully exploited in this framework. Anslow et al. [7] developed *SourceVis*, a collaborative visualization system for co-located environments based on multi-touch tables. The table provides a horizontal display on which one viewport per user is created on opposite sides. Single tasks can be performed on the individual's viewport and collaboration is possible due to the same-location setting. The number of active users is limited to the size of the table. Similar to the work of Borchers et al. [42], Finke et al. [100] extended an interactive large public display with small devices. User interfaces are distributed across the differently scaled devices, and one can take advantage of the input and output capabilities of both devices. Unfortunately, only single user interaction was considered.

Keefe et al. [157] combined a hand-held multi-touch device with six degrees of freedom with a large-scale visualization display. The interaction with a large display is improved, and group work tasks can be performed. Single-task performance and integration is not covered. Seifert et al. [256] introduced *MobiSurf*, integrating interactive surface capabilities and information exchange for team members' personal and mobile devices, and supporting co-located collaborative tasks. Cooperative task execution is currently not possible and work done simultaneously by several users involving multi-role perspectives is not considered.

Cooperative task execution with the systems, as discussed in these papers, is not provided. Simultaneous activities performed by several users and multi-role perspectives are not covered. Existing frameworks assume that all participants have the same view on the data or the same tasks to perform. Multivariate data requires the observation of the data in various ways, keeping in mind that diverse tasks are performed by experts. Existing systems do not fully support the collaboration of different user groups with different foci, different tasks, and different privileges. Our system focuses on supporting the active collaboration where smart devices are remotely controlled, see [172], and consider independent application clients.

A user's sense of immersion in VR environments, the perception of being physically present in a non-physical world, increases when the used devices are efficient, intuitive, and as "natural" as possible. The most natural and intuitive way to interact with data in a VR environment is to perform the actual real-world interaction [169]. For example, gamers are typically clicking the same mouse button to swing a sword in different directions. However, the natural interaction to swing a sword in a VR application is to actually swing the arm in the physically correct direction as the sword is an extension of the user's arm. Therefore, intuitive and natural interaction techniques for VR applications can be achieved by using the human body as an input device [18].

Common technologies – like a flight stick, 3D mouse, or 3D controller with joystick and buttons – do not support body movement gestures and require the investment of a significant amount of time for learning. A DataGlove supports the detection of finger movements, position tracking via body suits with in-sewed trackers. Exo-skeletons even make possible full body tracking, but restricting a user when performing interactions, as the user is tethered to the system, cannot walk around, and might fear to damage hardware [189]. Position tracking done with 3D cameras is relatively cheaper, but it restricts a user's natural behavior as the tracking area is limited and the user must face the camera to avoid occlusion. VR devices are usually specialized to support one interaction modality used only in VR environments. Substantial research has been done in this field, yet VR input devices still lack highly desirable intuitive, natural, and multi-modal interaction capabilities, offered at reasonable, low cost.

4.1.3 Gesture Recognition Based on Smart Device Motion Sensors

Current research covers many aspects of interaction in VRs, that is of importance to our work. Bergé et al. [31] stated that mid-air hand and mid-air phone gestures perform better than touchscreen input implying that users were able to perform the tasks without training. Tregillus et al. [289] affirmed that walking-in-place as a natural and immersive way to navigate in VR potentially reduce VRISE (Virtual reality induced symptoms and effects [260]) but they also address difficulties that come along with the implementation of this interaction technique. Freeman et al. [104] addressed the issue of missing awareness of the physical space when performing in-air gestures with a multi-modal feedback system. In order to overcome the lack of current display touch sensors and to equip a user with further input manipulators, Wilkinson et al. utilized wrist-worn motion sensors as additional input devices [309]. Driven by the limited input space of common smart watches, the design of non-touchscreen gestures is examined [15]. Houben et al. [131] prototyped cross-device applications with a focus on smartwatches. In their work, they provided a toolkit to accelerate application development process by using hardware emulations and a UI framework.

Similar to this work, several investigations concerning interaction techniques with wrist-worn devices such as smartwatches and fitness trackers have been made. In general, two types of recognizing techniques can be differentiated: (1) machine learning techniques base on a (high) number of training samples from which features are extracted and gestures identified with the use of probability classifiers and (2) simple pattern recognition with predefined features. Mace et al. [187] compared naive Bayesian classification with feature separability weighting against dynamic time warping. The extremely differing gesture types (circle, figure eight, square, and star) could be recognized with an average accuracy of 97% for the feature separability weighted Bayesian Classifier, and 95% for the dynamic time warping with only five gesture samples. Mänyjärvi et al. [192] presented a hidden Markov model in order to define continuous gesture recognition for primitive gestures used to remotely control a DVD player. Their model could reach an accuracy value of 90-95%.

The investigation by Schlomer et al. performed in [246] employs a hidden Markov model for user dependent gesture recognition. The models are used for training and recognition of user-chosen gestures performed with a wii controller. Only few gestures are tested, which differ extremely. Shortcomings of high computational power are mentioned. Compared to the wii remote control [99], the smartwatch data does not show comparable high peaks. Therefore, the model is not suitable for the underlying kind of data. Methods from machine learning have a high flexibility and find application especially at end user side, when users do not know the data nor are able to identify features. These techniques can lead to excellent accuracy rates with an exceeding number of training samples. However, classifying the training data and identifying the correct gesture with machine learning techniques are resource-intensive. Considering, the less computation power of a smartwatch, machine learning techniques are not applicable in our investigations. Work done in

[311] presents a frame-based feature extraction stage to accelerometer-based gesture recognition performed with the wii remote control. A gesture descriptor combining spectral features and temporal features is presented. However, the recognition starting point is activated with a mouse click and not only due to the recognition. Chu et al. [70] used a fuzzy control technique to classify different gestures based on acceleration data from smartphones. The gestures defined in the study from Chu et al. are totally different, therefore, it would be interesting if there is still a precision rate of 91% with gestures which are similar to each other.

4.1.4 Smart Devices Used as Multi-Modal Interfaces

Several studies have been performed in investigating and enhancing the interaction capabilities of smart watches. Work performed by Vlaenderen et al. [294] is using the smartwatch camera to provide enhanced input capabilities for the smartwatch whereby three different types of text input mechanisms are recognized. Migratory interfaces described in [33] enable users to switch between devices while seamlessly continuing their ongoing work. Another multi-modal system is described by Blumendorf et al. [38], consisting of smart devices and large displays, where the devices' interfaces dynamically adapt to new contexts. Speech and gesture recognition as input modalities are one of the most natural ways to interact with a system. The advantages of speech regarding to Bernsen [32] can be cited as following: 1) it is natural and so, people communicate as they normally do; 2) it is fast (150-250 word per minute); 3) it requires no visual attention; and d) it does not require the use of hands. Almeida et al. [6] combined in one use case speech input and in air gestures, tracked with the Microsoft Kinect, in order to provide multi-modal interaction for enhanced user experience and usability in a news reader scenario.

The fusion of multiple modalities of interaction has been the focus of several research studies, one of the earliest project to showcase these concept is the "Put That There" (PTT) project [39], which swiftly combines gesture, and speech as input modality for communicating with large displays. This showcases a way of man machine interaction where users can point to space while addressing the computer with speech (and not typed symbols) to perform actions in relation to the referent-space[39]. Although our research work also fuses multiple modalities as did PTT, this research did not implement a pointing gesture, for obvious reasons.

Our approach incorporates tilt gestures alongside the speech recognition. Tilt gesture as a means of interaction has been used in several research studies. One of the earliest being [237], which highlights the difficulties of sensing motion in comparison with rotation and evaluates the suitability for one handed usage of portable devices. Another investigation in [119] concerns the usage of tilt for interaction. Although both papers were focused on interaction using tilting; their work focuses on interacting with menus displayed on the device providing the tilt data (i.e. navigating menus on the watch), which slightly differs from our aim of interacting with a large remote display devices. A similar project in terms of aim is Tilt and Touch [317]. This project explores gesture based interaction with a 3d environment, rendered on a secondary display like a projector or monitor. However, gesture data was provided by a mobile phone and not directly extracted by the smart watch. Most similar to the work performed here is Unified Remote [1]. It recently added support for smartwatches. However, its functionality is limited to just tilt input and it lacks speech support. Speech as an input modality in a multi-modal interaction system can be seen in several other projects, e.g. MATCH [151], which provides a navigation system with multiple interaction modalities such as touch, speech, and pen input.

4.2 Design Objectives for Collaborative Design and Assessment in Virtual Reality

In order to support multiple diverse decision makers in the collaborative design and assessment in virtual reality, a system is designed, which basically consists of three components:

1. World simulation,
2. Output device,
3. Input device.

The first component, the **world simulation**, is described by scenes, i.e., a realization of real world scenarios, deduced from the users' descriptions. The second component, the **output device**, serves to embody the virtual world with adequate sensory output. The optimal objective is to provide the same amount of sensory output in the virtual world like in the real world in order to enable immersion, which affects the perception of being physically present in the virtual world. **Input device**, as the third component, is used to gather the sensorial information of user interactions, which are further sent to the world simulation. This incoming information executes the corresponding scene manipulation of the world simulation. The optimal objective is to ensure the user's ability to act naturally like in the real world by providing adequate large number of actions to support natural behavior and immersion.

The first design objective is to understand the scope of users' diversities in order to model dedicated users and activities with a high level of detail, including scenarios and aimed functionality for the world simulation. The second design objective is the design and prototypical realization of a collaborative design and assessment system focusing on multi-modal input- and output devices for multiple diverse decision makers. The gathered findings serve as reference for the combined proof-of-concept system IN²CO that will be presented in Chapter 5.

4.2.1 Understanding the Scope of Users' Diversities

In a collaboration process, team members often switch between tasks to achieve successful cooperation. Changes, performed by team members, affect the entire system and individual performed tasks. According to Johnson et al. [149] it is necessary that team-members understand this impact and adjust tasks and operations accordingly. Therefore, the availability of a common framework to support the decision-making process is highly demanded. Ideally, experts can work independently or jointly on certain aspects/subsystems – still focusing on the design goal that must be achieved. The system supports the entire process by visualizing connections, dependencies, or system changes. More and more activities can be delegated to the system, but it is crucial to identify which activities and tasks are performed in the collaborative work and who performs those activities. Detailed task-models under the rules of user centered design and dependency graphs need to be modeled and visualized to enable the data exchange and to consider the dependencies between different users.

Therefore, a user taxonomy has to be deployed and is used as the foundation to describe those task models. Sample visualizations of a chosen dataset are shown in Figure 4.1. Based on the task models and dependency graph, explicit visualization and interaction techniques can be created and integrated into a virtual collaboration environment.

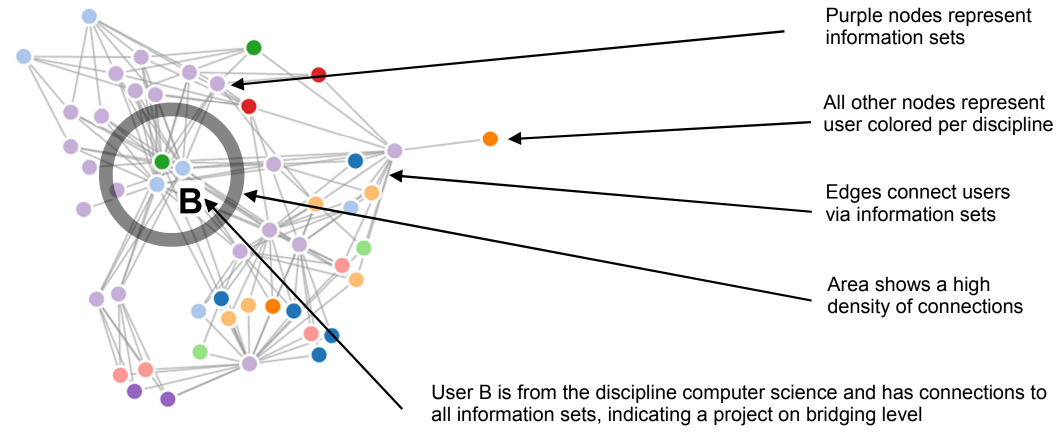


FIGURE 4.1: Dependency graph of a dataset, representing projects within the IRTG 2057 college [164], visualized with the use of a force-directed graph. The interactive 2-dimensional layout uses draggable nodes to facilitate the recognition of existing connections.

A user taxonomy, used to clearly define entities and relationships in a system in order to model a network of their relations is introduced. Afterwards, a set of dependency parameters is proposed, to display a meaningful relation between objects and to give an easy understandable overview of all relationships. Thus, aiming to solve complex tasks and improve a groups' performance. Following, several methods that give the possibility to visualize class data and show relations, connections and interdependencies of group members will be introduced and analyzed. The visualization techniques used in this investigation are: Force directed graphs, Sankey diagrams, and ring based radial visualization with hierarchical edge bundling. Those differing visualization techniques are applied to make a complex system architecture easily understandable for users, fading out unnecessary information, and illustrated in a case study. The given data of the case study, containing different user groups that are acting in a virtual, industrial corporation, is related to the IRTG 2057 college, see Figure 4.1. The system itself is represented by dependency parameters, using three different types of visualization. The representations are embedded in an interactive application, which allows users to explore the recognized dependencies and links.

The findings and gathered information of the investigation are used to model dedicated users and activities with a high level of detail. Together with the deducible dependency graph, explicit visualization and interaction techniques can be created.

4.2.2 Input- and Output Devices for Multiple Diverse Decision Makers

After inferdependencies and synergies between users are explored and aimed scenarios and functionality are established, a system combining those multiple diverse decision makers with the simulation world is needed. Providing one user with adequate visualization and interaction capabilities in VR environments is challenging as most input devices incorporate special hardware which is cumbersome to use and extremely expensive. It is even more challenging to support multiple users. Common systems like tracking systems or flight sticks are not designed for multiple users. Thus, there is a need for such a system. Smart devices have the potential to increase the number of interaction capabilities by providing intuitive input mechanisms and simultaneously scale with the number of users.

Rapid advances in computer networks and visualization-based human-computer interaction technologies are impacting a large spectrum of graphics-based design-simulations. For instance, conceptual design and critical assessment of complex systems generally requires large teams of scientists, engineers and planners to work together. The conceptual design process is extremely time-consuming, typically involving several iterations of different solutions before a generally acceptable solution is obtained. The collaborative framework presented in Chapter 5 is aimed at pointing out the efficiency gained when bringing diverse areas of expertise together. Examples where next-generation network-enabled collaborative environments, connected by visual and mobile interaction devices, can have significant impact are: design and simulation of automobiles [123] and aircraft [205]; urban planning and simulation of urban infrastructure (e.g., transportation, electricity, water, and communication grids)[165]; or design of complex and large buildings, including efficiency- and cost-optimized manufacturing buildings [198]. The conceptual design and simulation-based evaluation of a new aircraft requires a manufacturer to bring together experts from mechanical engineering, electrical engineering, computer engineering, ergonomics, material science, air quality, health, and even more fields. Team-members often have to switch between tasks to achieve successful collaboration [254]. When members make changes, they affect the entire system and individually performed tasks. It is important to understand this impact and adjust tasks and operations accordingly [150]. The desire for a common framework to support decision-making in this process was a main motivation for our effort.

We introduce a framework for distributed or co-located teams to collaborate efficiently using diverse mobile smart devices for the design and assessment of complex systems. Our framework enhances the efficiency of collaborations arising in design, simulation, or data analysis including visualization. Based on the requirements stated in Chapter 2.2, a prototypical framework taking into account the influences between the deduced task models is designed and implemented. The devices provide three views of data to be processed collaboratively: (1) a simulation view; (2) a status report view; and (3) a status update view. These views serve the purpose of providing overview, detail, and performance views. A distributed viewport on a smartwatch view shows at-a-glance information, the environment or the process being inspected, possibly influenced by a user. Users can use their mobile devices as control interfaces. The framework is especially effective for combining the synergistic, complementary competencies of a team. The design of the framework is presented and specific applications are discussed.

Traditionally, the most common way to interact with large display devices has been through the keyboard/mouse interaction model. However, more recently, there has been an increasing adoption of natural communication such as speech, touch, or gesture (captured using sensors) for interacting with digital devices. Large display devices usually do not come with embedded sensors required for tracking natural interaction. Hence, users have to purchase dedicated devices, in order to make use of these modern interaction means, which are more often than not, useless outside the reason for which they were purchased.

Mobile devices are almost ambiguous today and feature a wide range of input and output capabilities like touch screens, cameras, accelerometer, microphones, speakers, near-field communication, Wi-Fi, etc. The usage of smart-devices is easy and intuitive, and they offer a wide range of interaction metaphors, which can lead to a more natural and intuitive interaction as well as a broad array of control elements. Especially the smartwatches, as the latest technology in that field, brings new possibilities of interaction techniques. As the watch is fixed on the wrist, the hands are free and this leads to a more natural interaction in the meaning of body gestures. Additionally, other technology like finger tracking can be combined and new interaction techniques can be enabled. Next to the common touch gestures which are performed very frequently on the smart device's display, we developed additional movement gestures which enriches the input capabilities of the smartwatch significantly.

The initial designed multi-modal Interface is combined with multi-modal interaction capabilities realized by smart devices: (1) touch, (2) in-air gestures, and (3) speech. Thus, a multi-modal interaction interface is presented, designed for smartwatches and smartphones for VR environments. The presented research was driven by the goal to enable users to invoke actions with their body physically, causing the correct corresponding action of the VR environment. A system that tracks a user's movements recognized as specific gestures is introduced. Smartwatches are promising new devices enabling new modes of interaction. They can support natural, hands-free interaction. The presented effort is concerned with the replacement of common touch input gestures with body movement gestures. In particular, missing or insufficiently precise sensor data are a challenge, e.g., gyroscope and magnetometer data. This data is needed, together with acceleration data, to compute orientation and motion of the device. A transformation of recorded smartwatch data to arm movement gestures is introduced, involving data smoothing and gesture state machines. The combination of (1) touch, (2) in-air gestures, and (3) speech enhances the efficiency of interaction in virtual worlds in a natural and intuitive way.

4.3 Identification of Users' Inferdependencies and Synergies

The overall goal of this section is to prepare a given dataset in a way that dependencies and activities are extracted and become recognizable. Initially independent projects are examined and overlapping (dependencies) can be found. First, we introduce a user taxonomy that describes users in a domain considering different aspects and characteristics. Based on the taxonomy, users and activities can be correspondingly modeled with a high level of detail as required for user centered design. Afterwards, dependency parameters are introduced in order to quantify the synergies and inferdependecies between users. The proposed taxonomy and the dependency parameters are applied and visualized with the use of different visualization techniques embedded in a visualization exploration framework.

The visualization techniques used in this framework are: Force directed graphs [108], Sankey diagrams [238], and ring based radial visualization [87] with hierarchical edge bundling [127]. Force directed graphs have the attributes of evenly distributed vertices, edges with uniform lengths, and reflecting in order to create higher level of aesthetic and better readability for the user [108]. This approach has been applied in numerous cases and served as basis for more enhanced visualizations like in [128, 94, 135, 137]. Traditionally, Sankey diagrams are applied to visualize energy flows or material flows. Those diagrams represent quantitative information about flows, their relationships, and their transformation depicted as directed, weighted graphs [238]. The possibly most famous Sankey diagram is Charles Minard's Map of Napoleon's Russian Campaign of 1812, which was created in 1869, even before the actual Sankey diagram has been introduced in 1898 [290]. Sankey diagrams are applied in a wide range of fields, e.g. [248, 134, 82]. The third visualization technique used in the visualization exploration framework is ring based radial visualization with hierarchical edge bundling as characterized by [87]: nodes are positioned around the circumference of a ring; line segments (edges) are used to connect the nodes; additional nodes optionally appear in ring's interior. Ring based radial visualizations are commonly used to depict relationships among disparate entities. The edge bundling is used to reduce the visual complexity [127]. Radial visualizations are widely used and enhanced by e.g. [288, 127, 30].

4.3.1 User Taxonomy in Industrial Corporations

Industrial corporations rely on interdisciplinary collaboration of integrated complex design and decision-making. The conceptual design and critical assessment of complex systems generally requires large teams of scientists, engineers, and planners who collaborate, bringing together different aspects and knowledge from different areas. The conceptual design process is extremely time-consuming, typically involving several iteration steps to improve conceptual designs before a generally acceptable solution is obtained. In this taxonomy, users in industrial corporations are described based on three categories: system-level, discipline, and task. These categories are explained in the following sections.

4.3.1.1 System-Level

Industrial corporations can be adapted to a layer model, where every layer is considered as one subsystem that is in an exchange relationship with the neighboring levels. Design decisions on the lowest level escalate to the highest level and vice versa. The following levels can be distinguished as shown in Figure 4.2: factory level, machine level, process level, and the bridging level, which combines all aspects and brings those levels together. This level model is related to the whole IRTG 2057 [164] structure, where every participant is assigned to one of those given levels.

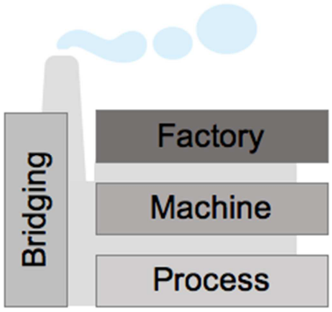


FIGURE 4.2: System-levels in industrial corporations related to the structure of the IRTG 2057 college with the title "Physical Modeling for Virtual Manufacturing Systems and Processes" for the successful realization of activities and tasks in an analogically industrial project.

The factory level comprises physical properties, features of the factory, and of the manufactured products. For example, factory-level transactions describe the material flow within the whole manufacturing process or conditions of the indoor environment. On the machine level the focus lies on machine tools and their components, tooling systems, and measuring instruments. Machine-level transactions specify manufacturing processes on the workbench level and include physical characteristics like deformation, stress, and temperature. On the process level, single machining processes (e.g., cutting, grinding, and milling) are considered. Material properties as well as material behavior of the work piece and the machining tools under machining conditions are investigated. Changes made on the lower levels have an influence on the overall production program, output, and quality to a higher degree. The exchange of transactions between levels and the connections between each other as well as the consideration of cross-references is investigated and developed on the bridging level. Correlations and interdependencies between the levels, as shown in Figure 4.1, are identified and connections are ensured in order to obtain a comprehensive view of the manufacturing system.

4.3.1.2 Disciplines

Industry is defined as the type of trade that is distinguished by the production and the processing of material goods. Mainly performed disciplines in many industrial corporations are mechanical engineering, industrial engineering, electronic engineering, and computer science. An example of disciplines and their corresponding sub-disciplines is depicted in Figure 4.3. This categorization was used for the case study conducted in this research, but it can easily be adapted to the disciplines that are important for different kinds of corporations.

Mechanical engineering is one of the main disciplines performed within an industrial corporation. The performed tasks within this discipline are related to the design, analysis, manufacturing, and maintenance of mechanical systems. The principles

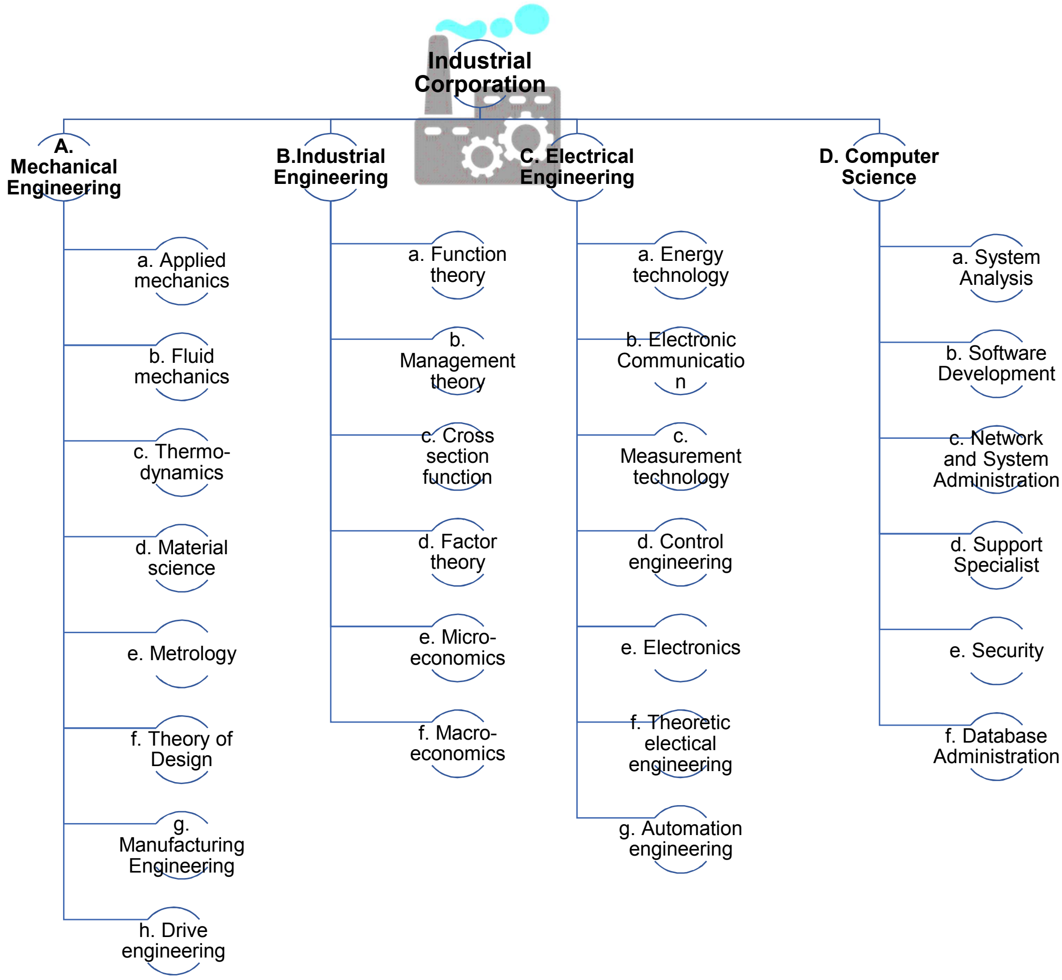
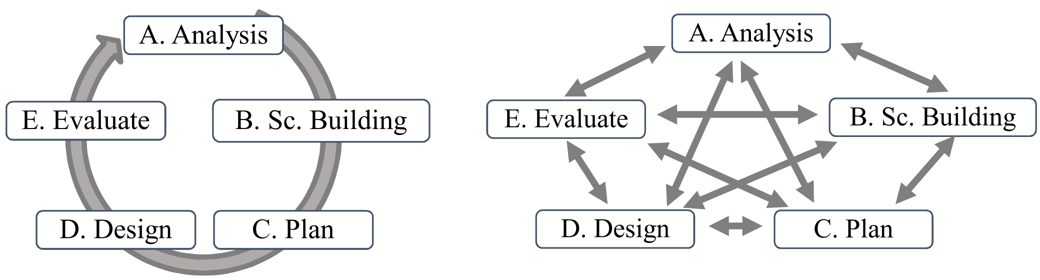


FIGURE 4.3: Overview and classification of industrial disciplines within a typical industrial corporation. It is shown a segmentation of the individual categories in dedicated ranges of tasks for the realistic modeling of an industrial corporation. Classification based on [305, 230, 173].

of engineering, physics, and materials science are assigned. Mechanical engineering is further subdivided into the disciplines shown in Figure 4.3, column 1. Mechanical engineering involves the design, production, and operation of machines. Business administration as task in industrial engineering has the goal to describe, explain, and support decision-making processes, mainly performed by several experts. Exemplary fields of activities in industrial corporations design the product program, optimize factory layouts, insurance of part deliveries, and many more. Sub-disciplines of industrial engineering are listed in Figure 4.3, column 2. Electrical engineering deals with the research, development, and production of electrical appliances. For example, electrical machines, components, and circuits need to be utilized in embedded systems and manufacturing systems. Figure 4.3, column 3 lists the sub-disciplines of electrical engineering. For example, measurement technology and automatic engineering are disciplines that have a close relation to mechanical engineering and are very frequently adapted in industrial corporations. Especially the field of computer science experienced a strong growth in industry. The reason is a growing demand of applications and systems that are computer-driven and necessarily need to be developed by computer scientists (networking, embedded systems, and data-management, etc.). The term "digital transformation" plays an important role for corporations to remain competitive and to open up new markets. Consequently, disciplines of computer science are deeply integrated into the daily business, as depicted in Figure 4.3, column 4. The proposed classification of industrial disciplines serves as an overview for incorporated disciplines in industrial corporations, but does not describe the generality for all companies. The diagram merely comprises exemplifying activity fields, which is not generalizable for all organizations. There might be corporations with a different discipline distribution and differing activity fields.

4.3.1.3 Tasks

The tasks in an industrial corporation can be centralized in a design cycle in Figure 4.4a illustrates the iterative optimum quality assurance concept and comprises the tasks analysis, scenario building (Sc. Building), planning, design, and evaluation. All tasks proceed on all levels, leading to cross-references between disciplines, tasks, and levels. Analysis deals with the observation of causalities, in which one decision is partly responsible for an effect. Responsibilities and dependencies are ascertained and formalized. The scenario building phase addresses the task of developing processes, strategies, and guidelines based on those causalities. These strategies are collected and classified into diverse use cases. In the planning phase, the current scenario is acquired and mapped with collected use cases. Different strategies are balanced and adapted into the current scenario. The chosen strategy is elaborated and transformed into an operating plan to achieve the overall business goals. Design addresses the actual task of manufacturing work-pieces, design of machines and their arrangements, composition of materials, and fabrication of products. Once all strategies are implemented and realized, performance indicators are acquired and analyzed within the evaluation phase. Quality assurance tasks and maintenance can be comprised in this phase. Based on the evaluation, the current scenario is newly valuated and the design cycle is reiterated.



(A) Optimal design cycle with a perfectly structured task sequence.

(B) Optimizable design process with many cross-references and feedback loops.

FIGURE 4.4: Visualization of tasks within an industrial corporation.

Typically, the design process in industrial corporations is not conducted as described. Interactions, adjustments, and spontaneous decisions take place, which might disturb the process structure and must be controlled. To steer the process flow, it is necessary to model it, visualize it, and positively influence it with the use of appropriated parameters, which will be described in this section. While the left design cycle shows an optimal process, the right one (Figure 4.4b) in contrast represents a more realistic design process. It is highly unstructured; cross-references between tasks exist or phases are continuously performed simultaneously to other tasks. Especially the task of evaluation and quality assurance is performed continuously over tasks, phases, and disciplines.

Similar to the classification of disciplines, this classification of tasks serves as an overview for performed tasks in industrial corporations, but does not describe the generality for all companies. The diagram merely comprises activity fields, which can differ in chronological order, scopes, and defined sub-activities within the category.

4.3.2 Dependency Parameters for Meaningful Visualization

To visualize the existing dependencies within the network, so-called Sankey plots are used. A Sankey plot is a well-known and commonly used method to highlight relations between several participants, objects, or elements within a group. Cubic splines can be used to visualize those connections. Nevertheless, there is the challenge to display a meaningful relation between those objects and to give an easily understandable overview over the whole relationship to help solving the process described earlier (see Figure 4.4b). Line plots quite often do not lead to an immediate understanding of a complex system as shown in Figure 4.1. Therefore, it is mandatory to define a set of parameters that can be visualized, using color-coded line plots or transparency definitions within a Sankey diagram. This problem will be solved by considering a case-dependent two- or three-dimensional representation of subcategories and a set of elements in a main category. Considering several categories whose relations between each other have to be investigated, the main category describes the entities which are in focus of the investigation and which are located in the center of the visualization. For example, the main category can represent a group of cooperating people where the focus is lying on their cooperation. System-levels, disciplines, and tasks are defined to be subcategories that need to be considered and help to optimize the upper process. The "system complexity" is an example of a parameter that characterizes the whole system. Before describing a set of parameters, view definitions will be made.

Let us assume that there are $P \in \mathbb{N} \setminus 0$ elements in the main category, which are in the focus and can be addressed by an index $p \in \{0, \dots, P-1\}$. Furthermore, it is assumed that there are $N \in \mathbb{N} \setminus \{0\}$ categories, which can be addressed by an index $n \in \{0, \dots, N-1\}$. Each category contains $K_n \in \mathbb{N} \setminus \{0,1\}$ individual elements and is addressed using the variable $k_n \in \{0, \dots, K_n - 1\}$.

System complexity
The number of connections between the categories and the main category defines the complexity of a system. A connection is described by the parameter
$\delta_{k_n,p} \in \{0,1\}$ where $\delta_{k_n,p} = \begin{cases} 1, \text{if connection between main and sub n exists} \\ 0, \text{otherwise.} \end{cases}$

The maximum number of possible connections normalizes the parameter. The complexity then leads to a scalar value and can be defined as

$$comp = \frac{\sum_{n=0}^{N-1} \left(\sum_{k_n=0}^{K_n-1} \sum_{p=0}^{P-1} \delta_{k_n,p} \right)}{P \cdot \sum_{n=0}^{N-1} K_n} \in [0;1] \tag{4.1}$$

comp = 100 % is the maximum possible complexity and comp = 0 % is the minimum complexity of the given system. This parameter does not contain any information on how homogeneous connections are distributed and therefore it is not very appropriate to visualize this parameter or make a judgment of the system's balance (work balance).

User's specialization regarding one category
Nevertheless, it is a good idea to have a closer look at the distribution of connections between a chosen category n and the main category. That is why it is necessary to focus on how many connections of an element within the main category exist for each subcategory. The parameter

$$v_{n,p} = 1 - \frac{\left(\sum_{k_n=0}^{K_n-1} \delta_{k_n,p}\right) - 1}{K_n - 1}, \text{where} \left(\sum_{k_n=0}^{K_n-1} \delta_{k_n,p}\right) \geq 1 \tag{4.2}$$

can be visualized using color coding. The parameter makes the assumption that there is at least one group-related connection for each element available. This parameter gives a value for the degree of specialization of an element of the main category with respect to a chosen subcategory. Per definition $v_{n,p} = 0$ is a low degree of specialization and $v_{n,p} = 1$ is a high degree.

User's specialization regarding all categories
Since $v_{n,p}$ is a subcategory based parameter it might be interesting to define a mean value that is calculated from this parameter. Summing up all categories and considering the number of categories leads to

$$\mu_p = \frac{1}{N} \cdot \sum_{n=0}^{N-1} v_{n,p} = \frac{1}{N} \cdot \sum_{n=0}^{N-1} \left(\frac{\left(\sum_{k_n=0}^{K_n-1} \delta_{k_n,p}\right) - 1}{K_n - 1} \right), \tag{4.3}$$

which gives the user an idea of the mean value. A low degree of specialization means that a selected user has connections to many subcategories. Consequently, the work balance is not well distributed and a system adjustment should be considered. Otherwise, it might be desired that a user works on several subcategories (e.g., he is a participant of the bridging level). To avoid an overload of this user, other subcategories should be taken into account as well. This can be done by calculating the empirical standard deviation of this parameter, which is described below.

Empirical standard deviation of a user's specialization
Consequently, the empirical standard deviation can also be visualized. This parameter gives the user an idea of the homogeneity for each object and leads to the information if there is a high deviation between degrees of specialization for all elements of the main category or not. The empirical standard deviation is defined as

$$s_p = \sqrt{\frac{1}{N-1} \cdot \sum_{n=0}^{N-1} \left(v_{n,p} - \mu_p\right)^2} \text{ if } N \geq 2. \tag{4.4}$$

If there is a low degree of specialization and a low standard deviation, this means that a user might be overloaded because he has connections to many subcategories. This must be avoided to improve the system's work balance and consequently the process flow (see Figure 4.4).

Group coverage
So far, the described parameters have had their focus on the main category. It might be interesting to set the focus on the subcategories as well. The group coverage is defined as a parameter that can be used to analyze how big the support of the main category actually is. This might be interesting within process planning if a big group must solve several kinds of tasks. This is the case within the IRTG 2057 group. The group coverage may be calculated with the equation

$$w_{n,k_n} = \frac{1}{P} \cdot \sum_{p=0}^{P-1} \delta_{k_n,p} \in [0;1]. \tag{4.5}$$

If w_{n,k_n} is a big value this means that many elements of the main category have a connection to the selected category. In conclusion, if the main category contains people working in a group, and if there are tasks defined within the subcategory, this value gives an idea on how good the coverage of the tasks actually is and might help to improve the decision-making processes.

Mean group coverage and empirical standard deviation
In analogy to the defined parameters μ_p and s_p, the category related mean value μ_n and s_n are defined as

$$\mu_n = \frac{1}{K_n} \cdot \sum_{k_n=0}^{K_n-1} w_{n,k_n} \tag{4.6}$$

and

$$s_n = \sqrt{\frac{1}{K_n-1} \cdot \sum_{k_n=0}^{K_n-1} \left(w_{n,k_n} - \mu_n\right)^2} \text{ if } K_n \geq 2. \tag{4.7}$$

4.3.3 Applied User Taxonomy onto Sample Data Set of IRTG 2057

The example dataset is constructed based on actual projects within the International Research and Training Group (IRTG 2057, "Physical Modeling for Virtual Manufacturing Systems and Processes"). Although production planning is not a new research topic, this field still offers potential for optimization. With the use of theoretical computer models, production is planned from different points of view: from a single machine, up to a complete factory. However, the computer models include actual physical properties, which enable them to calculate key properties of a production line. Product quality or the energy consumption are calculated and targeted improvements can be performed.

As stated earlier, the IRTG 2057 program covers disciplines from mechanical engineering, industrial engineering, electrical engineering, and computer science allocated on given levels: process, machine, factory, and bridging (see Figure 4.2). According to the discipline classification (see Figure 4.3) and tasks (see Figure 4.4) from above, users can be identified (see Table 4.2). The following types of information, which are needed and provided by a user, can be examined.

TABLE 4.1: Identified information types within the dataset.

• Process requirements	• Material composition	• Material properties
• Material behavior	• Process properties	• Process behavior
• Nominal-actual comparison	• Machine properties	• Machine behavior
• Machine mechanisms	• Machine setting guidelines	• Production program alternatives
• Facility properties	• Decision making support	• Optimized production program
• Facility layout guidelines	• Sustainability indicators	• Indoor condition guidelines
• Quality parameter	• Service guideline	• Indoor environment properties

Two users are connected with each other if they receive or provide the same kind of information. There are two types of connections available: indirect and direct dependencies. A connection is assigned as direct, if a user receives information from another user. Indirect connections mean that information is shared by those users and can be manipulated by both of them. Furthermore, Figure 4.5 represents the information flow and data exchange between other users and a common database system among all levels based on the described dataset. For a better readability, the users are connected via graphical elements, which represent the information type (also highlighted in Figure 4.5). The cylindrical shapes depict the continuous read and write transactions from and towards the database system.

Although the diagram might be overwhelming for the reader, we decided to include it to demonstrate the high complexity of the dataset and the consequential challenging design decisions for the system architecture and task modeling. With the use of the defined dependency parameter we can confirm this statement. The complexity of the system is $comp = 0.411$, implying that the overall system complexity is within an acceptable range. For more reliable statements further analysis needs to be considered. The specialization of user B for each individual category are: $v_{B,0} = 1$, $v_{B,1} = 0$, $v_{B,2} = 1$. It can be deduced, that user B has only one connection to category 1 and category 3 which means that he is specialized to one of the given subcategories (of each category, levels, and disciplines). Regarding the second category (tasks), the user is not specialized but rather completely interdisciplinary. Hence, the user shows a mean specialization value over all categories of $\mu_{p=2} = 0.667$. The empirical standard deviation of a user B's specialization is $s_p = 0,5774$, implying that the degree

TABLE 4.2: Identified users in the dataset related to IRTG 2057 college.

ID	User	Discipline	Task	S.-Level	ID	User	Discipline	Task	S.-Level
1	A	Applied mechanics	Scenario building	Process	16	Q	Computer science	Analysis	Process
2	B	Computer science	Analysis	Bridge	17	R	Fluid mechanics	Design	Machine
3	D	Manufacturing Eng.	Design	Process	18	S	Applied mechanics	Scenario building	Process
4	E	Computer science	Analysis	Factory	19	T	Applied mechanics	Design	Machine
5	F	Theory of design	Scenario building	Process	20	U	Theory of design	Evaluate	Process
6	G	Computer science	Design	Factory	21	V	Material science	Plan	Process
7	H	Computer science	Evaluate	Factory	22	W	Fluid mechanics	Analysis	Process
8	I	Management theory	Plan	Factory	23	X	Material science	Scenario building	Machine
9	J	Measurement technology	Analysis	Machine	24	Y	Theory of design	Scenario building	Process
10	K	Function theory	Plan	Factory	25	Z	Applied mechanics	Design	Machine
11	L	Function theory	Plan	Factory	26	AA	Theory of design	Evaluate	Process
12	M	Material science	Design	Process	27	AB	Material science	Plan	Process
13	N	Management theory	Evaluate	Factory	28	AC	Applied mechanics	Analysis	Process
14	O	Manufacturing Eng.	Plan	Machine	29	AD	Theory of design	Evaluate	Factory
15	P	Computer science	Scenario building	Factory	30	AE	Measurement technology	Design	Machine

of specialization has a comparable big variance over all considered categories. The user might have a good work balance. The mean group coverage values per category are $\mu_{n=0} = 0.111$, $\mu_{n=1} = 0.4933$, and $\mu_{n=2} = 0.25$. With the use of these values, the number of connections per category can be calculated: $\mu_{n=0} \cdot P \cdot K_{n=0} = 0.111 \cdot 30 \cdot 930$. This number of connections indicates that each user is connected to exactly one subcategory. Consequently, category 2 (tasks) has 74 connections and category 3 (levels) 30 connections. The bigger the empirical standard deviation the higher the disparity of the distribution of the users to subcategories. The calculated values $s_{n=0} = 0.0707$, $s_{n=1} = 0.0596$, and $s_{n=2} = 0.1774$ are considerably low, meaning the system has equal distributions among all categories.

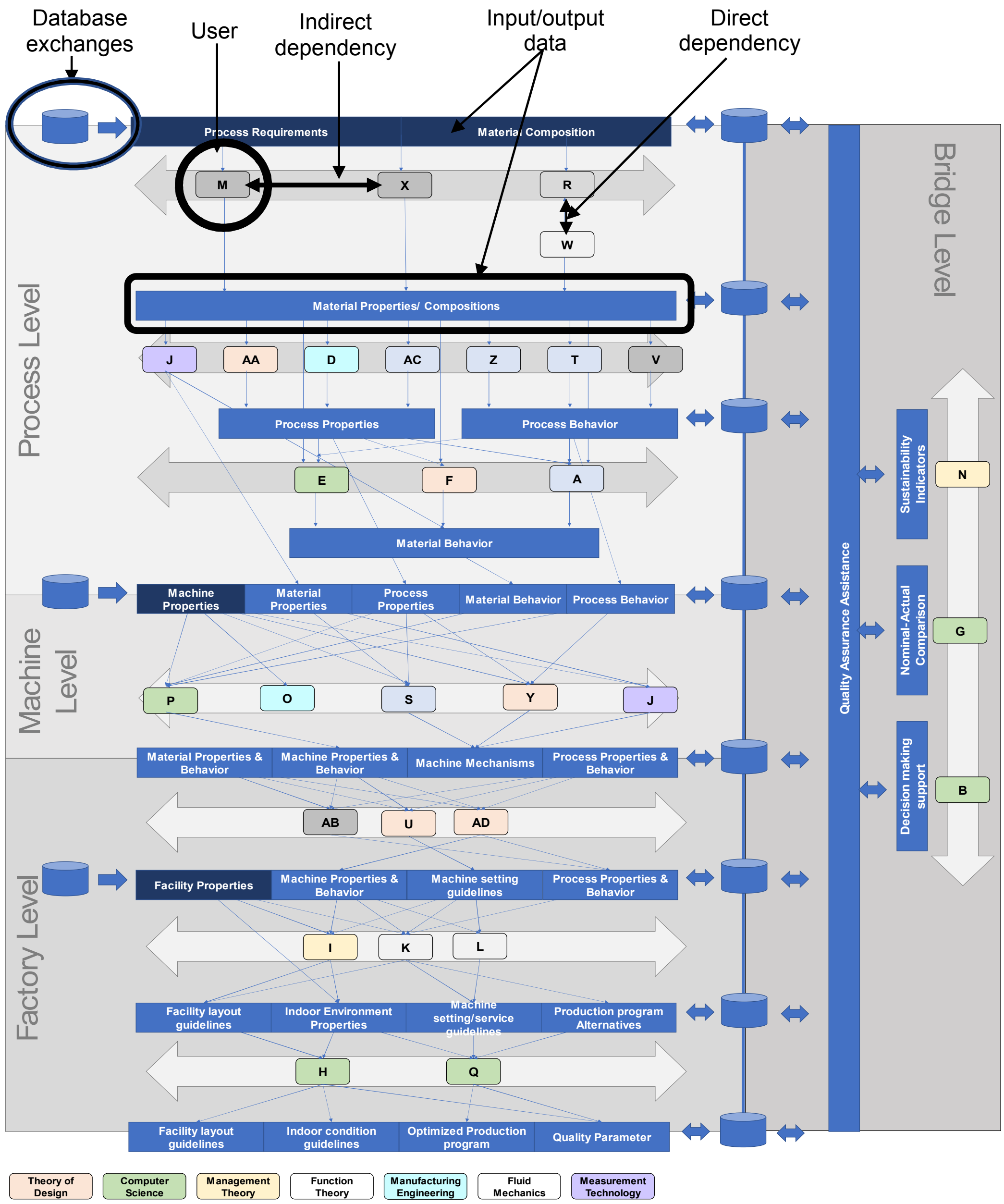


FIGURE 4.5: Connecting of users with databases for detailed information exchange within an industrial environment. The whole information flow is a complex network, which is hard to visualize and to understand.

4.3.4 Visualization Framework Facilitating Dependency Exploration

The whole dependency graph that is shown in Figure 4.5 describes a very complex and hard to understand system. To get a more generalized overview of the given dependencies, alternative visualization techniques need to be applied. Such an alternative visualization technique has already been shown at the very beginning (see Figure 4.1). The given graph in Figure 4.1 visualizes the identified dependencies using a force directed graph. The high amount of edges and crossing lines indicates a higher number of connections between all entities. While Figure 4.1 only shows the connections between the participants, Figure 4.5 gives a detailed insight into the system structure, which is interesting especially for system's architects. It provides

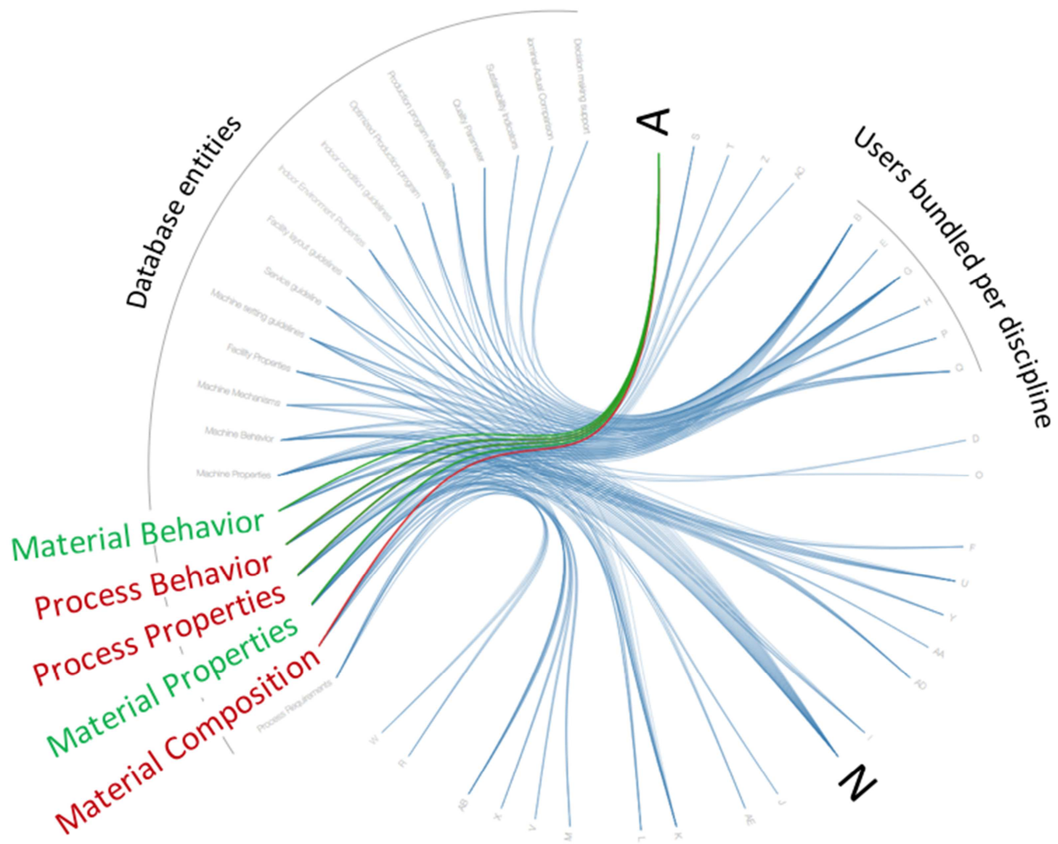


FIGURE 4.6: Radial visualization of the dataset, which represents incoming information (red) and outgoing information (green). The nodes describe database entities and users.

information on entities, data, or material flow and the database structure. Nevertheless, the whole visualization might be overwhelming for other user groups, which do not necessarily need such a deep insight into the architecture. Therefore, an appropriate visualization method for other user groups is introduced in Figure 4.6. Figure 4.6 describes the relation between input and output data provided and needed by the users. Users who correspond to the same discipline are bundled as seen on the right-hand side. On the left-hand side database entities are represented. Hovering over nodes highlights the connections. For example, the selected and highlighted user A has the green connections as input data and the red connections as output data. A red label combined with a green line indicates that this information type is both input and output data. Highlighted user N shows a high amount of connections to all database entities, which indicates that user N is on the bridging level, where it is necessary to have access to all available information. Hovering over a database entity highlights all users who are connected to that information type. Users that receive the information are colored in red, while users that provide the information are colored in green. Therefore, direct (green) and indirect (red) connections between users are indicated across the database entities.

4.3.4.1 *Graphical User Interface*

In the course of finding simplified visualization methods for complex systems, an application to explore the dependencies and to get insights into parameters, which have been described in section 4.3.2, has been developed using Matlab. The result is an interactive environment that implements the Sankey methodology, force directed graphs, ring based radial visualization with hierarchical edge bundling, and the deployed dependency parameter. The aim of this separate platform is to enable users to explore the recognized dependencies and links, which are deduced from the deployed user taxonomy. To support an efficient interaction with the model and easy information gathering, the application focuses the most important interdependencies by fading out irrelevant information and allows the user to switch between different visualizations without losing the focus. The initial view of the graphical user interface (GUI) is shown in Figure 4.7.

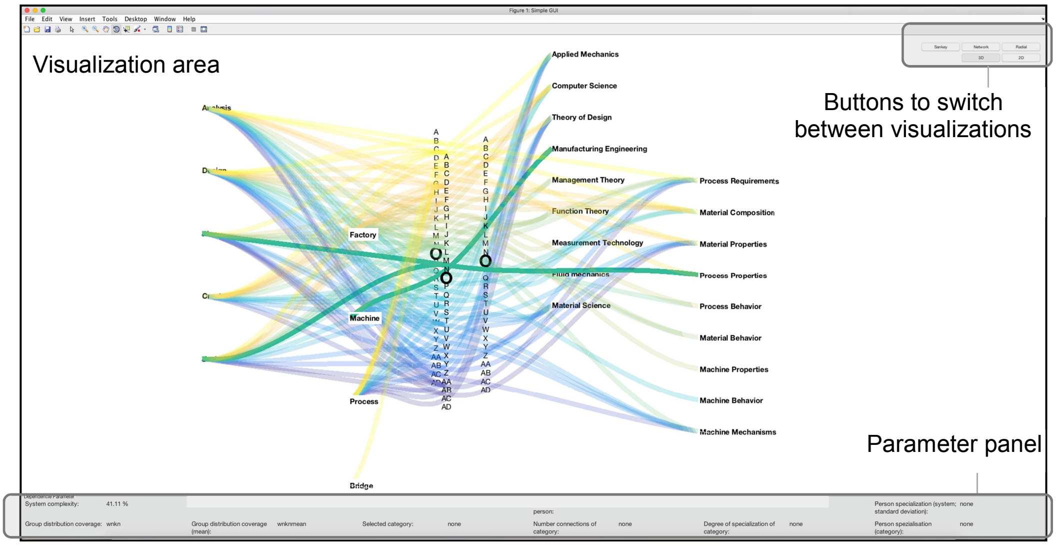


FIGURE 4.7: The user interface of the application comprises mainly three components: the parameter panel, buttons to switch between different visualizations, and the visualization area. By initialization the dependency parameters are calculated and a 3-D Sankey diagram is plotted. Highlighted in the visualization are the connections of the selected user O.

The earlier described network visualizations (see Figure 4.1 and Figure 4.6) have been implemented, giving the possibility to set the focus on different aspects of the dataset. These are based on d3.js [45] and the programming language Python, respectively embedded as web-view in the application. The user interface itself consists of a visualization area, selection buttons, and a parameter panel, and is shown in Figure 4.7. Pressing the buttons aligns the visualization as explained below.

The dependency parameters, which have been discussed above, are calculated in the background and can be visualized individually using the interactive parameter panel (see Figure 4.7, highlighted area). They provide a powerful tool to make the whole system structure better understandable using color-coded visualizations. Per default, a 3- dimensional Sankey diagram is shown. A 2-dimensional representation can be selected as well. All possible visualizations will be described and shown in the next section. The focused category is displayed at the center of the visualization and it is surrounded by a given number of subcategories. These are linked to the focused main group. In our example, the main group is represented by users (compare to Table 4.2), which are surrounded by the categories disciplines, tasks, and levels (compare to Figure 4.2, Figure 4.3, and Figure 4.4).

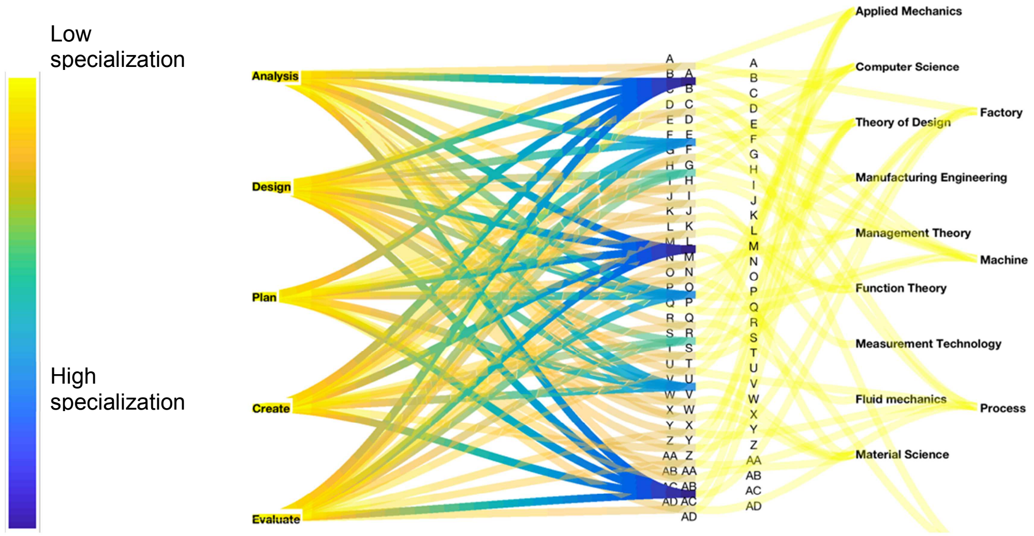


FIGURE 4.8: Colorization of the diagram based on calculated specialization parameter $v_{n,p}$ according to Equation 4.3.2. The visualization elucidated an inhomogeneous distribution of the group respective of the assigned scope of duties of the cooperating user. The color gradients from yellow to blue as well as the transparency elucidate the number of the existing connections.

Both the 3-dimensional and the 2-dimensional Sankey diagrams are interactively connected with the dependency parameters. Selecting one user or one category entity by clicking on the label, highlights the label and all connected edges. The colors are defined per user, evenly distributed over the complete spectrum of the parula-colormap, which gives a wide range of colors and is available within the Matlab libraries. Clicking on one of the defined dependency parameters results in a colorization of the Sankey diagram according to the calculated parameter, as shown in Figure 4.8. The selected and highlighted labels and connections remain unchanged when switching between the Sankey visualizations. In Figure 4.8, the individualized parameter $v_{n,p}$ is displayed, which gives the applicator a feedback on the users' specialization regarding disciplines, tasks and levels. The degree of the specialization of a user regarding one category is visualized with color-coded gradients from one connection (yellow) to the maximal number of existing connections (blue). While all users only have one connection to the other categories (system-levels and disciplines), users can have up to five connections to entities of the category *tasks*. The meaningful colorization makes it easy to see the number of existing connections and the degree of specialization of each user.

4.3.4.2 *Comparison and Utilization of the Implemented Visualization Techniques*

Default view – 3D Sankey visualization. The 3D Sankey visualization shows all entities and connections in one view. By rotating the diagram the connections with all categories of one user can be explored. Any amount of categories can be visualized and be interactively explored.

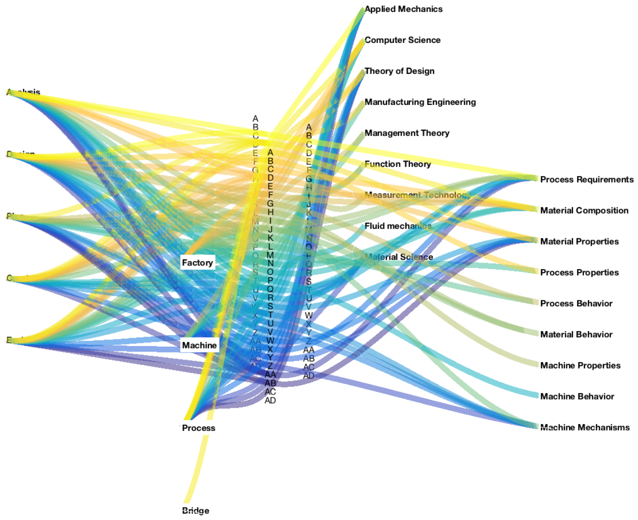


FIGURE 4.9: 3D Sankey visualization shows all entities and connections in one view.

Benefits & Shortfalls:

- ✓ All connections in one view.
- ✓ Visualization and comparing of individual parameters.
- ✗ Might be overwhelming.
- ✗ Impact depends on selected parameter.

2D visualization of Sankey diagram. 2D visualization sets the focus on two categories. The links can be easily recognized and the user can observe the relevant data. User interdependencies are indirectly connected via categories. Entities and highlighted parameters do not change when switching between visualizations.

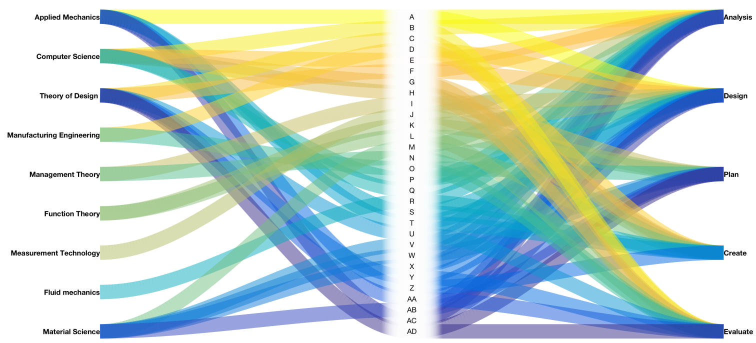


FIGURE 4.10: 2D visualization of Sankey diagram focus on two categories for easy connection identification.

Benefits & Shortfalls:

- ✓ Selected features retain and focus remains.
- ✓ Only relevant data is visualized.
- ✓ Connections can be tracked.
- ✗ No information on other connections.
- ✗ No direct interdependencies between users.

2D/ 3D graph visualization. Network visualizations are used to identify dependencies and connections at a glance. The high-level visualization of connected nodes represents an alternative to the 3D graph visualizations. Entities with a very high or very low amount of connections can be easily recognized.

FIGURE 4.11: Compiled 2D/ 3D graph visualization identifying dependencies and connections at a glance.

Benefits & Shortfalls

- ✓ Can be combined with interaction capabilities.
- ✓ Densities are captured easily.
- ✗ Details on the connections and entities are not recognizable.
- ✗ No detailed information available.

Radial visualization of information flow. Method is used to get insights into the information flow and connects users via existing information types. Input and output data of users are easily recognizable. Information flow and data transfer between provider and receiver can be highlighted interactively.

Benefits & Shortfalls

- ✓ Clear visualization.
- ✓ Contains relevant information in multiple dimensions.

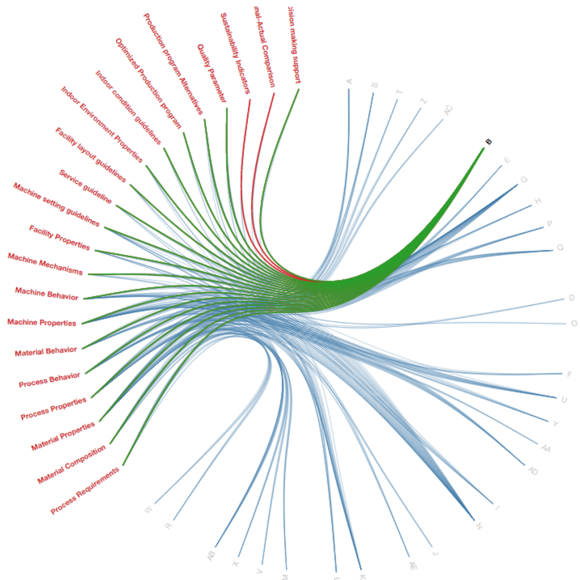


FIGURE 4.12: Radial visualization used to get insights into the information flow and connects users.

- ✗ Degree of simplification in some cases might be to high.
- ✗ No direct connections recognizable without user interaction.

4.3.5 Creation of Dedicated User and Task Models

Examples of users and tasks from the mechanical engineering discipline are selected in order to create dedicated user and task models. Using a real-world scenario in production control, we illustrate the impact and interdisciplinary collaboration requirements. Two domain-specific tasks of Event-driven Production Control (EDPC) and layout planning were chosen, both tasks being relevant for factory planning. Two different task models were created using the task world ontology introduced in Chapter 2.2 and the inferdependencies are explored.

4.3.5.1 Event-Driven Production Control and Factory Layout Planning

Companies in high-wage countries must be highly efficient and innovative in manufacturing to remain competitive. Market competition has increased through openings in economic regions, e.g., in Eastern Europe or the Far East [25]. Mass products are offered for lower prices by countries in these regions. Companies in high-wage countries have to adapt in this evolving competitive setting, as they might become obsolete and be destroyed otherwise [53].

The domain of factory planning tackles these problems. A strategy to address this challenge is a shifted focus on highly specialized, customized products [25]. By moving to a more individual customer-oriented production setting a company follows a make-to-order processing paradigm, where the production of a part starts with the arrival of an order [146]. Nevertheless, customers still require products with high quality to be offered at a low price and a short delivery time. For a company this means that it must be flexible in offering individual products in a short amount of time while still remaining economically sustainable [168]. High flexibility within a production process makes it necessary to concentrate on production planning and control. A production process can be planned but only in rare cases the production is planned. Events that lead to such a deviation within the production process, are, for example, machine breakdowns, missing parts or manufacturing of unusable parts/products [167]. In such circumstances, a planned optimal production process cannot be realized.

A company must react as quickly as possible to events (new customer orders or deviation from a planned optimal production process) to minimally deviate from a stated production plan. Such events cause a gap between the current state of a production process and the planned state of production. An alternative production plan must be created in this situation, leading to the concept of "production control." Production control regulates such conditions within the order processing, i.e., it determines the sequence of sub-processes that should be executed [252]. If changes in the production plan arise, the production control will be responsible to carry out modifications. The first step identifies the current state of production. This step must be executed rapidly to minimize, as much as possible, the deviation between the planned and the current state of production [168]. To react rapidly, a continuous view of the state of production is necessary and all related information must be continually recorded and available. Such a setup makes it possible to adapt a production plan quickly once a disturbing event is recognized.

Production control must be an automatic process to detect undesired events and adapt the production plan accordingly. An event-driven production control (EDPC) was developed to meet these requirements. The EDPC system uses an extended bill of materials in order to shorten the reaction time in case of occurrence of an (undesired) event. The bill of materials is extended with additional information for each part (e.g., required production station, size, mass, set-up time, and production time). This extension makes it possible to store information within the bill of materials that is necessary for the production control.

The system approach taken by Kasakow et al. [155] uses the production of a turbo charger as an example. By placing an order, the EDPC uses the content of the order and creates an appropriate extended bill of materials. Necessary production tasks to be done to satisfy the customer order derive from this bill of materials. This approach demonstrates that it is possible to derive all necessary actions to be taken within a production process (e.g., creation of production orders, arrangement of the production sequence), based on a customer's bill of materials and the information of the current state of production.

A disadvantage of this EDPC is the acceptance by users. The acceptance of a system depends on the experience of a user. Here, a user is a planner of a production. The more experience a planner has with automation errors, the more she/he wants to supervise and monitor the system. But a planner trusts automation only when it is fully reliable. The reliability of an automated system leads to the required or desirable amount of supervision and monitoring effort. A lack of reliability reduces the acceptance of an automated system [300]. In case of misbehavior or breakdown of the automated EDPC, a planner needs to have access to an uncomplicated solution to interfere with this production control to ensure a smooth operation of the production process or to carry out tasks that are not part of the EDPC (e.g., implementation of a rushed order) [29, 155]. An ideal system allows the planner to monitor the status of the current production and provides all the necessary information to optimize production.

4.3.5.2 Task Model Inferdependency

Factory planning, as domain field of EDPC, is characterized by the parallel consideration of multiple aspects such as production resources, production process and technology, and products while anticipating uncertainty and future developments over the factory life-cycle [286]. These aspects usually result in different partial-models with specific information content (e.g., layout model, process model) and components of the factory (e.g., building, machinery, foundation, media), which need to be analyzed in combination. The different partial solutions are usually developed by various stakeholders, but typically interfere and require each other [258]. The major tasks regarding collaborative factory planning are [307]:

1. Assembling multiple, domain-specific points of view.
2. Bilateral problem introduction.
3. Joint discussion and integrated decision making.

Appropriate visualization tools to support collaborative factory planning must be able to coordinate different layouts and viewpoints on the factory as well as exchange and manage information and models from different domains. The tasks are summarized in the following:

- **Creation:** Combination of different part models and information content;
- **Perform:** Adjustments on the layout to develop optimizations;
- **Coordination** of various models, information sets, and planning perspectives;
- **Verification** of layout through immersion and analytics;
- **Consideration** of efficiency, usability, and extendibility constraints.

VR-supported workflows are proposed to foster collaboration, establishment of a joint problem understanding, and exchange of different points of view [307].

In order to implement a real-world collaboration process, two domain-specific tasks of EDPC and layout planning were chosen, both tasks are relevant for factory planning. Two different task models were created using the task world ontology introduced in Chapter 2.2: Factory layout planning and event-driven production control. The concur task tree (CTT) diagram [301] depicts the simplification of both task models and their inferdependency (see Figure 4.13[1]).

[1]Usually, in a CTT, several nodes are not connected to the same child node. For better readability, the user sub-tasks *Move*, *Rotate*, *Delete* and *Insert* are connected to the grouped task *Adjust Model*. More precisely, each of the user sub-tasks should have a connection to the system activities *Update model* and *Update simulation*.

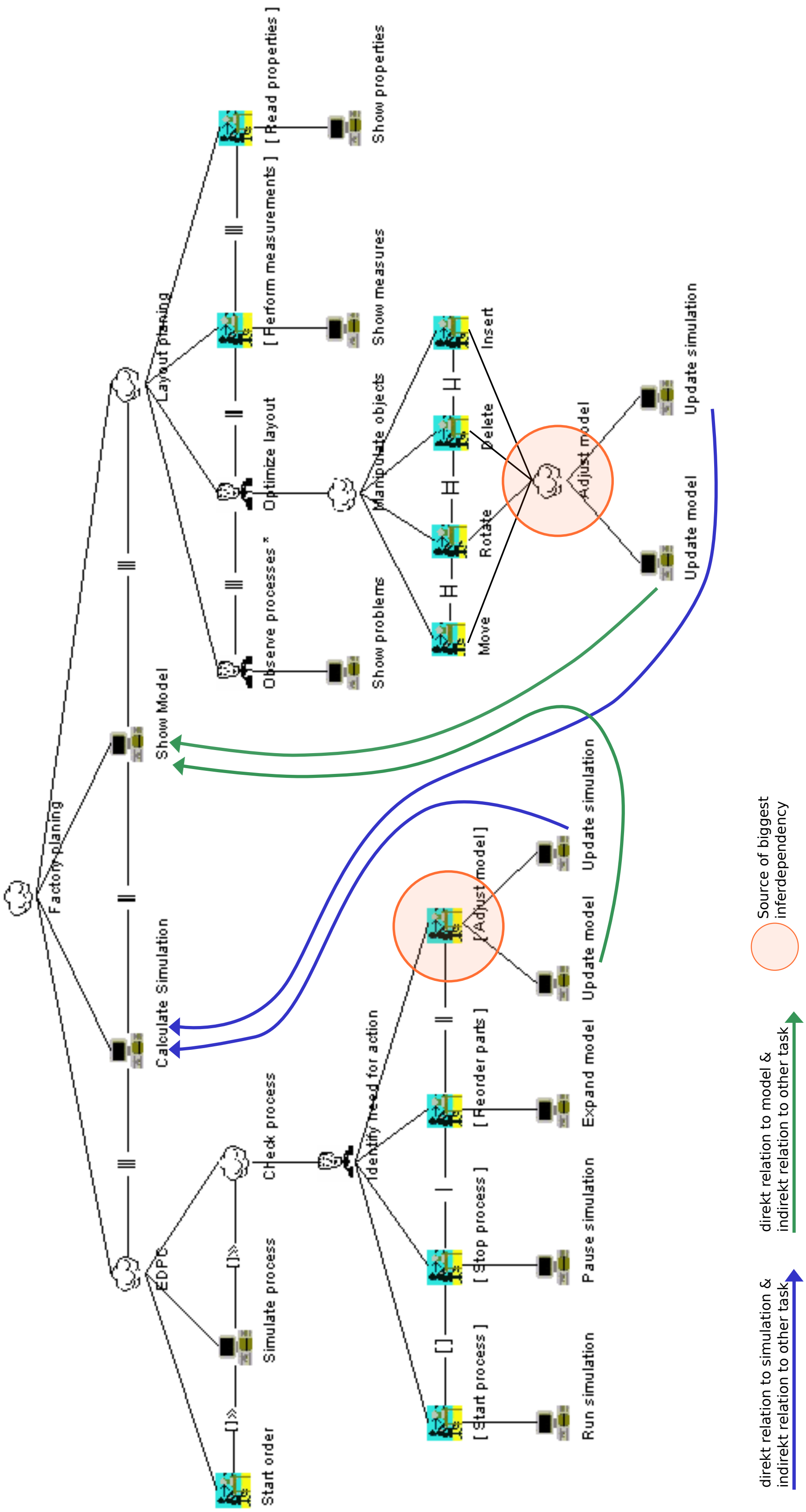


Figure 4.13: CTT diagram of combined task model representing task structure and inferdependencies.

All shown tasks are further refined as task world models, formulated as tasks performed by an actor having a specific role. Tasks can use objects and trigger, or be triggered, by events (see Figure 2.1). The CTT diagram shown in Figure 4.13 shows a refined definition of the tasks based on the task world ontology. The CTT diagram depicts a simplification of the tasks that are performed in factory planning.

Inferdependencies are found in sub-tasks incorporating the combination of influence and dependence between two elements, one- or bi-directional. Adjusting the model as sub-task in EDPC changes the underlying dataset for EDPC and factory layout planning, either due to changes in the simulation or manipulation of the model itself. Manipulating objects in the course of a factory layout planning sub-task has a direct influence on the production flow and simulation within EDPC, revealing the inferdependencies between both tasks. Layout planning concerns the task of *deciding on the best physical arrangement of all resources that consume space within a facility*. This task is performed when there is a change in the arrangement of resources [236].

Table 4.3 summarizes the parameters affected and incorporated by the task models for factory layout planning and event driven production control. Improvements of the overall production performance are achieved concerning those parameters due both tasks.

Table 4.3: Affected and incorporated improvement parameters of factory layout planning and event driven production control

Factory layout planning	Event driven production control
Time	Time
Energy consumption	Energy consumption
Cost	Cost
Organization	Quality
Efficiency	Utilization of tools
Productivity	Utilization of machines
Information flow	Factory layout suggestions
Optimal path of material flow	Optimal path of material flow

Accordingly, other processes and tasks also influence these parameters. Impact on material flow and overall production performance occurs when changes of the order are performed, but also between material flow and layout changes (two differing task models). Based on the listing of improvement parameters of both tasks in Table 4.3, it is easy to see, that several inferdependencies between the task models exist, e.g., changing machine positions and paths has an direct impact on material flow, production time, transportation time, waiting time, and path utilization.

Figure 4.14 visualizes the dependencies of this simple dependency graph, as described in Section 2.

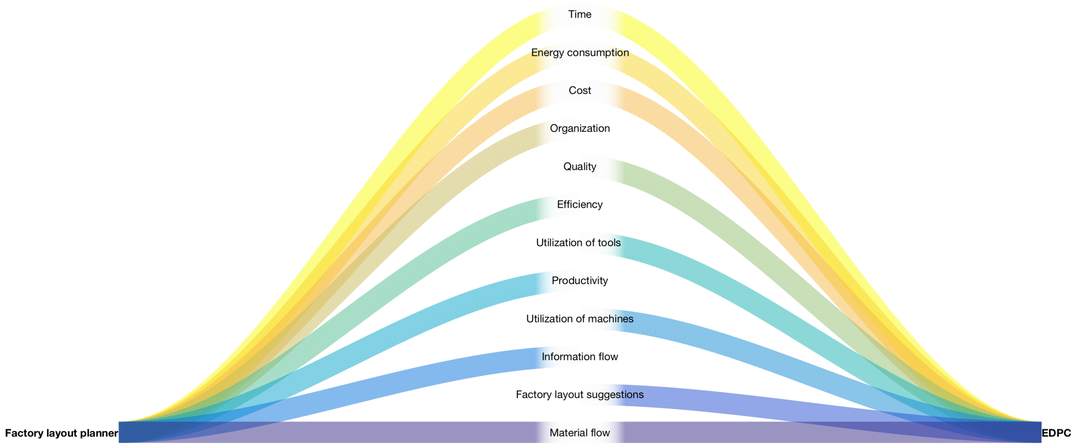


FIGURE 4.14: 2D Sankey diagram visualized inferdependencies between factory layout planning and event driven production control.

In the next sections, we will examine how to support the work-flow of the two user models. Thus, those user-models serve as an example of user groups for the development of novel and intuitive interaction and visualization techniques for computer-aided collaboration and decision making.

4.4 Multi-Modal Interface for Collaborative Design and Assessment

Ideally, in a distributed and collaborative networked environment, experts can work independently or jointly on sub-systems of an overall design [285]. Adoption of existing and available device and network technologies is still in its early stages and the integration into a collaborative system, as described here, is not realized. We introduce a framework enabling a team working in a distributed setting to collaborate via computer networks using various mobile interfaces and visualization devices. The framework makes possible the effective and synergistic combination of team members' complementary competencies and expertise. We address relevant challenges in the design and realization of an efficient, effective, and satisfactory collaboration framework.

The framework presented in this section can be adapted to the specific requirements of any application to support collaborative design done simultaneously. Mobile phones are used as secondary displays to provide a private and detailed view of data. Different aspects of the data can be represented in task-driven views (second display). Impact on a system caused by changes applied to it by another user is visualized in the main view (first display). However, the impact on particular tasks is transparent to a single user. The presented framework can substantially enhance efficiency of a distributed collaboration environment for design, simulation, and analysis efforts.

The devices we use provide three views of the data processed collaboratively: (1) a simulation view; (2) a status report view; and (3) a status update view. These views provide overview, detail, and performance information, see Figure 4.16. A large display device, acting as a public viewport, provides an overview of the data for all participants in a simulation view containing a virtual reality application. Therefore, a smart watch is used to redistribute the status update view, while the status report view is presented on the smart phone. Locating the status update on a smart watch is done analogously to using a usual watch, where users capture information (time) at a glance.

First, we summarize the requirements stated in Chapter 2.2 on such a collaboration framework and the influences that have to be taken into account. Afterwards we transfer the perceptions into a prototypical system. We initially present our framework as a general framework, from an application-independent perspective. Later, we demonstrate the specific adaptation and utilization of it for a mechanical engineering scenario that documents the various benefits offered by our framework.

4.4.1 Methodology of a Collaboration Supportive Framework

In this section, the requirements of a collaboration framework will be summarized and following, the components of such a system are defined.

4.4.1.1 Requirements for Efficient Collaboration and Decision Making

Based on the assumptions of collaborative work stated in Chapter 2.2, the requirements for efficient collaboration and decision making are summarized as follows:

R1 Participants can coordinate activities.

R2 Participants can communicate with each other.

R3 Participants can perform activities.

R4 Participants can switch between collaboration styles.

R5 Participants are satisfied with/in group work.

R6 Participants expectations are accomplished.

R7 Participants can determine others' activities.

R8 Participants can understand the impact of changes made.

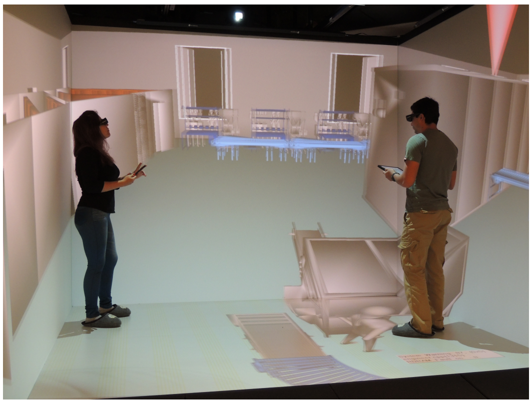
R9 Participants can make adjustments based on impact.

R10 Participants can exchange information.

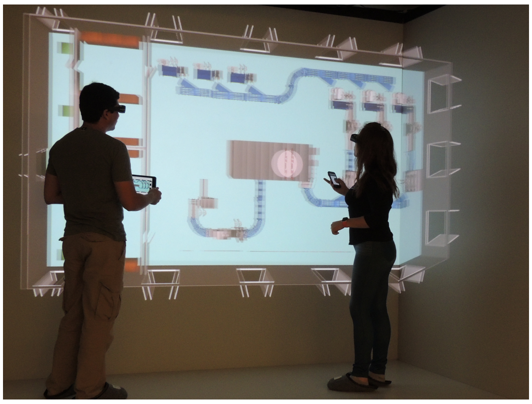
The aim of this section is to fulfill these requirements in order to support efficient collaboration and decision making in design and assessment in Virtual Reality.

4.4.1.2 Collaboration Environment

To support a collaboration environment, we use a setup consisting of (1) a large display device used as public viewing device and (2) smart devices as input devices while simultaneously holding private views of the collaborative task. That system is enhanced in order to fulfill the requirements stated above. In analogy to *Overview and Detail* views, a large display device presents an overview of the complete data in form of a public viewport (see Figure 4.15).



(A) Participants observing the scene in first person view.



(B) Participants observing the scene from the top.

FIGURE 4.15: Initial Collaboration Environment: Large display device used as public viewing device, smart devices enabling private views of the collaborative task.

Supplementary, mobile devices are used to provide detailed information of task-driven aspects of the data and they also act as input and control interfaces. In a user study performed for a typical, simplified factory-planning problem, it was demonstrated that team members could focus on the problem-solving task itself, instead of concentrating on interaction issues. By using the intuitive interaction capabilities provided by smart devices, focusing on the actual task at hand was made possible. A virtual representation of the data on the shared viewport facilitated communication and decision making in a team-oriented manner.

Simultaneous work development is an important aspect to support efficient real time collaboration. Similarly, different aspects and interpretations of the data are also important. Team-members have different interests, and consequently, the data must be shown in multiple views. The collaboration framework combines different task models together with visualizations for a shared public view on a large display device as well as interaction and visualization capabilities for mobile devices. Elements of mobile devices are used to support particular tasks, while the public view visualization enhances the existing visualizations combined in one view. Impacts caused by changes of another autonomous team-member are considered in the public viewport, where changes of the entire system are visualized. Impacts on particular tasks are hidden in the complete system view and cannot be identified by a single team-member. *Overview and Detail* techniques as well as *Context and Focus* techniques are not sufficient to support collaborative work while considering the inferdependencies.

Three different views are necessary to give insights about group performance, individuals' performance, and the visualization data. Our aim is to overcome the before mentioned limitations by proposing a general framework holding three views: Simulation view, status update view, and status report view indicating overview, performance view, and detail view. The simulation view enacts as an overview; holding a Virtual Reality application that generates realistic images and depictions of the processes. The status report view provides insights about individuals' aspect of the data. The status update view indicates individuals' performance. The simulation view is located on a public large display device; thus all team-members are able to observe the same view and have the same base knowledge on which they can investigate. Both status views are private elements visualized on smart devices. In this way, detailed information of single processes are removed from the public screen in order to not overwhelm the users with unnecessary information or even occlude more relevant information beneath. As powerful communication technology has become increasingly pervasive, collaboration between people has moved in the direction of computer-supported cooperative work (CSCW), where computers are now additional actors in the collaborative processes. Roles can be exchanged easily between actors, and activities can be delegated to systems [296].

To cover the design of collaborative task models and a supporting system, activities to be performed must be clearly defined. Who is performing what activities? What objects are needed? How should the information be presented to user-groups? How can interaction with the systems be enabled? What are the dependencies involved? These are the most important questions that have to be answered. As such, it is necessary to have a clear understanding of the requirements for the intended system. Services, users, environment, and associated constraints, for example, must be defined and connected. Users in the system are actors performing tasks using the task world ontology. One must include dedicated task models into the system together with rules and rights of data access and functionalities. Participants must be able to choose profiles that are connected with tasks when they register smart devices in the main system. Users can coordinate their activities and assign tasks (R1✓). The implemented visualization and interaction techniques are adequately realized for different scaled devices providing differing ranges of capabilities. Thus, participants are able to perform activities (R3✓). Setting up the co-located environment as depicted above, team-members share the same location and make use of a combined shared viewport on a large display device. Team members can determine others' activities (R7✓) and are enabled to communicate with each other (R2✓) in a natural manner. However, the postulated requirements R4 (Member can switch between collaboration styles), R8 (Member can understand the impact of changes made), and R7 (Member can make adjustments based on impact) are not ensured and will be tackled with the distribution of the user interface across different scaled devices.

4.4.1.3 *Distribution of User Interface Capabilities across Devices*

Dividing the simulation view in order to enable *Overview and Detail* techniques in one viewport is not sufficient. First, virtual reality is used for the simulation to generate realistic images and a mental image of the process. Splitting the view into two parts would decrease the level of immersion and the perception of being physically present in a non-physical world [226]. Second, positioning a second view in the simulation

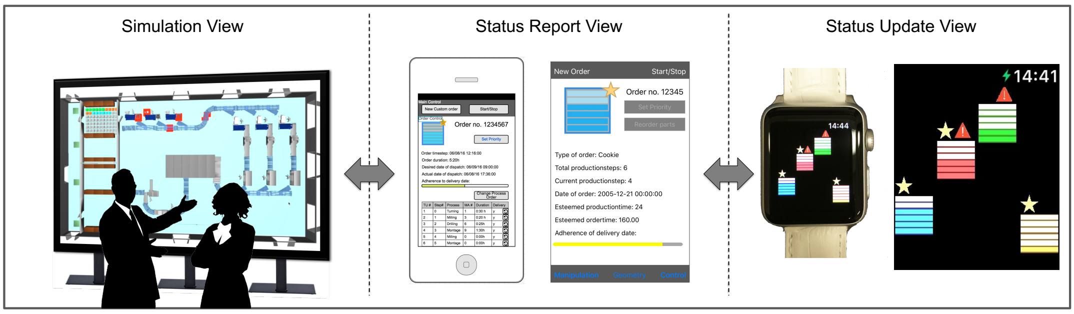


FIGURE 4.16: Collaboration setup: Simulation view containing a virtual reality application as public display; status update view enables monitoring of own process; status report view provides explanations of performance and interaction with the system.

view leads to a reduction of the visualization area.

Mobile devices enable the implementation of many interaction metaphors, leading to more natural and intuitive interaction. Tablet computers as well as smart phones offer input capabilities due to touch input mechanisms and other sensors, and we can use them as secondary displays. Tablet computers offer a bigger screen compared to smart phones, where the status update and status report views can be juxtaposed in a split-view. A split-view is not sufficient on a small display like those on smart phones. We use a smartwatch to re-distribute the status update view, while the status report view is shown on the smart phone, as depicted in Figure 4.16. Locating the status update view on a smart watch is analogous to the usage of a usual watch. In the following, the three different views are examined.

Simulation view. The simulation view provides an overview of the complete data and combines individual task model visualizations in one view. The simulation view holds a virtual reality application in which realistic images and depictions of the processes are generated. Virtual reality leads to a high level of immersion and the perception of being physically present in a non-physical world [226]. The purpose of virtual reality is the facilitation of reception and understanding of complex data due to simulation and visualization of the data in a real-world perspective [52]. Users experience and observe the scene "from inside" and are able to concentrate their attention on the task exclusively [48].

Status update view. The status update view provides an overview of the task performance/progress at a glance. To overcome the limited display size of the smart watch, we use a glyph-based visualization. This visual design is a commonly used technique, where data is represented by a collection of glyphs. The data set is typically multivariate. Related work is, for example, performed by Steiger et al. [270], who described "zoomable" glyphs. Viewed from a distance, glyphs are recognizable in shape in color; zooming in brings out the information captured by each glyph in detail. Relationships between variables and explanations of a glyph's appearance can be seen. A glyph-based visualization on mobile devices for the notional analysis in sport was successfully used to establish collaboration between different analysts on event-based visualization [179]. The major strength of glyph-based visualization is, that: patterns of multivariate data can be easily perceived in the context of a spatial relationship [44]. According to Borgo et al. [44] a glyph is defined as: *"a small independent visual object that depicts attributes of a data record"*. Characterization of those visual objects can be done as follows:

- Glyphs are discretely placed in a display space.
- Glyphs are a type of visual sign but differ in form.

Next to the number of dimensions that will be represented with the glyphs, their placement (positioning inside the display area; relationships between glyphs) on the display indicates significant information regarding the data values [304]. Taking into account the inferdependencies of task models implying a multivariate nature of the data in the presence of only having limited screen size available, glyph-based visualization matches the requirements of a visualization technique for our setting and its conditions.

Status report view. Touch input applied to a glyph in the status update view on the smart watch, opens a dedicated status report view on the smart phone. While the status update view provides a quick overview of the task performance, the status report view is designed to provide detailed information about the ongoing processes and the data visually presented via glyphs. Glyphs are repetitively visualized in the status report view to reflect the affiliation of both views. Detailed information is presented as text together with graphical control elements. Interaction capability in form of adjustments to the underlying data is provided, which has a direct impact on the main application and the visual representation of the glyphs.

4.4.2 Applied Methodology on Collaboration in Production Control

Using a real-world scenario in production control, we illustrate the impact and interdisciplinary collaboration requirements, and discuss the implemented system components in detail.

The dynamic model in the existing prototype combines the two task models of event-driven production control and factory layout planning, as introduced in Section 4.3.5. Based on the listing of improvement parameters of both tasks above, one might see, that both tasks have several inferdependencies between each other, e.g., changing machine positions and paths has an direct impact on material flow, production time, transportation time, waiting time, and path utilization. In the following subsections, we describe the information/data that is displayed in each of the views and devices for this case study.

4.4.2.1 User Roles and Rights

The simulation view of the framework provides an overview of the underlying manufacturing system, consisting of the building, storage areas, machines, human resources, and conveyors, see Figure 4.17. The smart-devices are used to control the scene and execute the functionalities in the large screen setup. The following functionalities are currently implemented for a desktop and CAVE setup, and for mobile devices. Supported functionalities include:

- **Manipulation:** rotate, pan, and zoom of single objects;
- **Navigation:** rotate, pan, and zoom of the whole model; first-person view and navigation; selection of pre-defined views; hide/show object-groups;
- **Examination:** measurement of distances and dimensions; textual output of object-information;
- **User feedback:** highlighting and vibration;
- **Collaborative features:** making annotations, inserting comments, marking areas, and creating a visual snapshot.

In order to identify different users, the following roles are defined and associated with the implemented functionalities. Each functionality is associated with exactly one role, implying existence of distinct roles. One or several roles can be associated to one user/actor.

Factory layout planner, basic

- measurement of distances
- measurement of dimensions
- getting machine information
- see facility information
- creation/ removal of machines

EDPC, basic

- start new product order
- stop production simulation
- re-order machining parts

Manipulator

- rotate, pan, and zoom object
- duplicate object
- delete object
- hide/show object groups

Collaborator

- making annotations
- marking areas
- creating visual snapshot
- show/hide highlights

Navigator

- rotate, pan, and zoom model
- first-person view and navigation
- selection of pre-defined views

4.4.2.2 *Distributed User Interface*

Simulation view. The simulation view is presented on the public large display device. This view shows, in this example, a manufacturing system in the context of dedicated work areas, machines, workstations, and transportation paths. Each object in the scene can be selected, moved, rotated, duplicated, or removed by a user. A selected object is highlighted in the user's color and locked for other users until it has been released. Coloring the selected object provides awareness of the various users. Locking an object ensures that the same object cannot be manipulated by several users at the same time.

In the simulation view each order is associated with several transport units. Each transport unit depicts one production step of the order, and it is visualized by cubes color-coded per production order (red), see Figure 4.17. The transport units start in the "commission site" where they load materials and the required production parts, then, they move to the machines where the production process takes place, and finally coming back to the commission site delivering the final products.

The simulation view visualizes material flow of the production, and it also provides hints about transportation time, transportation paths, waiting times, machine capacity, and supports modeling and simulating the material flow. The simulation explains path utilization and suggests possible layout changes. Users can highlight specific areas and make annotations in the user's color. Those markings and annotations can be shown and hidden in the visualization view.

Status update view. This view in the task model of EDPC must quickly provide an overview of the production progress, indicating potential problems and status of production goals. While the simulation view sheds light on overall production performance, the status update view shows explanations concerning a single order's production progress.

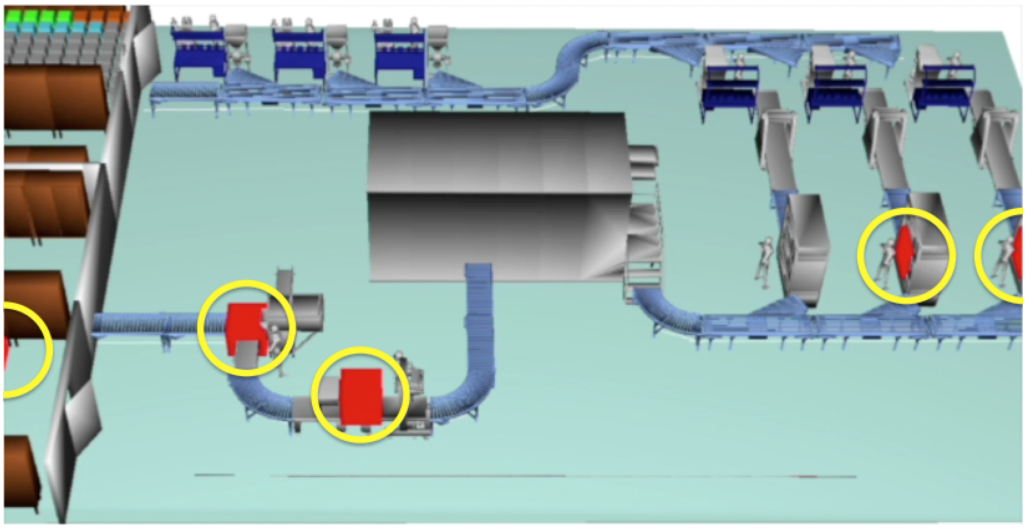


FIGURE 4.17: Transport units in virtual manufacturing system are dedicated to one product order.

It is important that a user has insight into the status of the various status points to be achieved, in our example related to the delivery date for orders and explanations regarding the goal achievement. One glyph represents the data set associated with one order of a customer. Figure 4.18 provides an overview of the glyph design.

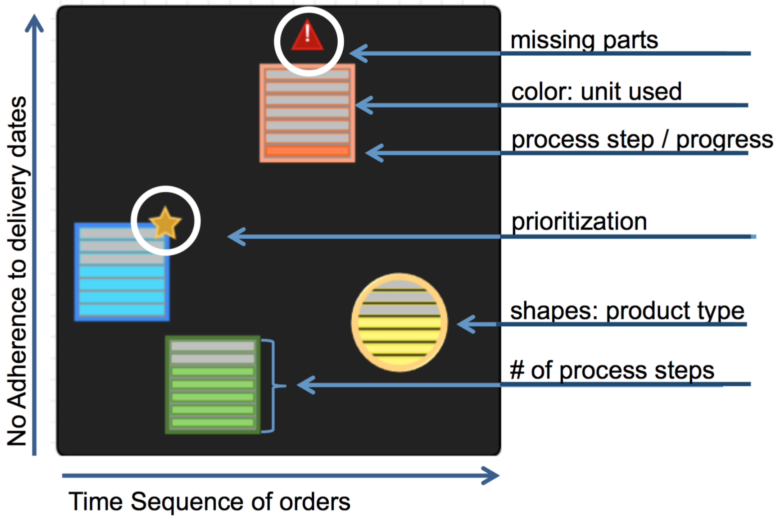


FIGURE 4.18: Glyph design for event-driven production control.

The color of the glyph is identical with that of the associated transport units in the simulation view. The position of the glyph indicates two dimensions of the data set. Positioning along the x-axis reflects the time-stamp of posted orders in sequence in analogy to the reading direction from left to right. Positioning along the y-axis shows status of the planned delivery date. A glyph positioned near the bottom represents the case where the planned delivery date is achieved, based on the accumulated performance; a glyph positioned near the top indicates that is not possible to satisfy the projected delivery. This positioning is based on the analogy to read from top to bottom. Glyphs positioned at the top are critical for the process and have to be detected quickly.

Different shapes of the glyphs reflect different product types. Analogous to the graphical control element status bar, stacked-up bar layers, representing transport units of the order, depict the glyph. The number of layers represents the number of production steps that have to be performed to produce the corresponding product type. Filled bars show the number of steps that have been performed, while unfilled bars show the number of process steps still to be performed. A production step is only feasible when needed parts and material are in stock and ready to be collected in the commission site. Missing parts lead to delays in the production of the order. This condition is indicated with a red triangle on top of a glyph. In general, orders are processed in the same order as they are placed. The first posted order has the first position in the machining process. The transport unit asks for the machining position after picking up materials and production parts. Missing parts or long transportation path can change this order, which can have again an impact on waiting time. To avoid this situation and to process new urgent postings, orders can be prioritized, which defines the transport units associated with that order for the first machining positions. A small star on the top of a glyph indicates this prioritization. Awareness indication of other users is not integrated in the status update view to reduce information over-load and, instead, a clear overview of the ongoing production processes is provided.

Status report view. A tap on a glyph in the status update view opens the corresponding status report view on the smart phone application, (see Figure 4.19).

FIGURE 4.19: Touch input applied to the status update view on the smart watch opens a dedicated status report view on the smart phone.

The status report view, in contrast to the status update view, provides detailed information about one individual order, and potential undesirable behaviors are described together with suggested possible adjustments.

The view is integrated in the smart device interaction application. The user interface is divided into three parts: Main control, order control, and view tab bar, see Figure 4.20. The *main control* contains buttons allowing to post a new order and start/stop the simulation of the material flow in the simulation view. The *view tab bar* contains bar buttons to the existing view widgets. When the status report is not visible, no updates of the report are performed to save computational resources. The status update view, however, is continuously updated and rendered. The *order control* constitutes the main part of the report. The glyph is visualized in the left-upper corner to reflect the relationship to the status update and simulation views. Next to the textual representation of the data set, a color-coded status bar is rendered as a graphical control element, located at the bottom of the interface. Instead of visualizing the progress of an order, the status bar represents the predicted probability of achieving the planned delivery date, predicted based on the accumulated performance. This is done similarly to the positioning of glyphs along axes in the status update view. The color of a status bar indicates the degree to what a desired state has been achieved.

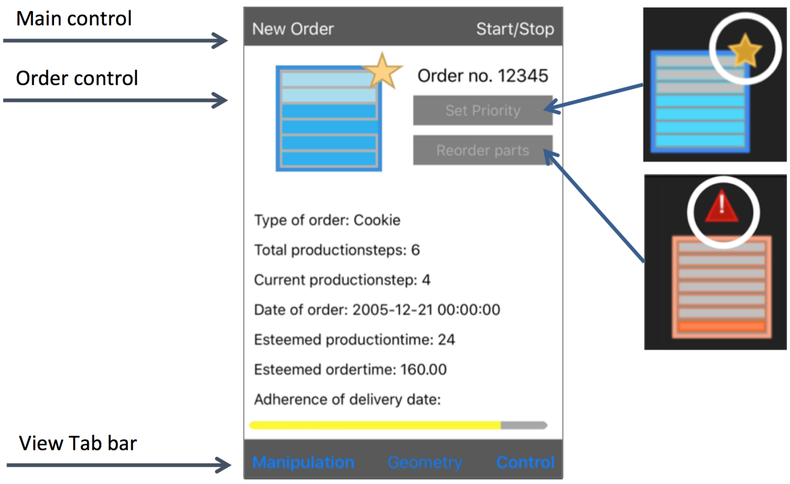


FIGURE 4.20: Status report and dashboard application contains referring custom order glyph, detailed information of the order production, and graphical control elements.

In this view, it is of interest to provide information about users observing the same production process, to avoid redundant re-ordering of parts or provide helpful information for discussion. However, buttons for re-ordering parts and prioritization are enabled when already selected, to ensure that the stock is filled and/or the production process is prioritized. We believe that it might be desirable to highlight single lines in the status report view, which could be synchronized with other users and devices to indicate potential bottlenecks or communicate interesting information.

4.4.2.3 Distributed Collaboration

A possible collaboration session within production planning is sketched as follows. Two users are jointly located in the CAVE-System and register with the system. One user is assigned the task of layout planning (further called planner) and associated the roles of *Factory layout planner basic*, *Manipulator*, and *Collaborator*; the other one is assigned the task of production control (further called controller) and is associated with the roles *EDPC basic*, *Manipulator, Collaborator,* and *Navigator* (R1 ✓). The joint goal is to find a layout of the facilities interior in which the resources are optimally arranged and the production program is optimized. While the planner has information of spacial constraints and environmental conditions like illumination or temperature, the controller has information about the machining sequence, production program, and inventory. For both users, it is important to find the shortest path between the single machining steps. Both users are able to move machines and the controller can furthermore initiate product orders (R3 ✓). The orders are visualized in the simulation view, which enables the user to track the material flow with the current layout. It can be easily identified if machines have to be rotated or if machines are not used and should be probably removed. At any time, production parts can be missing, leading to waiting-phases of the product in the machine. In the simulation view it can just be recognized, that a product is in the machine, but not if it is getting produced. The controller gets a haptic feedback on the watch, if parts are missing. With a quick look on the watch, they can identify problems in the processing and reorder parts to counteract the potential delay (R4 ✓)(R9 ✓). Both users could detect that waiting products in machines blocks the machine for further usage. If following orders are also delayed, the machine describes a bottleneck initiating a demand of action. In joint discussions, in which the ideas and thoughts of both users are considered (R2 ✓) (R4 ✓), they decide to install a waiting area next to "bottleneck-machines" and an alternative conveyor system, so that missing parts can be delivered to the machine. The planner performs the installation of the new resources, which is commented and observed by the controller (R7 ✓) who can track the impacts of these changes as positive or negative influences on the production program on the watch (R8 ✓).

4.4.3 Expert Evaluation and Discussion

In order to verify the system, we performed an expert evaluation in two steps. First, we conducted an assessment based on the property checklist method [152] and afterwards performed the method of assessment of experience [61] in close collaboration with three domain experts.

4.4.3.1 Property Checklist

The method of property checklist is a structured way to do an evaluation, in which the expert goes through a checklist of design goals for different product properties. In our case, the product properties correspond to the ten stated requirements in Chapter 2.2 summarized in Section 4.4.1. As described in Section 3, all requirements, except for **R5, R6,** and **R10** are realized and the users are facilitated to perform collaborative work in an efficient and natural manner.

- ✓ [**R1**] Member can coordinate activities.
- ✓ [**R2**] Member can communicate with each other.
- ✓ [**R3**] Member can perform activities.
- ✓ [**R4**] Member can switch between collaboration styles.
- ✗ [**R5**] Member is satisfied with/in group work.
- ✗ [**R6**] Members expectations are accomplished.

- ✓ [**R7**] Member can determine others' activities.
- ✓ [**R8**] Member can understand the impact of changes made.
- ✓ [**R9**] Member can make adjustments based on impact.
- ✗ [**R10**] Member can exchange information.

4.4.3.2 *Qualitative Assessment*

Nestler et al. [212] proposed a qualitative assessment approach used when a comparative evaluation cannot be performed, and when baseline efficiency values/rates cannot be used to benchmark the system. No other existing setup could serve for benchmarking when less functionality and interaction capabilities are provided. If one were to compare the usability of the proposed multi-modal interface with the usability of the earlier EDPC system, the new approach would be inferior. Therefore, the evaluation method of Nestler et al. is a viable alternative, based on a reliable questionnaire for assessing technology acceptance. According to Nestler et al., general considerations useful for qualitative usability evaluations are: (1) Most usability problems are detected with three to five subjects. (2) It is unlikely that additional subjects reveal new information. (3) Most severe usability problems are detected by the first few subjects. Due to the limited number of available experts in our domain, we involved three participants for the qualitative usability evaluation.

In close collaboration with three EDPC experts, we performed an experimental user study. The experts performed the collaboration session as described in Section 3. Each expert performed the experiment three times - two times with two participants involved, and once with three participants involved, leading to four experiments. The number of participants was not sufficient to gather significant results concerning efficiency or effectiveness of the prototype. Nevertheless, we were able to gain insight into general users' experiences with our system. We measured a general usability score U as introduced in [212]. After performing the experiments, we solicited expert feedback using questionnaires as proposed in [212] concerning four categories, see [61]: (1) ease-of-use; (2) user satisfaction; (3) usefulness; and (4) intention to use the system. For the qualitative evaluation, the interviewer used open-ended questions and did not interrupt the subject. The aim of the interview was to discuss all perceived problems with the subject in order to detect usability issues.

4.4.3.3 *Quantitative Assessment*

According to [197] the qualitative results of the assessment provide a performance quantification basis, resulting in a scalar usability value U. The usability categories are adjusted on a three-point scale: (1) positive comment (1.0); (2) neutral comment (0.5); and (3) negative comment (0.0). The mean of the values is calculated. We obtain a quantitative rating of all categories on a scale from 0.0 to 1.0. Moreover, the usability categories are weighted, summing up to 100, to express the importance of a category in a specific application. Formally, the usability score is calculated by multiplying all weights $w(s)$ with the quantitative scores $v(s)$, where the value of U is between 0.0 and 1.0:

$$U = \sum v(s) \cdot w(s) = \sum U_c \cdot \frac{w_c}{100} \tag{4.8}$$

4.4.3.4 *Results*

During the experiment execution, a brief look at the watch provided insight into the production process, allowing experts/users to determine whether actions were needed to interfere or not. At the same time, the team-members could observe the material flow in the simulation view, detect potential bottlenecks, and make suggestions for an optimized factory layout. As both task models of EDPC and factory layout planning are integrated in the prototype, the experts were able to adjust the layout and track the impact on production on the watch (at a glance), and establish priorities and initiate re-ordering of parts with the smart phone application. It was simple to stop/pause the simulation view and balances the current outcomes. Furthermore, the experts could leave the physical setup and track changes on the watch for monitoring purposes due to the WIFI connection with the server and database system.

(1) Ease-of-use. Since our experts were familiar with the general control capabilities of smart devices they could focus on the task instead of the control mechanisms. The implemented user interface elements are easy to learn and match the dedicated functionalities. Using the watch to get immediate feedback of the progress and potential delays was stated intuitively. The users were able to recognize if they had to intervene in the process or if they could focus on the group-work. Switching between group- and individuals' tasks was performed smoothly and without interruptions of others' performances. Only the interaction capabilities with the smartphone could be enhanced with more intuitive techniques.

(2) User satisfaction. Each user in the evaluation could control the scene within the restrictions of his task and was able to actively participate in the collaboration. The users did not feel strained using the collaboration environment and could focus on solving their tasks. On the contrary, the users took delight in using the system and felt immersed in the scene. Overall, each user was satisfied, which increased the motivation of the users to perform the tasks and the collaboration in the team.

(3) Usefulness. The system supported the users to perform both group and individuals' tasks of both use cases and facilitated the recognition of impacts and the cooperation between the users. We have identified several questions that could be answered by using our system in this specific application domain. Example questions are:

- How many orders do exist?
- Where do you spot potential hazards?
- How many production steps does an order have?
- How many missing parts do exist?
- Which order should be prioritized?
- How many priorities exist?
- Do orders have the same number of production steps?
- How many steps have been performed already?

- Which order was placed most recently?

The usefulness of the system for the specific use case and for collaboration activities was demonstrated.

(4) Intention-to-use. All experts expressed their desire to use our setup in the future, recognizing our system's value for a planner when checking on production status and intervening to optimize it. One participant provided a neutral comment, stating that the setup was too sophisticated for the problems he was concerned with.

The resulting quantitative scores and weights leading to a usability value $U = 0.902$ are summarized in Table 4.4:

TABLE 4.4: Weights and scoring of usability categories to calculate usability score U of the system.

Category	(1)	(2)	(3)	(4)
$w(s)$	0.25	0.25	0.25	0.25
$v(s)$	0.833	1.0	1.0	0.833
$w(s) \cdot v(s)$	0.201	0.25	0.25	0.201
$\sum$	0.902			

The qualitative and quantitative assessment results shown are quite good. Summarizing, the requirements **R5** and **R6** are fulfilled.

✓ [**R5**] Member is satisfied with/in group work.

✓ [**R6**] Members expectations are accomplished.

It is possible to conclude that our framework leads to a more efficient and successful collaboration in the case of factory planning. Due to the modular concept of the framework, different scenarios and task models can be integrated, facilitating collaborative work and decision-making in other domains.

As such, a setup is designed that increases the efficiency of collaborative design and assessment in virtual reality. Adequate visualization techniques have been applied and proven in a collaborative task. In the following, we will focus on the aspect of active interaction and participation. Thus, novel natural and intuitive interaction techniques are developed.

4.5 Multi-Modal Interaction for Collaborative Design and Assessment

As a reminder, VR visual interaction environments make possible the sensation of being physically present in a non-physical world. The benefit of this experience is a better perception and comprehension of complex data, based on simulation and visualization from a near-real-world perspective. A user's sense of immersion, the perception of being physically present in a non-physical world, increases when the used devices are efficient, intuitive, and as "natural" as possible. The most natural and intuitive way to interact with data in a VR environment is to perform the actual real-world interaction. For example, gamers are typically clicking the same mouse button to swing a sword in different directions. However, the natural interaction to swing a sword in a VR application is to actually swing the arm in the physically correct direction as the sword is an extension of the user's arm. Therefore, intuitive and natural interaction techniques for VR applications can be achieved by using the human body itself as an input device. VR devices are usually specialized to support one interaction modality used only in VR environments. Substantial research has been done in this field as described in Section 4.1, yet VR input devices still lack highly desirable intuitive, natural, and multi-modal interaction capabilities, offered at reasonable, low cost.

4.5.1 Methodology of Body Movement Gestures in VR Environments

In this section, a multi-modal interaction interface is introduced, implemented on a smartwatch and smartphone for fully immersive environments. We use a head-mounted display (HMD) for a high degree of immersion. Our approach improves the efficiency of interaction in VR by making possible more natural and intuitive interactions. We have designed and implemented methods for seven gestures and evaluated them comparatively to common VR input technology, specifically body tracking enabled by a 3D camera. We present our approach initially from an application-independent perspective. Later, we demonstrate and discuss its adaptation and utilization for a real-world scenario of factory planning, as shown in Figure 4.21 and Figure 4.25. Current research covers many aspects of interaction in VR, being of great interest to our work. Similarly, there have been several investigations concerning interaction techniques with wrist-worn devices such as smartwatches and fitness trackers as mentioned in section 4.1. However, present literature does provide very little insights about eyes-free interaction in VR as well as combination of VR technology, which is crucial when it comes to the utilization of HMDs as an interface to the virtual world. With this research, we go one step further in closing this gap, employing everyday available low-budget hardware.

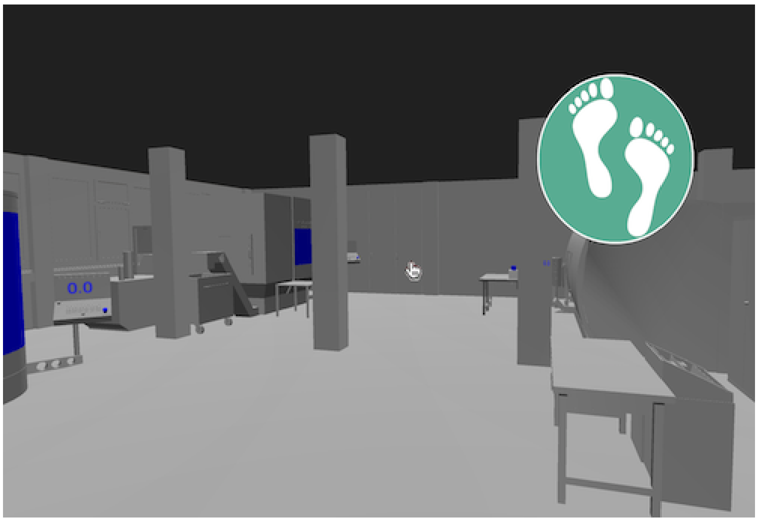


FIGURE 4.21: Virtual plant floor as seen through the HMD during the user study.

4.5.1.1 *Conceptual Design*

Our approach uses common technologies, at relatively low cost, supporting intuitive, basic interaction techniques already known. A smartphone fixed in an HD viewer serves as fully operational HMD and allows one to experience a virtual environment in 3D space. The smartphone holds the VR application and communicates directly with a smartwatch. Wearing a smartwatch with in-built sensors "moves" the user into the interaction device and leads to a more natural interaction experience.

In order to support control capabilities to a great extent, we consider all input capabilities supported by the smartphone and the smartwatch. In addition to touch display and smartwatch crown, we considered accelerometer, gyroscope, and magnetometer, as they are built-in sensors. In discussions with collaborating experts, we determined what types of interaction could and should be realized with the input devices and their capabilities. As the smartwatch has a small display and a user cannot see it, touch input is only used for inaccurate gestures (tap). Most smartwatches have several integrated sensors, e.g., to trace orientation and motion. To obtain platform independence, we decided to focus on accelerometer data as a feature of all smart devices during design and implementation of our system. We designed seven distinct gestures dedicated to VR modes of orientation, movement, and manipulation.

We built two setups to enable body gesture interaction. While the first setup relies on body tracking based on a 3D camera, the second features a smartwatch and its built-in sensors as basic interaction component. To make the two approaches fully comparable, the underlying concept of both setups is the same: while hands-free gestures are used to interact within the virtual environment (VE), a HMD provides visual access to the virtual world. The input devices used to capture gestures differ in flexibility and have different limitations discussed in the following sections.

4.5.1.2 *Setups*

For both setups, we decided to use a smartphone, the Apple iPhone 6+, in combination with a leap HD VR viewer. The smartphone is fixed in the viewer, which, in combination, is fully operational as HMD and allows users to experience a virtual environment in 3D space.

(A) 3D camera setup: Asus Xtion Pro Live 3D camera and VR viewer fixing iPhone 6Plus (left). Viewing angle of camera limits user's movement ability (right).

(B) Watch setup: Apple watch sport 38mm and VR viewer fixing iPhone 6Plus (left). Allows usage of the entire physical space for a user's movement (right).

FIGURE 4.22: 3D camera and watch setup in comparison of user's movement abilities.

Camera Setup. Our 3D camera-based configuration essentially requires two components: (1) A 3D camera, Asus Xtion Pro Live, tracks a user's skeleton posture and provides the system with a continuous stream of RGB color images and corresponding depth images and (2) an HMD. The 3D camera is tethered to the main system. A user must remain in small distance to and in field of view of the camera, to be tracked entirely. The tracking radius and the minimal distance of the user enforce a narrowed range of allowable movement, see Figure 4.22a. More specifically, the camera features a 58 horizontal and 45 vertical field-of-view while the tracking distance ranges from .8m to 3.5m. Another limitation must be applied to a user's orientation to ensure accurate tracking. A user must face the camera to avoid occlusion, preventing the possibility of misinterpretation of body parts or gestures.

Watch Setup. Our watch setup consists of two components: (1) A smartwatch, the Apple Watch Sport 38mm Generation 1 and (2) an HMD. The watch's dimensions are 38.6mm x 33.3mm x 10.5mm. Neither watch nor HMD are tethered, and there is no technical limitation to the tracking area. Also, the battery is not a limiting factor in our investigation. A user's range of movement is defined by the actual physical space, see Figure 4.22b. One considerable limitation is the fact that body movement gestures are limited to one arm. This limitation implies that all other body parts cannot be utilized for gesturing. Body movements and gestures involving more body parts, like legs, both arms, or torso, could enable a more natural user interface experience.

4.5.1.3 *Software Design and Implementation*

Camera Gesture Recognition. In order to enable the system to detect gestures, a framework combining OpenNI 2 [218] with NiTE 2 [220] was designed. While OpenNI handles low-level image processing requirements, NITE serves as a middleware library for detecting and tracking body postures. It supports an easy-to-extend gesture detection framework. Gesture recognition is algorithmically handled via a finite state machine (FSM). Each detectable gesture is represented by a corresponding sequential FSM. In order to trigger the detection of a particular gesture, one or more user's detected joints are tracked in a certain absolute position and/or relative position to one another. When a body posture indicates a starting condition of an implemented gesture, the system continuously checks for subsequent satisfaction of additional states of the underlying FSM. Once the FSM reaches its final state, the associated gesture is considered as complete. In addition to the gestures available in NITE, we expanded the system by adding several new gestures to satisfy additional needs. For detection, it was crucial to design the additional gestures in such a way that they do not interfere with each other.

Accelerometer-based Pattern Recognition. Smart watch and smartphone are connected in our framework via Bluetooth, making possible a continuous communication. Accelerometer data collected by the watch are communicated to the phone that computes and detects defined gestures, making use of the smartphone's computation power. It is challenging to devise an algorithm to transform the raw stream of accelerometer data into explicit gestures. Gestures should not interfere with each other, and the system must compute and detect gestures in real time. The resulting data stream to be transmitted as well as the computation time required for data processing can lead to potential bottlenecks. Applying a low-pass filter to the data stream and dedicated gesture patterns makes it possible to detect necessary changes and to reduce the "jittering" of the watch greatly . Thus, the system can effectively distinguish between gestures, which are described in the following.

4.5.1.4 Interaction Mechanisms

Both setups support the same application, but they differ in input mechanisms. The application is created with Unity3D [292], which is a cross-platform game engine. VR interaction modes can be grouped into (1) orientation, (2) movement, and (3) manipulation modes.

(1) Orientation is implemented through head-tracking. A user can look around for orientation. The smartphone uses built-in sensors, like accelerometer and gyroscope, to determine orientation and motion (of the devices), permitting translation, done by the game engine, into the user's viewpoint in a virtual scene.

(2) Movement is implemented by two interaction techniques: (a) In the watch setup, a user looks in walking direction, and single-touch taps the watch to indicate begin or end of movement. (b) In the 3D Camera setup, a user "walks on the spot", see Figure 4.23a.

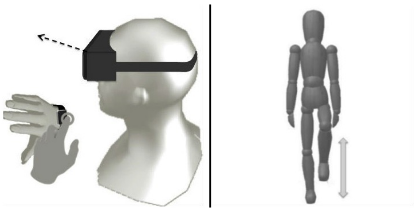
(3) Manipulation refers to the interaction with objects in a scene. For example, we designed and implemented the following six additional body movement gestures: swipe in left and right direction; vertical shaking; circle gesture; slider-value setting; and button push, see Figures 4.23b - 4.23g.

4.5.1.5 User Study

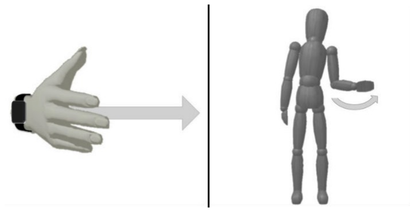
In order to find out to what extent working with the 3D Camera-based environment compared to the watch-enabled setup has an effect on a user's task performance, we conducted a preliminary user study. While performing the experiment, the user is located in a virtual environment of a factory building. This building is an accurate 3D model of a machine hall existing in the real world.

Design. There were a total of 20 participants (5 females and 15 males) taking part in the evaluation within an age range between 20 to 32. While all of them were used to work on a computer on regular basis, only a few of them had any prior experience concerning HMDs and VRs. Each participant performed the experiment in both within subject design setups. Half of the user group began evaluating the 3D-Camera setup while the other half started in the smart watch environment in order to cancel out learning effects while the assignment occurred randomly. Subsequent to the experiment, the participants were asked to fill out a questionnaire consisting of 24 questions considering their user satisfaction.

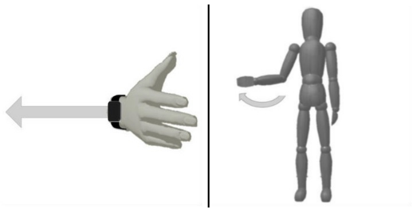
Realization. In the course of the experiment the subjects were asked to perform several authentic tasks in VR, which are all performed by actual field experts in real life on a daily basis. In total, we considered five machines (stations) in the virtual factory and realistically mapped their control to a sequence of gestures to be performed by the evaluation participant (see Figure 4.24). Table 4.5 describes the tasks and gestures of all stations. When users reach the machine, they are asked to perform the tasks in the sequences as described in Table 4.5. After having all gestures recognized in the correct way and order, all tasks on the station are considered as finished.



(A) **Walk** gesture for watch setup (L) and 3D camera setup (R).

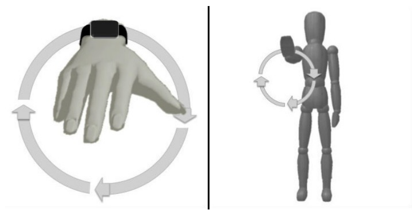


(B) **Swipe-left** gestures. (L) Watch display faces wall; moving arm horizontally, first in left and then in right direction. (R) Perform swipe gesture with left arm.

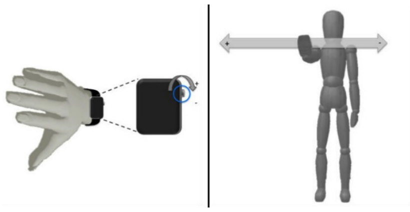


(C) **Swipe-right** gestures. (L) Watch display faces wall; moving arm horizontally, first in right and then in left direction. (R) Perform swipe gesture with right arm.

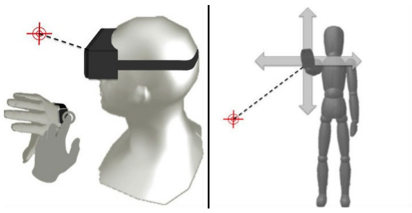
(D) **Vertical shaking** gesture. (L) Watch display faces wall; fast arm movement in vertical direction. (R) Fast arm movement in vertical direction with right arm.



(E) **Circle** gesture. (L) Watch display faces ceiling; arm movement in small circles clockwise. (R) Arm movement in big circles clockwise.



(F) Gesture for **value setting**. (L) Using scroll wheel of watch; accepting value by tapping on watch display. (R) Sprawling out right arm; moving in horizontal direction sets value; holding position for three seconds.

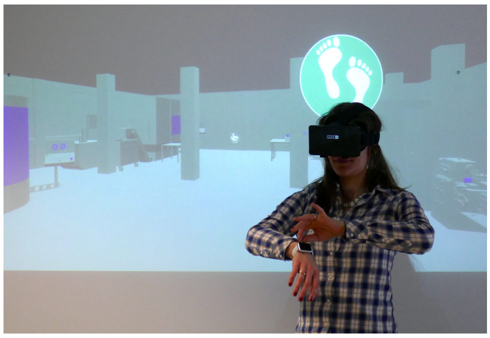


(G) **Push** gesture. (L) Small point symbolized center of viewpoint; position point on object; approve by tapping on watch display. (R) Sprawl out right arm; cursor symbolized hand position; position hand on object; hold position for 3 seconds.

FIGURE 4.23: Body gestures for orientation and manipulation in VR.

For the purpose of keeping the whole experimental scenario as realistic as possible, the subjects had to virtually walk to the next station in the sequence before they were able to perform the necessary gestures. Hence, it was possible to perform the whole experiment in one go, without having the users distracted or having their level of immersion lowered. As soon as the user reaches a specific station, they are standing in front of the corresponding machine in VR. Since we wanted the

(A) User performs swipe gesture.



(B) User performs value setting gesture.

FIGURE 4.24: User performs gestures in user study. User's view is mirrored to a wall in background in order to provide positional information to experiment instructor.

distraction and external input to be as low as possible, the users were supported by a pictogram illustrating the gesture that had to be performed at that moment. After completion of a sub-task, the pictogram instantly displays the upcoming task. In order to investigate possible differences between the two setups in terms of task performance, we documented the time a participant needed to complete the tasks in each station (i.e., completion of all corresponding gestures). Note that the measured times does not include walking from one station to another.

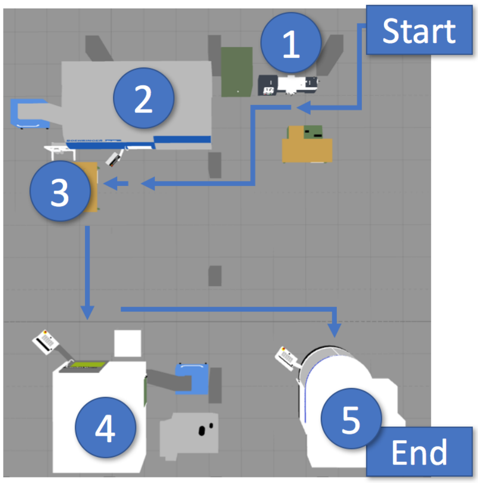


FIGURE 4.25: Top view of plant floor consisting of 5 stations.

Results. Since each participant performed both of the experiments (within-subject design), we performed a paired t-test on the measured times of both, each station separately and cumulated execution. We then tested the null hypothesis (H_0: there is no significant difference between the given setups) for its tenability with each condition. Regarding the times of stations 1, 2 and 5 exclusively, we found no significant difference in task performance, meaning that the task performance in both setups was equally good, therefore we can not reject the hypothesis H_0. However, we found a significant effect at the stations 3 and 4 solely, as well as in the total time, with the watch setup outperforming the 3D-Camera setup. The complete results are stated in Table 4.6.

As a result, we have a significant difference in task performance in the above cases, which allows us to legitimately reject the null hypothesis H_0. Therefore we can state

TABLE 4.5: Gestures translated in user tasks for all stations.

STATION 1 - WENDING MACHINE
1. Winding gesture to rotate the workpiece inside the machine
STATION 2 - TURNING MACHINE
1. Winding gesture to rotate the workpiece inside the machine
2. Swipe right to close the sliding door
3. Sliding gesture to set a specific value at the control panel of the machine
4. Push button gesture to start the machine
STATION 3 - HAMMER
1. Hammer gesture to clamp the workpiece
STATION 4 - MILLING MACHINE
1. Swipe left to open the machine's sliding door
2. Hammer gesture to clamp the workpiece
3. Swipe right to close the sliding door
4. Push button gesture to start the machine
STATION 5 - GRINDING MACHINE
1. Swipe left to open the machine's sliding door
2. Winding gesture to rotate the workpiece inside the machine
3. Hammer gesture to clamp the workpiece
4. Swipe right to close the sliding door
5. Sliding gesture to set a specific value at the control panel of the machine
6. Push button gesture to start the machine

TABLE 4.6: Efficiency results comparing smartwatch setup with 3D camera setup.

Station	t(19)	p	Cohen's d
Station 1	0.12814	> 0.05	0.02865257
Station 2	0.84168	> 0.05	0.1882063
Station 3	6.7031	< 0.05	1.498849
Station 4	2.8739	< 0.05	0.6426245
Station 5	1.1956	> 0.05	0.267355
Total	2.4031	< 0.05	0.5373388

that the interactions performed with the watch setup are equally good or better than the performance with the 3D camera setup. We could not find a significant difference at stations where the circle gesture was performed. A possible explanation could be found in the questionnaire: the only gesture subjects preferred within the 3D-camera setup over the watch setup was this particular circle gesture. With respect to the questionnaire, there were some interesting findings. Although, it has been assured, that the gestures for both setups are equally comfortable, natural, and intuitive for the users, 5 gestures were more preferred to perform with the watch setup (walk, push, value setting, swipe left, and vertical shaking). The swipe right gesture performance is nearly identical in both setups, which is also confirmed by the questionnaires. Overall, there was a low degree of motion sickness with no significant difference in both setups. These findings lead to the justified assumption that the novel approach presented in this section is at least as good as currently used techniques.

Thus, the combination of smartphone and smartwatch capabilities provides the potential to outperform a comparable common VR input device. We have demonstrated the effective use for a simple application. The full immersive system as combination of smartphone and smartwatch contributes advantages for highly effective and intuitive gesture-based interaction as following:

- Location independence;
- Simplicity-of-Use;
- Intuitive usability;
- Eyes-free interaction capability;
- Support for several different inputs;
- High degree of flexibility;
- Potential to reduce motion sickness;
- Elegant combination with existing input technology.

4.5.2 Body Movement Gesture Recognition for Low-Cost Technology

In this section the accelerometer-based pattern recognition from the last section is enhanced and the adaption of enhanced gestures is improved. the Apple Watch (watchOS2) is used for the investigation to develop different arm movement gestures to enable new natural interaction mechanisms. Unfortunately, the only sensor data that can be extracted from the Apple Watch are the acceleration data as gyroscope and magnetometer was not accessible at the time of our investigation. The above-mentioned sensors are used to calculate the orientation and motion dynamics of the device. The challenge is described by missing sensor data, precisely gyroscope and magnetometer data, which need to be compensated in order to calculate orientation and motion dynamics of the device.

The aim of this research is to create a system that is able to recognize arm gestures only using the accelerometer data of the device. This system has to allow people, skilled in programming, to define their own gestures. Based on six key values and a statement sequence we are able to define precise arm movement gestures, demonstrated with seven different gestures. More gestures are conceivable and easily adoptable with our approach. Subsequently, we are able to transform the given signal-processing from the smartwatch into arm movement gestures with the use of smoothing algorithms and gesture state machines which lead to the actual gesture recognition.

4.5.2.1 *System Model*

In order to describe the system model, a gesture is defined as a pattern of wrist movements. Patterns of interest are characterized as intentionally performed, easily memorable, and easily performed by a wide range of users. In the system model a pattern is recognized by a sequence of states based on key information of the sensor data. As such, we will further describe the concept of state machines, the extracted key values from the sensor data, and corresponding states that are used to define a gesture.

State Machine. A state machine is a model that describes the output of a control system based on the incoming stimuli from the past. States represent all possible situations in which the state machine may ever be. The incoming inputs and the sequence of inputs in the past determine the state in the system and lead to the corresponding output of reaching that state [303]. The visual definition of a state machine is depicted in Figure 4.26. If the number of distinguishable situations for a given state machine is finite, the number of states is finite and the model is called a finite state machine.

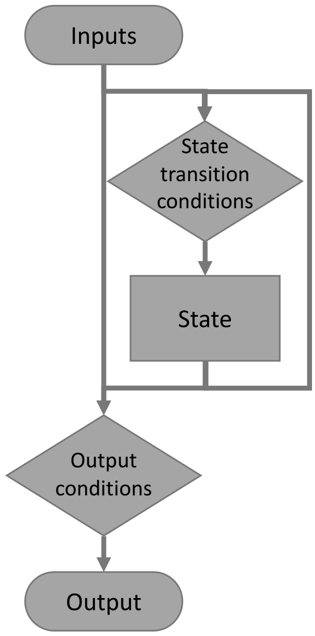


FIGURE 4.26: Definition of a state machine according to [303]

Key values. Every time a state machine gets updated, the following key values are used to determine the state and corresponding transition:

- Direction of Acceleration
- Direction of Velocity
- Direction of Gravity
- Value of Acceleration
- Value of Velocity
- Time

In each frame the extracted sensor data are sent to all state machines (SMs) as new incoming inputs. In general, an update of the SMs occurs every time new input arrives; some SMs only get updated if a key value changes. An update of the SM does not imply a state change.

Two different types of SMs are used in the system model. The first type of SM defines states based on segment positions. The second type of SM defines states based on the number of reached segments. For the purpose of the definition, they are both true SMs, but we can use a lot less states this way, as some gestures do not need an exact position but a specific number of direction changes. If one gesture is recognized, all SMs get reset and a message is sent to execute the desired interaction in the VR environment. An example state machine is depicted in Figure 4.27.

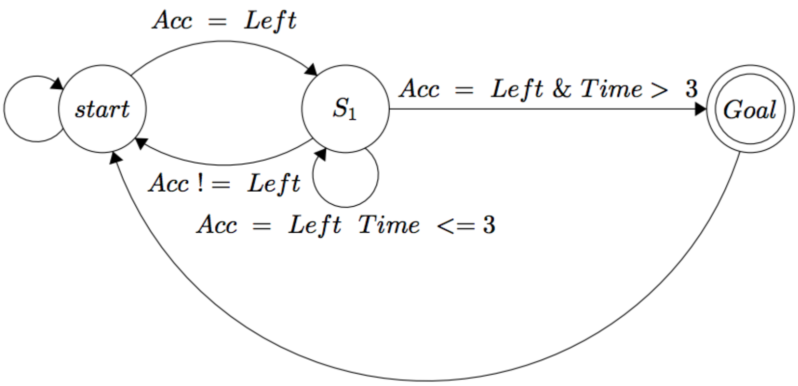
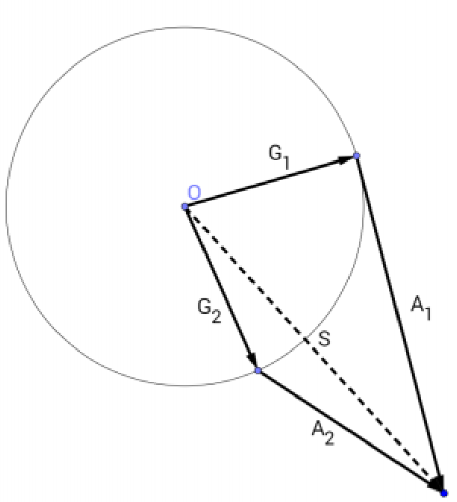


FIGURE 4.27: A State Machine that recognized acceleration to the left, that is present for an extended period of time

As condition of our approach, we merely use the patterns generated by a single 3-axis accelerometer and try to extract the information needed to define gestures. The key values are calculated based on the following sensor data: (1) gravitation, (2) acceleration, and (3) velocity. The gravitation value is extracted in order to be used as reference of the watch's posture with which we can align the sensor data and ultimately because the gravitation is polluting the sensor-data. Acceleration data is used to compute the path of the wrist in 3D space monitored over time. The velocity is derived from the acceleration data and used to define additional state transition conditions. An overview of all forces and their dependencies are depicted in Figure 4.28.



$\vec{S}$ = SensorData
$\vec{G}$ = Gravitation
$\vec{A}$ = Acceleration
$\vec{V}$ = Velocity

What we know
$\vec{S} = \vec{G} + \vec{A}$
$||G|| = 1$

FIGURE 4.28: Key values of gesture recognizer. Different $\vec{G}$ and $\vec{A}$ add up to $\vec{S}$

If we measure $\vec{S}$ then we know that $\vec{G}$ also points in the same direction, meaning $\vec{G} = \frac{\vec{S}}{||S||}$. However, to adopt the direction of $\vec{G}$ deriving from $\vec{S}$ is only correct if the watch stands still and is not moved in any direction. Therefore, it is challenging to find the right moment to calculate the direction. Our approach is to implement an adoption-rate describing the degree of how much we trust in $\frac{\vec{S}}{\|\vec{S}\|}$ to be the same as the actual/real gravitation force with the following steps:

$$\vec{G}_{new} = \vec{G}_{old}(1 - weight) + weight\frac{\vec{S}}{\|\vec{S}\|} \tag{4.9}$$

$$weight = 0.3 * \exp(-(\|\vec{S}\| - 1)^2 * 14) \tag{4.10}$$

The $weight$ distribution corresponds to a Gaussian bell curve, which has its peak in $\|\vec{S}\| = 1$. This function guarantees that $\vec{G}$ is rapidly corrected once the user stays still and that the changing rate from $\vec{G}$ is lowered while gestures are performed. Computations according to Equation 4.9 are performed every frame with a rate of 20 frames per second; after merely 8 frames G_{old} is only covered by $(1 - 0.3)^8 = 5{,}7\%$ if $\|\vec{S}\|$ is close to 1. After computing $\vec{G}$ we know:

$$\vec{A} = \vec{S} - \vec{G}_{new} \tag{4.11}$$

This approximation still does not consider the position of the smartwatch on the wrist and therefore $\vec{G}_{new}$ could be pointing anywhere. Taking this into account, we apply a rotation R to $\vec{A}$ with the property (0,0,-1) = $\frac{\vec{G}_{new}}{\|\vec{G}_{new}\|} \times R$. Thus, the robustness of the gesture recognition is enhanced, as it is independent on which wrist the smartwatch is worn.

FIGURE 4.29: 28 defined sectors on the sphere correspond to states used by the state machines.

States. In order to define states for the state machine in an easy processable form, areas on a sphere are defined into sectors. Every vector is transformed into an identifier representing if this vector lies in that sector. Figure 4.29 depicts the defined sectors of the sphere. In total we defined 26 sectors: 5 sectors in longitude axis, whereby the 3 middle slices are divided into 8 sectors in latitude axis.

Tested in a preliminary study, we figured out that those 28 sectors are the optimal number of segmentations, which are comfortable to reach and provide an adequate number of possible permutations, and therefore, gesture states.

Based on the defined sectors, the following refined denotations of the key values are used:

Key values

$$Acceleration = \text{Sector that } \vec{A} \text{ points to;}$$
$$Velocity = \text{Sector the } \vec{V} \text{ points to;}$$
$$Gravity = \text{Sector the } \vec{G} \text{ points to;}$$
$$||Acceleration|| = \text{Value of Acceleration} = \|\vec{A}\|;$$
$$||Velocity|| = \text{Value of Velocity} = \|\vec{V}\|.$$

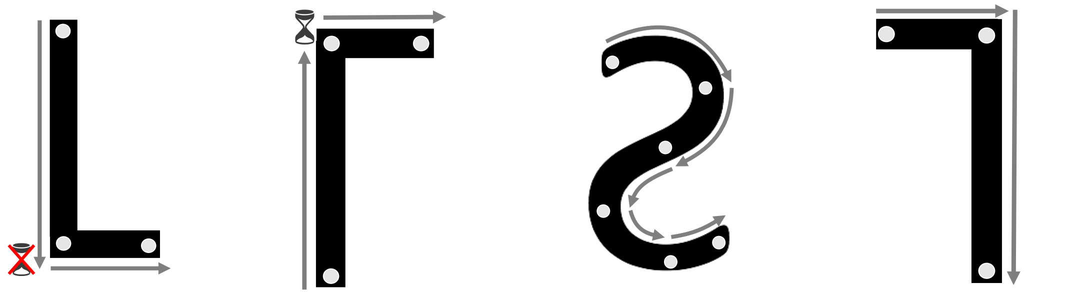
Next, due to the keen accelerometer sensors and repetitive assimilation of data inaccuracies over time, the so-called drift of the computed acceleration occurs. This well known problem appears by using sensors without the ability to re-calibrate. Multiple factors lead to this inaccuracy. The system is often slightly lagging behind a movement and hence it computes faulty values. If the hand is rotated by 180 degree, the sign of the number changes. As the systems sensor is slightly lagging behind, the sign shift is not recognized for a short time. But we can not avoid the sensor lag, as we cannot trust that the direction of $\vec{G}$ and the direction of $\vec{S}$ is the same. As defined $\vec{S} = \vec{A}_{real} + \vec{G}$ and $\vec{A}_{polluted} = \vec{S} - \vec{G}$, for that short moment, one has $\vec{A}_{polluted} = \vec{S} + \vec{G} = \vec{A}_{real} + 2 * \vec{G}$, until $\vec{G}$ gets adjusted. Furthermore, over time the $\sum \vec{A} \neq 0$, and therefore the velocity of the object, also is $||Vel|| \neq 0$. In long movements the velocity drifts extremely as the sensor can not be calibrated $\vec{G}$ is not adjusted. It can not be stressed enough that rotating the wrist causes anomalies due to above mentioned problem, which implies that $\vec{A}$ and $\vec{V}$ can not be trusted for 1/2 of a second after a full rotation is performed. To fix the drift problem, the following adjustment to the velocity vector is performed in every frame:

$$\begin{aligned} \vec{V}_{new} = &\vec{V}_{old} * reductionFactor_1 + \\ &\vec{A} * timeFactor - reductionFactor_2 \end{aligned} \tag{4.12}$$

Hence, we achieve a higher accuracy of the gesture recognition.

4.5.2.2 *Gesture Definition*

We defined seven gestures that cover the full range of possibilities to evaluate the gesture recognition algorithm. Hereby, it is implied that single states are recognized as well as sequences of states and changes of the key values. These gestures also try to prove that a series of movements can make up a recognizable pattern. We show this for a realistic amount of steps. The classification of the gestures follows along the attributes of movement and shape. We differentiate motion between continuous and partitioned movements. If a gesture is performed without breaks, it has a continuous movement while partitioned gestures are made up of a series of continuous sub-gestures with sufficient breaks in-between. The second differentiation between gestures is related to the gesture form. Gestures have either a curvy or angular paths describing their shape. Figure 4.30 depicts all four gesture forms.



(A) Continuous movement (B) Partitioned movement (C) Path follows curved shape (D) Path follows angular shape

FIGURE 4.30: Taxonomy of gestures along the attributes movement and shape.

Following the classification along the attributes movement and shape, the seven gestures can be described with the categorization in Table 4.7:

TABLE 4.7: Classification of the designed gestures into movement and shape.

	Circle	Shake	Swipe	Hamme	Z	Lever	Ladle
Curvy	√		√	√			√
Angular		√			√	√	
Continuous	√	√		√	√		
Partitioned			√			√	√

As shown in Table 4.8, every key value has at least been used twice and in combinations with other values that correspond to a particular gestures. Additionally the "Amount of Steps" indicates how many different motions have to be performed in series for this gesture to be recognized. We created gestures that have multiple steps, which demonstrates the capability of the system to recognize series of motions[2].

TABLE 4.8: The usage of key values in the gestures and the amount of states that have to be transitioned.

	Circle	Shake	Swipe	Hamm	Z	Lever	Ladle
Acceleration	√	√	√	√	√		
Velocity			√	√			
Gravity			√			√	√
‖*Acceleration*‖			√	√			
‖*Velocity*‖			√	√	√		√
Time		√	√				√
# *steps*	5	4	2	2	4	3	6

In the following, we give detailed information of each gesture with the aim to transfer the knowledge so that users are able to create their own gestures. For better readability, the following diagrams are simplified into a 2-dimensional abstraction of the real state machines.

The *Circle Gesture* is defined with continuous movement and a curvy shape, see Figure 4.31. This gesture seems to be intuitive and is supposed to be used when something has to be rotated. In order to perform this gesture users have to perform

[2]Swipe only uses the *Gravity* value to forbid rotating

a clockwise circular motion with their arm. The corresponding SM counts to five transitions, which is at least half of a circle. Start point of the circle can be at any point and due to the self-recovering nature of this particular SM, the recognition works always if the user does not stop circling. Axis and direction of turn can vary in the definition of the SM. The SM shown in Figure 4.31 is not parametric.

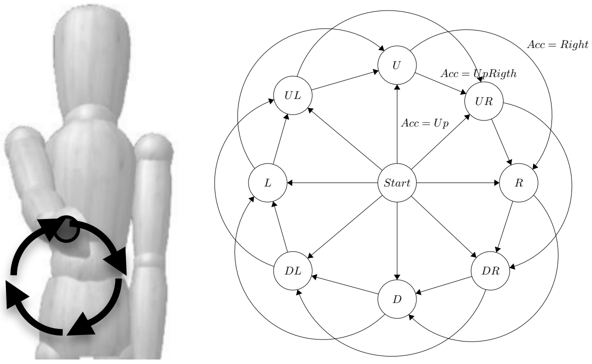


Figure 4.31: Motion and state machine of *Circle Gesture*: Every arrow that points to state "L" has the condition $Acc = Left$.

The *Shaking Gesture* is defined as a continuous movement with an angular shape. This intuitive gesture is an analogy to shake things. An example would be shaking a dice cup. In order to perform this gesture the executed motion is described by an alternating up and down of the user's arm. The *Shaking Gesture* can either be defined in vertical direction or horizontal direction. The corresponding state machine is depicted in Figure 4.32 and shows the definition of this gesture in vertical direction by counting transitions between up and down. The state machine is parameterized in a way that each two sectors on opposite directions can be used.

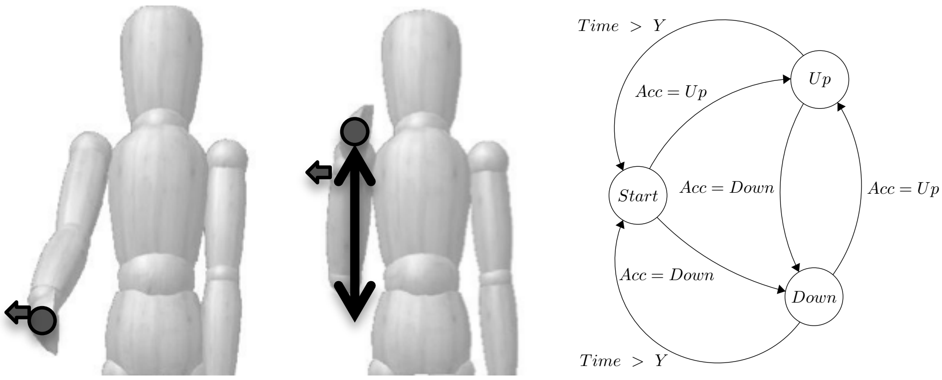


Figure 4.32: Motion and state machine of *Shaking Gesture*; the gray arrow indicates the direction of the watch.

The *Swipe Gesture* is defined as partitioned gesture following a curvy path, see Figure 4.33. This gesture is in analogy to the swipe touch gesture on smartphones that could find use in interactions where something has to be moved into the direction of the swipe. The gesture can be performed unintentionally in the natural interaction without the intention to, therefore the definition of the state machine is designed in a quite restrictive manner with the use of all key values. In order to avoid the performance of the motion by accident, before performing the gesture the user has to hold his hand still for around 0,3 seconds, after that he has to move his hand in the wanted direction and hold the speed for a given number of frames. This gesture also detects false alarms which is caused by rotating the wrist, that is done by checking $Grav = Grav_{atStart}$.

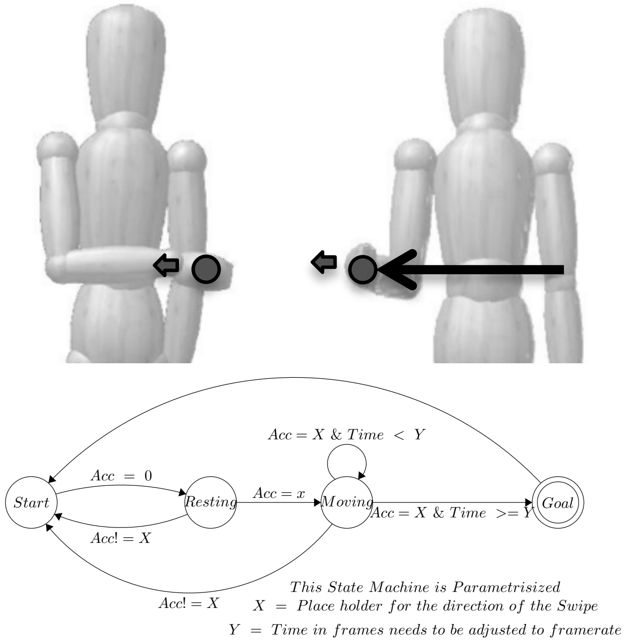


FIGURE 4.33: Motion and state machine of *Swipe Gesture* uses all defined key values to avoid unintentionally gesture performance.

The *Hammer Gesture* follows the definition of a curvy and continuous gesture, see Figure 4.34. Knocking the watch wearing hand in a hammer-swing-like motion onto the other hand performs this gesture. That way the de-acceleration is high enough to make a special pattern that we detect. It is supposed too be used for pushing buttons, smashing objects, or forging.

The *Lever Gesture* was made to evaluate the usefulness of recognizing gestures just by the alignment of the watch to the gravity, see Figure 4.35. No other key values are used for the recognition of this gesture. Therefore, we designed a partitioned gesture following an angular path with two stages in which the watch is rotated in two different directions using the gravity in those directions. First, the watch is rotated along the longitudinal axis and in the second state, along the lateral axis.

This *Ladle Gesture* is defined by partitioned movement following a curvy shape, see Figure 4.36. The gesture demonstrates the combination of gravity elements and further key values of velocity and time. The motion of the gesture is in analogy to scooping fluid and pouring it into another container. Due to the additional key values, merely rotating the wrist does not trigger the gesture recognition.

The *Z Gesture* is defined as a continuous movement along an angular shape and was made to test the limits of the system by combining arbitrary motions into one recognizable gesture. The path of the *Z Gesture* is depicted in Figure 4.37.

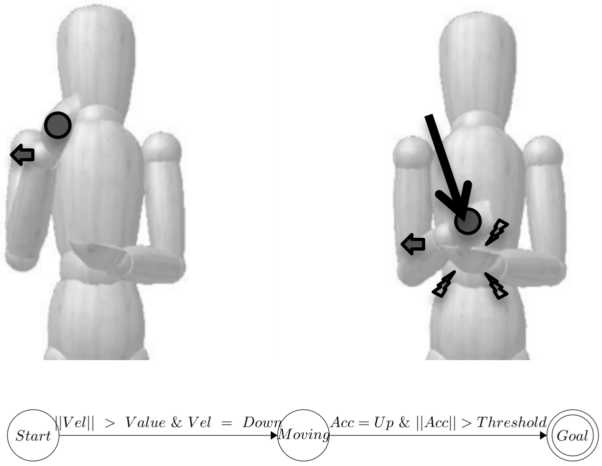


FIGURE 4.34: Motion and state machine of *Hammer Gesture* demonstrates the usage of negative acceleration and velocity.

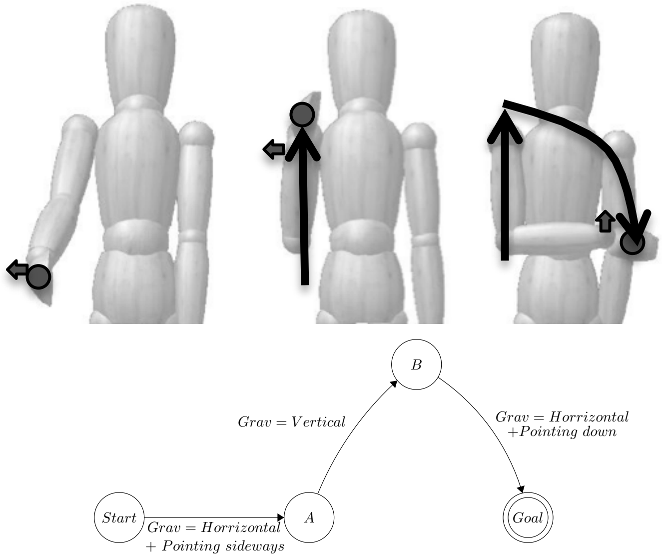


FIGURE 4.35: Motion and state machine of two staged *Lever Gesture* only uses gravity as key value.

Following the system description above, two possible ways to create a gesture recognizing state machine can be deduced.

4.5.2.3 Experimentation and Analysis

We propose our multi-modal interaction interface for improving the efficiency of task performance in virtual reality. In particular, we provide a natural and intuitive interaction technique that can be used with increasingly popular networked systems, like smart homes or in virtual reality environments. As shown in Chapter 4.5.1, we could prove that our approach is at least as good as common visual gesture recognizer. In this study, we wanted to figure out if the efforts we performed are real improvements of the system. In related work, there are no experiments to which our system/gesture recognizer could be compared. Therefore we perform a usability

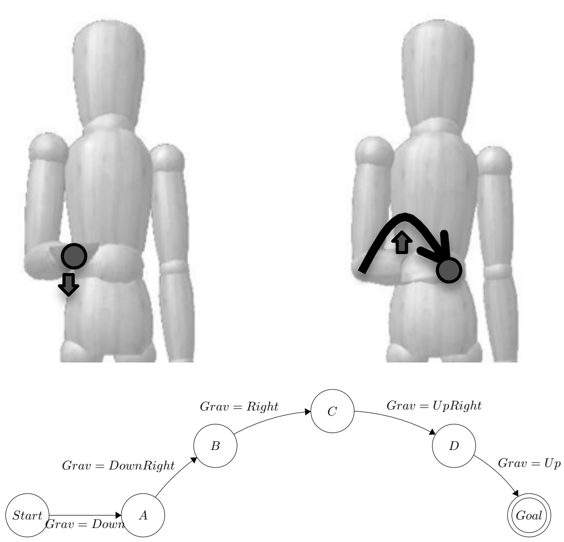


FIGURE 4.36: Motion and state machine of *Ladle Gesture* combines gravity with velocity and time values.

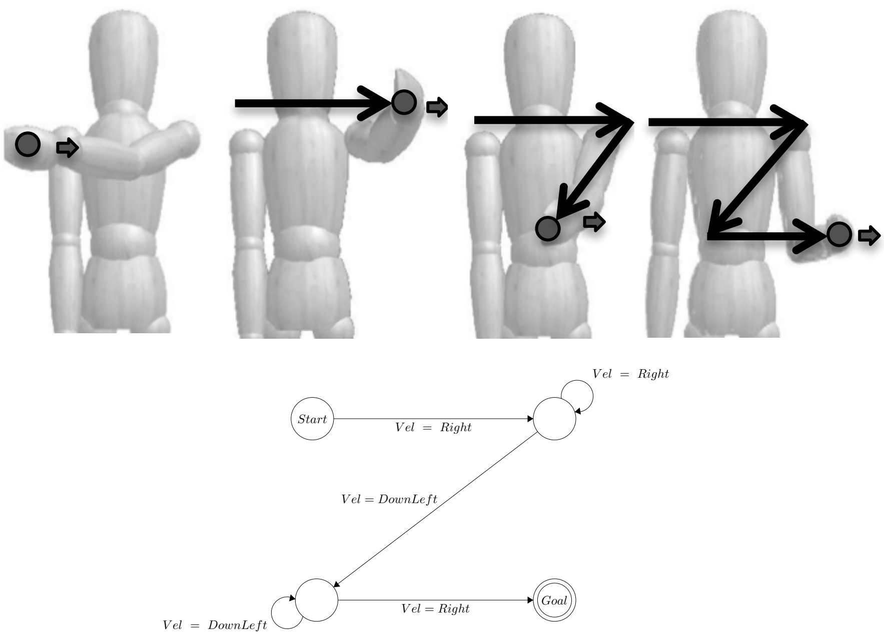


FIGURE 4.37: Motion and state machine of *Z Gesture*, which failed in our system.

study to measure:

1. Learnability;
2. Effectiveness;
3. Acceptability.

We expect, that our interaction-interface will get highly accepted and perceived as useful. For testing these assumptions, we designed and performed a controlled experiment.

Operationalization. In order to measure the above usability types, we operationalized three variables of interest and designated the tasks to be performed. The variables of interest were operationalized as follows:

1. Learnability is measured by the amount of trials a subject has to perform until the gesture is recognized with a rate of 75%.
2. Effectiveness is measured by the accuracy of the gesture recognizer describing the proportion of all measures which are correctly classified.
3. Acceptability is measured using the Technology Acceptance Model (TAM), which is a valid and reliable questionnaire for assessing technology acceptance and use [298]. Out of 5 categories in the original TAM, we selected 12 questions out of 3 categories focusing on performance expectancy, effort expectancy, and attitude towards using the technology. All questions were rated using a five-point Likert scale (1: I strongly disagree, 5: I strongly agree).

Subjects. The controlled experiment was conducted with 13 participants: undergraduate and graduate students in computer science from the Technical University of Kaiserslautern participated in the study. The subjects were aged between 22 and 38 years old; 10 were male and 3 female; 3 participants had experience with wearable devices and 4 participants are performing gesture control frequently.

Experimental setup and data collection. The setup consists of a desktop computer, used to provide feedback, and an Apple Watch Sport 38mm (first generation). Participants could move freely in the room (no cables attached) in order to provide an environment as natural as possible. The experimental procedure was performed in the following order:

1. Training the subjects. Each subject learns how to perform the specific gestures. The subject can make use of visual presentations of the gesture and personal advice by the experimenter. After the subject knows how to perform the gesture, the number of trials until the gesture is correctly performed (and recognized) for 3 times in a row is recorded.
2. Executing the accuracy test. Each user performs each gesture 20 times. In particular, the following gestures are tested: *Circle, Left Swipe, Right Swipe, Saw* (variation of shaking), *Lever, Laddle, Complete Square, and Right Bracket* (both variations of z-gesture). For each trial, it is recorded if the gesture was correctly recognized and if other gestures also were recognized.
3. Subjects fill out the acceptance and usability questionnaire.

Data analysis. A transcript of the collected learnability, effectiveness, and acceptability data was compiled in excel. The subject data is kept anonymous and confidential. Regarding effectiveness, we used statistical evaluation of binary classifiers based on confusion matrix. We applied descriptive statistics methods such as the sample mean, standard deviation, and median.

4.5.2.4 Results

Learnability. Figure 4.38 displays the distribution of the data for all performed gestures. It can be seen in the diagram in Figure 4.38, that all gestures except for *Complete Square* and *Right Bracket* are easy to learn with a mean number of trials of 6.2 to reach an accuracy rate of 75%. Only for the two mentioned gestures the subjects needed 12 to 15 trials on average. The *Saw* gesture is the most easy gesture to learn with an average number of 3 trials with a maximum number of 5 trials of one subject.

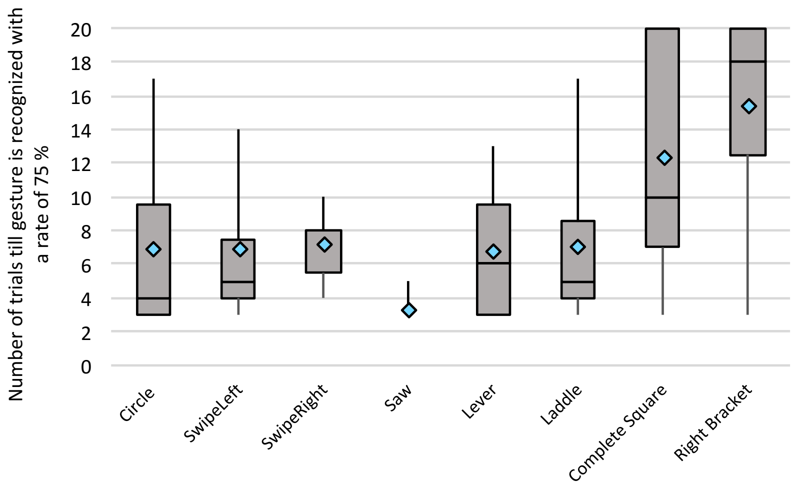


FIGURE 4.38: Boxplot showing the data distribution of all gestures including the average number of trials per gesture displayed as blue rhombus.

The descriptive statistic values of mean, median, and standard deviation of all performed gestures are listed in Table 4.9:

TABLE 4.9: Statistic values of learnability time measures in seconds

Gesture	Mean	Median	σ
Circle	6.2	4	4.5
Swipe Left	6.2	5	3.7
Swipe Right	7.1	8	2.1
Saw	3.2	3	0.6
Lever	6.6	6	3.8
Laddle	7	5	4.5
Complete Square	12.2	10	6.7
Right Bracket	15.2	18	5.8

Effectiveness. Effectiveness is measured by the accuracy of the gesture recognizer describing the proportion of all measures, which are correctly classified. A statistical evaluation of binary classifiers based on confusion matrix is performed, bringing into relation true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), considering the following values:

1. Sensitivity/Hitrate = TP/(TP+FN), describes the probability to which a performed gesture is recognized correctly.
2. Specificity = TN/(FP+TN), describes the probability that the gesture is not recognized by performing of another gesture.
3. PPV = TP / (TP+FP), describes the probability that a recognized gesture is actually performed.
4. NPV = TN / (TN+FN), describes the probability that the not recognized gestures are actually not the performed gesture.
5. Accuracy = (TP+TN)/(TP+TN+FP+FN), describes the amount of correctly classified objects.

TABLE 4.10: Binary classifiers based on confusion matrix

Value	Circle	Swipe Left	Swipe Right	Saw	Lever	Laddle	Complete Square	Right Bracket
Sensitivity	89,62%	87,31%	86,54%	99,23%	89,62%	91,54%	48,85%	53,08%
Specificity	99,51%	99,29%	98,96%	99,56%	99,45%	100,00%	99,89%	98,57%
PPV	96,28%	94,58%	92,21%	96,99 %	95,88%	100,00%	98,45%	84,15%
NPV	98,53%	98,21%	98,09%	99,89%	98,53%	98,81%	93,18%	93,63%
Accuracy	98,27%	97,79%	97,40%	99,52%	98,22%	98,94%	93,51%	92,88%
1-Specificity	0,49%	0,71%	1,04%	0,44%	0,55%	0,00%	0,11%	1,43%

The diagram in Figure 4.39 shows the values and differences between gestures clustered per effectiveness value in a bar chart. The gestures *Complete Square* and *Right Bracket* have an accuracy of 93.51% and 92.88%, but merely show a sensitivity of 48.85% and 53.08%, which are far distanced to the remaining gestures. The high accuracy values result in the high specificity values. As the gestures are really hard to recognize, they are also not recognized while performing other gestures, leading to really low false positive values. Excepting the already mentioned gestures, an av-

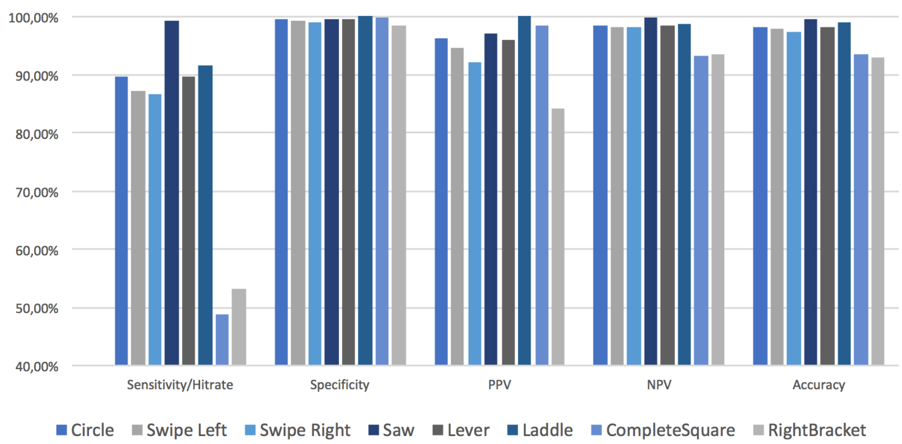
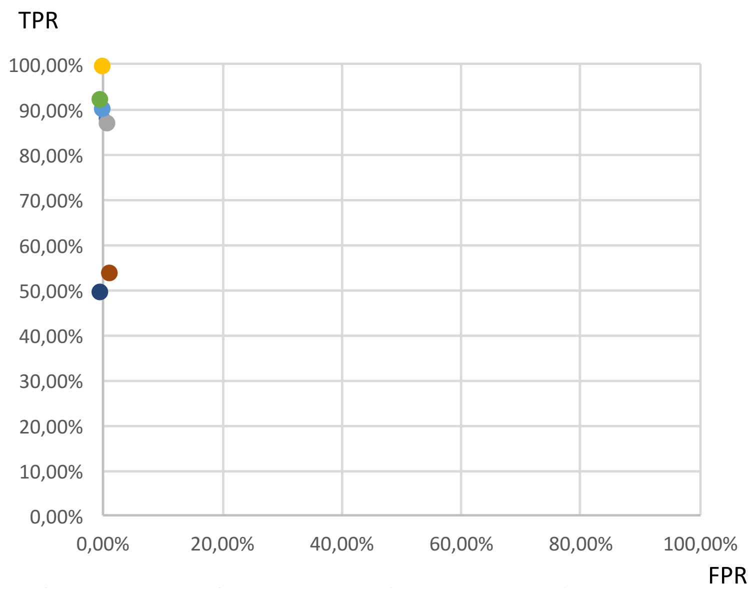


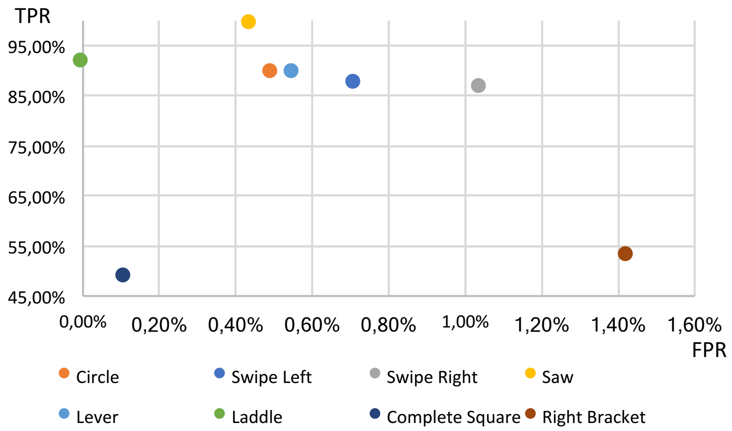
FIGURE 4.39: Bar chart visualizes effectiveness values and comparisons of gestures.

erage sensitivity rate of all performed gestures of 90.64% is achieved, with an average specificity rate of 99.46% and an average accuracy rate of 98.36%. The best performed gesture (*Saw*) shows a accuracy of 99.52%, while the weakest gesture (*Swipe Right*) still shows a accuracy of 97.40%. The corresponding values of all performed and tested gestures are listed in Table 4.10.

Figures 4.40a and 4.40b show the accuracy values of all gestures in a receiver operating characteristic (ROC)-curve plotting the true positive rates (TPR) against the false positive rates (FPR) (1-Specificity) for the chosen threshold in our recognizer. Optimal data points in a ROC curve are located in the upper left corner, implying highest sensitivity and highest specificity. Figure 4.40a encompasses the complete range of TPR and FPR from 0.00% to 100.00%, displaying all data points close to the optimal position. Figure 4.40b, however, is zoomed into the range of particular interest for better readability.



(A) ROC-curve in full range from 0.0 till 100.0%.



(B) ROC-curve in range of particular interest from 45.0 till 98.0% and from 0.0 till 1.60%.

FIGURE 4.40: ROC-curve plotting the TPR against the FPR (1-specificity) for all gestures.

Figure 4.41 shows the ratio between the cumulated true positives and false positives of each gesture. All graphs except of those for the gestures *Complete Square* and *Right Bracket* show a linear behavior, describing the distances between correctly recognized gestures and incorrectly classified data points with their maximum in a range of 73% and 83%. Incorrectly classified gestures lower the gradient of the curve up to a static behavior with a gradient of Zero.

Acceptability. For the acceptability, the descriptive statistic values mean, median, and standard deviation based on 5 point likert scale are calculated and listed in Table 4.11.

The average acceptability over all questions is 3.18. The average value is neither exceptionally good nor bad. From direct feedback with the user, an overall satisfaction was stated, the subjects felt that learning the system was easy, using the system was fun, and the system would make their work more interesting. The attitude toward using the technology has an average value of 3.31, higher than the effort expectancy

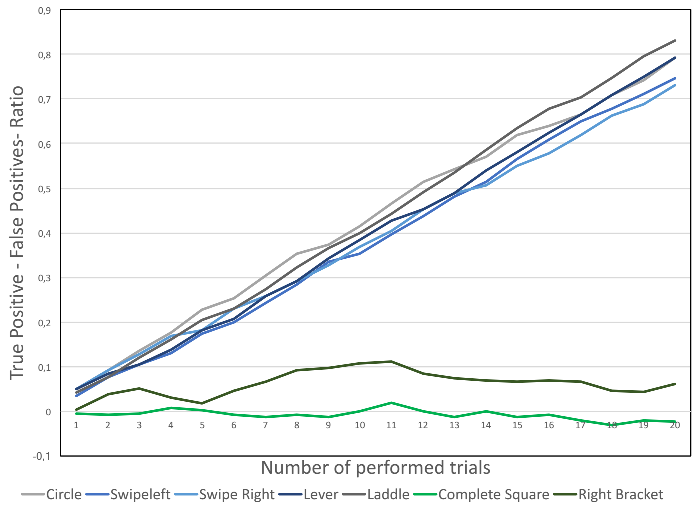


FIGURE 4.41: Ratio between true positives and false positives of all performed gestures.

TABLE 4.11: Statistic values of acceptability measures

Question	Mean	Median	σ
1	3.00	3	1.18
2	2.62	2	0.74
3	2.77	2	0.97
4	2.85	2	1.09
5	3.23	3	1.31
6	3.46	3	1.01
7	3.08	3	0.92
8	3.23	3	1.12
9	3.00	3	1.24
10	3.54	4	1.44
11	3.54	4	1.33
12	3.15	3	1.29

with 3.26, and the performance expectancy with 2.89 in average. The Boxplot in Figure 4.42 shows the distribution of the collected acceptability measures sorted per asked question.

4.5.2.5 *Discussion*

The gestures we provided through the high flexibility of our system are easy to learn, effective, and users show a positive attitude towards using the technology. Primitive gestures as *Swipe Left* and *Swipe Right* seem simple, however it is challenging to design those gestures in a way, that they are easy to learn, easy to recognize, but not recognized while performing other gestures. More complex gestures, like *Lever*, *Laddle*, or *Shaking* incorporate less key values and are easier to design. Although, the circle gesture integrates up to nine potential states, not all of those have to be reached making the design and recognition of those gestures easier. Table 4.8 confirms our thoughts.

Some limitations discovered during the process of designing gestures need to be taken into consideration.. It can be stated that gestures following continuous movement

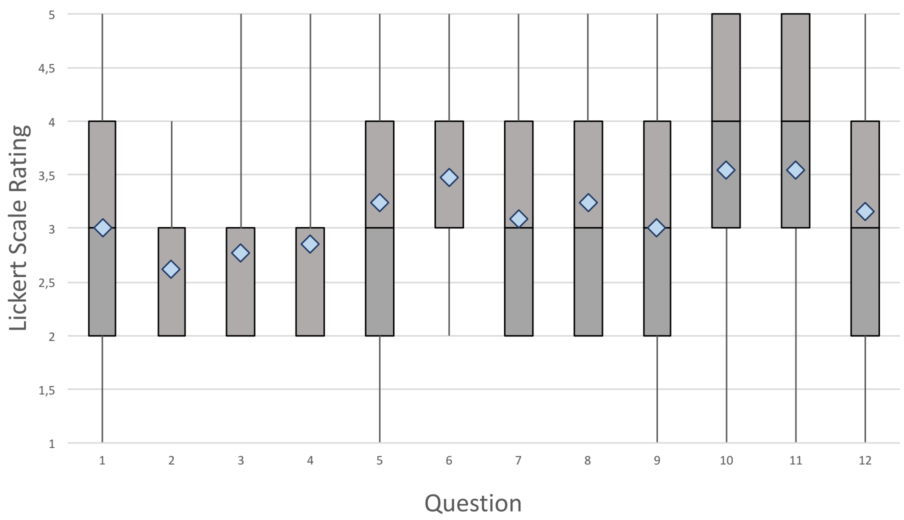


FIGURE 4.42: Boxplot displaying the data distribution of acceptability measures.

along angular shapes, like *Z Gesture*, *Complete Square*, or *Right Bracket* are hard to learn and hard to recognize and should not be used. The reason is that a deceleration of the hand movement is easily recognized as movement into the opposite direction progressing the state machine into the next state too early. Combining arbitrary motions into one recognizable gesture is not possible in any case. Especially, for designing continuous gestures, reversing movements should be avoided.

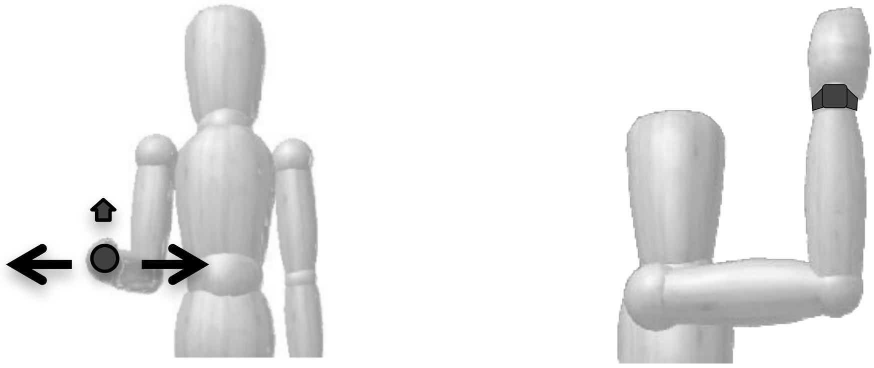
One limitation can be stated by performing rotations only along the longitudinal and lateral axis[3]. The rotation detection depends on the direction of the gravitational force describing the position of the watch. It is not possible to perceive any rotation along the perpendicular axis, where $\vec{G}$ points exactly along the z-axis. As depicted in Figure 4.43 (a), rotating the watch along the perpendicular axis, while the watch is facing to the ceiling, does not lead to any rotation recognition.

Another challenge was found in calculating the acceleration $\vec{A}$ for a specific position. Here, small changes to $\vec{G}$ can have enormous influence on $\vec{A}$ if $\frac{\vec{G}_{new}}{\|\vec{G}_{new}\|}$ is in a certain area of the unit circle. The problem causative position is in $\frac{\vec{G}_{new}}{\|\vec{G}_{new}\|} = (0,0,1)$, as depicted in Figure 4.43 (b).

We cannot state, that incorporating less or more states/key values are easier or harder to recognize. But we can state that the combination of those key values in a keen matter are highly usable to recognize easy learnable unique, primitive, and even complex gestures while overcoming the missing gyroscope sensor in the used technology.

Comparability to common technology. Compared to common technology like other smartwatches or electromyographic armbands, the used device in this investigation uses less expensive sensors, which can easily lead to inaccurate signals and measurements. Nevertheless, our approach is able to overcome this limitation and it led to satisfying results with low budget devices. Compared to optical tracking systems like 3D cameras, we could already prove in Chapter 4.5.1 that the usage of

[3]By using the Apple watch 1; This statement is not generalized.



(A) Rotation detection problem along perpendicular axis. Watches faces the ceiling.

(B) Acceleration calculation problem by (0,0,1)

FIGURE 4.43: Positions describing limitations of the system.

smartwatches are promising alternatives to common gesture based interaction technology. Furthermore, our approach is able to identify more diverse gestures and even small movements like rotating the wrist, which would not be recognizable with common optical trackers.

4.5.3 Enhancing Multi-Modal Interaction in Virtual Reality with Smartwatches and Speech

By the mid of 2017, we have access to a wide variety of inter-connected devices that communicate with their surroundings and expand interaction possibilities. For example, smartwatches have embedded sensors and decent processing units, and they have been considerably improved and become broadly available. Despite the increase of power from these ubiquitous devices, the amount of information they can display and the input capabilities via touch gestures are defined by their display sizes and are therefore limited. In spite of the small and usually poor displays on smartwatches, big-screen TVs and display monitors are becoming cheaper and more prevalent. As a result, display technologies are becoming less expensive as well, and there has been a steady increase in the use of the large screen displays. However, these displays lack input sensors and are controlled via traditional input devices such as a remote control, keyboard or mouse.

Technologies for large-screen displays and smartwatches have limitations, and the lack of capabilities of one device can be compensated by the capabilities of another (display/screen-estate vs. sensors). The possibilities for natural interaction to be achieved with these devices has been explored, see [237] for example. The sensors and input capabilities of the smartwatch can be exploited to support the interaction with large-display devices, using natural interactions such as touch, gesture or speech. Speech enhances overall interaction capabilities enormously. The use of speech is often the easiest, most natural way to interact with other humans but also computers [210]. In many of today's systems, using speech and gestures is supported in the user interface, creating a concerted, and natural interaction modality [39]. Such a natural interaction modality enables innovative interaction possibilities, going far beyond those offered by a remote control or desktop interaction model.

In this section an approach for fusing multiple interaction modalities such as speech, gesture, and touch by using a smartwatch for user interaction with large display devices is introduced. We investigate in depth the concepts of multi-modal interaction, speech recognition, different usage contexts, and we also design a prototype. We first present concepts of multi-modal interaction and speech recognition in a general manner. Subsequently, we demonstrate the adaptation and utilization of these concepts for three different applications, and then perform a user evaluation to document the benefits offered by the combination of the various input modalities.

4.5.3.1 Multi-Modal Interaction

The term multi-modal interaction refers to the combination of several natural methods of communication for the purpose of interacting with a system. Natural modalities of communication are, amongst others, gesture, gaze, speech, and touch [142]; thereby making it more intuitive to untrained users. This interaction interface allows a user to employ their skilled and coordinated communicative behavior to control systems in a more natural way. Hence, multi-modal systems incorporate different modalities.

Modality refers to the type of communication used to convey or acquire information. It is the way an idea is expressed or the manner in which an action is performed [217], and it defines the type of data exchange. The state that determines the way information is interpreted in order to extract or convey meaning is referred to as *mode*. For example, gesture modality can provide data that can be interpreted into different modes of communication such as *tilt* or *shake*. When multiple modalities are in use, it is paramount to fuse them in a way that is most suitable and natural. The available modalities can either be used simultaneously or sequentially. A system that allows simultaneous use of different modalities does so by concurrently processing the information attained from the systems I/O channels in real time. Concurrency may be viewed from the actions perceived at I/O level (microphone, touch-screen, accelerometer) or from the source performing the actual tasks (tilt and speech). Central to this concept is the ability to combine data perceived by the user, *fusion*. While on the output end, multiple channels (mostly independent of one another) can also be used to convey information, which is called *fission*.

In multi-modal systems, the decision to fuse or not to fuse the data from different modalities depends on the suitability of the intended usage of the data. The absence of multi-modal fusion is called *independent* multi-modal interaction whereas the presence is termed *combined* [217]. Fusion can be viewed from different levels such as the data level, feature level, and decision level. Combination of audio from two microphones or a microphone-array for a stereo effect can be said to be fusion on data level. Fusion on the feature level can be seen as combining speech and lip movement, while combination of gestures and speech is on the decision level. Fission on the other hand, is the splitting and dissemination of information through several channels [5], used for outputting information in more immersive ways. This could be the transmission of text, speech, haptic feedback, and audio cues concurrently, to allow a more accurate interpretation.

4.5.3.2 Concept of a Multi-Modal Interaction Interface

For a proof-of-concept demonstration of multi-modal interaction using smartwatch and speech, we had to fully understand the general concept and explore its feasibility. Related work such as [237]illustrates the principles/foundation of multi-modal interaction, guiding us to determine viability of some of our envisioned approaches.

Modes of interaction. As multi-modal systems become more prevalent, new and novel ways of interacting with systems are continuously being discovered and improved, techniques such as gaze, smile, gesture, speech, and touch among others, are not uncommon in modern studies in Human Computer Interaction (HCI).

Speech Input – The use of speech as an interaction means is not a new technique in HCI. Actually, it has gone through numerous evolutions to attain the level of stability it currently supports, with some systems almost enabling free form communication. Several speech based interaction systems exist today. Ranging from software based speech input systems (e.g., Siri) to dedicated standalone devices (e.g., Xperia Ear). Although speech has proven very useful for hands-free interaction, it can, however, be hindered by problems such as ambient noise, privacy, and limited support for accents. Numerous *Software Development Kits* (SDKs) have been developed from research projects aiming to improve the process of speech recognition and analysis. They can be classified into two main categories: online and offline analysis. The online-based analysis engines are very powerful and leverage powerful cloud architecture for speech recognition thereby offloading processing from the device. They provide access to SDKs or *Application Programming Interfaces* (APIs) for easy adoption by developers. Examples are: Google Speech API [110] or Microsoft's Bing Speech API [202]. Offline analysis engines allow analysis from within the system/application without the need of a network connection, an example is CMU Sphinx open source speech recognition [71].

Gesture Input – Gestures used as interaction modes, add more meaning to interaction with systems or interfaces by empowering users to achieve goals in more natural ways that current modalities such as mouse and keyboard [36]. Gesture input allows more natural communication modes such as pointing, body movement, tilt, shake, etc. to interact with systems. A popular example is movement tracking with the Kinect camera [201], which uses a RGB camera, depth sensor and a microphone array for acquiring user's complete body interaction and voice commands. Another popular gesture based input interface is the Wii remote control [99], which enables numerous ways of interacting with systems using gesture and movement [251]. Most mobile phones and smartwatches of recent age, come equipped with sensors that can be used to easily detect gestures of various forms which can range from a mere shake, down to imitation of steering wheel tilt for racing mobile games.

Touch Input – Touch is the most common input method for smart devices [65]. However, smartwatches compared to common smart devices have an even smaller form factor and are worn on a wrist, which demands a reconsideration of common smart device input techniques. Touch is more difficult on smartwatches, which typically have small screens, exacerbating the fat finger problem [247] and no multi-touch capabilities. Touch input on smartwatches should be designed in a way that even inaccurate touches are successful but also that the full capacities of the display are used, e.g., really precise touch points (e.g., too small buttons) should be avoided. Often used User Interface (UI) elements, can be distributed to the smartwatch's display and accelerate the task completion.

Interaction model. An overview of the interaction model is shown in Figure 4.44, based on the multi-modal interaction concept, showing the main components

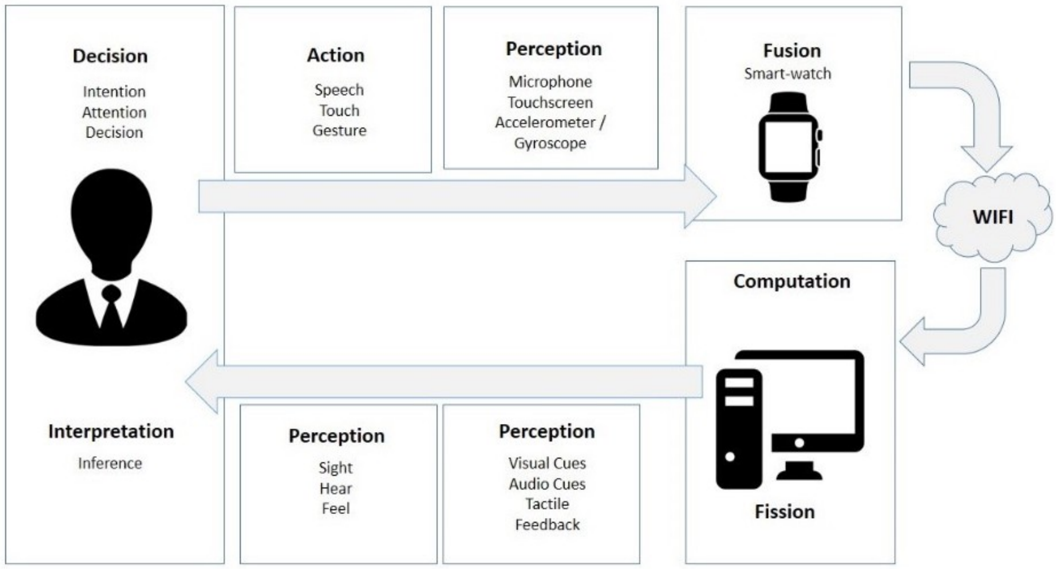


FIGURE 4.44: Multi-modal interaction system. The main components are a smartwatch and a large display enabling speech, touch, and gesture modalities perceived by the system through microphone, touchscreen, accelerometer, and gyroscope. Visual and audio cues, as well as tactile feedback are perceived by the user due sight, hear, and feel.

needed to achieve the desired level of interaction capabilities. The figure emphasizes which modalities are used to support a user in the decision making process. We designed and implemented a smartwatch application that enables users to interact with large display devices, allowing them to interact with a PC using the provided combination of interaction modalities, e.g., speech, tilt gesture, and touch. Since the smartwatch is physically separated from the display or PC, to which someone wants to communicate with, a system architecture is needed that enables transfer of user interaction data and interpretation of this data on the receiving, large-display side. In order to capture speech, gesture and touch input, the smartwatch must have a microphone, gyroscope, accelerometer, and touch screen. Besides, the smartwatch must be capable of communicating with the PC using a wireless network, requiring the smartwatch to have its own board and a wifi communication chip.

Scenarios. Multi-modal interaction can be used in many contexts, ranging from navigating a map on a white-board for controlling a robot, or navigating on a smartwatch menu. However, a "near-perfect" interaction paradigm used in one context could be inappropriate in another context; different contexts have different requirements in terms of precision and responsiveness [60]. Our targeted scenario includes a large display device, e.g., a projector or large-screen TV device, and a user with a smartwatch. We discuss three interaction contexts. They are:

1. Interaction with a standard operating system interface, surfing the Web, and searching or typing a note where all input done via speech and gesture from the smartwatch.
2. Menu-driven navigation on a large screen device using the smartwatch.
3. Providing a windows interface to control games on a large display using the smartwatch.

Especially text input poses challenges when performed with a smartwatch. Nevertheless, it is possible to tackle these challenges by using touchscreen-based soft keyboards [90]. Alternatively, one can utilize advanced skin-based and around-device interaction technologies [109]. To support our targeted scenarios, the application provides several options: toggling orientation to match wrist-use mode and air mouse hand-held mode considering a user's preference, as depicted in Figure 4.45.

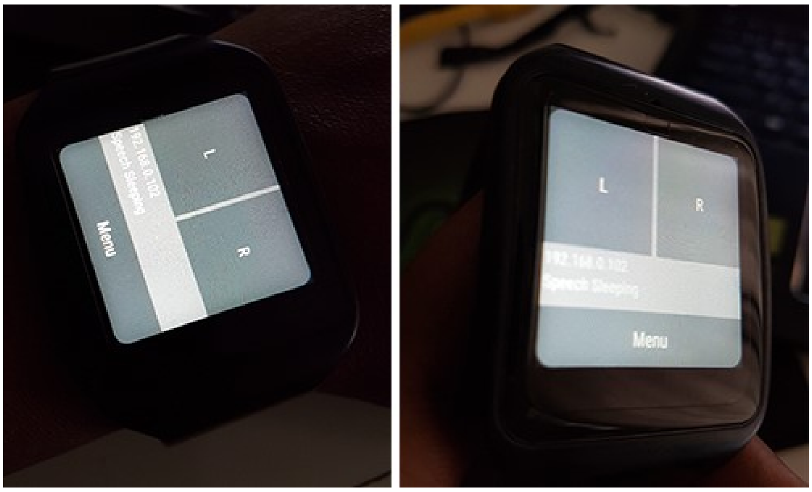


FIGURE 4.45: Landscape (left) and portrait mode (right) of smartwatch worn on wrist and held in hand, respectively.

4.5.3.3 *Prototype Implementation*

For the implementation of our concept, we adopted a *component-based architecture*. The overall system was divided into two separate components. The system uses a *smartwatch app* component, included in the smartwatch module, and a *server* component, part of the large display module, that is executed on the PC end. Both components only transmit to one another but do not rely on each other for processing capabilities, as computations and data transformations are done locally in both components.

Smartwatch application. For the development of the smartwatch application, a watch with the components satisfying our input requirements, see Figure 4.44, was required. The watch should support communication capability with the server component, for efficiency and lag-free setup. A Wi-Fi chip would be ideal, as it supports direct communication between the watch and server component without the need to proxy data through a companion mobile phone. Latency is reduced as much as possible. The Sony smartwatch 3 [267], called SWR50 in the following, was used as a test device. It is equipped with a microphone, accelerometer, gyroscope, Bluetooth, and Wi-Fi for direct Internet communication. SWR50 also runs the Android Wear O.S 1.5, which enables a direct socket connection to the network, without proxying network or socket calls through a mobile phone. As shown in Figure 4.46, the app depends on data provided by the O.S' Sensor Manager and Media Recorder API.

Gesture Implementation – Two classes of gestures are handled by our prototype, (1) tilt and (2) face-down. As shown in Figure 4.48, users are alerted to calibrate the app to detect their watch's central position, which is the reference point for interpreting sensor data. In order to support the tilt functionality, gravity and data from the magnetic

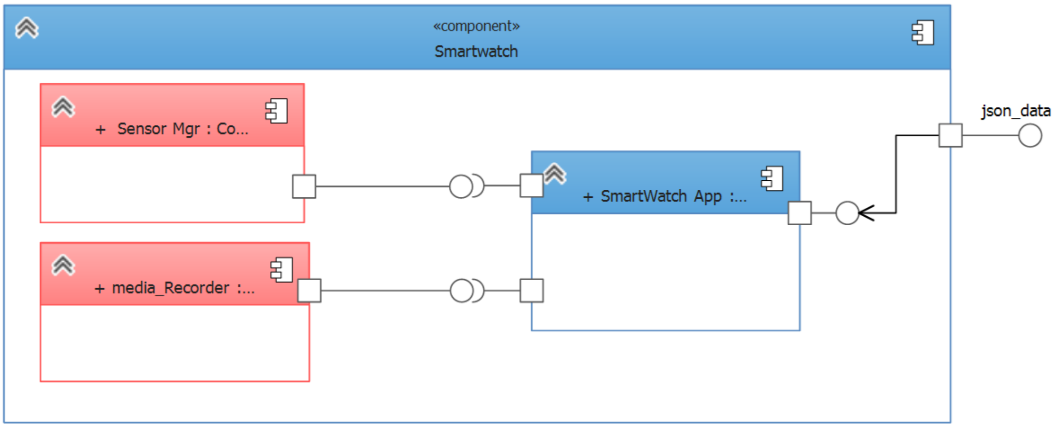


FIGURE 4.46: Smartwatch app and explicitly accessed system APIs (red).

sensor are combined. Although the SWR50 does not have a gravity sensor, gravity data is generated by using accelerometer data and applying a low-pass filter to it, see Figure 4.48. In order to obtain tilt information, the procedures `getRotationMatrix` and `getOrientation` of Android SDK's SensorManager are used. The procedure `getRotationMatrix` combines the gravity and extracted geomagnetic data generated by the magnetic sensor to compute the inclination and rotation matrices. This step transforms a vector from the device coordinate system to the world coordinate system [112], defined as an orthonormal basis system, see Figure 4.47.

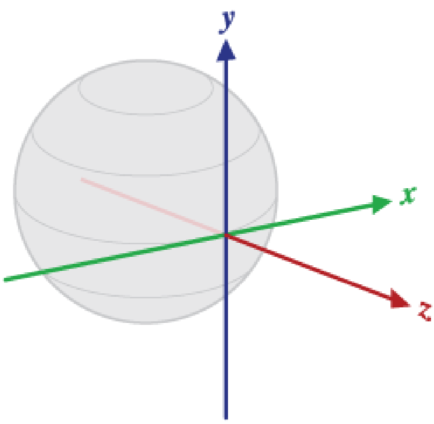


FIGURE 4.47: The z-axis is perpendicular to the ground, pointing to the sky; the y-axis is tangential to the ground, pointing towards the magnetic North pole; and the x-axis is defined by the vector product of the y-axis and z-axis basis vectors [112].

The rotation matrix resulting from this process is passed on to the `getOrientation` of the `SensorManager`, returning angular vector data (in radian) as an array of `Azimuth`, `Pitch`, and `Yaw` values. This data is converted to degrees and normalized before sent to the server, see Figure 4.48.

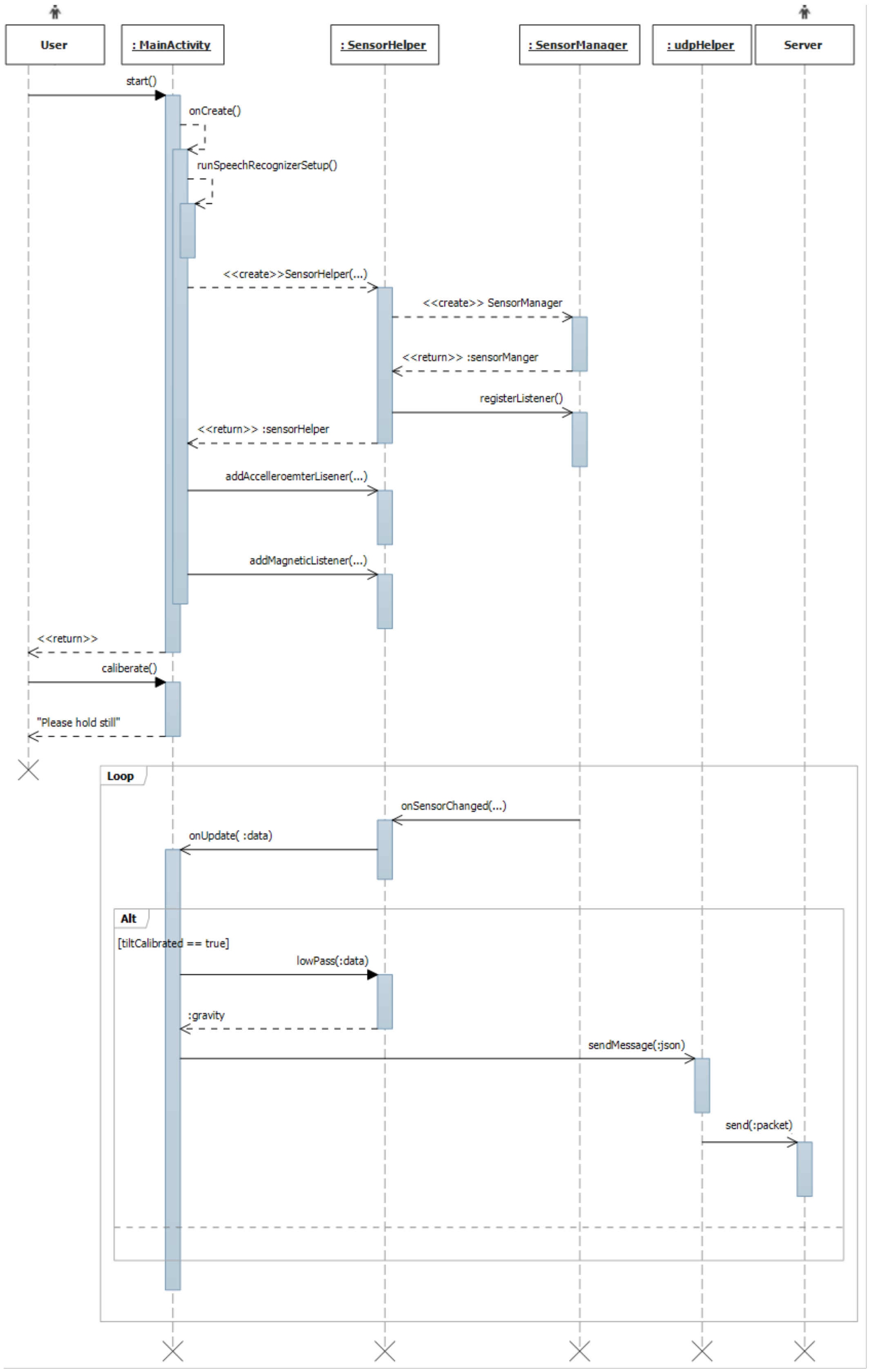


FIGURE 4.48: Sequence diagram of gesture-capturing process. The diagram shows transactions between user, sensor manager and server.

The **face-down gesture** relies on the use of gravity data. The desired effect is achieved when gravity readings are mainly in z-axis direction, with values being $v \leq -9.0m/s^2$ and readings close to 0 in the other two axis-directions. The face-down gesture is only used to enable listening to speech. Before issuing a speech command, a user must twist the watch, forcing it to face down towards the floor, invoking the listening module and causing the watch to vibrate indicating readiness for speech input. We considered different methods to determine a very good solution for speech recognition. *Shake-to-Speak* is an operational mode where the smartwatch is quickly shaken in any direction to activate the speech listener. Another solution for the speech listening module is to continuously listen to all spoken words. Both approaches lead to many false positives and require high computation times. Using a dedicated gesture to wake up the listening module is less resource-intensive than listening continuously, and a dedicated gesture-based approach also produces less false positives.

Speech Implementation – Several speech recognition engines were considered. The CMU pocket sphinx was adopted, mainly due to its lightweight form and portability. Speech processing is done off-line. Two types of speech recognition were implemented, *keyword-targeted translation* and *free-form speech* for typing. Free-form speech recognition is made possible through dedicated keywords. Text synthesized from speech is transmitted as normal text in a JSON format to the server. Figure 4.49 shows the components involved in our setup.

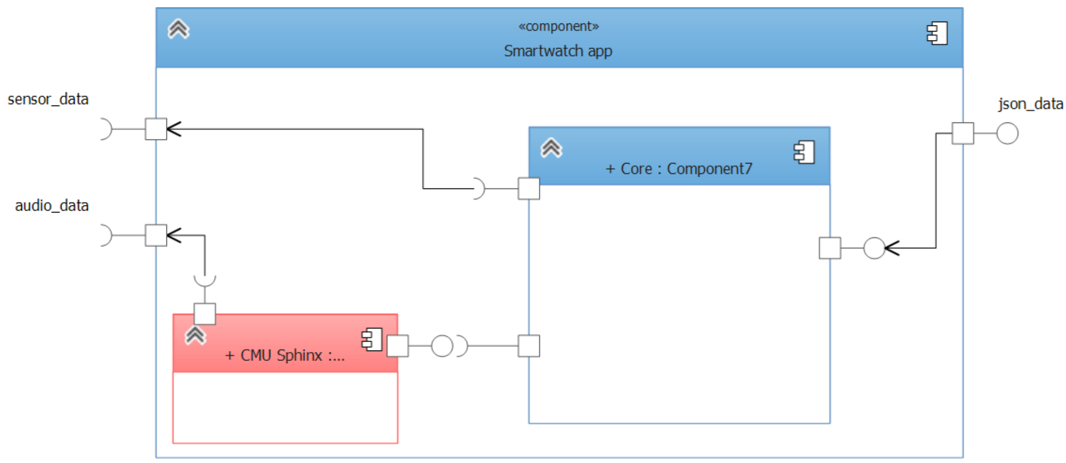


FIGURE 4.49: Internal component dependencies and the CMU Sphinx speech recognizer SDK.

Large Display Device. Interaction with a large display would require a component capable of interpreting the packets sent from the smartwatch to execute the intended action(s). This component is the *server*. These components can communicate after a connection between them has been established.

Communication Protocol – Communication between the app and the server is enabled by a User Datagram Protocol (UDP) socket connection. Although UDP lacks reliability and congestion control, it is energy-efficient [60]. The absence of reliability logic and status packets (ACK/NACK) was the reason why we chose it as our means of communication. Almost real-time transmission and extremely low latency are ensured, and data is sent at the fastest possible rate. Packet loss would be almost negligible in the gesture data-based approach. Regarding our system, packet loss can be tolerated. Lag is detrimental for a smooth user experience. Watch data are serialized using the JSON notation, see Figure 4.48.

Server – The server application is implemented in c#. It interprets the data from the watch and decides what actions to execute. The server itself is not a device driver, but it enacts the user's intention through the appropriate components (dependencies), see Figure 4.50. The server app's User Interface (UI) was designed in a straightforward

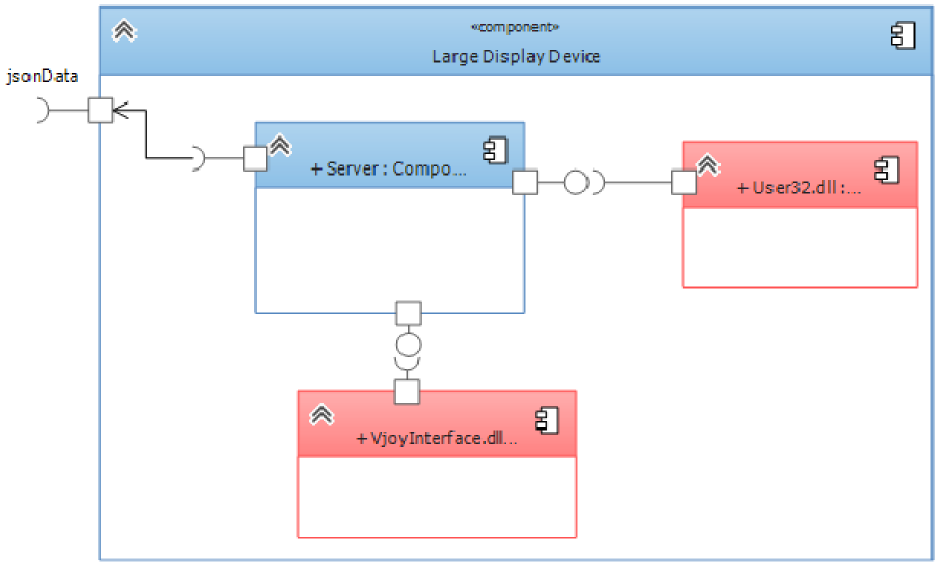


FIGURE 4.50: System components and main server dependencies.

manner and provides useful capabilities, including axis inversion, speed-of-mouse control and a drop-down box that allows a user to switch between three contexts to support a specific scenario via an appropriate mode. The three modes are: *mouse mode*, *key navigation mode*, and *game controller mode*.

In *mouse mode*, mouse movement is emulated by tilting the smartwatch in the corresponding direction. Angular tilt data of smartwatch motion is mapped to mouse velocity. Hence, a steep tilt causes the mouse to move at high speed. In *keyboard mode*, speech is used as text input and the keyboard's cursor keys are simulated by mapping tilt angle and direction of smartwatch motion. In *controller mode*, the server acts as a feeder to the Vjoy controller, interpolating the angular values from tilt data to match an analog stick axis. This mode imitates a virtual joystick's movement, mimicking a controller analog stick with the smartwatch's tilting motions.

Setting Companion App – The tiny displays used on watches, and the lack of suitable input mechanisms for them pose challenges for system design choices. Clearly, there exists a need for an app that can synchronize with the watch and update preferences. All transactions are shown in Figure 4.48 in form of a sequence diagram.

4.5.3.4 *Evaluation*

We conducted a lab-based experiment to evaluate the usability of our system. We presented two different applications covering examples for data exploration and immersive navigation, which are adequate application to demonstrate large display

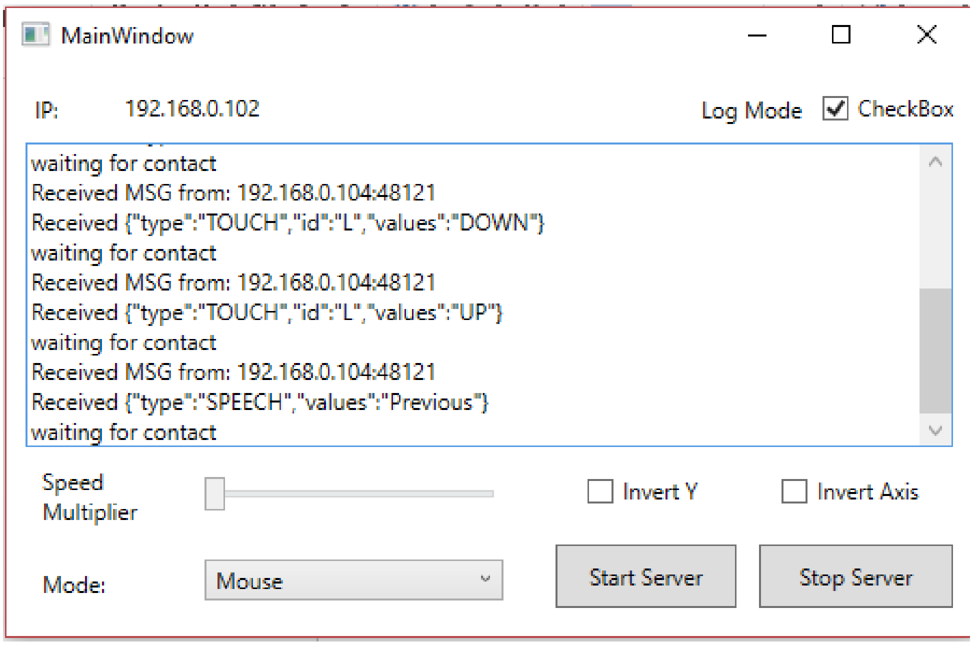


FIGURE 4.51: User Interface of server application. A user can start a session and activate handshaking between devices.

interaction. Participants performed 18 tasks in total, whereby we measured their success rates in order to determine the systems effectiveness. Afterwards, surveys and interviews about usability and user satisfaction were carried out.

Study setting. 7 university students participated in the study (undergraduate and graduate students, 21-29 years old; 5 were male and 2 female; 3 participants had experience with wearable devices and 3 participants are using speech commands frequently). In order to measure the usability and adaptability of the setup we followed the taxonomy of tasks for large display interaction, according to Foley [101]. Thus, the following task types are realized in both applications: (1) Position, (2) Orient, (3) Select, (4) Path, (5) Quantify, (6) Text. If we can demonstrate, that all tasks types according to the task taxonomy are successfully executed, it can be stated that the system is usable and adaptable for large display interaction.

Case 1: Visual analytics - Data exploration around the globe over the years – The visual analytic application is based on the Unity3D-Globe provided by Aldandarawy [3]. A 3D globe showing the worlds population is centered on the screen, as shown in Figure 4.52. Area's population values are discrete data sets shown as color-coded bars attached to the country/area. The height of the bar and the color denotes the amount of people residing in that area.

The application is initially created for a mouse/keyboard setting but could be easily enhanced for improved interaction technology. The stated task types are mapped to the following actions inside the application as shown in Table 4.12.

The following speech commands, as listed in Table 4.13 are integrated.

Tasks for case 1 – Users are given a labeled world map and a list of country names. At the beginning, users were ask to perform simple navigation tasks in order to explore the control capabilities. In the next step, users were asked to use the learned

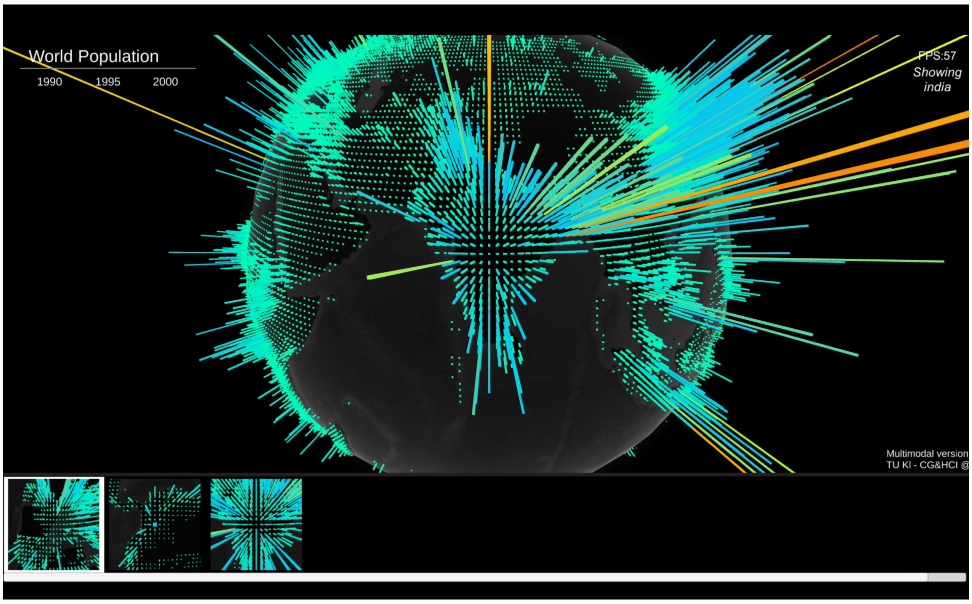


FIGURE 4.52: Unity3D-Globe application controlled with multi-model interaction techniques.

TABLE 4.12: Case 1: Action mapping

No.	Action	Interaction	Task type
1	Rotate globe	Touch	Position, Orientation
2	Change data per year	Touch & Gesture	Select
3	Zoom in/out	Speech	Quantify
4	Get specific data	Speech	Text
5	Make a screenshot	Speech	Select
6	Change the mode	Gesture	Path

TABLE 4.13: Speech commands for data exploration application

Command	Action
"Go to <Country name>"	Locate the globe to the desired location
"Zoom in" / "Zoom out"	Zoom in or zoom out in the current location
"Capture"	Take a screenshot of current location
"Remove"	Remove selected screenshot

interaction techniques in order to explore the data. The following tasks had to be performed:

Control exploration phase:

1. Rotate the globe in all directions (watch control - touch & tilt).
2. Show the data for the year 1995 (watch flickering in year mode).
3. Zoom in and out (voice control).
4. Locate and view each of the countries (alternate voice and/or watch control), capture the view in few locations (voice control).
5. Remove a selected capture (voice control).

Data observations:

1. Observe the population growth in Hong Kong from year 1990 to 2000.
2. Capture the view of the current location.
3. Compare the population between Europe and Asia in the year 2000 using captures.
4. Remove existing captures (voice control).
5. Compare the population between France, Colombia, and India in the year 2000 using captures.

Case 2: Immersive navigation - Heliborne The application for immersive navigation (see Figure 4.53), called Heliborne [145], is also initially created for mouse/keyboard setup but could be enhanced for improved interaction technology. The application is a simple helicopter simulator controlled with multi-model interaction techniques provided by the smartwatch. Heliborne is a helicopter combat game that simulates combats and terrains, helicopter and gunships from 1950 to modern day machines. It is not a real helicopter flight simulator game, but a flight game with flight physics toned down to a control scheme make flying and playing simple and fun. Although complex maneuvers may still require some degree of expertise, the basics can be easily picked up and enjoyed. Playable from both 3rd and 1st person view perspective.

FIGURE 4.53: Heliborne – a helicopter simulator controlled with multi-model interaction techniques.

The stated task types are mapped to the following actions inside the application as shown in Table 4.14 and linked with the speech commands listed in Table 4.15.

Tasks for case 2 – Users have a print out copy of the map in the application, highlighting specific locations. Analog to the first application, introductory users were ask to perform simple navigation tasks in order to explore the control capabilities. In the next step, users were asked to use the learned interaction techniques in order to explore the simulation world and to perform combined tasks. The following tasks had to be performed:

TABLE 4.14: Case 2: Action mapping

No.	Action	Interaction	Task type
1	Raise Altitude	Speech	Quantify
2	Reduce Altitude	Speech	Quantify
3	Control flight direction	Touch & Tilt	Position
4	Move Camera	Touch & Gesture	Orientation
5	Fire/Stop Fire	Speech & Touch	Select
6	Roll left/right	Touch & Speech	Position, Quantify
7	Switch Weapon	Flick wrist	Selection
8	Select gun	Speech	Selection

TABLE 4.15: Speech commands for immersive navigation application

Command	Action
"Go up" / "Go down"	Raise/reduce altitude of the helicopter
"Enough"	Clears previous command
"Bank left" / "Bank right"	Role the helicopter left/right
"Open fire"	Starts fire
"Give me guns"	Selects gun as weapon
"Give me rockets"	Selects rockets as weapon

Control exploration phase:

1. Raise/Reduce altitude of the helicopter (voice control).
2. Control helicopter in all directions (watch control).
3. Move the camera in left/right direction.
4. Roll left/ right (touch and voice control).
5. Select a gun and fire (voice control & flickering).

Simulation world observation:

1. Visit the camp located around longitude 6.8 and latitude 35; count the number of silos in that settlement (from the starting point behind you).
2. Visit the other camp located around longitude 6 and latitude 45; count the number of silos situated there.
3. Travel to the rendezvous point at longitude 4.8 and latitude 65: locate the orange signal, destroy as much of the surrounding structure around the location as you can, before landing.

Procedure. To conduct the whole experiment took about 45 minutes per participant. We determined 5 minutes to introduce the setups and basic interfaces. Then the participants carried out the tasks described underneath. Before each task, the concept and input modalities have been introduced and demonstrated. The participants were asked to get familiar with the corresponding device before the actual tasks have been conducted (10 minutes per application).

After the task execution session, we conducted a survey and interview. The survey included 5 aspects listed in Figure 4.55. During the interview, we asked for the reasons for their ratings. We also asked about general usability issues and solicited detailed feedback about the system and the experience of multi-modal interaction with the smartwatch.

Results.

Effectiveness – In order to measure the effectiveness of the system, we measured the users' success rate for each task. The averaged success rate defines the total accuracy value of the executed tasks. The effectiveness value is calculated by multiplying the success rate with the normalized task difficulty. Table 4.16 summarizes the accuracy and effectiveness results for both demonstrated applications.

TABLE 4.16: Accuracy and Effectiveness values of both applications.

Application	Tasks	Accuracy	Effectiveness
Visual analytics	10	96.25	95.17
Immersive navigation	8	82.5	76.73

Acceptabiliy – For the acceptability, the descriptive statistic values mean, median, and standard deviation based on 5 point likert scale are calculated. In total we asked 21 questions, covering the usability aspects Suitability, Learnability, Controllability, Error Tolerance, and Attitude toward using the technology. The Boxplot in Figure 4.54 shows the distribution of the collected acceptability measures sorted per asked question.

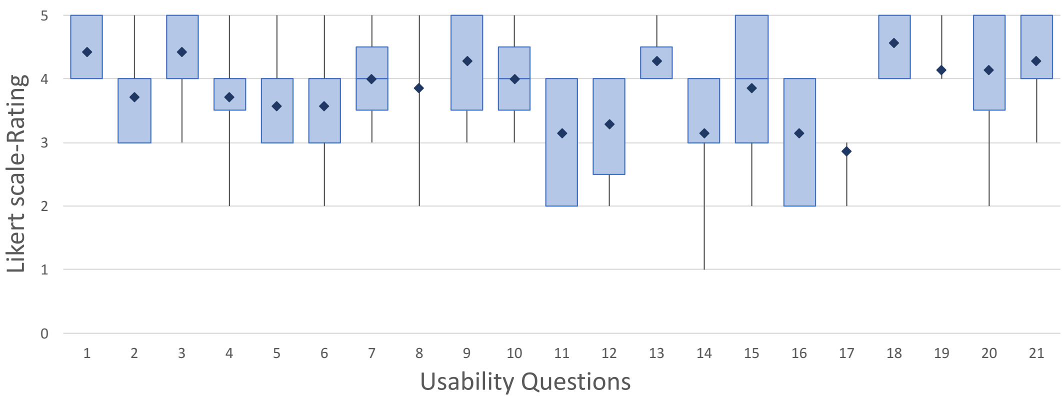


FIGURE 4.54: Boxplot displays the data distribution of the usability measures.

The average acceptability over all questions is 3.81, showing a quite good result. From direct feedback with the user, an overall satisfaction was stated. The subjects felt that learning the system was easy, using the system was fun, and the system would make their work more interesting. Users mentioned not having the feeling of complete control over the scene but also stated that it would be easy to become skillful in using the system. The attitude toward using the technology has an average value of 4.28, higher than the suitability value with 4.07, and the controllability with 3.71 in average. Table 4.17 summarizes the acceptability measures per usability category, which are visualized in Figure 4.55.

Quantitative Assessment – As described in [197], the qualitative results of the assessment provide a performance quantification basis, that results in a scalar usability value U as introduced in section 4.4.3.

TABLE 4.17: Average rating of usability categories

Usability categories	Average rating
Suitability	4,07
Learnability	3,71
Controllability	3,71
Error Tolerance	3,28
Usage attitude	4,28

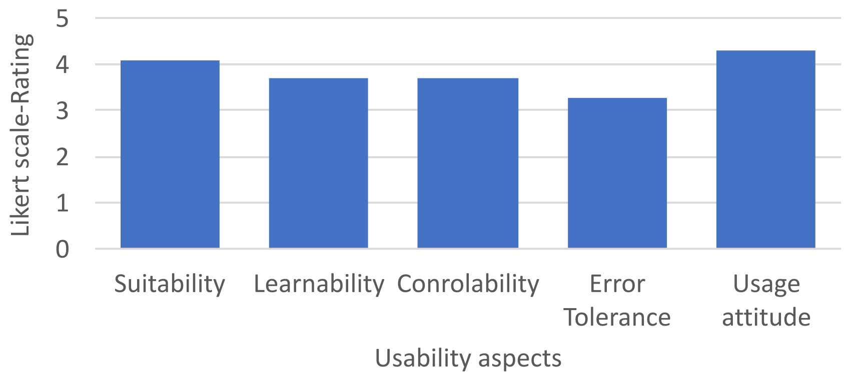


FIGURE 4.55: Average user ratings of usability categories from questionnaire.

TABLE 4.18: Weights and scoring of usability categories to calculate usability score U of the system.

Category	(1)	(2)	(3)	(4)	(5)
$w(s)$	0,3	0,1	0,2	0,1	0,3
$v(s)$	1	0,66	0,78	0,66	1
$w(s) \cdot v(s)$	0,3	0,06	0,15	0,06	0,3
$\sum$	0.8905				

The aim of this evaluation was to proof that the system is suitable for those kinds of applications and large display interaction. Thus, the usability categories: suitability and users attitude towards using system, were more important implying the association of a higher weight in the usability score calculation. As we focused less on evaluating the quality of the implemented tasks as well as the interaction techniques themselves, the categories error tolerance and learnability are less weighted. Table 4.18 summarizes the weights and scoring of each usability category, leading to a satisfactory overall usability score of $U = 0.8905$.

4.5.3.5 Discussion

We could demonstrate, that all task types according to the task taxonomy are applicable in an adequate way. It can be stated that the system is usable and adaptable for large display interaction. The visual analytics application, compared to the immersive navigation application, incorporates less degree of freedom, making the control easier. As expected, the accuracy value of the visual analytics application (96.25 %) is higher than the one form the immersive navigation application (82.5 %). Analog observations were found for the effectiveness value (95.17 % vs. 76.73 %). Thus, the first application shows very good results; implying suitability of the system for this kind of application. The immersive navigation application, however, was in total more difficult. It could be observed that the control techniques felt less cumbersome towards the end of the evaluation compared to the beginning. After executing the interviews, it is expected that the effectiveness of the system with this kind of applications will increase after a longer training phase as users felt they could become easily skillful at using the system.

We could proof that multi-modal interaction realized with the use of a single smartwatch is suitable for exploration tasks and adaptable as large display interaction. With overall satisfactory user feedback and an usability score of 0.8905, the presented system demonstrates a more natural and novel way of interaction.

4.6 Discussion and Summary

In the course of this chapter, a clearly defined dependency graph of users in industrial corporations was designed and demonstrated. Based on the introduced user taxonomy, clearly defined entities and relationships of that network can be modeled. A set of dependency parameters were proposed to display a meaningful relation between objects and to give an easy understandable overview of the whole relationship with the goal to solve complex tasks and improve a groups' performance. A case study based on the dataset of projects within the IRTG 2057 college was performed. In the course of this observation, initially independent projects were examined and overlaps were found. Consequently, dependencies between users were clearly identified and connected across the corresponding information-flows. An overview of all existing relationships within that group is visualized in a complex all-embracing diagram. During the work of this research, it turned out, that this way of illustration is too complex for getting insights into a specific entity. However, the visualization is not detailed enough to gather any information about the calculated dependency parameters as defined. Therefore, we used several state-of-the-art methods that give the possibility to visualize classified data and show relations, connections and interdependencies of group members that perform a common task. Those different visualization techniques were applied to make complex system architecture easily understandable for users. The representations are embedded in an interactive application, which allows users to explore the recognized dependencies and links. To support an efficient interaction with the model and easy information gathering, the application focuses on the most important interdependencies by fading out irrelevant information and allowing users to switch between different visualizations without losing the focus.

According to the findings and gathered information of the investigation, two dedicated user groups (event driven production control and factory layout planning) and associated activities with a high level of detail have been modeled and explored. Those user-models served as scenarios for the development of novel and intuitive interaction and visualization techniques of a collaborative design and assessment system.

Afterwards, we have described and prototyped a general-purpose framework that can be adapted to the specific requirements of a specific application to support efficient collaborative design, simulation, or visual data analysis – done simultaneously by distributed or co-located teams using diverse mobile devices. The devices provide three views of the data to be processed collaboratively: (1) a simulation view; (2) a status report view; and (3) a status update view. These views serve the purpose of providing overview, detail, and performance views. Our approach goes beyond the known characteristics of existing "Overview-plus-Detail" techniques. The watch analogy employed by us provides a user with the information explaining the impact of user-induced changes made to a production process in a natural and intuitive manner. Comparable frameworks do not support an active manipulation of a simulation considering different task models and inferdependencies. When geographically separated from the collaboration system, users can monitor processes and actively perform changes to them to improve process progression. Visualizations provide high-level insights into a process' status, and the status report view leads to a deeper understanding of the effects resulting from optimizing the production process. The performance view shown on the watch display depicts whetheran action must be taken for an ongoing process or not. We have implemented an event-driven production control (EDPC) application as modeled earlier in a case study and successfully demonstrated the use and advantages of our framework, while achieving an overall usability value of 0.902.

As such, a setup is designed that increases the efficiency of collaborative design and assessment in virtual reality. Adequate visualization techniques have been applied and proven in a collaborative task. Afterwards, the aspect of active interaction and participation have been explored. Thus, novel natural and intuitive interaction techniques were developed.

Following, a signal processing approach for enhanced multi-modal interaction interfaces was presented. The approach is designed for smartwatches and smartphones for fully immersive environments that enhance the efficiency of interaction in virtual worlds in a natural and intuitive way. The combination of smartphone and smartwatch capabilities is introduced, outperforming a comparable common VR input device. This research deals with the replacement of the common touch input gestures with actual body movement gestures. We have demonstrated the effective use for a simple application. The full immersive system as combination of smartphone and smartwatch contributes advantages for highly effective and intuitive gesture-based interaction as following:

- Location independence;
- Simplicity-of-Use;
- Intuitive usability;
- Eyes-free interaction capability;
- Support for several different inputs;
- High degree of flexibility;
- Potential to reduce motion sickness;
- Elegant combination with existing input technology.

The challenge of processing the smart watch signals is described by missing sensor data, precisely gyroscope and magnetometer data, which are used together with acceleration to calculate orientation and motion dynamics of the device. We present a transformation of the given signal-processing from the smartwatch into arm movement gestures with the use of smoothing algorithms and gesture state machines.

Based on six key values and statement sequences we are able to define precise arm movement gestures, demonstrated with seven different gestures. More gestures are conceivable and easily adoptable with our approach. The findings of the user study prove that the system is able to recognize unique, primitive, and even complex gestures in an easy learnable way, while overcoming the missing/inaccurate sensor data in low budget technology. The presented approach performs quantitatively better compared to the existing gesture recognition technology. Besides, it allows the easy creation of diverse gestures by incorporating different kind of states and key values. The evaluation showed that complex gestures with many consecutive states are just as easy to design as more primitive ones.

The combination of touch gestures, non-touch gestures, and speech leads to a more natural and novel ways of interaction. Speech interaction as the most natural way of interaction enhances the range of common interaction techniques significantly. Together with touch- and non-touch gesture a wide range of natural and intuitive interaction capabilities are provided. The lightweight and portability of a smartwatch makes it very convenient to handle and fuse all the modalities into one single system. Based on first prototype combining touch, non-touch gestures, and speech as interaction techniques performed with a smartwatch, and the performed evaluation, we could identify shortfalls of our initial design.

Following, we could improve the system for better performance and usability. The results are described and incorporated in the final system. The performed user study of the final system provided some useful ways of combining speech, gesture, haptic, and touch interaction modes with a smartwatch showing an effectiveness value of 95.71 % and 76.73 %. As such, the system is suitable and adaptable for controlling large displays. We could gather overall satisfactory user feedback resulting in a usability score of 0.8905. Following, the presented system demonstrates more natural and novel way of interaction for large displays.

Summarizing, the investigations in this chapter answer the question of how to design dedicated interaction and visualization techniques in order to support single users as part of a collaboration team.

Contents of this chapter have lead to the following publications:

F. Rupprecht, F. Torner, J. Seewig, A. Ebert, "Dependency graph based on user taxonomy and related parameters for more efficient collaborative work", *Applied Mechanics and Materials 869 - Proceedings of the 1st Conference on Physical Modeling for Virtual Manufacturing Systems and Processes*, AMM 869:195-211, 2017.

F. Rupprecht, G. Kasakow, J. Aurich, B. Hamann, A. Ebert, "Improving Collaboration Efficiency via Diverse Networked Mobile Devices", *Journal of Multimodal User Interfaces*, Springer, 2017.

F. Rupprecht, A. Ebert, A. Schneider, B. Hamann, "Virtual Reality Meets Smartwatch: Intuitive, Natural, and Multi-Modal Interaction", *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, (pp. 2884-2890), ACM, 2017.

Accepted for publication:

F. Rupprecht, B. Heck, B. Hamann, A. Ebert, "Signal-processing Transformation Proceed from Smartwatch into Arm Movement Gestures", submitted at *the conference track "Human Factors and Systems Interaction" of the 9th International Conference on Applied Human Factors and Ergonomics*, AHFE, 2018.

Submitted:

F. Rupprecht, O. Joseph, C. Naranjo Valero, B. Hamann and A. Ebert, "Natural and Multi-Modal Interaction with Large Displays Using a Smartwatch: Speech, Touch, and Non-Touch Gestures", submitted at *The 10th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS, 2018.

Chapter 5

Collaborative Computer-Aided Decision Making and Design in Virtual Reality

According to the UCD principle 1, finally the performed examinations about *the product* and about *the user* are *brought together*. In this chapter, all findings and results from the chapters above are combined by examining the resulting proof-of-concept implementation as one single system. As such, making possible collaborative computer aided decision making and design in virtual reality. First, the framework's concept will be introduced. Following, a detailed description of the technical realization and integration as well as supporting functionalities and roles are explained. An evaluation of the complete prototype concludes this chapter.

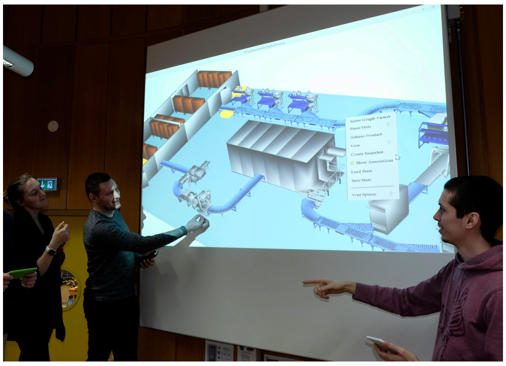
5.1 IN^{2}CO – INtuitive and INteractive COllaboration

The need for interaction and visualization techniques to fulfill user requirements for collaborative work is ever increasing. Current approaches do not suffice since they do not consider (1) the simultaneous work of participating users, (2) the provision of different views of the data being analyzed, or (3) the exchange of information between different data emphases and implicit inferdependencies. We introduce **IN**tuitive and **IN**teractive **CO**llaboration (IN^{2}CO), a scalable visualization framework that supports decision-making processes concerning multiple levels and multiple roles. IN^{2}CO improves the state of the art by integrating ubiquitous technologies and existing techniques to explore and manipulate data and dependencies collaboratively. Especially for decision-making of complex interrogation, no single person takes sole responsibility. Thus, considering and integrating the ideas and expertise from several experts is crucial. Real-time simultaneous multi-user software is common in gaming communities, where it is now much more routinely used than in other communities [106]. Such collaborative software can also be useful in other fields such as engineering settings with various simultaneous contributors. Here, collaboration is essential to identify and solve design conflicts in an early stage and, consequently, to reduce development lead-time and manufacturing costs. Common collaboration technologies mostly address work of distributed teams. There exist a wide range of tools undertaking mind mapping, file sharing, messaging, etc. These tools are mainly developed as single desktop applications. Co-located collaboration is often performed by one presenter and several spectators, whereby active participation is strongly limited. Our research focuses on an environmental setup for co-located and distributed collaborative work.

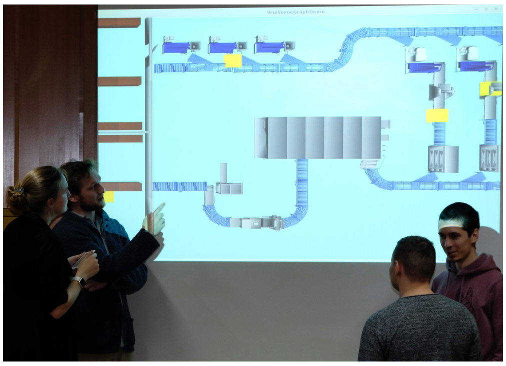
Large display devices (LDDs) enable the reproduction of large datasets in one view, by providing a large size and high resolution [249]. However, most LDD's interaction capabilities are designed for single users, so powerful and intuitive visualization and interaction capabilities are needed to support a larger number of users. On the other hand, mobile smart devices offer a wide range of interaction metaphors, leading to natural and intuitive interaction. In addition, mobile smart devices come with a display that can be used as secondary output device.

Complex data often comprises several levels on which different activity emphases exist (e.g., machine energy consumption or production rate). These emphases can have inferdependencies that must be identified and collaboratively solved. Changing attributes in one level might have an unaware or undesirable impact in another level of the same data. With the number of participants, the requirements for the visualization tool and techniques accumulate. Combining different core-competences and supporting intuitive data exploration for and between different activity emphases is still a challenging task.

IN²CO is a human-centric visualization framework for intuitive and collaborative data exploration and manipulation, as shown in Figure 5.1. Specifically, it's contribution is the integration of ubiquitous technologies and existing techniques to explore data and inferdependencies in collaborative decision-making for co-located and distributed participants. The ubiquitous technologies used in this proof-of concept system range from smartwatches via smart phones to tablets.



(A) Collaborative group task solving by rearranging participants – Performance of active discussion and exchange between all participants.



(B) Sub-teams solve individual tasks under consideration of other's inferdependencies. Face-to-face arrangement are useful for discussions.

FIGURE 5.1: Active collaboration of all participants. Participants optimizing the factory layout and material flow visualized on public viewport by controlling the scene with diverse smart devices.

5.1.1 Applying eHoQ Task Ontology onto Collaboration Framework

A challenge in the system design is the support of active participation of all users, starting with the question how active participation can be achieved in general. Thus, the established catalog of general-purpose user needs for collaborative work and environments, presented in Chapter 3, is used within the eHoQ approach in order to classify tasks according to the task taxonomy of the approach (also presented in Chapter 3). Thus, a feasible software architecture for collaborative computer-aided decision making in virtual reality will be designed. The tool needs to be highly scalable, in order to provide the ability to change the application and the hardware setup, user groups, and constellations. All user needs and corresponding task definitions are listed in Table 5.1.

Table 5.1: Applied task taxonomy onto general criteria for efficient collaboration and decision making. The horizontal lines separate the eight capacities of collaborative work, as described in Chapter 3.3.1.

#	Requirement	Task
1	Content integration	task (data manipulation, user, system)
2	Move	task (data manipulation, user, system)
3	Judge	task (info. acquisition, system, user)
4	Quickly retrieve information	task (info. acquisition, system, user)
5	Access to shared objects	task (data acquisition, system, user)
6	Accentuating	task (data manipulation, user, system)
		task (info. acquisition, user, user, system)
7	Track others approach	task (data acquisition, system, user)
		task (info. acquisition, user, user, system)
8	Screen sharing	task (data acquisition, system, user)
9	Individual and/or shared workspaces	task (data acquisition, system, user)
10	Jurisdiction	task (info. processing, user, user, system)
11	Transformations	task (data transfer, user, user, system)
		task (information transfer, user, system)
12	Alert mechanisms	task (info. acquisition, user, user, system)
13	Awareness Support	task (info. acquisition, user, user, system)
14	Community Support	task (external condition, user, user*)
15	Team structure and size	task (info. processing, user, system)
16	Changing work styles	task (info. processing, user, system)
17	Discussion tool	task (info. acquisition, user, user, system)
18	Communication in group/individual	task (information transfer, user, system)
19	Encrypted Communication	task (info. transfer, system, user)
20	Avoid team debates	task (external condition, user, user*)
21	Group process training	task (external condition, user, user*)
22	Reflecting all notions/opinions	task (info. processing, user, system)
23	Use guidelines/restrictions	task (external condition, user, user*)
24	Involving all actors	task (info. processing, user, system)
25	Team self-managing behaviors	task (external condition, user, user*)
26	Action parameter	task (data manipulation, system, user)
27	Access Control	task (data acquisition, system, user)
28	Session Persistence	task (data acquisition, system, user)
29	Consistency and interactivity	task (data acquisition, system, user)
30	Reliability	task (data manipulation, system, user)
31	Reusability	task (data manipulation, system, user)
32	Transferability of skills	task (info. transfer, user, system)
33	Flexible actor arrangements	task (info. processing, user, system)
34	Guidance	task (info. acquisition, system, user)
35	Generalizable & ease of maintenance	task (data acquisition, system, user)
36	Natural interpersonal interaction	task (info. processing, user, user, system)
37	High user satisfaction and motivation	task (info. processing, user, user, system)
38	Intuitive and simple technology	task (info. processing, user, user, system)
39	Reduced cognitive load of actors	task (info. processing, user, user, system)
40	No all-embracing knowledge needed	task (info. processing, user, user, system)

The absolute weights of all system attributes for the case of a general-purpose collaboration framework are calculated with the demonstrated procedure in section 3.4.2. The final results are listed in Table 5.2.

TABLE 5.2: Absolute weighting and importance ranking of system attributes

System attribute j	w_j	Rank
Data acquisition	179	**2**
Data manipulation	125	**4**
Data transfer	18	**7**
Information acquisition	131	**3**
Information processing	216	**1**
Information transfer	58	**6**
External conditions	90	**5**

Following, a clear identification of the absolute weightings and rankings of the importance values of the system attributes is easily identifiable. The system attributes are sorted according to their importance as listed in Table 5.2. Accordingly, designers/developers should take the most attention on the attribute "*Data acquisition*", while "*External conditions*" needs less attention. In analogy to Chapter 3, the corresponding rated system architecture is depicted in Figure 5.2

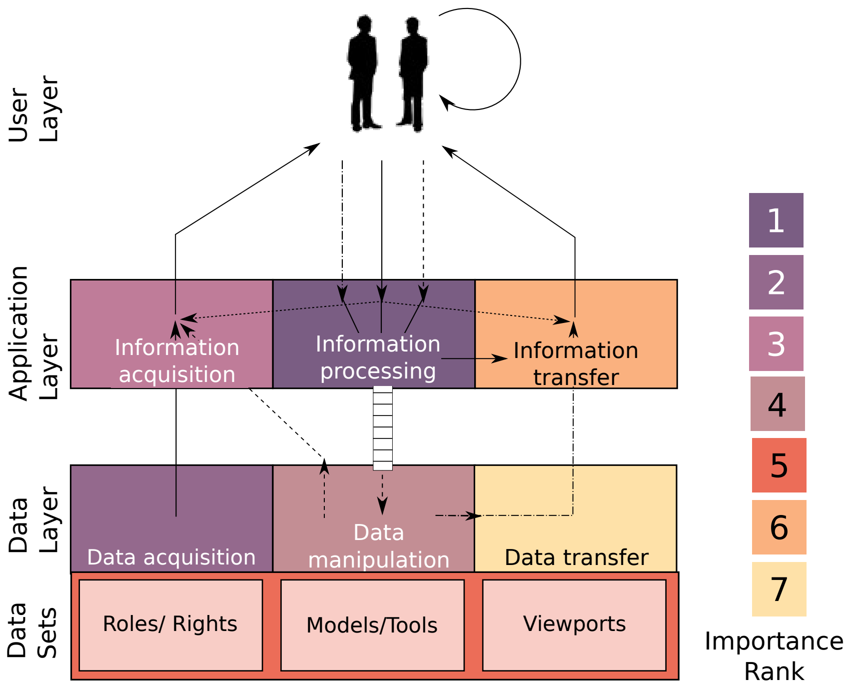


FIGURE 5.2: Rated components for a general purpose collaboration framework.

5.1.2 Software-Components and Architecture

All requirements stated in Table 5.1 will be covered by refining the system architecture in the following section.

5.1.2.1 *Basic System*

The basic system IN²CO consists of four components as seen in Figure 5.3: (1) Vrui, (2) CIMT, (3) a large display device, and (4) smart devices running with a Vrui Remote Application.

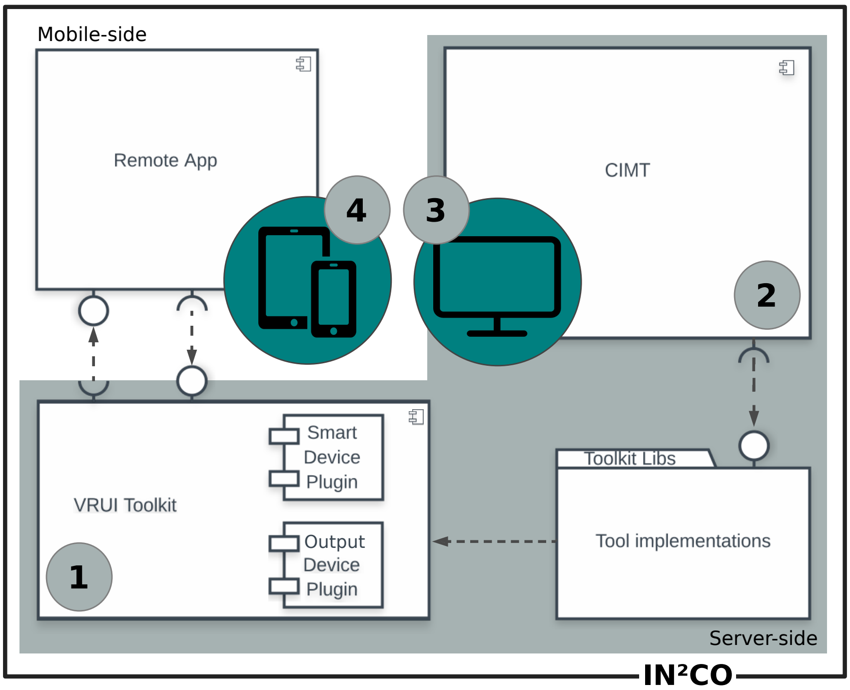


Figure 5.3: IN²CO Components: (1) Vrui, (2) CIMT, (3) a large display device, and (4) smart devices running with a Vrui Remote Application.

A foundational building-block of our system is Vrui [171], a VR development toolkit, which supports rendering and interface capabilities for common types of input and output devices. The task of the Vrui VR development toolkit is to shield an application developer from the particular configuration of a VR environment, such that applications can be developed quickly and in a portable and scalable fashion. Vrui contains class interface definitions and software components which need to be defined and implemented in an actual application. Vrui, initially designed for single user systems, is enhanced with additional tools, implemented in Collaboration Interaction Manipulation Tool (CIMT), enabling collaborative manipulation of the data and a smart device plug-in that enhances the input and output capacity of the system.

IN²CO integrates ubiquitous technologies representing a domain-oriented visualization framework with a focus on multilevel data analysis and multi-role perspectives, and interaction capabilities. The framework combines a large display device (LDD), used as output device and several mobile smart devices used as input and secondary output devices. Using smart devices as independent clients enables co-located cooperative tasks, while simultaneously offering interactive viewports using semantic snarfing for individual use. Thus, other devices – such as desktop systems, CAVE systems, or smart-devices in distributed locations – can be connected to the main server and join a session.

Smart devices offer a wide range of interaction metaphors, which can lead to natural and intuitive interaction as well as a broad array of control elements. As users can be explicitly identified, smart devices as interaction devices scale with the number of users. Furthermore, the smart device offers the possibility to use the screen as secondary output. Therefore, the overall design objective is to provide two viewports to each user. The LDD represents a shared viewport for all users, on which everyone can track the observation of the others and cooperatively discuss the same scene. Additionally, each user owns a private viewport on the smart device. On this private view, users see exclusively the information relevant to their domain. Symbolic input is a usual task of smart devices; notes and markings are made on the private view and synchronized with the shared view if desired. With the latest developments in wristwatch computers, new techniques (as described in Chapter 4) can be used to make the interaction more natural and intuitive. Next to smartphones and tablet computers with different sizes, smartwatches are used to interact with the model and support decision-making processes. Therefore, "Vrui Remote App" is the actual implementation on the smart device side. Here, the interaction and visualization techniques on the mobile side are defined, as well as message handler and observer interpreting incoming messages from the application.

The enhanced smart device plug-in located in Vrui provides the actual message handling and defines what kind of messages can be exchanged. The interpretation and utilization of the messages is done in the application layer in CIMT. The VR simulation world and the actual implementations of the tools are located in the application layer in CIMT. Communication interfaces, message handling and observer, proving write and read access rights, are implemented in order to associate incoming messages with the corresponding callbacks of data manipulation and inferdependence notifier. Designed as modular expandable and generally applicable, the CIMT application layer determines what is visualized on the large display device and what on the smart devices. Refinements of data manipulation, interaction tools, and information access is performed due to input parameters, which need to be provided before running the application. Before we go further into technical details, the coverage of the system attributes depicted in Figure 5.2 will be examined in the following section.

5.1.2.2 *Coverage of System Attributes*

Information acquisition, information processing, and information transfer is mainly covered by output and input device interfaces connected via communication interfaces, message handlers, and observers.

Information acquisition. Defines how the accessed/manipulated data is visualized on the LDD and smart devices and how changes are communicated to the users. Visualization and communication/notification techniques in the Vrui Remote App facilitates this task on the mobile devices side. Included and refined visualization techniques in CIMT's application layer realizes this attribute.

Information processing. Defines how uses can interact with the system. By providing each user with a smart device in order to actively control and participate in the collaboration process the simultaneous communication between user and server needs to be defined and handled. The smart device plug-in, the tool implementations, and Vrui Remote app cover these properties. The implemented message handler and observer transfer requests to data manipulation.

Information transfer. Information transfer refers to the exchange of information between users. On the one side, natural interpersonal communication between users is enabled due to the shared physical location. On the other side, an external message handler, which transfers information that do not directly influencing the data set, is provided by communication interfaces between devices and refined in the CIMT application layer.

Data acquisition. As everyone can actively participate and control the state of the application, roles and rights have to guarantee which content/data is provided to the user. The message handler in CIMT interprets the messages and the observers prove the users' read access. Predefined viewports and dedicated visualization call backs are transferred to information acquisition.

Data manipulation. Similar to data acquisition, as everyone can actively participate and control the state of the application, roles and rights have to specify which user can perform what kind of data manipulation. CIMT manipulation tools are informed (by observers) when the user has access to perform the manipulation, which executes the manipulation of the underlying simulation model. If an actual data manipulation is triggered, the information is transferred to the data acquisition module in order to provide feedback to the user. Also, system internal tasks like transaction handling are performed. These transactions do not directly manipulate the data but guarantee the simultaneous manipulation by several users. These transactions do not require an additional read access proof in the data acquisition module, thus the information are directly transferred to the information acquisition module.

Data transfer. Refers to transitions between personal and group work, between activities, and between the system and external systems. Interfaces between data layers, message creators in the Vrui Remote App, and communication interfaces enable these data transfers. The transfer of data either triggers further transactions in the system attribute information or leads to the depart use of the data from the system to an external system.

Thus, the designed IN²CO components cover all system modules as proposed in Section 5.1.1.

5.1.3 Realization of System Attributes

Build upon Vrui, the IN²CO framework has been extended with the following schematic modules, as sketched in Figure 5.4:

- **Smartdevice interface** links smart devices and triggers message-exchanges;
- **Graphical User Interfaces (GUI)** register smart devices with the environment;
- **Basis module** undertakes supportive activities like parsing for import and export and further data manipulation tools;

- **Collaboration module** triggers user registry, object distribution, data exchange and transaction handling;
- **Application interface** holds user specific viewports; user roles; tool and functionality collection for the tasks/usable devices;
- **Data storage** collects all application-specific values with impact links between processes, and contains all session logs for recording and recovering.

The functionality and application of these modules are described in the following sections. Therefore, we examine the order of events fielded into events before runtime and during runtime.

5.1.3.1 *Before Runtime*

The graphical user interface assists choosing and aggregating the needed plug-ins and devices, triggering the *system registry* (black data flow in Figure 5.4), the *user registry* (blue data flow in Figure 5.4), and finally program execution.

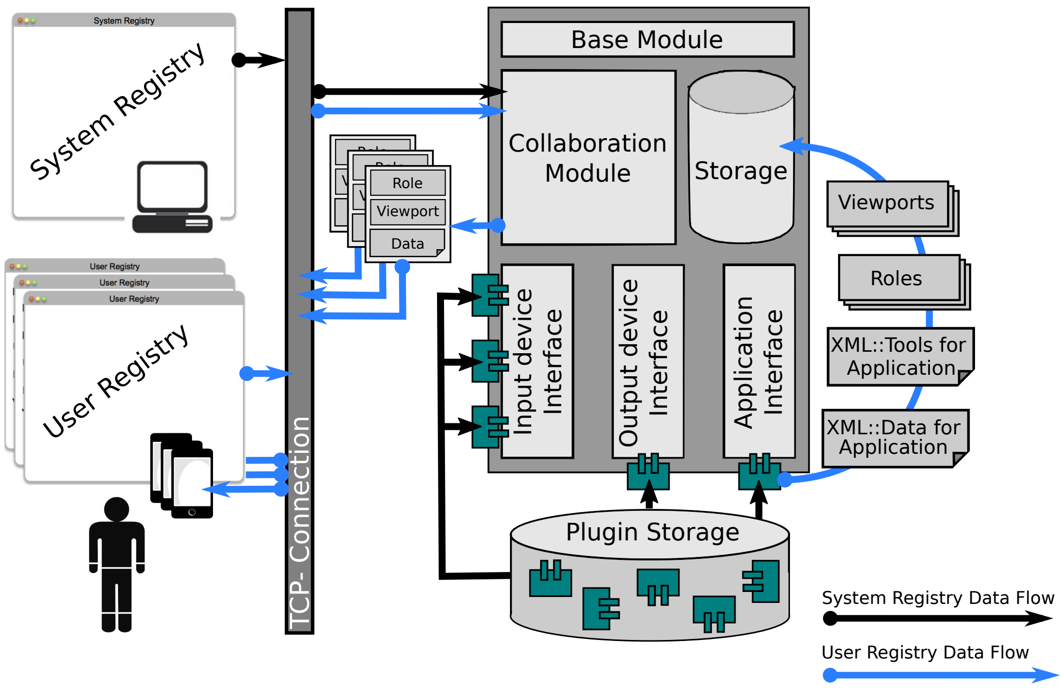


FIGURE 5.4: Schemed system modules connected with graphical user interfaced via tcp connection. Data flows of system registry (black) and user registry (blue) represent the initial configuration of the system.

The framework is highly scalable and generalized, thus, it is configurable to combine any kinds of input devices, output devices, and applications. Therefore, the setup has to be configured in advance. Originally, a *config file* is manually created signalizing which tools and interfaces are required. For better usability, a graphical user interface is appended in order to allow non-experts to configure such immersive system. An indispensable assumption is the existence of an application. In order to differentiate between users, this application holds different user roles that are assigned with privileges. The user roles and rights are integrated in our prototype system and will be described in the following sections.

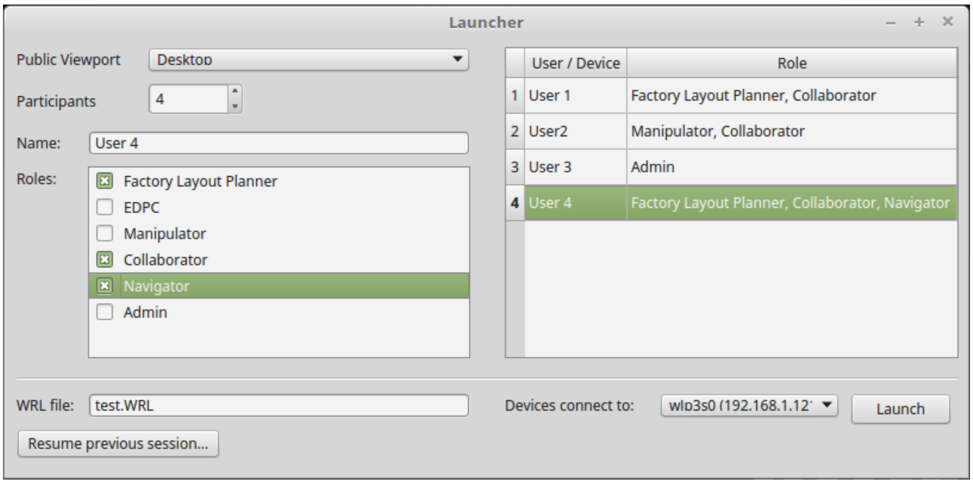


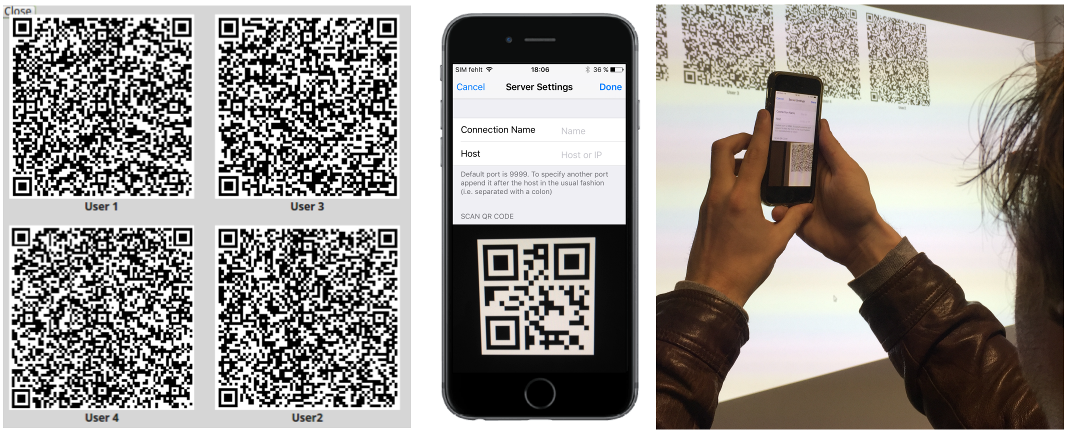
Figure 5.5: Default graphical user interface facilitates the registration of users and devices, associated with roles.

The graphical user interface (Launcher) in Figure 5.5 facilitates the user to choose and aggregate the available hardware with the created application. In order to configure the collaboration setup, the hardware to be used as public viewport and the number of participants have to be defined. In the Figure a simple desktop environment is used. By increasing the number of participants, a user is added in the table on the right side. The launcher uses the predefined roles of the application, which are assignable to the created user in the table. One user can be associated with several roles. Once all users are created, the applications scene can be chosen. In this example a scenegraph in form of a *wrl file* is selected. Finally, the IP of the server has to be selected.

The corresponding system registry module links all appropriated resources and plugins to the program, starts the application and initiates the user registry. Therefore, one QR code per user is generated (see Figure 5.6a) holding the user's information (name and roles) and system's information (IP). A QR code reader is integrated on the mobile application on smart device side (see Figure 5.6b), making it easy for the participants to register and join the collaboration session. Users merely have to scan the QR code with their user name, as shown in Figure 5.6c. Associated roles, viewports, and rights are linked within the profile on both sides, smart device side and application side, and connects the registered user/devices with the server (application). All configuration information like the system IP, user name, and roles are set and the user is automatically connected with the system.

The mobile application sends a handshake message to the server side and receives user associated viewports and manipulation rights. A detailed description of the input- and output capabilities of smart devices will be given in the next section.

Once the QR codes are generated, the configuration file for the setup is automatically generated and the participants are registered. The QR codes can be shown and hidden to the participants at any time. Thus, additional participants can join an ongoing collaboration session without a complete new configuration of the setup. The sequence diagram in Figure 5.7 gives an overview of the described registry processes and initializing of the program execution.



(A) Users QR codes (B) Mobile App (C) User connects with system

FIGURE 5.6: User friendly connection with the system via QR codes and predefined configuration user interface.

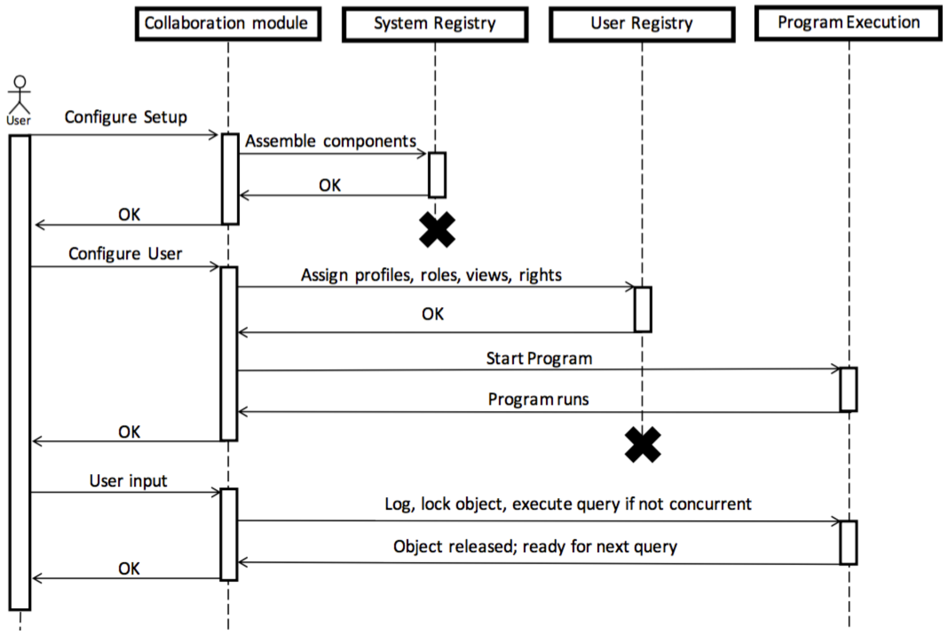


FIGURE 5.7: Sequence diagram describing information flow during registry processes.

5.1.3.2 *During Runtime*

The deployment diagram in Figure 5.8 represents the connection between the initially described IN^{2}CO components during run time.

During runtime, the application is running on the server side in CIMT. The application uses the visualization and manipulation tools from the Vrui toolkit library, provided by the Vrui software. The mobile application is running on the smart devices, which act as independent clients. The enhanced smart device plug-in is built on top of Vrui and the corresponding visualization and manipulation tools are integrated in the Vrui toolkit library. The clients communicate via Vrui with the application on server side in CIMT.

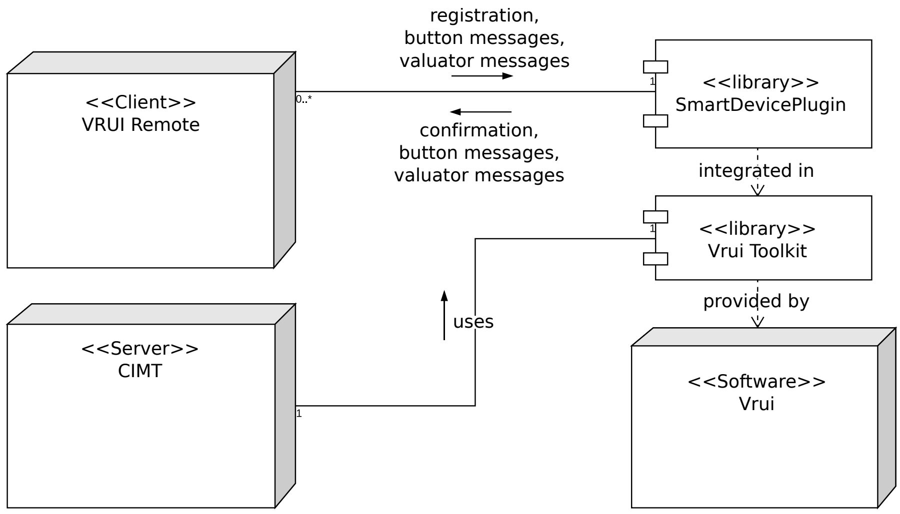


FIGURE 5.8: Deployment diagram of IN²CO components: (1) Vrui, (2) CIMT, (3) Vrui Remote

Communication smart devices and basis. Once connected to the main application, the smart devices send dedicated messages (based on JSON files [93]) to the main application. A message-handler, included in the main application, transfers the incoming messages from the smart devices to scene manipulation tools. These tools are included within the selected task model and provide functionalities and visual representations for both the simulation view and the clients (smart devices). Initially, these tools are created and listen for incoming messages, which triggers the functionality in the main application. Both the main application and the smart devices are directly connected to the database system and can trigger update of the database entities. It is crucial to not only update the visual representation in the simulation view but also to update the underlying data structures and information for all other clients (smart devices). The sequence diagram in Figure 5.9 shows the message exchange between the simultaneous living processes.

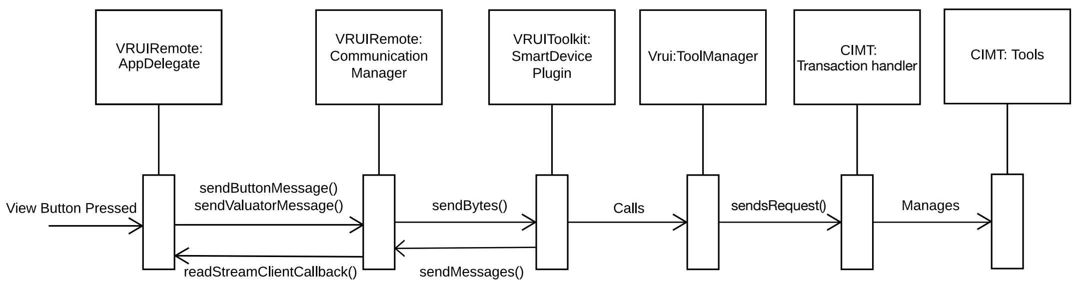


FIGURE 5.9: Sequence diagram of message handling in the system initiated by a smart device client.

Communication across smart devices. In the IN²CO framework, the smartwatch is not an independent client of the system, but the enlargement of the smart phone as distributed viewport. As independent device, the watch does not represent enough information and does not directly communicate with the main application or the database system. However, this small scale device is used additionally to enhance the interaction capabilities by including the performance of natural and intuitive gestures in the VR world based on arm-movements (see Chapter 4). Distributing the status-update view to smart watches leads to several challenges. All task model-dedicated data and information are requested by the phone and continuously sent to the watch via Bluetooth connection [83].

5.2 Implementation and Integration

5.2.1 Input and Output Technologies

The main application, running on a LDD, communicates via Wi-Fi [140] using TCP/IP protocols [102] with the mobile device application. The main application starts the server and initiates message handling. The devices get connected to the server. Interaction with the main application is made possible via mobile devices that directly communicate with the main server through a local WIFI network. Task model-dedicated data and information are stored in a MySQL database [221], which is updated when changes in the main application are performed, causing continuously incoming requests from mobile device applications, see Figure 5.10.

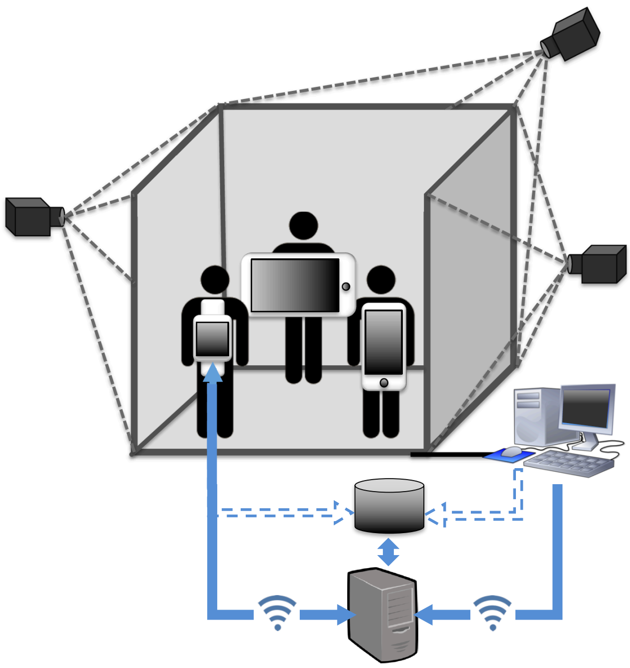
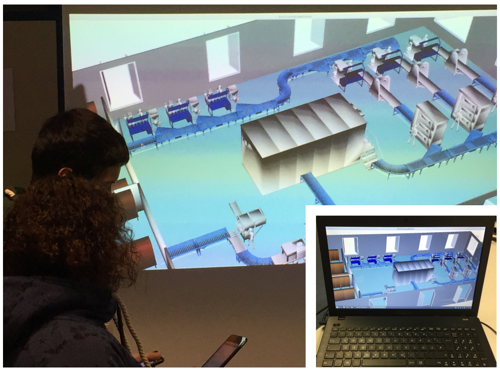


FIGURE 5.10: Setup and communication channels of collaboration framework.

The database is persistent, i.e., database tables are created once and can be used in each collaboration session without prior creation of the database structures. MySQL includes an InnoDB storage engine as consistency model that adheres closely to the ACID model [221]. The ACID model describes the four properties atomicity, consistency, isolation, and durability, used as major guarantees of transaction paradigms within database applications. Data is not corrupted and results are not distorted by exceptional conditions, such as software crashes or hardware malfunctions. Consistency checking and crash recovery mechanisms are included, and data reliability for several users is ensured.

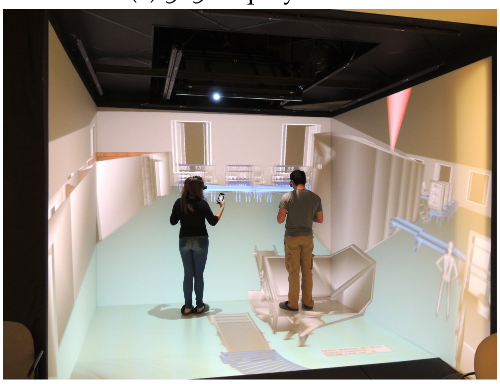
5.2.1.1 *Large Display Device as Output Device*

The large display device is used as public viewport on which the simulation world is presented to all users simultaneously.



(A) Laptop application with projection

(B) 3x3 display tiled wall



(C) 4-sided CAVE system

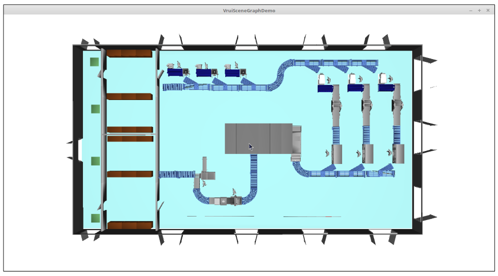
FIGURE 5.11: High immersion LDDs used as public viewports

The framework has been implemented and tested with three different large display devices as public viewports: (1) A simple desktop application extended with a projection device, (2) a tiled wall, and (3) a Cave Automatic Virtual Environment (CAVE) system. Figure 5.11 shows the used large display devices. Figure 5.11a shows the same scene and application running on a laptop (right corner), enhanced with a full HD projection device.

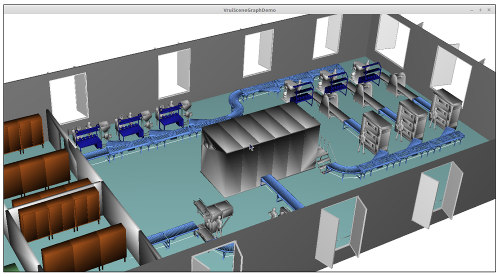
Figure 5.11b shows a 3x3 display high-resolution tiled wall located at the computer graphics and HCI group at the University of Kaiserslautern, driven by personal computers, each equipped with an Intel Core Duo 6600 processor and dual NVIDIA GeForce 7950 GX2.

Figure 5.11c shows a four sided Mechdyne CAVE, an immersive visualization environment consisting of three 10′ x 8′ walls and a 10′ x 8′ floor, located at the University of California, Davis. Stereoscopic images are projected onto all four surfaces using one 3-chip DLP projector (Christie Mirage S+4K) each. A user perceives a seamless three-dimensional environment that can be explored by manipulating data within the 10′ x 10′ x 8′ CAVE. The CAVE is driven by a cluster of 6 high-end graphics workstations running Linux and the custom virtual reality operating system Vrui. A head node with an Intel Core i7-870 CPU at 2.9GHz, 8GB of main memory, and 12TB of RAID disk storage controls the system, while the images for the projection are generated by four render nodes, each with an Intel Core i7-920 CPU at 2.67GHz, 6GB of main memory, and an Nvidia Quadro FX 5800G 3D graphics card.

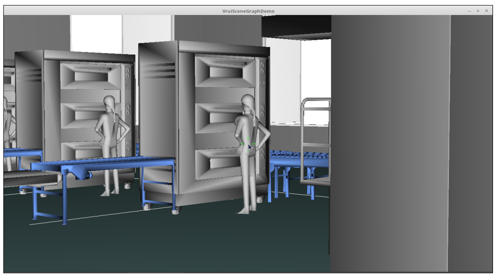
All three setups hold exactly the same application and interaction capabilities. Merely the degree of immersion is scaled. The tiled wall and projection provide the same degree of immersion as both are two-dimensional output devices providing no depth perception. The white stripes on the displays as seen in Figure 5.11b are used to project onto the display bezels that leads neither to discontinuities nor to significant loss of information. This technique, as presented and described in [92], eliminates ambiguities that commonly occur on tiled displays and improves the usability of multi-monitor systems by virtually eliminating the bezels. The CAVE system, however, provides depth perception on 4 sides. Thus, participants are not distracted by physical elements of the real location, which increases the perception of being physically in a non-physical world.



(A) Top view



(B) 3D view



(C) First person view

FIGURE 5.12: Visualizations of simulation world on public viewport

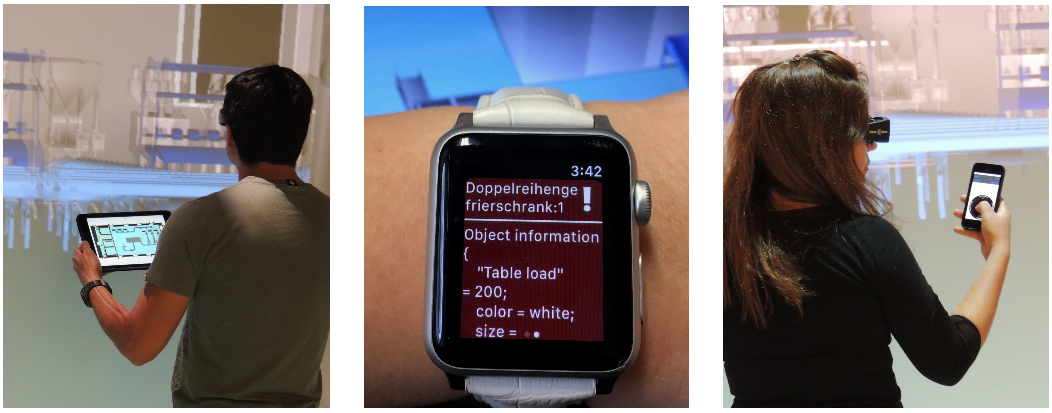
This public viewport is used to establish a common knowledge basis for all participants, enhancing collaborative task solving. Only common data and information is presented, while user/role specific information is visualized via private viewports on the smart devices. In the example, a virtual factory layout, used for the task models of factory layout planning and EDPC as described in Chapter 4, describes the common knowledge basis for all participants. The public viewport of the framework provides an overview of the underlying manufacturing system, consisting of a factory building, storage areas, machines, human resources, and conveyors.

The 3D simulation world is visualized in three different modes as depicted in Figure 5.12. Predefined 2D representation of all six angles (top, bottom, left, right, front, back) of the scene is provided. To exemplify, the top view of the scene is shown in Figure 5.12a. 3D navigation allows the 3D observation of the scene, scaling the degree of immersion as depicted in Figure 5.12b. The highest degree of immersion is established by a first person view, as shown in Figure 5.12c. This view is qualified for jointly observing a specific object. For the simultaneous observation of different objects, the first person view is less qualified.

5.2.1.2 *Smart Device as Output Device*

While the public viewport is used to establish a common knowledge basis for all participants, user/role specific information is visualized via private viewports on the smart devices. The support of multiple diverse user groups implies the existence of multiple diverse interest and focus on the data set. An essential task for decision makers is the rapid extraction of relevant information from the flood of data [159]. However, the bandwidth of available information is higher than individually needed. According to the information seeking mantra "Overview first, zoom and filter, then details-on-demand" proposed by Shneidermann [264], uninteresting items should be filtered out and hidden in order to efficiently process information. In some scenarios, it can be useful to maintain the awareness that more information is available than shown. In other scenarios, irrelevant data should be filtered out and made invisible in order to focus the visualization entirely on relevant data [287]. As different users have highly varied needs for filtering features, smart devices are personally used to perceive the individually needed information. Thus, users, acting in differing roles, are not distracted by role specific visualizations on the public viewport.

In our prototype implementation, we provide connection, message exchanges, and task dedicated interaction and visualization techniques on six different scaled smart devices: iPhone5 (3,5") [11], iPhone SE (4") [13], iPhone 6 plus (4,7") [12], iPad mini (7,9") [10], iPad2 (9,7") [9], and Apple Watch Sport [8]. The implementation on client side is done with native user elements and integrated web-views that allow platform independence. The used programming language is Objective-C [14]. Figure 5.13 gives an overview of the used device types in the framework.



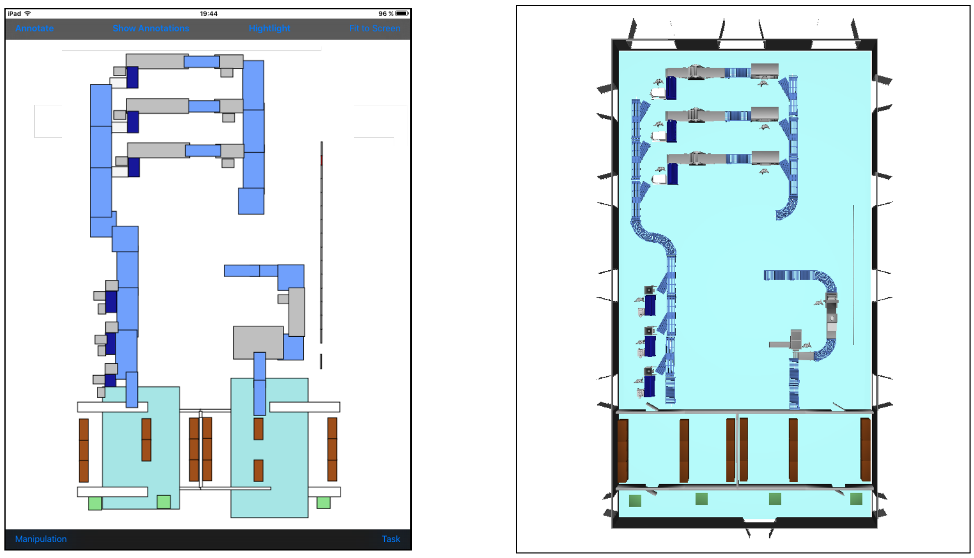
(A) Tablet (B) Smartwatch (C) Smart phone

FIGURE 5.13: Integrated smart devices acting as input and secondary output-devices scaling from smartwatch to tablet computers

As shown in Figure 5.13, the smart devices used as output devices scale from really small displays (Figure 5.13b) to relatively big ones (Figure 5.13a). Thus, the information provided to the user is adjusted to the available display size. While the bigger displays are used to represent any of the used visualizations, the smartwatch is only used for dedicated information visualization as a complementary display that enhances the smart phone visualizations. On bigger scale smart devices (smartphones and tablets), three individual views are provided. A view renders content and handles any interactions with that content. The provided views are seperated into (1) Navigation view, (2) Geometry view, and (3) Task view.

Navigation view. The initial start view, called navigation view, as shown in Figure 5.17b represents no visual elements. By touching anywhere on the screen, a radial menu pops up enabling interaction with the system, as described below.

Geometry view. For individual examination (without changing the view/ manipulating the scene on the public viewport) a reduced visualization of the scene is shown on smartphone and tablet devices. By connecting the device with the framework, simple bounding boxes of the objects are calculated and the coordinates are transferred to the smart devices. Thus, a simplified 2D visualization of the top view of the scene is rendered on the mobile side.



(A) Scene simplified visualized on smart device

(B) Scene as visualized on public viewport

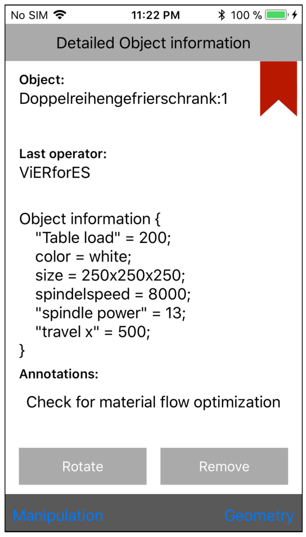
FIGURE 5.14: Scene in simulation world, visualized on public viewport, is send to and visualized on smart device

Figure 5.14 depicts the scene from the simulation world visualized on smart device side (Figure 5.14a) and on the large display device (Figure 5.14b). Although, only a simplified representation of the scene is rendered on mobile side, a connection between the objects on the mobile side and the public viewport is easily conceived.

Task view. Dedicated task views provide specific information and visualizations in order to support explicit tasks as stated in Chapter 4.3.5. Next to the simplified geometry data, also object specific information are transferred from the server to the clients during the connection process. On smart device side, the task view of the factory layout planning task model shows the transferred information. If an object within the geometry view is selected, specific machine information (like dimensions, capacities, speed) is extracted from a dictionary, and displayed on the smart phone and smartwatch (as at-a-glance information). Furthermore, on the smartphone, it is indicated who was the last operator, who manipulated the object, and if the object got highlighted (red flag in right upper corner). In order to discuss performed changes and alternative configurations, the awareness indication of the last operator is provided. In other situations, it might be necessary to highlight specific objects in order to accentuate that there is a need for discussion at a later juncture not interrupting the actual task execution.

According to the "Information Seeking Mantra" by Shneidermann [264], the smart devices are used to filter irrelevant data and facilitate details-on-demand functionality. Details are provided by changing the view within the private viewport or enhancing the view by using a smartwatch.

(A) Task view for EDPC on private viewport



(B) Task view for factory layout planning on private viewport

FIGURE 5.15: Private viewport shows task specific information

(A) Enhanced private viewport for EDPC

(B) Enhanced private viewport for layout planing

FIGURE 5.16: Enhanced private viewport on smartwatch

The small display size of the smartwatch makes it challenging to provide comprehensive information or visually complex elements. Therefore, merely simple and easy to capture visualization are used as well as textual output. Figure 5.16a shows the enhanced viewport for the task model of EDPC, while Figure 5.16b shows the enhanced viewport for the task model for factory layout planning. In the first example, the enhanced viewport is used to provide an overview of the ongoing production process. The user is informed at a glance about problems and the production process as a whole. According to the principle "details-on-demand", more specific information of one production orders (glyphs on the smartwatch representing production orders as described in Chapter 4.4) is accordingly provided on the smartphone. In the second example, the enhanced viewport on the smartwatch is used to provide at-a glance information of the selected machine, while details-on-demand are shown on the dedicated task view. This additional viewport allows the user to focus and concentrate on other tasks, while still being updated on his ongoing process. The object's information is updated when selecting a new object on the geometry view. The background color indicates, if the object was highlighted by another participant. And the exclamation mark in the right upper corner indicates, if an annotation has been created to the selected object. Detailed information of the last manipulator or the annotation text are shown in the dedicated task view. To keep the focus and overview, users do not have to switch views or tasks. Thus, users with different roles are able to discuss and jointly reflect results and possible configurations or situations in order to find a comprehensive solution without getting completely distracted from their own individual tasks. Collaborative decision making and individual task solving is facilitated and, likewise, switching between collaboration styles is completely hassle-free and in a natural manner.

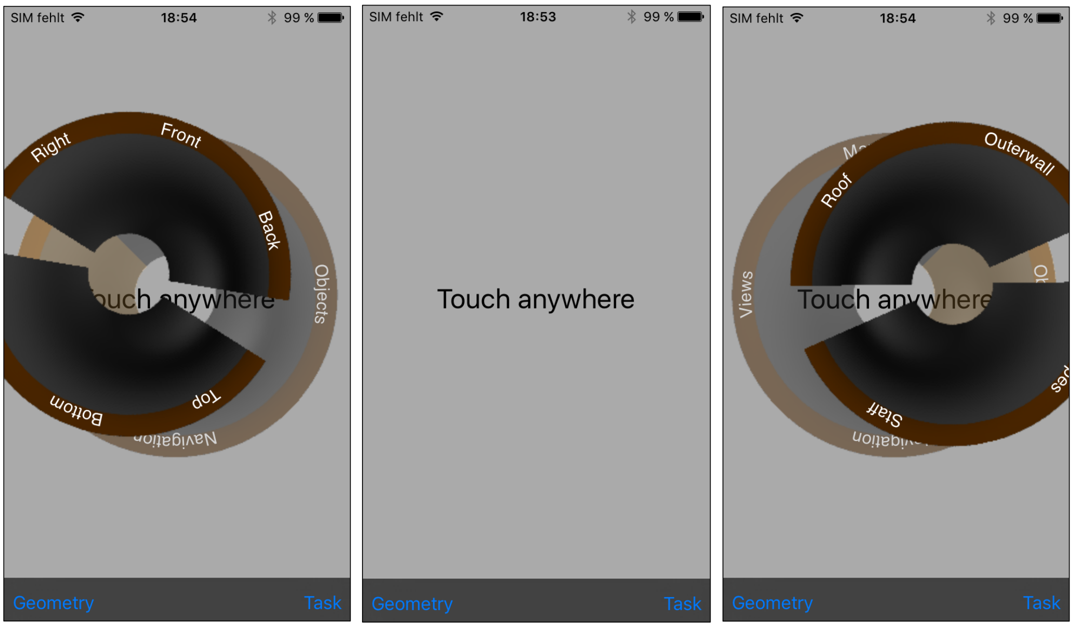
5.2.1.3 *Smart Device as Input Device*

In a study from Birnholtz et al. [37] it is stated, that multiple input devices lead to higher groupwork while in a single input device setup, one participant dominates the task and a high frustration level is recorded for those participants that do not control the system. In order to support the active participation of all users, each user is provided with a smart device with which everyone has active control of the scene and is able to manipulate the data according to their assigned roles and rights. In the context of a multi-touch wall, investigations by Marshall et al. [195] recorded higher participation and more equal interactive participation with touch input and multiple entry points compared to stationary input devices. In analogy to a multi-touch display wall, the wireless mobile devices in our setup allow the user to move around freely. Thus, participants can observe the data literally from different angles and change freely their communication partners. Interactions are more fluid and interferences can be resolved more quickly, as stated in [130]. In the following, the realized input capabilities are examined separately in analogy to the three views already introduced above.

Navigation view. The initial view, as shown in Figure 5.17b does not provide any task related visual representations or indication. By positioning a finger on the screen, a marking menu as introduced by Bauer et al. [24], pops up. According to the authors, eyes-free interaction for experienced users who do not need the visual feedback from the mobile device is enabled. Thus, the efficiency for expert users is increased. While at the same time the menu structures are kept visible, in case they are needed.

As implemented in our example, the radial menu facilitates the user to connect with the system, either manual by typing in the server information or automatically by opening the QR code reader. After connecting the device with the system, moving the finger to the left side (see Figure 5.17a) facilitates the user to select one of the predefined views, triggering the corresponding transformation on the public view. Moving the finger to the right side (see Figure 5.17c) opens a menu containing object groups. The selection of an object group triggers the function to hide/show the object in the public viewport. Moving up enables panning and rotating of single selected objects and moving down enables equivalent functionality for the complete model. The menu-bar on the bottom of the view enables to switch between the different views within the smart device application.

Geometry view. An important task while interacting with the simulation world is the selection of objects. A complete natural behavior is to point at a specific object. Pointing gestures with smart devices, however, are not sufficient. Thus, the visual representation of the model inside the geometry view is used to select single objects by simply tapping on the object. After selecting an object, further functionalities that enable the manipulation of single objects in the simulation world become visible. The implemented functionalities are described in detail in the next sections. The



(A) Marking menu facilitates to change the simulation view (B) Start view containing the marking menu (C) Marking menu facilitates to show/hide object groups

FIGURE 5.17: Marking menu on initial view facilitates user to manipulate the complete scene

performed manipulations are transferred to the public viewport and further send to each client, which adjusts the transformations of the simulation model on each node, leading to the exact same views on public viewport and private viewports of every client. As described in Chapter 4.5.1, smartwatches are integrated in order to perform object transformations. This is leading to a highly intuitive and natural interaction with the system. Exemplified, the circle gesture as introduced in Chapter 4.5.2 is applied, which performs a stepwise rotation of the selected object.

Task view. The task view provides next to task specific information also task dedicated functionality. Depending on the user role and corresponding rights that are associated with the device, different information and functionalities are enabled, as depicted in Figure 5.15. A detailed description of user roles and associated rights is provided in the next sections.

5.2.2 Supported Computer-Aided Design Functionality

Next to task specific functionalities, generally required functionalities are provided. Supported functionalities are listed below and explained underneath.

- **Navigation:** rotate, pan, and zoom of the whole model; first-person view and navigation; selection of pre-defined views;
- **Manipulation:** rotate, pan, and zoom, duplicate, delete of single objects, hide/show object-groups;
- **Examination:** measurement of distances and dimensions; textual output of object-information;
- **User feedback:** highlighting and vibration;

- **Collaborative features:** making annotations, marking areas, and creating a visual snapshot, show/hide accentuates.

5.2.2.1 *Navigation*

Navigation and manipulation are distinguished in order to differentiate between control functionality of the complete scene and transformations of single objects within the scene. Navigation functionality changes what is shown on the public viewport, as depicted in Figure 5.12. In a collaborative setting, navigation functionality should be performed with the agreement of all participants. If participants perform individual tasks while observing the scene on the public viewport, it is obstructive when other participants change the public viewport. By adjusting the view in order to change the focus, other participants might lose their focus and get interrupted on performing their own tasks. Collaborative group task solving performed with all participants, however, is facilitated by these navigation functionalities. The navigation capabilities for smart devices are implemented with eyes-free interaction marking menus in the navigation view, see [24].

5.2.2.2 *Manipulation*

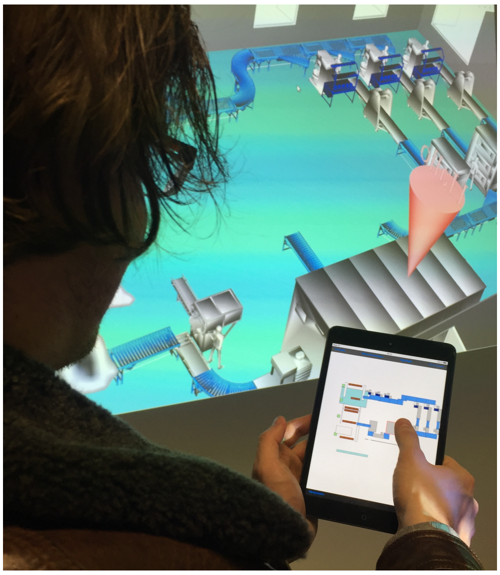


Figure 5.18: Object transformation as manipulation functionality implemented on smart devices

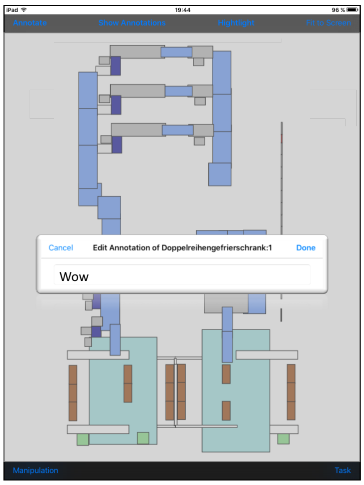
Manipulation functionalities lead to transformations of single objects within the scene. These functionalities are synchronized with all devices in real time. Participants can select objects, which further can be moved or rotated within the scene. The performed transformations (including the new object coordinates) are sent to the server and forwarded to all clients. The corresponding object on each side is transformed accordingly to its new properties. While navigation functionality in a collaboration setting should be performed rarely, manipulation functionality implies less restrictions.

Only one participant at a time can select and transform a specific object. When selected, the object is locked and cannot be selected or transformed by other participants. Transformation handling grants the participants with the transformation privileges and triggers corresponding functionalities. Adjustments performed by one participant should not be reversed without previous discussion or agreement by another participant. But, in order to facilitate open and active collaboration of diverse participants, manipulation functionality should not be restricted and it should allow the creation of diverse configurations and alternatives. Figure 5.18 shows the movement of a selected object as manipulation functionality.

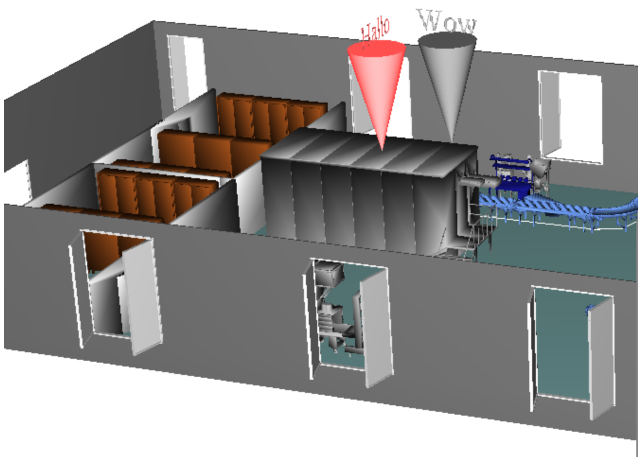
5.2.2.3 *Collaborative Features and User Feedback*

Collaborative work is enabled as each user has their own control device and everyone can track the changes of others. All transactions and requests are handled on the server side. Particularly important is to provide awareness on ongoing processes and inferdependencies.

Participants can perform accentuation by creating annotations or highlighting particular objects. These accentuations are visualized on the public viewport but also on smart device side as indicated in Figure 5.15b. Participants are associated with an individual color, then, selected objects and annotation indicator are highlighted in the corresponding user color. This makes all participants aware of the ongoing processes of others. Figure 5.19 shows the creation of annotations (Figure 5.19a) and the indication of the existence of an annotation (Figure 5.19b). To prevent visual clutter on the public viewport, all accentuations can be hidden or shown as demanded. On the smart device visualization, accentuations are still indicated to ensure awareness. After discussing the purpose of the created accentuations, those artifacts can be deleted by everyone.



(A) Creation of annotations enabled with the use of smart devices.



(B) Visualized annotations in public viewport, color-coded in creator's user color.

FIGURE 5.19: Creation and visualization of annotations

User feedback is enabled in various forms. Object selection and clashed objects are color-coded, indicating that objects cannot be positioned on those coordinates. The haptic engine of the smart-watch is used to signal inferdependencies between settings of different users. For example, clashing objects sends a notification to clients which leads to a haptic feedback on the smart watch.

Users can observe the manipulation of a specific object performed by other participants. Once selected, detailed information of an object is shown in the task view and the smartwatch task view enhancement. After releasing the object, the object information is still shown and updated on the task views. Conflicts and synchronizations are solved in a straightforward manner. The participant who selects an object first has exclusive rights for this object.

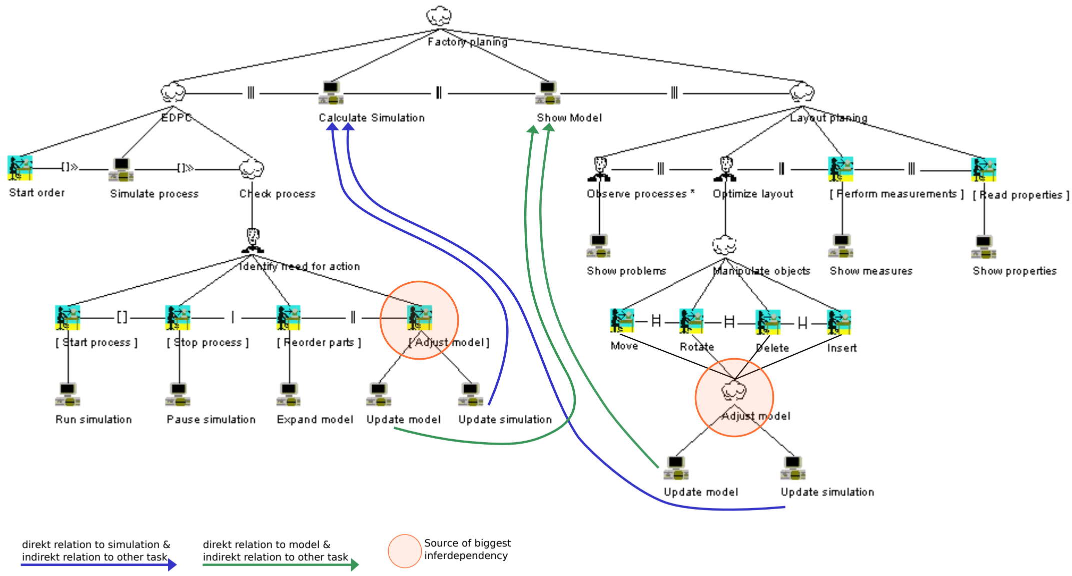


FIGURE 5.20: Concur task tree diagram of combined task model representing task structure and inferdependencies of event driven production control and factory layout planning.

5.2.3 Roles, Tasks, and Rights

Different participants have different foci or desire to execute different activities or emphasize various aspects of the data. Inferdependencies between these activities can exist and lead to a prior unknown and potentially undesirable impact on each other. To avoid unauthorized data manipulation, multiple user-roles (and associated privileges) are used. User roles refer to role-based access control of objects or services. Domain-specific tasks and interactions are defined in our system, and corresponding viewports are designed. We also established an ontology that defines user-roles, together with task-specific viewports and interactions that are assigned to the users. The definition of user roles (and privileges) and viewports is done at the API level but the declaration is done at user level.

The IN^{2}CO-prototype followed the user-centric design methodology, starting with a user and task analysis involving engineers from factory planning, which represents an appropriate application to demonstrate the usefulness and benefits of the desired system. The implemented and supported user- and task-models of event driven production control and factory layout planning are described in Chapter 4.3.5. As a reminder, the concur tree diagram in Figure 5.20 visualizes the task structures and inferdependecies of both tasks models. Next to the already mentioned functionalities for manipulation, navigation, and collaboration, few task-specific functionalities have been added as described in Chapter 4.4.2.

5.2.3.1 *Mapping Functionality to User Roles and Rights*

In order to identify different users, the following roles are defined and associated with the implemented functionalities. Each functionality is associated with exactly one role, implying the existence of distinct roles. One or several roles can be associated with one user/actor.

Factory layout planner, basic
• measurement of distances
• measurement of dimensions
• getting machine information
• see facility information
• creation/ removal of machines

EDPC, basic
• start new product order
• stop production simulation
• re-order machining parts

Manipulator
• rotate, pan, and zoom
• duplicate object
• delete object
• hide/show object groups

Collaborator
• making annotations
• marking areas
• creating visual snapshot
• show/hide highlights

Navigator
• rotate, pan, and zoom
• first-person view and navigation
• selection of pre-defined views

Each participant is at least associated with the role of a collaborator. For both described task-models the corresponding role should be selected together with the manipulator role, as in both use cases it is required to change objects' positions.

5.2.3.2 *Mapping Functionality to Devices and Views*

Although all functionalities are also implemented for a desktop setup, in the following Table 5.3 a mapping of the realized functionalities onto smart device capacities is performed.

Table 5.3: Mapping of integrated functionalities onto smart device capacities.

Functionality	Navigation view	Geometry view	Task view	Technique
Factory layout planner, basic				
textual output of machine and facility			✓	Text
creation and removal of machines			✓	Buttons
Event driven production control, basic				
start new product order			✓	Button
start/stop production simulation			✓	Button
re-order machining parts			✓	Button
Manipulator				
rotate, move (single objects)		✓		Button, Gesture
delete object		✓		Button
hide/show object groups	✓			Marking menu
Collaborator				
making annotations		✓		Button
marking areas		✓		Button
creating visual snapshot		✓		Button
show/hide highlights		✓		Button
Navigator				
rotate, pan, and zoom (whole model)	✓			Marking menu
selection of pre-defined views	✓			Marking menu

5.3 Evaluation of the Integrated System

The usability of the framework was analyzed based on a preliminary user study. Since existing frameworks do not cover the identified needs, there is no baseline for making a comparative study of IN^{2}CO to other frameworks. Instead, Nestler et al.'s approach [212] was followed to design the experiment. The approach is based on the Technology Acceptance Model [80] and common usability questionnaires. In the end, we can judge the usability and effectiveness of IN^{2}CO.

Therefore, we conduct a usability study to measure:

1. Effectiveness;
2. Acceptability.

We expect, that our framework will get highly accepted and perceived as useful. For testing these assumptions, we designed and performed a controlled experiment.

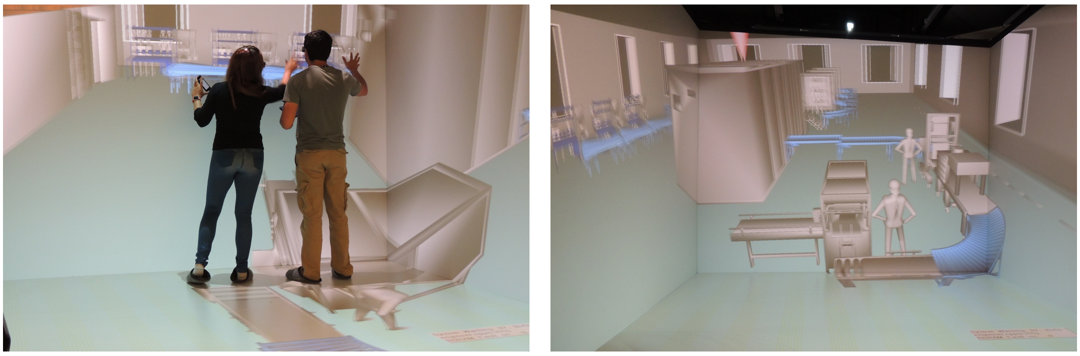
Operationalization. In order to measure the above usability types, we operationalized the two variables of interest and designated the tasks to be performed. The variables of interest were operationalized as follows:

1. Effectiveness is measured by the degree of correct solved tasks.

2. Acceptability is measured using the Technology Acceptance Model (TAM), which is a valid and reliable questionnaire for assessing technology acceptance and use [298]. We selected 40 questions out of 8 categories. All questions were rated using a five-point Likert scale (1: I strongly disagree, 5: I strongly agree).

5.3.1 Controlled Experiment

Subjects. The experiment was conducted with 12 subjects: undergraduate and graduate students in computer science and mechanical engineering from the Technical University of Kaiserslautern. The subjects, 11 males and 1 female, are between 26 and 31 years old. In average, the participants have a high experience level (3.9/5) of smartphone control and are familiar with the handling of 3D scenes (3.75/5). The number of participants satisfies the minimal number of six subjects required for a statistical significance of usability tests [124].



(A) Collaborative planning process (B) Virtual manufacturing system

FIGURE 5.21: Virtual manufacturing system in IN²CO prototype as tested in the CAVE system at the University of California, Davis

Experimental setup and data collection. The setup consists of a 3x3 high-resolution tiled wall displays and diverse smart devices ranging from smartwatches to tablet computers, similar as shown on Figure 5.23. The virtual manufacturing system's initial layout (as shown in Figure 5.21, right) had been prepared in advance and was provided on the public viewport. Participants could walk around or sit and were not restricted by any wired devices in order to provide a natural environment. The experimental procedure was performed in the following order:

1. Training the subjects. The main capabilities of the framework are introduced and subjects had to perform several tasks from the categories navigation and manipulation, as well as collaborative functionalities like highlighting areas, and insertion of annotations to become familiar with the setup.

2. Collaborative design of a factory layout. Participants were asked to rearrange objects' position, so that an additional machine could be integrated. A subject performing this task was monitored in great detail to gather information about the subject's use of the system and its supported tools.

3. Gathering user feedback. Subjects fill out the acceptance and usability questionnaire.

Data analysis. A transcript of the collected effectiveness, and acceptability data was compiled in excel. The subject data is kept anonymous and confidential. We applied descriptive statistics methods such as the sample mean, standard deviation, and median.

5.3.2 Results

Effectiveness. All participants individually and in teams have been able to perform the training tasks and the main task of designing a new factory layout collaboratively. On average, the teams needed 7.4 minutes to redesign the layout with an acceptable solution for all participants.

Acceptability. For the acceptability, the descriptive statistic mean values based on a 5 point Likert scale are calculated and listed in Table 5.4 and visualized in Figure 5.22.

TABLE 5.4: Average rating of acceptability measures per usability category

Category	Mean-Rating
(1) Individual satisfaction in team work	4.183
(2) Self-efficacy for teamwork	3.937
(3) Collectivism	3.611
(4) Decision comprehensiveness	3.083
(5) Suitable for the task	3.901
(6) Self descriptiveness	3.833
(7) Controllability	3.853
(8) Suitability for learning	4.15

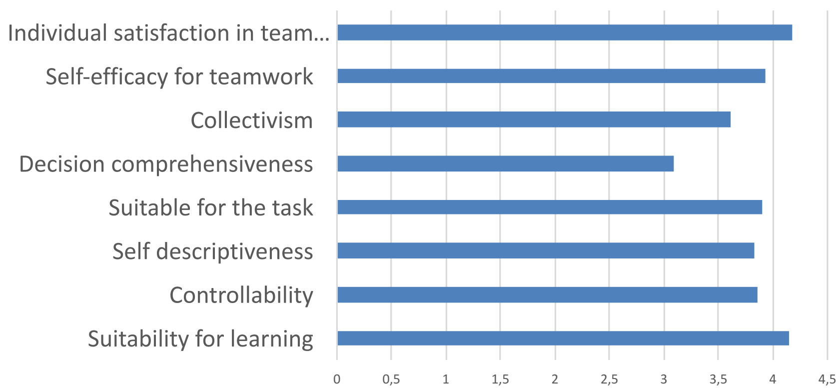


FIGURE 5.22: Average rating of all eight usability categories

All participants appreciated the additional information provided on the private viewports. Especially small objects were difficult to track exclusively on the large screen, so the smart device served as facilitating device. Additionally, some users with a solid factory layout planning background were more intensively confronted with the setup and collaboration features. The majority of the users provided encouraging feedback: On a 5 point-Likert scale, 75 % of the users estimated that the way in which the team worked together has been adequate, the way in which data has been visualized suited the task they wanted to perform, and that the setup met their requirements. Overall, all participants evaluated the prototype positively and would like to use it in the future.

Usability. Accordingly, as described in Chapter 4.4.3, a quantitative usability score is calculated. The resulting quantitative scores and weights led to a usability value $U = 0.888$ are summarized in Table 5.5:

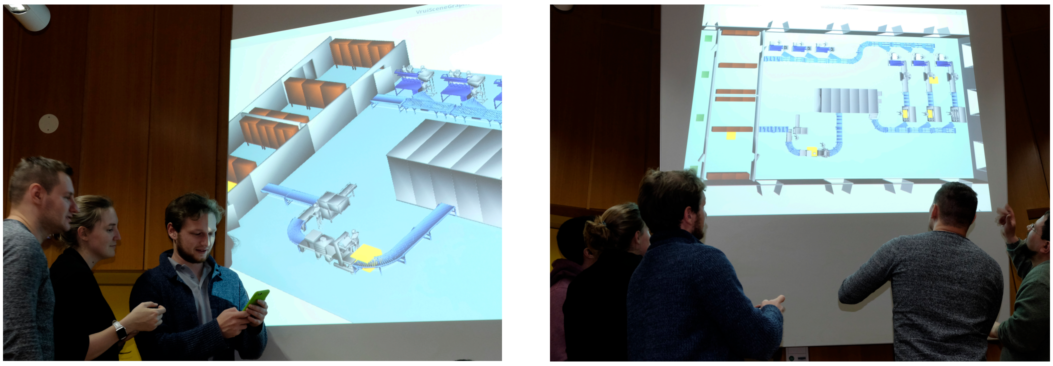
TABLE 5.5: Weights and scoring of usability categories to calculate usability score U of the system.

Category	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
$w(s)$	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
$v(s)$	1.0	1.0	0.83	0.7	0.9	0.875	0.8	1.0
$w(s) \cdot v(s)$	0.125	0.125	0.1037	0.0875	0.1125	0.1094	0.1	0.125
$\sum$	0.888							

In the following, we will discuss the results of the user study and summarize the findings.

5.4 Discussion and Summary

IN²CO was successfully applied to an sample factory-planning problem. The intuitive interaction that is provided by the smart devices allowed users to focus on the problem description, instead of concentrating on interaction issues.



(A) Exchange between all participants – collaborative group tasks solving.

(B) Sub-teams solve individual tasks under consideration of other's inferdependencies.

FIGURE 5.23: Active collaboration of all participants. Participants optimizing the factory layout and material flow visualized on public viewport by controlling the scene with diverse smart devices.

Thus, communication and decision-making based on the virtual representation of the factory could be enhanced in a team-oriented manner. The co-located teamwork is encouraged, as the provided functionalities enable planners to examine and modify the given factory layout immediately. In contrast to traditional planning tools, no privileged master-controller is defined, the participants can perform tasks in parallel which implements an equal balance of power. Hence, IN²CO empowers a creative and collaborative factory planning process, as exemplified in Figure 5.23.

In the course of a university event, called "*Die Nacht, die Wissen schafft – A Night out with Science*", the framework has been presented and opened to the public. In this annual event, the Science Alliance, an association of ten internationally renowned research facilities in Trippstadter Strasse, Kaiserslautern (PRE-Uni-Park), the University of Kaiserslautern, the University of Applied Sciences and several notable companies open their doors to all those interested in science and technology [191].

Instead of a factory layout, two mazes have been visualized on the public viewport (see Figure 5.24). The setup consisted of a simple desktop system enhanced with a projection device and four smart devices (smartphones and tablets). The smart devices were merely used as input device. Each two participants were asked to collaboratively solve the maze, by transporting a cube from the start node to the end node, in competition with another team of two. Within a team, one participant was responsible to move the cube vertically (up and down) while the other participant was responsible to move the cube horizontally (left and right).

FIGURE 5.24: Maze application in IN²CO facilitates collaboration at *A Night out with Science*

Event visitors ranging in the age from 8 years to 65 years, faced the challenge and participated in the competition. Overall, the framework was stable and easily adjustable for that application. The participants feedback was throughout positive and the visitors had noticeable fun with the collaboration framework.

The goal of this chapter was the integration and combination of all individual designed and examined visualization and interaction techniques in a proof-of concept system. First, the task ontology defined in Chapter 3.2.2 was applied onto the criteria catalog as stated in Chapter 3.3.1 in order to define the system's software architecture. The single software components and the architecture are described. Following, it has been examined how the systems attributes have been realized. Afterwards, the implementation and integration of the presented aspects into IN²CO is performed and exemplified based on manufacturing disciplines. All independently designed components, visualization- and interaction techniques as described in Chapter 4 were integrated and proved to work. Merely, the speech recognition was not completely integrated into IN²CO.

The tasks of factory layout planning and event driven production control, both ideally suited as application domain to address collaboration requirements, have been integrated to demonstrate how flexible but also easily customizable the framework is. The evaluation results have shown that the usage of smart devices is a beneficial approach to enable joint interaction with the model and also does not impair the natural interaction between users. The results demonstrated an overall usability value of 0.89 that leads to promising implications for the real application in industry, education, and research.

Summarizing, the requirements stated in Chapter 3.3.1 are checked for application in the collaboration framework.

Content support

- ✓ **[R1]** Content integration by each user
- ✓ **[R2]** Move functionalities by everyone
- ✓ **[R3]** Judge through visual indication and inferdependency awareness

Information sharing

- ✓ **[R4]** Quickly retrieve information via diverse viewports
- ✓ **[R5]** Access to shared objects within simulation world
- ✓ **[R6]** Accentuating due diverse functionalities
- ✓ **[R7]** Track others approach by observing participants
- ✓ **[R8]** Screen sharing by presenting own private viewport and public viewport
- ✓ **[R9]** Individual and shared workspaces due public and private viewports

Coordination support

- ✓ **[R10]** Jurisdiction through Launcher UI
- ✓ **[R11]** Transformations enabled for saving and loading sessions and scenes
- ✓ **[R12]** Alert mechanisms due visual and tactical feedback
- ✓ **[R13]** Awareness Support due visualization and dependency network
- ✗ **[R14]** Community Support is not provided
- ✓ **[R15]** Team structure and size can be varying and adjusted even during a session
- ✓ **[R16]** Changing work styles due to taskviews and status/update views with watch enhancements

Communication support

- ✗ **[R17]** Discussion tool; rather natural interpersonal discussion
- ✓ **[R18]** Communication in group/individual due same physical location

- ✓ **[R19]** Encrypted Communication is not considered

Compliance support is difficult to proof in general

- ✓ **[R20]** Team debates did not happen during user studies
- ✗ **[R21]** Group process training is not a requirement of the framework
- ✓ **[R22]** Reflecting all notions/opinions is facilitated
- ✓ **[R23]** Use guidelines/restrictions in form of user guidance in UI
- ✓ **[R24]** Involving all actors is ensured
- ✗ **[R25]** Team self-managing behaviors is not a requirement of the framework

Content management is ensured due to message handlers and the ACID Database technology.

- ✓ **[R26]** Action parameter are enabled
- ✓ **[R27]** Access Control is ensured
- (✓) **[R28]** Session Persistence in prototypical state
- ✓ **[R29]** Consistency and interactivity is ensured

Usability

- (✓) **[R30]** Reliability in prototypical state
- ✓ **[R31]** Reusability of system, session, and scene
- ✓ **[R32]** Transferability of skills by observing other participants
- ✓ **[R33]** Flexible actor arrangements for input and output devices
- ✓ **[R34]** Guidance through visual indication
- ✓ **[R35]** Generalizable and ease of maintenance due modular system design

User experience

- ✓ **[R36]** Natural interpersonal interaction is ensured
- ✓ **[R37]** High user satisfaction and motivation could be recorded
- ✓ **[R38]** Intuitive and simple technology is evaluated
- (✓) **[R39]** Reduced cognitive load of actors could not proofed in an experiment
- ✓ **[R40]** No all-embracing knowledge needed as user can have diverse roles

Nearly all stated needs for efficient computer-aided decision making and collaborative work are covered within the presented framework. Thus, the efficiency of co-located real-time simultaneous collaboration for complex tasks can be enhanced, and at the same time provide intuitive and natural interaction techniques. Due to the modular and flexible setup connected via diverse network connections, distributed collaboration is also enabled to a great extent.

A check-mark in brackets indicates that these requirements could not be completely proved in the user study. Crossings in the table indicate that these requirements are not covered within the framework. Namely a discussion tool in form of, e.g., chat widgets, enabling communication support was not fully integrated. Compliance support is difficult to evaluate and cannot be necessarily formulated as a requirement of the framework. However, the framework facilitates good team climate and gives everyone the ability to participate and present their ideas and opinions.

Acknowledgements:

I would like to express my special gratitude to the students who helped me to perform the user studies, to take photographs, and to bug fix the software: Nadine Bonte, Florian Schmidt, Stefan Wiesen, Viktor Leonhardt, Felix Claus, Johannes Schultz, Jonas Zeunert, and Andreas Schneider.

Contents of this chapter have lead to the following publications:

F. Rupprecht, B. Hamann, C. Weidig, J. Aurich, A. Ebert, "IN2CO - A Visualization Framework for Intuitive Collaboration", *The Eurographics Association EuroVis 2016 Short Paper Track*, Groningen, Netherlands, 2016.

Chapter 6

Conclusions

This dissertation makes contributions in (1) the design and architecture of a collaboration framework based on multi-criteria decision making; (2) the deployment and evaluation of more natural and intuitive interaction and visualization techniques in order to support multiple decision makers; and (3) the integration of novel techniques into a single proof-of-concept system.

The topics that have been inspected in this dissertation focus on supporting the collaboration process of multidisciplinary teams through a user-centered approach. One of the major problems of collaborative systems is reasoned by the complexity of interpersonal interaction and the absence of joint collaboration by active participation. While most collaboration systems focus their attention on the functionality for multiple users, less attention is payed on the development or improvement of interaction and visualization techniques to satisfy the actual requirements of a truly collaborative system.

The main goal of this dissertation is the exploration and development of novel natural and intuitive interaction- and visualization techniques through a user-centered approach, combined in a collaboration framework which enhances collaborative decision making for co-located and distributed participants. Thus, the structure of this dissertation follows the user centered design principles, namely *(1) learn about the product, (2) learn about the user,* and *(3) bring it all together.*

Learn about the Product

First, the task of collaborative work is analyzed before a conceptual framework is deduced, integrating activities and structure of collaborative tasks. An overview and classification of collaboration styles is given and the concept of inferdependencies concerning task dependencies in complex activities is introduced. While observing the domain and background, requirements for collaborative work and decision making are identified and pointed out. Subsequently, general requirements for efficient collaboration and decision making are deployed, elucidated, and transferred into six capacities of collaborative work.

The enhanced House of Quality is presented as a methodology to illuminate collaborative computer-aided decision making. Based on collected user needs, this new approach defines a quality assurance tool used for two application cases: (1) definition and rating of software components and (2) benchmarking and ranking of software/frameworks. The traditionally HoQ matrix is enhanced with four additional components: (1) a task ontology that serves to translate user needs into the design attributes in a structured way, (2) system attributes' importance rate based on multi-attribute utility theory considering correlations in between, (3) a utility function for precise benchmarking; and (4) a clear ranking between alternative solutions. The proposed methodology is applied to both application examples. As a result, it could be proved, that (software) designers can easily identify which components/-modules/system attributes have to be developed and with which priorities, in order to enhance user satisfaction, quality of the product, and improvements of the design process. Secondly, it is shown that the combination of eHoQ and MAUT results in a more accurate evaluation of alternative collaboration solutions, making (software) evaluation more precise and reliable. Summarizing, those investigations answers the question of how to design and evaluate a product that enables and supports collaborative computer-aided decision making.

Learn about the User

In order to understand the impact between diverse user groups, a clearly defined dependency graph of users in industrial corporations was designed and demonstrated. Applied on an example, clearly defined entities, and relationships of a user network are defined and meaningful relation are displayed with the use of introduced dependency parameters. State-of-the art visualization techniques are integrated in a simple data exploration framework in order to make complex system architecture easily understandable. According to the identified attributes, two dedicated user groups and associated activities with a high level of detail have been modeled and explored. These user-models serve as example user groups for the development of novel and intuitive interaction and visualization techniques for computer-aided collaboration and decision making.

A general-purpose framework that supports efficient collaborative work executed simultaneously by distributed or co-located teams using diverse mobile devices (smartphones, tablets, and smart watches) is presented. The devices provide three views of the data to be processed collaboratively: (1) a simulation view; (2) a status report view; and (3) a status update view. These views serve the purpose of providing overview, detail, and performance views. This approach goes beyond the known characteristics of existing "Overview-plus-Detail" techniques. The watch analogy provides a user with the information explaining the impact of, for example, user-induced changes made to a production process in a natural and intuitive manner. As is, simultaneous monitoring of ongoing high and low level processes is provided and the awareness of other's impacts is improved.

A signal processing approach for enhanced multi-modal interaction interfaces was presented, designed for smartwatches and smartphones for fully immersive environments. By replacing common touch input gestures with actual body movement gestures, realizable with smart watches, comparable common VR input devices are outperformed. Thus, enhancing the efficiency of interaction in virtual worlds in a natural and intuitive way. Advantages of the framework for highly effective and intuitive gesture-based interaction could be proved. The challenge of designing accurate body gesture recognizer is described by missing sensor data, precisely gyroscope and magnetometer data, which together with acceleration is used to calculate orientation and motion dynamics of the device. A transformation of the given signal-processing from the smartwatch into arm movement gestures with the use of smoothing algorithms and gesture state machines is deployed. The system is capable of recognizing unique, primitive, and even complex gestures in an easy learnable way, while overcoming the missing sensor data in low budget technology. Thus, precise arm movement gestures can be designed while allowing more diverse gestures simultaneously.

Speech, as the most natural interaction technique, combined with touch input, haptic, and arm gestures enhances the interaction capabilities of a smartwatch, thus making the smartwatch extraordinarily versatile. We could prove that multi modal interaction realized with the use of a single smartwatch is suitable for exploration tasks for large display interaction while leading to novel and more natural ways of interaction. Summarizing, the investigations answer the question of how to design dedicated interaction and visualization techniques in order to support single users as part of a collaboration team.

Bring it all together

Finally, the integration and combination of all individual designed and examined visualization and interaction techniques in one proof-of concept system is established. Based on the identified criteria of efficient collaboration and decision making, the system's software architecture is defined. All presented components and techniques are designed and implemented, for an example based on manufacturing disciplines, and integrated into a prototype system, called IN^{2}CO. IN^{2}CO is a human-centric visualization framework for intuitive and collaborative data exploration and manipulation. Specifically, its contribution is the integration of ubiquitous technologies and existing techniques to explore data and dependencies in collaborative decision-making for co-located and distributed participants. Real-time simultaneous multi-user interaction in order to support active collaboration for complex tasks as decision making and design in virtual reality is enabled. Exemplifying, the tasks of factory layout planning and event driven production control, both ideally suited as application domain to address collaboration requirements, have been integrated to demonstrate how powerful but also easily customizable the framework is.

Summarizing, the requirements defined in the general criteria catalog are checked for application in the collaboration framework. Nearly all stated needs for efficient computer-aided decision making and collaborative work are covered within the presented framework. Thus, the efficiency of co-located real-time simultaneous collaboration for complex tasks can be enhanced while at the same time providing intuitive and natural interaction techniques.

Summary

Collaboration is the combination and exchange of different core competencies and expertise, with the goal of creating a joint outcome in agreement, considering ideas and objectives of all participants. Thus, collaboration leads to comprehensive decisions.

Motivation for innovative use of VR technology is founded in the fact that disputes in practical realization are easily overcome and potential benefits are premature identifiable without the common challenges of the actual implementation: high costs, implementation risks (including risk of injury or death), inflexibility to adapt alternate scenarios, and difficulty to replicate. Not only in the industrial context, but there especially endangering, those challenges can be immense and can lead to the failure of the corporation. Virtual technology elaborates comprehensive planning steps before the actual construction or reconstruction of new manufacturing facilities. Virtual reality technology applied for validation and prototyping enables a user to step into the virtual production system and experience it as if it already exists. Thus, users can observe the production system from inside and also actively interact with it. Errors or possible problems in the planned system are discovered early in the development stage and a safe way of testing the integration of new technology and software is enabled with an unlimited number of prototype alternatives while excluding the risk of physical destruction. Thanks to virtual prototyping and corresponding simulation results the need for costly mockups are eliminated and engineering analysis becomes more efficient.

Due to increasing market competition and labour market change, companies must be highly efficient and innovative to remain competitive. The ability to learn, collaborate and solve problems in a digital information environment has become necessary. As is, computer-aided collaboration and decision making in Virtual Reality allowing intuitive and natural interaction, as proposed in this dissertation, leads to comprehensive decisions in an efficient way, while preventing implementation risks and saving money and time.

Bibliography

[1] AB - Unified Intents, *Unified remote: Remote control app for your computer*, 2017. [Online]. Available: https://www.unifiedremote.com/.

[2] Y. Akao and G. H. Mazur, "The leading edge in qfd: Past, present and future", *International Journal of Quality & Reliability Management*, vol. 20, no. 1, pp. 20–35, 2003.

[3] A. Aldandarawy, *Unity3d-globe*, (Accessed on 12/31/2017), 2017. [Online]. Available: https://github.com/Dandarawy/Unity3D-Globe/blob/master/README.md.

[4] E. Alepis, M. Virvou, and S. Drakoulis, "Human smartphone interaction: Exploring smartphone senses", in *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, 2014, pp. 44–48. DOI: 10.1109/IISA.2014.6878825.

[5] A. E. Ali, *Multimodal interaction*, 2011. [Online]. Available: https://www.slideshare.net/Abd0/multimodal-interaction-an-introduction.

[6] N. Almeida, S. Silva, and A. Teixeira, "Design and development of speech interaction: A methodology", in *International Conference on Human-Computer Interaction*, Springer, 2014, pp. 370–381.

[7] C. Anslow, S. Marshall, J. Noble, and R. Biddle, "SourceVis: Collaborative software visualization for co-located environments", *2013 1st IEEE Working Conference on Software Visualization - Proceedings of VISSOFT 2013*, 2013.

[8] Apple Inc., *Apple watch (1st generation) - technical specifications*, (Accessed on 04/07/2017), 2017. [Online]. Available: https://support.apple.com/kb/SP735?locale=en_US.

[9] Apple Inc., *Ipad 2 - technical specifications*, (Accessed on 04/18/2017), 2017. [Online]. Available: https://support.apple.com/kb/SP622?viewlocale=en_US&locale=de_DE.

[10] Apple Inc., *Ipad mini - technical specifications*, (Accessed on 04/18/2017), 2017. [Online]. Available: https://support.apple.com/kb/SP661?viewlocale=en_US&locale=de_DE.

[11] Apple Inc., *Iphone 5 - technical specifications*, (Accessed on 04/07/2017), 2017. [Online]. Available: https://support.apple.com/kb/SP655?viewlocale=en_US&locale=de_DE.

[12] Apple Inc., *Iphone 6 plus - technical specifications*, (Accessed on 04/18/2017), 2017. [Online]. Available: https://support.apple.com/kb/SP706?viewlocale=en_US&locale=de_DE.

[13] Apple Inc., *Iphone se – apple (de)*, (Accessed on 01/09/2018), 2018. [Online]. Available: https://www.apple.com/de/iphone-se/.

[14] Apple Inc., *Objective-c | apple developer documentation*, (Accessed on 01/22/2018), 2018. [Online]. Available: https://developer.apple.com/documentation/objectivec.

[15] S. S. Arefin Shimon, C. Lutton, Z. Xu, S. Morrison-Smith, C. Boucher, and J. Ruiz, "Exploring non-touchscreen gestures for smartwatches", in *Proceedings*

of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, 2016, pp. 3822–3833.
[16] Ascensio System SIA, *Onlyoffice - online document editors, crm, projects, mail*, 2016. [Online]. Available: `https://www.onlyoffice.com/features.aspx` (visited on 11/07/2016).
[17] S. K. Badam, E. Fisher, and N. Elmqvist, "Munin: A peer-to-peer middleware for ubiquitous analytics and visualization spaces", *IEEE transactions on visualization and computer graphics*, vol. 21, no. 2, pp. 215–228, 2015.
[18] R. Ball, C. North, and D. A. Bowman, "Move to improve: Promoting physical navigation to increase user performance with large displays", in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2007, pp. 191–200.
[19] S. Bandini, F. De Paoli, S. Manzoni, and P. Mereghetti, "A support system to cots-based software development for business services", in *Proceedings of the 14th international conference on Software engineering and knowledge engineering*, ACM, 2002, pp. 307–314.
[20] M. S. Bandor, "Quantitative methods for software selection and evaluation", DTIC Document, Tech. Rep., 2006.
[21] W. D. Barnett and M. Raja, "Application of qfd to the software development process", *International Journal of Quality & Reliability Management*, vol. 12, no. 6, pp. 24–42, 1995.
[22] V. R. Basili and H. D. Rombach, "Tailoring the software process to project goals and environments", in *Proceedings of the 9th international conference on Software Engineering*, IEEE Computer Society Press, 1987, pp. 345–357.
[23] J. Bathelt and J. Meile, "Computer aided methods supporting concurrent engineering when designing mechatronic systems controlled by a plc", in *Proceedings of International Conference on Manufacturing Automation (ICMA07), Singapore*, 2007.
[24] J. Bauer, A. Ebert, O. Kreylos, and B. Hamann, "Marking menus for eyes-free interaction using smart phones and tablets", in *International Conference on Availability, Reliability, and Security*, Springer, 2013, pp. 481–494.
[25] T. Bauernhansl, M. Ten Hompel, and B. Vogel-Heuser, *Industrie 4.0 in produktion, automatisierung und logistik: Anwendung· technologien· migration*. Springer-Verlag, 2014.
[26] S. Baumgärtner, A. Ebert, M. Deller, and S. Agne, "2d meets 3d: A human-centered interface for visual data exploration", in *CHI'07 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2007, pp. 2273–2278.
[27] M. Beaudouin-Lafon and A. Karsenty, "Transparency and awareness in a real-time groupware system", in *Proceedings of the 5th annual ACM symposium on User interface software and technology*, ACM, 1992, pp. 171–180.
[28] G. S. Becker, "Nobel lecture: The economic way of looking at behavior", *Journal of political economy*, vol. 101, no. 3, pp. 385–409, 1993.
[29] B. Beckert, D. Buschak, B. G. IPA, and M. Hägele, "Automatisierung und robotik-systeme", *Studien zum deutschen Innovationssystem Nr. 11-2016*, 2016.
[30] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer, "Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees", *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1693–1702, 2014.
[31] L.-P. Bergé, M. Serrano, G. Perelman, and E. Dubois, "Exploring smartphone-based interaction with overview+ detail interfaces on 3d public displays",

in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, ACM, 2014, pp. 125–134.

[32] N. O. Bernsen, "Towards a tool for predicting speech functionality", *Speech Communication*, vol. 23, no. 3, pp. 181–210, 1997.

[33] S. Berti and F. Paternò, "Migratory multimodal interfaces in multidevice environments", in *Proceedings of the 7th international conference on Multimodal interfaces*, ACM, 2005, pp. 92–99.

[34] M. Betts, "Qfd integrated with software engineering", in *Transactions from the Second Symposium on Quality Function Deployment*, 1990, pp. 18–19.

[35] R. Beuscart, C. Grave, M. Wartki, S. Serbouti, and M. Beuscart-Zephir, "Computer supported cooperative work for medicine imaging", in *Engineering in Medicine and Biology Society, 1992 14th Annual International Conference of the IEEE*, IEEE, vol. 3, 1992, pp. 1213–1214.

[36] S. Bhandari and Y.-K. Lim, "Exploring gestural mode of interaction with mobile phones", in *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2008, pp. 2979–2984.

[37] J. P. Birnholtz, T. Grossman, C. Mak, and R. Balakrishnan, "An exploratory study of input configuration and group process in a negotiation task using a large display", in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 2007, pp. 91–100.

[38] M. Blumendorf, D. Roscher, and S. Albayrak, "Dynamic user interface distribution for flexible multimodal interaction", in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, ACM, 2010, p. 20.

[39] R. Bolt, "Put-that-there: Voice and gesture at the graphics interface", in *ACM Computer Graphics 14, 3*, 1980, pp. 262–270.

[40] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*. Citeseer, 1976, vol. 290.

[41] C. C. Bonwell and T. E. Sutherland, "The active learning continuum: Choosing activities to engage students in the classroom", *New directions for teaching and learning*, vol. 1996, no. 67, pp. 3–16, 1996.

[42] J. Borchers, M. Ringel, J. Tyler, and A. Fox, "Stanford interactive workspaces: A framework for physical and graphical user interface prototyping", *Wireless Commun.*, vol. 9, no. 6, pp. 64–69, Dec. 2002, ISSN: 1536-1284. DOI: `10.1109/MWC.2002.1160083`. [Online]. Available: `http://dx.doi.org/10.1109/MWC.2002.1160083`.

[43] U. M. Borghoff and J. H. Schlichter, "Computer-supported cooperative work", in *Computer-Supported Cooperative Work*, Springer, 2000, pp. 87–141.

[44] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications", *Eurographics State of the Art Reports*, pp. 39–63, 2013.

[45] M. Bostock, "D3.js", *Data Driven Documents*, vol. 492, p. 701, 2012.

[46] C. E. Bozdağ, C. Kahraman, and D. Ruan, "Fuzzy group decision making for selection among computer integrated manufacturing systems", *Computers in Industry*, vol. 51, no. 1, pp. 13–29, 2003.

[47] S. E. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. S. Warren, and A. Kaufman, "Toward a multi-analyst, collaborative framework for visual analytics", in *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, IEEE, 2006, pp. 129–136.

[48] W. Bricken, "Virtual reality: Directions of growth notes from the siggraph'90 panel", *Virtual Reality: Directions of Growth*, 1990.

[49] R. O. Briggs, G.-J. De Vreede, and J. F. Nunamaker Jr, "Collaboration engineering with thinklets to pursue sustained success with group support systems", *Journal of Management Information Systems*, vol. 19, no. 4, pp. 31–64, 2003.

[50] R. O. Briggs, G. L. Kolfschoten, G.-J. De Vreede, C. C. Albrecht, and S. G. Lukosch, "Facilitator in a box: Computer assisted collaboration engineering and process support systems for rapid development of collaborative applications for high-value tasks", in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, IEEE, 2010, pp. 1–10.

[51] P. J. Brown, "Process-oriented guided-inquiry learning in an introductory anatomy and physiology course with a diverse student population", *Advances in physiology education*, vol. 34, no. 3, pp. 150–155, 2010.

[52] S. Bryson, S. K. Feiner, F. P. Brooks Jr, P. Hubbard, R. Pausch, and A. van Dam, "Research frontiers in virtual reality", in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, ACM, 1994, pp. 473–474.

[53] H.-J. Bullinger, D. Spath, H.-J. Warnecke, and E. Westkämper, *Handbuch unternehmensorganisation: Strategien, planung, umsetzung*. Springer-Verlag, 2009.

[54] L. Bunch, M. Breedy, J. Bradshaw, M. Carvalho, D. Danks, and N. Suri, "Flexible automated monitoring and notification for complex processes using karmen", in *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.*, IEEE, 2005, pp. 443–448.

[55] J. Burdett and B. Hastie, "Predicting satisfaction with group work assignments.", *Journal of University Teaching and Learning Practice*, vol. 6, no. 1, pp. 61–71, 2009.

[56] S. Burigat and L. Chittaro, "Visualizing references to off-screen content on mobile devices: A comparison of arrows, wedge, and overview+ detail", *Interacting with Computers*, vol. 23, no. 2, pp. 156–166, 2011.

[57] S. Burigat and L. Chittaro, "On the effectiveness of overview+ detail visualization on mobile devices", *Personal and ubiquitous computing*, vol. 17, no. 2, pp. 371–385, 2013.

[58] T. Büring, J. Gerken, and H. Reiterer, "Usability of overview-supported zooming on small screens with regard to individual differences in spatial ability", in *Proceedings of the working conference on Advanced visual interfaces*, ACM, 2006, pp. 233–240.

[59] Carnegie Mellon University, *The pittsburgh pebbles pda project*, 2009. [Online]. Available: `http://www.pebbles.hcii.cmu.edu`.

[60] S. Carrino, A. Péclat, E. Mugellini, and O. A. Khaled, "Humans and smart environments: A novel multimodal interaction approach", in *ICMI '11 Proceedings of the 13th international conference on multimodal interfaces*, Alicante, Spain, 2011, pp. 105–112.

[61] E. Cavalcante, L. Rivero, and T. Conte, "Evaluating the feasibility of max: A method using cards and a board for assessing the post-use ux", *International Journal of Software Engineering and Knowledge Engineering*, vol. 25, no. 09n10, pp. 1759–1764, 2015.

[62] L.-K. Chan and M.-L. Wu, "Quality function deployment: A literature review", *European journal of operational research*, vol. 143, no. 3, pp. 463–497, 2002.

[63] R. Chebil, W. L. Chaari, S. A. Cerri, and K. Ghedira, “A causal graph based method to evaluate e-collaboration scenarios”, in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd International Workshop on*, IEEE, 2013, pp. 225–230.

[64] X. Chen, M. Wang, and Q. Wu, “Research and development of virtual reality game based on unreal engine 4”, in *2017 4th International Conference on Systems and Informatics (ICSAI)*, 2017, pp. 1388–1393. DOI: 10.1109/ICSAI.2017.8248503.

[65] X. Chen, T. Grossman, D. J. Wigdor, and G. Fitzmaurice, “Duet: Exploring joint interactions on a smart phone and a smart watch”, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2014, pp. 159–168.

[66] X. Cheng, Y. Li, J. Sun, and X. Zhu, “Easy collaboration process support system design for student collaborative group work: A case study”, in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, IEEE, 2014, pp. 453–462.

[67] S. K. Chin and A. W. Yeo, “Computer supported cooperative work (cscw) in orthography system development”, in *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*, IEEE, 2010, pp. 579–583.

[68] L. Chittaro, “Visualizing information on mobile devices”, *Computer*, vol. 39, no. 3, pp. 40–45, 2006.

[69] A. Christopoulos, M. Conrad, and M. Shukla, “How do students' really'interact with virtual worlds?-the influence of proper induction for virtual interactions.”, in *CSEDU (1)*, 2016, pp. 43–54.

[70] H.-C. Chu, S.-C. Huang, and J.-J. Liaw, “An acceleration feature-based gesture recognition system”, in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2013, pp. 3807–3812.

[71] CMUSphinx Group, *Cmusphinx open source speech recognition*, (Accessed on 01/30/2018), 2018. [Online]. Available: https://cmusphinx.github.io/.

[72] L. Cohen, “Quality function deployment: An application perspective from digital equipment corporation”, *Global Business and Organizational Excellence*, vol. 7, no. 3, pp. 197–208, 1988.

[73] K. Collier, B. Carey, D. Sautter, and C. Marjaniemi, “A methodology for evaluating and selecting data mining software”, in *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, vol. Track6, 1999, 11 pp.–.

[74] E. Colombo and C. Francalanci, “Selecting crm packages based on architectural, functional, and cost requirements: Empirical validation of a hierarchical ranking model”, *Requirements Engineering*, vol. 9, no. 3, pp. 186–203, 2004.

[75] S. Comella-Dorda, J. C. Dean, E. Morris, and P. Oberndorf, “A process for cots software product evaluation”, in *ICCBSS*, Springer, vol. 2, 2002, pp. 86–96.

[76] B. Councill and G. T. Heineman, “Definition of a software component and its elements”, *Component-based software engineering: Putting the pieces together*, pp. 5–19, 2001.

[77] C. Courage and K. Baxter, *Understanding your users: A practical guide to user requirements methods, tools, and techniques*. Gulf Professional Publishing, 2005.

[78] M. A. Cusumano, *Japan's software factories: A challenge to us management*. Oxford University Press, 1991.

[79] M. Dahl, A. Albo, J. Eriksson, J. Pettersson, and P. Falkman, “Virtual reality commissioning in production systems preparation”, in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–7. DOI: 10.1109/ETFA.2017.8247581.

[80] L. Davis and G. Williams, "Evaluating and selecting simulation software using the analytic hierarchy process", *Integrated manufacturing systems*, vol. 5, no. 1, pp. 23–32, 1994.

[81] S. De Freitas, G. Rebolledo-Mendez, F. Liarokapis, G. Magoulas, and A. Poulovassilis, "Learning as immersive experiences: Using the four-dimensional framework for designing and evaluating immersive learning experiences in a virtual world", *British Journal of Educational Technology*, vol. 41, no. 1, pp. 69–85, 2010.

[82] C. Deng, H. Li, and Y. Shao, "Research on the drawing method of energy sankey diagram based on java", in *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, IEEE, 2014, pp. 88–91.

[83] M. Dideles, "Bluetooth: A technical overview", *Crossroads*, vol. 9, no. 4, pp. 11–18, Jun. 2003, ISSN: 1528-4972. DOI: 10.1145/904080.904083. [Online]. Available: http://doi.acm.org/10.1145/904080.904083.

[84] H. Donker and M. Blumberg, "Computer support of team work on mobile devices", in *International Conference on Human-Computer Interaction*, Springer, 2011, pp. 38–47.

[85] D. Doroftei, G. De Cubber, R. Wagemans, A. Matos, E. Silva, V. Lobo, G. Cardoso, K. Chintamani, S. Govindaraj, J. Gancet, *et al.*, "User-centered design", in *Search and Rescue Robotics-From Theory to Practice*, InTech, 2017.

[86] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces", in *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, ACM, 1992, pp. 107–114.

[87] G. M. Draper, Y. Livnat, and R. F. Riesenfeld, "A survey of radial methods for information visualization", *IEEE transactions on visualization and computer graphics*, vol. 15, no. 5, pp. 759–776, 2009.

[88] R. Drath, P. Weber, and N. Mauser, "An evolutionary approach for the industrial introduction of virtual commissioning", in *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, IEEE, 2008, pp. 5–8.

[89] I. Duncan, A. Miller, and S. Jiang, "A taxonomy of virtual worlds usage in education", *British Journal of Educational Technology*, vol. 43, no. 6, pp. 949–964, 2012.

[90] M. D. Dunlop, A. Komninos, and N. Durga, "Towards high quality text entry on smartwatches", in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2014, pp. 2365–2370.

[91] J. S. Dyer and R. F. Miles Jr, "An actual application of collective choice theory to the selection of trajectories for the mariner jupiter/saturn 1977 project", *Operations Research*, vol. 24, no. 2, pp. 220–244, 1976.

[92] A. Ebert, S. Thelen, P.-S. Olech, J. Meyer, and H. Hagen, "Tiled++: An enhanced tiled hi-res display wall", *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 1, pp. 120–132, 2010.

[93] S. Ecma, "404: The json data interchange format", *Ecma-international*, vol. 9, 2015.

[94] D. Egorov and A. Bezgodov, "Improved force-directed method of graph layout generation with adaptive step length", *Procedia Computer Science*, vol. 66, pp. 689 –696, 2015, 4th International Young Scientist Conference on Computational Science, ISSN: 1877-0509. DOI: http://dx.doi.org/10.1016/j.procs.2015.11.078. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915034274.

[95] K. El Emam, N. Moukheiber, and N. H. Madhavji, "An empirical evaluation of the g/q/m method", in *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: Software engineering-Volume 1*, IBM Press, 1993, pp. 265–289.

[96] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems", in *Acm Sigmod Record*, ACM, vol. 18, 1989, pp. 399–407.

[97] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: Some issues and experiences", *Communications of the ACM*, vol. 34, no. 1, pp. 39–58, 1991.

[98] S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, *et al.*, "Liveboard: A large interactive display supporting group meetings, presentations, and remote collaboration", in *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 1992, pp. 599–607.

[99] ERP – Enterprise Resource Planning, *Nintendo wii specifications*, (Accessed on 01/24/2018), 2018. [Online]. Available: `http://www.tech-faq.com/nintendo-wii-specifications.html`.

[100] M. Finke, N. Kaviani, I. Wang, V. Tsao, S. Fels, and R. Lea, "Investigating distributed user interfaces across interactive large displays and mobile devices", in *Proceedings of the International Conference on Advanced Visual Interfaces*, ser. AVI '10, Roma, Italy: ACM, 2010, pp. 413–413, ISBN: 978-1-4503-0076-6. DOI: `10.1145/1842993.1843091`.

[101] J. D. Foley, V. L. Wallace, and P. Chan, "The human factors of computer graphics interaction techniques", *IEEE computer Graphics and Applications*, vol. 4, no. 11, pp. 13–48, 1984.

[102] B. A. Forouzan, *Tcp/ip protocol suite*, 2nd ed. New York, NY, USA: McGraw-Hill, Inc., 2002, ISBN: 0071199624.

[103] A. Forrester, A. Keane, *et al.*, *Engineering design via surrogate modelling: A practical guide*. John Wiley & Sons, 2008.

[104] E. Freeman, S. Brewster, and V. Lantz, "Do that, there: An interaction technique for addressing in-air gesture systems", in *Proceedings of the 34th Annual ACM Conference on Human Factors in Computing Systems-CHI'16*, ACM Press, 2016.

[105] R. B. Freeman, "The labour market in the new information economy", *Oxford Review of Economic Policy*, vol. 18, no. 3, pp. 288–305, 2002.

[106] D. J. French, B. Stone, T. T. Nysetvold, A. Hepworth, and W. E. Red, "Collaborative design principles from minecraft with applications to multi-user cad", in *ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2014.

[107] K. A. French and J. L. Kottke, "Teamwork satisfaction: Exploring the multilevel interaction of teamwork interest and group extraversion", *Active Learning in Higher Education*, vol. 14, no. 3, pp. 189–200, 2013.

[108] T. M. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement", *Software: Practice and experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[109] M. Funk, A. Sahami, N. Henze, and A. Schmidt, "Using a touch-sensitive wristband for text entry on smart watches", in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, ACM, 2014, pp. 2305–2310.

[110] Google Developers, *Google cloud speech api documentation*, (Accessed on 01/30/2018), 2018. [Online]. Available: `https://cloud.google.com/speech/docs/?hl=en`.

[111] Google Inc., *Google docs - create and edit documents online*, 2016. [Online]. Available: `https://www.google.com/intl/en/docs/about/` (visited on 11/07/2016).

[112] Google Inc., *Sensor manager*, 2017. [Online]. Available: `https://developer.android.com/reference/android/hardware/SensorManager.html`.

[113] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design", *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2051–2055, 2010.

[114] J. D. Gould and C. Lewis, "Designing for usability: Key principles and what designers think", *Communications of the ACM*, vol. 28, no. 3, pp. 300–311, 1985.

[115] G. Grau, J. P. Carvallo, X. Franch, and C. Quer, "Descots: A software system for selecting cots components", in *Euromicro Conference, 2004. Proceedings. 30th*, IEEE, 2004, pp. 118–126.

[116] P. Griffin, E. Care, and B. McGaw, "The changing role of education and schools", in *Assessment and teaching of 21st century skills*, Springer, 2012, pp. 1–15.

[117] F. Grivokostopoulou, I. Perikos, K. Kovas, M. Paraskevas, and I. Hatzilygeroudis, "Utilizing virtual reality to assist students in learning physics", in *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2017, pp. 486–489. DOI: `10.1109/TALE.2017.8252385`.

[118] J. Grudin and S. E. Poltrock, "Computer-supported cooperative work and groupware", *Advances in computers*, vol. 45, pp. 269–320, 1997.

[119] A. Guo and T. Paek, "Exploring tilt for no-touch, wrist-only interactions on smartwatches", in *13th International Conference on Human-Computer Interaction*, Florence, Italy, 2016, pp. 17–28.

[120] C. H. Han, J. K. Kim, S. H. Choi, and S. H. Kim, "Determination of information system development priority using quality function development", *Computers & industrial engineering*, vol. 35, no. 1, pp. 241–244, 1998.

[121] D. Harwood and A. Farrow, "Validation of reported events from a vertical chute marine evacuation study", *Applied ergonomics*, vol. 43, no. 6, pp. 1105–1109, 2012.

[122] J. R. Hauser and D. Clausing, "The house of quality", *Harvard business review*, vol. 66, no. 3, 1988.

[123] C. C. Heckscher, *The collaborative enterprise: Managing speed and complexity in knowledge-based businesses*. Yale University Press, 2007.

[124] M. Hegner, *Methoden zur evaluation von software*. 2003, vol. 29, p. 98.

[125] V. Hlupic and A. S. Mann, "Simselect: A system for simulation software selection", in *Simulation Conference Proceedings, 1995. Winter*, IEEE, 1995, pp. 720–727.

[126] P. Hoffmann, R. Schumann, T. M. Maksoud, and G. C. Premier, "Virtual commissioning of manufacturing systems a review and new approaches for simplification.", in *ECMS*, Kuala Lumpur, Malaysia, 2010, pp. 175–181.

[127] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data", *IEEE Transactions on visualization and computer graphics*, vol. 12, no. 5, pp. 741–748, 2006.

[128] D. Holten, Van, and J. W. Jarke, "Force-directed edge bundling for graph visualization", in *Computer graphics forum*, Computer graphics forum, vol. 28, 2009, pp. 983–990.

[129] B. L. Hooey, J. C. Da Silva, and D. C. Foyle, "A design rationale capture tool to support design verification and re-use", in *Proceedings of the 4th International Conference on Applied Human Factors and Ergonomics, San Francisco*, Jul. 21, 2012.

[130] E. Hornecker, P. Marshall, N. S. Dalton, and Y. Rogers, "Collaboration and interference: Awareness with mice or touch input", in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, acm, 2008, pp. 167–176.

[131] S. Houben and N. Marquardt, "Watchconnect: A toolkit for prototyping smartwatch-centric cross-device applications", in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 2015, pp. 1247–1256.

[132] C. L. Howell, "Student perceptions of learner-centered education.", *Online Submission*, 2006.

[133] E. B. Hsu, Y. Li, J. D. Bayram, D. Levinson, S. Yang, and C. Monahan, "State of virtual reality based disaster preparedness and response training", *PLoS currents*, vol. 5, 2013.

[134] G. Hu, X. Ou, Q. Zhang, and V. J. Karplus, "Analysis on energy-water nexus by sankey diagram: The case of beijing", *Desalination and Water Treatment*, vol. 51, no. 19-21, pp. 4183–4193, 2013.

[135] J. Hua, M. L. Huang, and Q. V. Nguyen, "Drawing large weighted graphs using clustered force-directed algorithm", in *Information Visualisation (IV), 2014 18th International Conference on*, IEEE, 2014, pp. 13–17.

[136] E. M. Huang, E. D. Mynatt, D. M. Russell, and A. E. Sue, "Secrets to success and fatal flaws: The design of large-display groupware", *IEEE Computer Graphics and Applications*, vol. 26, no. 1, pp. 37–45, 2006.

[137] W. huangHuang, P. Eades, S.-H. Hong, and H. B.-L. Duh, "Effects of curves on graph perception", in *Pacific Visualization Symposium (PacificVis), 2016 IEEE*, IEEE, 2016, pp. 199–203.

[138] A. E. Hühn, V.-J. Khan, A. Lucero, and P. Ketelaar, "On the use of virtual environments for the evaluation of location-based applications", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2012, pp. 2569–2578.

[139] IEEE Computer Society, *Ieee recommended practice for software requirements specifications*, Institute of Electrical and Electronics Engineers, 1998.

[140] IEEE Standards Association and others, "802.11-2012-ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications", *IEEE Std*, vol. 802, 2012.

[141] A. S. Jadhav and R. M. Sonar, "Evaluating and selecting software packages: A review", *Information and software technology*, vol. 51, no. 3, pp. 555–563, 2009.

[142] A. Jaimes and N. Sebe, "Multimodal human–computer interaction: A survey", *Computer vision and image understanding*, vol. 108, no. 1, pp. 116–134, 2007.

[143] M. R. Jakobsen and K. Hornbæk, "Up close and personal: Collaborative work on a high-resolution multitouch wall display", *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 21, no. 2, p. 11, 2014.

[144] K. Jansson, M. Höynälänmaa, I. Karvonen, and M. Ollus, "Bridging virtual models and reality for lifecycle support in process industries", in *2007 IEEE International Technology Management Conference (ICE)*, 2007, pp. 1–8.

[145] Jet Cat Games Interactive Entertainment, UAB, *Heliborne – helicopter combat game*, (Accessed on 01/18/2018), 2018. [Online]. Available: `http://www.jetcatgames.com/`.

[146] H. Jodlbauer, *Produktionsoptimierung*. Springer Science & Business, 2008.

[147] R. Johansen, *Groupware: Computer support for business teams*. The Free Press, 1988.

[148] M. Johnson, J. M. Bradshaw, P. J. Feltovich, R. R. Hoffman, C. Jonker, B. van Riemsdijk, and M. Sierhuis, "Beyond cooperative robotics: The central role of interdependence in coactive design", *IEEE Intelligent Systems*, vol. 26, no. 3, pp. 81–88, 2011.

[149] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, B. V. Riemsdijk, and M. Sierhuis, *The fundamental principle of coactive design: Interdependence must shape autonomy*. Springer, 2011, pp. 172–191.

[150] M. Johnson, J. M. Bradshaw, P. J. Feltovich, C. M. Jonker, M. B. Van Riemsdijk, and M. Sierhuis, "Coactive design: Designing support for interdependence in joint activity", *Journal of Human-Robot Interaction, 3 (1), 2014*, 2014.

[151] M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor, "Match: An architecture for multimodal dialogue systems", in *ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania, 2002, pp. 376–383.

[152] P. W. Jordan, *Designing pleasurable products: An introduction to the new human factors*. CRC press, 2002.

[153] M José, V. Ferrón-Vílchez, and N. Ortiz-de Mandojana, "Team decision making and individual satisfaction with the team", *Small Group Research*, vol. 45, no. 2, pp. 198–216, 2014.

[154] M. Kamalrudin and J. Grundy, "Generating essential user interface prototypes to validate requirements", in *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '11, Washington, DC, USA: IEEE Computer Society, 2011, pp. 564–567, ISBN: 978-1-4577-1638-6. DOI: 10.1109/ASE.2011.6100126. [Online]. Available: http://dx.doi.org/10.1109/ASE.2011.6100126.

[155] G. Kasakow and J. C. Aurich, "Realising digital connectivity by using interdependencies within a production process", *Procedia CIRP*, vol. 52, pp. 80–83, 2016.

[156] R. Kathuria, M. Anandarajan, and M. Igbaria, "Selecting it applications in manufacturing: A kbs approach", *Omega*, vol. 27, no. 6, pp. 605–616, 1999.

[157] D. F. Keefe, A. Gupta, D. Feldman, J. V. Carlis, S. K. Keefe, and T. J. Griffin, "Scaling up multi-touch selection and querying: Interfaces and applications for combining mobile multi-touch input with large-scale visualization displays", *International Journal of Human-Computer Studies*, vol. 70, no. 10, pp. 703–713, 2012.

[158] R. L. Keeney and H. Raiffa, *Decisions with multiple objectives: Preferences and value trade-offs*. Cambridge university press, 1993.

[159] D. A. Keim, F. Mansmann, J. Schneidewind, and H. Ziegler, "Challenges in visual data analysis", in *Information Visualization, 2006. IV 2006. Tenth International Conference on*, IEEE, 2006, pp. 9–16.

[160] T. Kersten, G. Büyüksalih, F. Tschirschwitz, T. Kan, S. Deggim, Y. Kaya, and A. Baskaraca, "The selimiye mosque of edirne, turkey-an immersive and interactive virtual reality experience using htc vive.", *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 42, 2017.

[161] K. Kessi, M. Oussalah, and Z. Alimazighi, "Requirement engineering of a cooperative information system using viewpoints", in *Advanced Information Systems for Enterprises (IWAISE), 2014 International Workshop on*, IEEE, 2014, pp. 51–57.

[162] C. S. Kim and Y. Yoon, "Selection of a good expert system shell for instructional purposes in business", *Information Management*, vol. 23, no. 5, pp. 249 –262, 1992, ISSN: 0378-7206. DOI: http://dx.doi.org/10.1016/0378-7206(92)

90056-L. [Online]. Available: http://www.sciencedirect.com/science/article/pii/037872069290056L.
[163] C. W. Kirkwood, “A case history of nuclear power plant site selection”, *Journal of the operational research society*, pp. 353–363, 1982.
[164] B. Kirsch, *Irtg 2057 graduate program*, 2014. [Online]. Available: http://www.irtg2057.de/ (visited on 02/13/2017).
[165] K. Kiyokawa, “Human communication in collaborative augmented reality systems”, in *Emerging Technologies of Augmented Reality: Interfaces and Design*, IGI Global, 2007, pp. 236–260.
[166] G. Klein, P. J. Feltovich, J. M. Bradshaw, and D. D. Woods, “Common ground and coordination in joint activity”, *Organizational simulation*, vol. 53, 2005.
[167] J. Kletti, *Mes - manufacturing execution system*. Springer Berlin Heidelberg, 2015.
[168] J. Kletti and J. Schumacher, *Die perfekte produktion: Manufacturing excellence durch short interval technology (sit)*. Springer-Verlag, 2011.
[169] W. A. König, R. Rädle, and H. Reiterer, *Squidy: A zoomable design environment for natural user interfaces*. ACM, 2009.
[170] G. Kotonya and I. Sommerville, “Requirements engineering with viewpoints”, *Software Engineering Journal*, vol. 11, no. 1, pp. 5–18, 1996.
[171] O. Kreylos, *Vrui vr toolkit*, 2009. [Online]. Available: http://idav.ucdavis.edu/~okreylos/ResDev/Vrui/.
[172] M. Krone, C. Müller, and T. Ertl, “Remote rendering and user interaction on mobile devices for scientific visualization”, in *Proceedings of the 8th International Symposium on Visual Information Communication and Interaction*, ACM, 2015, pp. 21–26.
[173] K. Küpfmüller, W. Mathis, and A. Reibiger, *Theoretische elektrotechnik: Eine einführung*. Springer-Verlag, 2013.
[174] D. Lanter and R. Essinger, “User-centered design”, in *International Encyclopedia of Geography: People, the Earth, Environment and Technology*. John Wiley & Sons, Ltd, 2016, ISBN: 9781118786352. DOI: 10.1002/9781118786352.wbieg0432. [Online]. Available: http://dx.doi.org/10.1002/9781118786352.wbieg0432.
[175] L. A. Le Blanc and M. T. Jelassi, “Dss software selection: A multiple criteria decision methodology”, *Information & Management*, vol. 17, no. 1, pp. 49–65, 1989.
[176] C. G. Lee and S. C. Park, “Survey on the virtual commissioning of manufacturing systems”, *Journal of Computational Design and Engineering*, vol. 1, no. 3, pp. 213–222, 2014.
[177] H.-S. Lee, P.-D. Shen, and W.-L. Chih, “A fuzzy multiple criteria decision making model for software selection”, in *IEEE International Conference on Fuzzy Systems*, 2004.
[178] S. Lee, N. Ezer, J. Sanford, and E. Y.-L. Do, “Designing together while apart: The role of computer-mediated communication and collaborative virtual environments on design collaboration”, in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, IEEE, 2009, pp. 3593–3598.
[179] P. A. Legg, D. H. Chung, M. L. Parry, M. W. Jones, R. Long, I. W. Griffiths, and M. Chen, “Matchpad: Interactive glyph-based visualization for real-time sports performance analysis”, in *Computer Graphics Forum*, Wiley Online Library, vol. 31, 2012, pp. 1255–1264.
[180] S. W. Lester, B. M. Meglino, and M. A. Korsgaard, “The antecedents and consequences of group potency: A longitudinal investigation of newly formed work groups”, *Academy of Management Journal*, vol. 45, no. 2, pp. 352–368, 2002.

[181] J. J. Li, T. Li, Z. Lin, A. Mathur, and K. Kanoun, "Computer supported cooperative work in software engineering", in *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International*, IEEE, 2004, 328–vol.

[182] A. Lieberman, "Collaborative work", *Educational Leadership*, vol. 43, no. 5, pp. 4–8, 1986.

[183] H.-Y. Lin, P.-Y. Hsu, and G.-J. Sheen", "A fuzzy-based decision-making procedure for data warehouse system selection", *Expert Systems with Applications*, vol. 32, no. 3, pp. 939 –953, 2007, ISSN: 0957-4174. DOI: http://dx.doi.org/10.1016/j.eswa.2006.01.031. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417406000509.

[184] H. Liu, M. Bowman, and F. Chang, "Survey of state melding in virtual worlds", *ACM Computing Surveys (CSUR)*, vol. 44, no. 4, p. 21, 2012.

[185] Y. Liu, Y. Chen, J. Zhou, and S. Zhong, "Fuzzy linear regression models for {qfd} using optimized h values", *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 45 –54, 2015, ISSN: 0952-1976. DOI: http://dx.doi.org/10.1016/j.engappai.2014.11.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197614002796.

[186] F. H. Lotfi and R. Fallahnejad, "Imprecise shannonâăźs entropy and multi attribute decision making", *Entropy*, vol. 12, no. 1, pp. 53–62, 2010.

[187] D. Mace, W. Gao, and A. K. Coskun, "Improving accuracy and practicality of accelerometer-based hand gesture recognition", *On Interacting with Smart Objects*, p. 45, 2013.

[188] J. Maeda, *The laws of simplicity*, (Accessed on 22/03/2017), 2005. [Online]. Available: http://lawsofsimplicity.com/los.html.

[189] C. Maggioni, "A novel gestural input device for virtual reality", in *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, IEEE, 1993, pp. 118–124.

[190] T. W. Malone and K. Crowston, "The interdisciplinary study of coordination", *ACM Computing Surveys (CSUR)*, vol. 26, no. 1, pp. 87–119, 1994.

[191] U. Manes-Korban, *Night of sciences- university of kaiserslautern*, (Accessed on 01/14/2018), 2016. [Online]. Available: http://www.uni-kl.de/en/university/pr-marketing/events/night-of-sciences/.

[192] J. Mäntyjärvi, J. Kela, P. Korpipää, and S. Kallio, "Enabling fast and effortless customisation in accelerometer based gesture interaction", in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, ACM, 2004, pp. 25–31.

[193] N. Marquardt, T. Ballendat, S. Boring, S. Greenberg, and K. Hinckley, "Gradual engagement: Facilitating information exchange between digital devices as a function of proximity", in *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ACM, 2012, pp. 31–40.

[194] N. Marquardt, K. Hinckley, and S. Greenberg, "Cross-device interaction via micro-mobility and f-formations", in *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM, 2012, pp. 13–22.

[195] P. Marshall, E. Hornecker, R. Morris, N. S. Dalton, and Y. Rogers, "When the fingers do the talking: A study of group participation with varying constraints to a tabletop interface", in *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, IEEE, 2008, pp. 33–40.

[196] E. Mayr, G. Schreder, M. Smuc, and F. Windhager, "Looking at the representations in our mind: Measuring mental models of information visualizations",

in *Proceedings of the Beyond Time and Errors on Novel Evaluation Methods for Visualization*, ACM, 2016, pp. 96–103.
[197] P. Mayring, "Qualitative sozialforschung", *Eine Anleitung zu qualitativen Denken*, vol. 5, 2002.
[198] N. Menck, X. Yang, C. Weidig, P. Winkes, C. Lauer, H. Hagen, B. Hamann, and J. Aurich, "Collaborative factory planning in virtual reality", *Procedia CIRP*, vol. 3, pp. 317–322, 2012.
[199] D. Mendes, M. Sousa, B. Araujo, A. Ferreira, H. Noronha, P. Campos, L. Soares, A. Raposo, and J. Jorge, "Collaborative 3d visualization on large screen displays", in *Powerwall-International Workshop on Interactive, Ultra-High-Resolution Displays–ACM CHI*, vol. 2013, 2013.
[200] Microsoft, *Collaborate and share - word*, (Accessed on 01/17/2018), 2018. [Online]. Available: `https://support.office.com/en-us/article/Collaborate-and-share-2d1d704d-7afb-4221-9301-fededf8150ff`.
[201] Microsoft, *Kinect for windows sensor components and specifications*, (Accessed on 01/24/2018), 2018. [Online]. Available: `https://msdn.microsoft.com/en-us/library/jj131033.aspx`.
[202] Microsoft, *Microsoft speech service*, (Accessed on 01/30/2018), 2018. [Online]. Available: `https://docs.microsoft.com/en-us/azure/cognitive-services/speech/home`.
[203] D. D. Mittleman, R. O. Briggs, J. Murphy, and A. Davis, "Toward a taxonomy of groupware technologies", in *Groupware: Design, Implementation, and Use*, Springer, 2008, pp. 305–317.
[204] M. Miyasugi, H. Akaike, Y. Nakayama, and H. Kakuda, "Implementation and evaluation of multi-user mind map authoring system using virtual reality and hand gestures", in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, 2017, pp. 1–5. DOI: `10.1109/GCCE.2017.8229432`.
[205] E. Moerland, S. Deinert, F. Daoud, J. Dornwald, and B. Nagel, "Collaborative aircraft design using an integrated and distributed multidisciplinary product development process", in *30th Congress of the International Council for Aeronautical Sciences*, 2016.
[206] A. Mohamed, T. Wanyama, G. Ruhe, A. Eberlein, and B. Far, "Cots evaluation supported by knowledge bases", in *International Workshop on Learning Software Organizations*, Springer, 2004, pp. 43–54.
[207] R. Moseley, "Deep immersion with kasina: An exploration of meditation and concentration within virtual reality environments", in *2017 Computing Conference*, 2017, pp. 523–529. DOI: `10.1109/SAI.2017.8252146`.
[208] N. Mundbrod, "Business process support for collaborative knowledge workers", Faculty of Engineering, Computer Science Institute of Databases, and Information Systems Master Thesis, Ulm University, Tech. Rep., 2012.
[209] B. A. Myers, C. H. Peck, J. Nichols, D. Kong, and R. Miller, "Interacting at a distance using semantic snarfing", in *International Conference on Ubiquitous Computing*, Springer, 2001, pp. 305–314.
[210] C. I. Nass and S. Brave, *Wired for speech: How voice activates and advances the human-computer relationship*. MIT press Cambridge, MA, 2005.
[211] NEC, IC, "Micon", *NEC IC Micon Deming Prize Application. NEC IC Microcomputer systems, Kawasaki City, Japan*, 1987.
[212] S. Nestler, E. Artinger, T. Coskun, Y. Yildirim-Krannig, S. Schumann, M. Maehler, F. Wucholt, S. Strohschneider, and G. Klinker, "Assessing qualitative usability in life-threatening, timecritical and unstable situations, 10", in *Workshop Mobile Informationstechnologien in der Medizin (MoCoMed 2010)*, 2010.

[213] J. W. Newman, *Management applications of decision theory*. Harper & Row, 1971.

[214] E. W. Ngai and E. Chan, "Evaluation of knowledge management tools using ahp", *Expert systems with applications*, vol. 29, no. 4, pp. 889–899, 2005.

[215] E. T. Ngai and D. Chow, "Icads: Intelligent car audio design system for product planning", *Expert Systems*, vol. 16, no. 1, pp. 19–32, 1999.

[216] R. Nibbelke, D. Ferro, and P. Hoogeboom, "Design and evaluation with the human in mind", *AIR & Space Europe*, vol. 3, no. 3, pp. 218–220, 2001.

[217] L. Nigay and J. Coutaz, "A design space for multimodal systems: Concurrent processing and data fusion", in *CHI '93 Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, Amsterdam, 1993, pp. 172–178.

[218] Occipital, *Openni 2 downloads and documentation – the structure sensor*, (Accessed on 01/26/2018), 2018. [Online]. Available: `https://structure.io/openni`.

[219] C. O'Loughlin, J. Walsh, and T. Mahon, "Collaboration in the development of a precision engineering programme at limerick institute of technology in response to a call from industry", in *2015 IEEE Frontiers in Education Conference (FIE)*, 2015, pp. 1–5. DOI: `10.1109/FIE.2015.7344141`.

[220] OpenNI Organization, *Nite 2.2.0.11 – openni*, (Accessed on 01/26/2018), 2018. [Online]. Available: `http://openni.ru/files/nite/`.

[221] Oracle Corporation, *Mysql :: Mysql 5.6 reference manual :: 14.2 innodb and the acid model*, 2016. [Online]. Available: `https://dev.mysql.com/doc/refman/5.6/en/mysql-acid.html` (visited on 11/09/2016).

[222] M. Otto, *Bootstrap âů the most popular html, css, and js library in the world.* (Accessed on 01/24/2018), 2018. [Online]. Available: `https://getbootstrap.com/`.

[223] P. S. Pande, R. P. Neuman, and R. R. Cavanagh, *The six sigma way*. McGraw-Hill, 2000.

[224] PARC – Palo Alto Research Center Incorporated, *Colab (electronic meeting room)*, (Accessed on 01/17/2018), 2018. [Online]. Available: `http://www.markstefik.com/?page_id=155`.

[225] H. Patel, M. Pettitt, and J. R. Wilson, "Factors of collaborative working: A framework for a collaboration model", *Applied ergonomics*, vol. 43, no. 1, pp. 1–26, 2012.

[226] R. Pausch, D. Proffitt, and G. Williams, "Quantifying immersion in virtual reality", in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 13–18.

[227] S. Pelurson and L. Nigay, "Multimodal interaction with a bifocal view on mobile devices", in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ACM, 2015, pp. 191–198.

[228] M. Pérez and T. Rojas, "Evaluation of workflow-type software products: A case study", *Information and Software Technology*, vol. 42, no. 7, pp. 489 –503, 2000.

[229] C. K. Pickering, "Synergizing people, process, and technology to motivate knowledge sharing and collaboration industry case study", in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 35–39. DOI: `10.1109/CTS.2013.6567200`.

[230] D. Piekenbrock and A. Hennig, *Einführung in die volkswirtschaftslehre und mikroökonomie*. Springer-Verlag, 2012.

[231] E. Prasolova-Forland, J. Molka-Danielsen, M. Fominykh, and K. Lamb, "Active learning modules for multi-professional emergency management training in

virtual reality", in *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2017, pp. 461–468. DOI: 10.1109/TALE.2017.8252380.

[232] QFD Institute, *Blitz qfd*, http://www.qfdi.org/what_is_qfd/blitz_qfd.html, (undefined 27/8/2017 18:50), May 2011.

[233] K. S. Raju and A Vasan, "Multi attribute utility theory for irrigation system evaluation", *Water resources management*, vol. 21, no. 4, pp. 717–728, 2007.

[234] H. Rampl, *Usability iso 9241*, (Accessed on 09/03/2017), 2017. [Online]. Available: http://www.handbuch-usability.de/iso-9241.html.

[235] W. A. Reed, "User-centered design: A guide to roi with iso 9241-210", *Texology Sci*, 2014.

[236] R. D. Reid and N. R. Sanders, *Operations management: An integrated approach, fourth edition*. Wiley Hoboken, NJ, 2010.

[237] J. Rekimoto, "Tilting operations for small screen interfaces", in *Proceedings of the 9th annual ACM symposium on User interface software and technology*, ACM, 1996, pp. 167–168.

[238] P. Riehmann, M. Hanfler, and B. Froehlich, "Interactive sankey diagrams", in *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, IEEE, 2005, pp. 233–240.

[239] V. Roto, A. Popescu, A. Koivisto, and E. Vartiainen, "Minimap: A web page visualization method for mobile phones", in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, ACM, 2006, pp. 35–44.

[240] V. Rousseau and C. Aubé, "Team self-managing behaviors and team effectiveness: The moderating effect of task routineness", *Group & Organization Management*, vol. 35, no. 6, pp. 751–781, 2010.

[241] A. Ruffert, *Rangeslider.js*, (Accessed on 01/24/2018), 2018. [Online]. Available: http://rangeslider.js.org/.

[242] D. Russell, C. Drews, and A. Sue, "Social aspects of using large public interactive displays for collaboration", *UbiComp 2002: Ubiquitous Computing*, pp. 663–670, 2002.

[243] R. Saaty, "The analytic hierarchy process what it is and how it is used", *Mathematical Modelling*, vol. 9, no. 3-5, pp. 161–176, 1987, ISSN: 0270-0255. DOI: http://dx.doi.org/10.1016/0270-0255(87)90473-8. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0270025587904738.

[244] J. Sarkis and D. H. Liles, "Using idef and qfd to develop an organizational decision support methodology for the strategic justification of computer-integrated technologies", *International Journal of Project Management*, vol. 13, no. 3, pp. 177–185, 1995.

[245] J. Sarkis and S. Talluri, "Evaluating and selecting e-commerce software and communication systems for a supply chain", *European journal of operational research*, vol. 159, no. 2, pp. 318–329, 2004.

[246] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller", in *Proceedings of the 2nd international conference on Tangible and embedded interaction*, ACM, 2008, pp. 11–14.

[247] D. Schmidt, J. Seifert, E. Rukzio, and H. Gellersen, "A cross-device interaction style for mobiles and surfaces", in *Proceedings of the Designing Interactive Systems Conference*, ACM, 2012, pp. 318–327.

[248] M. Schmidt, "The sankey diagram in energy and material flow management", *Journal of industrial ecology*, vol. 12, no. 1, pp. 82–94, 2008.

[249] A. Schneider, D. Cernea, and A. Ebert, "Hmd-enabled virtual screens as alternatives to large physical displays", in *Information Visualisation (IV), 2016 20th International Conference*, IEEE, 2016, pp. 390–394.

[250] J. Scholtz and M. P. Steves, "A framework for real-world software system evaluations", in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, ACM, 2004, pp. 600–603.

[251] M. Schreiber, M. von Wilamowitz-Moellendorff, and R. Bruder, "New interaction concepts by using the wii remote", in *13th International Conference on Human-Computer Interaction*, San Diego, CA: Springer-Verlag Berlin, Heidelberg, 2009, pp. 261 –270.

[252] G. Schuh, T. Brosze, U. Brandenburg, S. Cuber, M. Schenk, J. Quick, C. Schmidt, J. Helmig, M. Schürmeyer, and N. Hering, "Grundlagen der produktionsplanung und-steuerung", in *Produktionsplanung und-steuerung 1*, Springer, 2012, pp. 11–293.

[253] T. Schümmer, S. Lukosch, and R. Slagter, "Using patterns to empower endusers - the oregon software development process for groupware", *International Journal of Cooperative Information Systems*, vol. 15, no. 02, pp. 259 –288, 2006.

[254] S. D. Scott, K. D. Grant, and R. L. Mandryk, "System guidelines for co-located, collaborative work on a tabletop display", in *ECSCW 2003*, Springer, 2003, pp. 159–178.

[255] B. Scott-Ladd and C. C. Chan, "Using action research to teach students to manage team learning and improve teamwork satisfaction", *Active Learning in Higher Education*, vol. 9, no. 3, pp. 231–248, 2008.

[256] J. Seifert, A. Simeone, D. Schmidt, P. Holleis, C. Reinartz, M. Wagner, H. Gellersen, and E. Rukzio, "Mobisurf: Improving co-located collaboration through integrating mobile devices and interactive surfaces", in *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, ACM, 2012, pp. 51–60.

[257] K. I. Shaikh, "Thrill your customer, be a winner", in *Transactions from a Symposium on Quality Function Deployment*, 1989, pp. 18–19.

[258] N. Shariatzadeh, G. Sivard, and D. Chen, "Software evaluation criteria for rapid factory layout planning, design and simulation", *Procedia CIRP*, vol. 3, pp. 299–304, 2012.

[259] A. I. Sharkey, "Generalized approach to adapting qfd for software", in *Transactions of the Third Symposium on Quality Function Deployment, June*, 1991, pp. 24–25.

[260] S. Sharples, S. Cobb, A. Moody, and J. R. Wilson, "Virtual reality induced symptoms and effects (vrise): Comparison of head mounted display (hmd), desktop and projection display systems", *Displays*, vol. 29, no. 2, pp. 58–69, 2008.

[261] C.-w. Shen, J.-t. Ho, T.-C. Kuo, and T. H. Luong, "Behavioral intention of using virtual reality in learning", in *Proceedings of the 26th International Conference on World Wide Web Companion*, International World Wide Web Conferences Steering Committee, 2017, pp. 129–137.

[262] T. B. Sheridan, "Teleoperation, telerobotics and telepresence: A progress report", *Control Engineering Practice*, vol. 3, no. 2, pp. 205–214, 1995.

[263] D. C. Sherrard and M. Narayanan, "The aid of virtual reality in the industry", in *IEEE Technical Applications Conference and Workshops. Northcon/95. Conference Record*, 1995, pp. 28–. DOI: 10.1109/NORTHC.1995.484951.

[264] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations", in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, IEEE, 1996, pp. 336–343.

[265] B. Shneiderman and C. Plaisant, "Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies", in *Proceedings of the 2006 AVI workshop on BEyond time and errors: Novel evaluation methods for information visualization*, ACM, 2006, pp. 1–7.

[266] T. W. Simpson, V. Toropov, V. Balabanov, and F. A. Viana, "Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not", in *12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, vol. 5, 2008, pp. 10–12.

[267] Sony Mobile Communications Inc., *Smartwatch 3 swr50 specifications*, (Accessed on 01/30/2018), 2018. [Online]. Available: `https://www.sonymobile.com/global-en/products/smart-products/smartwatch-3-swr50/specifications/`.

[268] State Research Institute of Aviation Systems, *Development of design and architecture – modern avionics of civil aviation*, (Accessed on 11/21/2017), 2017. [Online]. Available: `http://www.modern-avionics.com/design-process/development-process/design-and-architecture/`.

[269] Statista, *Smartphones - prognose zu den weltweiten anschlüssen bis 2020*, (Accessed on 01/20/2018), 2018. [Online]. Available: `https://de.statista.com/statistik/daten/studie/312258/umfrage/weltweiter-bestand-an-smartphones/`.

[270] M. Steiger, J. Bernard, S. Mittelstädt, H. Lücke-Tieke, D. Keim, T. May, and J. Kohlhammer, "Visual analysis of time-series similarities for anomaly detection in sensor networks", in *Computer Graphics Forum*, Wiley Online Library, vol. 33, 2014, pp. 401–410.

[271] M. P. Steves and J. Scholtz, "A framework for evaluating collaborative systems in the real world", in *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, IEEE, 2005, pp. 29c–29c.

[272] N. A. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz, "I-land: An interactive landscape for creativity and innovation", in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM, 1999, pp. 120–127.

[273] L. Sullivan, *Quality function deployment*. AMERICAN SOCIETY FOR QUALITY, 1986, pp. 39–50.

[274] C. Sun and C. Ellis, "Operational transformation in real-time group editors: Issues, algorithms, and achievements", in *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, ACM, 1998, pp. 59–68.

[275] E. Sundstrom, K. P. De Meuse, and D. Futrell, "Work teams: Applications and effectiveness.", *American psychologist*, vol. 45, no. 2, p. 120, 1990.

[276] M. K. Tageldeen, I. Elamvazuthi, N. Perumal, and T. Ganesan, "A virtual reality based serious games for rehabilitation of arm", in *2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA)*, 2017, pp. 1–6. DOI: `10.1109/ROMA.2017.8231737`.

[277] C. Temponi, J. Yen, and W. A. Tiao, "House of quality: A fuzzy logic-based requirements analysis", *European Journal of Operational Research*, vol. 117, no. 2, pp. 340–354, 1999.

[278] R. Thackeray and G. V. Treeck, "Applying quality function deployment for software product development", *Journal of Engineering Design*, vol. 1, no. 4, pp. 389–410, 1990.

[279] The Etherpad Foundation, *Etherpad*, (Accessed on 01/09/2018), 2018. [Online]. Available: `http://etherpad.org/`.

[280] E. Thomas, M. Ravachol, J. B. Quincy, and M. Malmheden, "Collaborative complex system design applied to an aircraft system", in *Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany*, Linköping University Electronic Press, 2012, pp. 855–866.

[281] D. M. Thompson and M. H. Fallah, "Qfd–a systematic approach to product definition", in *ASQC Quality Congress Transactions-Toronto, l989*, 1989.

[282] K. D. Thoondee and A. Oikonomou, "Using virtual reality to reduce stress at work", in *2017 Computing Conference*, 2017, pp. 492–499. DOI: `10.1109/SAI.2017.8252142`.

[283] K. Thyng, *Beautiful colormaps for oceanography*, (Accessed on 12/23/2017), 2017. [Online]. Available: `https://matplotlib.org/cmocean/`.

[284] E. Timberg, T. Linsley, and S. Brunel, *Chart.js | open source html5 charts for your website*, (Accessed on 01/24/2018), 2018. [Online]. Available: `http://www.chartjs.org/`.

[285] M. Tobiasz, P. Isenberg, and S. Carpendale, "Lark: Coordinating co-located collaboration with information visualization", *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1065–1072, 2009.

[286] T. Tolio, D. Ceglarek, H. ElMaraghy, A. Fischer, S. Hu, L. Laperrière, S. T. Newman, and J. Váncza, "Species–co-evolution of products, processes and production systems", *CIRP Annals-Manufacturing Technology*, vol. 59, no. 2, pp. 672–693, 2010.

[287] C. Tominski, *Interaction for visualization*, ser. Synthesis Lectures on Visualization. Morgan & Claypool Publishers, 2015, ISBN: 9781627057745. [Online]. Available: `https://books.google.de/books?id=Qpr8CQAAQBAJ`.

[288] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko, "Stacking-based visualization of trajectory attribute data", *IEEE Transactions on visualization and Computer Graphics*, vol. 18, no. 12, pp. 2565–2574, 2012.

[289] S. Tregillus and E. Folmer, "Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments", in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 2016, pp. 1250–1255.

[290] E. R. Tufte, "The visual display of quantitative information.", *Journal for Healthcare Quality*, vol. 7, no. 3, p. 15, 1985.

[291] C. J. Turner, W. Hutabarat, J. Oyekan, and A. Tiwari, "Discrete event simulation and virtual reality use in industry: New opportunities and future trends", *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 6, pp. 882–894, 2016, ISSN: 2168-2291. DOI: `10.1109/THMS.2016.2596099`.

[292] Unity Technologies, *Unity - products*, (Accessed on 01/26/2018), 2018. [Online]. Available: `https://unity3d.com/de/unity`.

[293] H. Väätäjä, J. Varsaluoma, T. Heimonen, K. Tiitinen, J. Hakulinen, M. Turunen, H. Nieminen, and P. Ihantola, "Information visualization heuristics in practical expert evaluation", in *Proceedings of the Beyond Time and Errors on Novel Evaluation Methods for Visualization*, ACM, 2016, pp. 36–43.

[294] W. Van Vlaenderen, J. Brulmans, J. Vermeulen, and J. Schöning, "Watchme: A novel input method combining a smartwatch and bimanual interaction", in *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 2015, pp. 2091–2095.

[295] M. Van Welie, G. C. Van der Veer, and A. Eliëns, "An ontology for task world models", in *Design, Specification and Verification of Interactive Systems' 98*, Springer, 1998, pp. 57–70.

[296] G. van der Veer, O. Kulyk, D. Vyas, O. Kubbe, and A. Ebert, "Task modeling for collaborative authoring", in *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics*, ser. ECCE '11, Proceedings of the 29th Annual European Conference on Cognitive Ergonomics, New York, NY, USA: ACM, 2011, pp. 171–178, ISBN: 978-1-4503-1029-1. DOI: 10.1145/2074712.2074746. [Online]. Available: http://doi.acm.org/10.1145/2074712.2074746.

[297] G. van der Veer and M. van Welie, "Task based groupware design: Putting theory into practice", in *Proceedings of the 3rd conference on Designing interactive systems: Processes, practices, methods, and techniques*, ACM, 2000, pp. 326–337.

[298] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis, "User acceptance of information technology: Toward a unified view", *MIS quarterly*, pp. 425–478, 2003.

[299] I. Vlahavas, I. Stamelos, I. Refanidis, and A. Tsoukiàs, "Esse: An expert system for software evaluation", *Knowledge-based systems*, vol. 12, no. 4, pp. 183–197, 1999.

[300] M. Vollrath, *Ingenieurpsychologie: Psychologische grundlagen und anwendungsgebiete*. Kohlhammer Verlag, 2015.

[301] W3C Working Group, *Concur task trees ctt*, accessed: 5/9/2017, Nov. 2012. [Online]. Available: https://www.w3.org/2012/02/ctt/.

[302] D. Wachsman, *Definition of smart device | new word suggestion | collins dictionary*, (Accessed on 01/16/2018), 2018. [Online]. Available: https://www.collinsdictionary.com/submission/11527/Smart+Device.

[303] F. Wagner, R. Schmuki, T. Wagner, and P. Wolstenholme, *Modeling software with finite state machines: A practical approach*. CRC Press, 2006.

[304] M. O. Ward, "A taxonomy of glyph placement strategies for multidimensional data visualization", *Information Visualization*, vol. 1, no. 3-4, pp. 194–210, 2002.

[305] J. Weber, "Überlegungen zu einer theoretischen fundierung der logistik in der betriebswirtschaftslehre", *Beiträge zu einer Theorie der Logistik*, pp. 43–65, 2008.

[306] S. Wei and Z. Wen-qi, "Virtual reality technology in modern medicine", in *2010 International Conference on Audio, Language and Image Processing*, 2010, pp. 557–561. DOI: 10.1109/ICALIP.2010.5684506.

[307] C. Weidig, P. Galambos, Á. Csapó, P. Zentay, P. Baranyi, J. C. Aurich, B. Hamann, and O. Kreylos, "Future internet-based collaboration in factory planning", *Acta Polytechnica Hungarica*, vol. 11, no. 7, 2014.

[308] P. E. Weiseth, B. E. Munkvold, B. Tvedte, and S. Larsen, "The wheel of collaboration tools: A typology for analysis within a holistic framework", in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, ACM, 2006, pp. 239–248.

[309] G. Wilkinson, A. Kharrufa, J. D. Hook, B. Pursgrove, G. Wood, H. Haeuser, N. Hammerla, S. Hodges, and P. Olivier, "Expressy: Using a wrist-worn inertial measurement unit to add expressiveness to touch-based interactions", in *Proceedings of the ACM Conference on Human Factors in Computing Systems:*, Association for Computing Machinery (ACM), 2016.

[310] Writelatex Limited, *Overleaf for authors - write and collaborate online with latex*, (Accessed on 12/23/2017), 2017. [Online]. Available: https://www.overleaf.com/benefits.

[311] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-d accelerometer", in *International Conference on Ubiquitous Intelligence and Computing*, Springer, 2009, pp. 25–38.

[312] M. A. Yalçin, N. Elmqvist, and B. B. Bederson, "Cognitive stages in visual data exploration", in *Proceedings of the Beyond Time and Errors on Novel Evaluation Methods for Visualization*, ACM, 2016, pp. 86–95.

[313] E. A. Yoshida, M. L. A. Castro, and V. F. Martins, "Virtual reality and fetal medicine – a systematic review", in *2017 XLIII Latin American Computer Conference (CLEI)*, 2017, pp. 1–10. DOI: 10.1109/CLEI.2017.8226468.

[314] T. Yoshizawa, Y. Akao, M. Ono, and H. Shindo, "Recent aspects of qfd in the japanese software industry", *Quality Engineering*, vol. 5, no. 3, pp. 495–504, 1993.

[315] H. Yu, H. Deng, Z. Zhao, and Y. Xiao, "Transformation of landscape planning and design in computer supported cooperative work era", in *Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on*, IEEE, 2008, pp. 1112–1115.

[316] U. Zaldívar-Colado, S. Garbaya, P. Tamayo-Serrano, X. Zaldívar-Colado, and P. Blazevic, "A mixed reality for virtual assembly", in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017, pp. 739–744. DOI: 10.1109/ROMAN.2017.8172385.

[317] Y. D. Zhejiang, H. Ren, G. Pan, and S. Li, "Tilt & touch: Mobile phone for 3d interaction", in *UbiComp '11 Proceedings of the 13th international conference on Ubiquitous computing*, Beijing, China, 2011, pp. 485–486.

[318] I. Zigurs and B. E. MUNKVOLD, "Collaboration technologies, tasks, and contexts", *Human-Computer Interaction and Management Information Systems: Applications. Advances in Management Information Systems*, p. 143, 2014.

[319] Z.-H. Zou, Y. Yi, and J.-N. Sun, "Entropy method for determination of weight of evaluating indicators in fuzzy synthetic evaluation for water quality assessment", *Journal of Environmental Sciences*, vol. 18, no. 5, pp. 1020–1023, 2006.

[320] R. Zultner, "Understanding customer needs: Software qfd and the voice of the customer", in *Design for trustworthy software: Tools, techniques, and methodology of developing robust software*, B. K. Jayaswal and P. C. Patton, Eds., Pearson Education, 2006, ch. 11, p. 160.

[321] R. E. Zultner, "Tqm for technical teams", *Communications of the ACM*, vol. 36, no. 10, pp. 79–91, 1993.

[322] R. E. Zultner, "Software quality function deployment", in *ANNUAL QUALITY CONGRESS PROCEEDINGS-AMERICAN SOCIETY FOR QUALITY CONTROL*, 1994, pp. 783–783.

[323] R. E. Zultner, "Software qfd: The north american experience", in *Proceedings of the 1st Pacific Rim Symposium on Quality Deployment*, 1995, pp. 163–174.

[324] R. E. Zultner, "Blitz qfd tutorial", in *Transactions from the 10th Symposium on QFD Tutorials*, 1998.

List of Figures

List of Tables

List of Equations

List of Abbreviations

A	AHP	Analytic Hierarchy Process
	API	Application Programming Interface
B	BYOD	Bring Your Own Device
C	CAD	Computer-Aided Design
	CAE	Computer-Aided Engineering
	CAVE	Cave Automatic Virtual Environment
	CIDP	Computer Integrated Design and Production
	CIMT	Collaboration Interaction Manipulation Tool
	CM	Content Management
	CO	Coordination Support
	COTS	Commercial Of The Shelf
	CS	Communication Support
	CSCW	Computer Supported Cooperative Work
	CT	Content Support
	CTT	Concur Task Tree
D	DMAIC	Define, Measure, Analyse, Improve, Control
E	ECS	Environment Control System
	EDPC	Event-driven Production Control
	eHoQ	enhanced House of Quality
F	FLR	Fisheries Library in R
	FN	False Negatives
	FSM	Finite State Machine
	FP	False Positives
	FPR	False Positive Rate
G	GD	Google Docs
	GUI	Graphical User Interface
H	HCI	Human Computer Interaction
	HMD	Head-Mounted Display
	HoQ	House of Quality
I	IN^{2}CO	INtuitive and INteractive COllaboration
	IR	Inconsistency Ratio
	IS	Information Sharing
J	JSON	JavaScript Object Notation
L	LDD	Large Deisplay Device
M	MAUT	Multi Attribute Utility Theory
	MCDA	Multi Criteria Decision Analysis
	MCDM	Multi Criteria Decision Making

O	OL	Overleaf
	OO	OPENOFFICE
P	PTT	Put That There
Q	QFD	Qualiry Function Deployment
R	ROC	Receiver Operating Characteristic
S	SDK	Software Development Kit
	SM	State Machine
	SPC	Statisitcal Progress Controll
	SRS	System Requirement Specification
T	TAM	Technology Acceptance Model
	TN	True Negatives
	TP	True Positives
	TPR	True Positive Rate
U	UCD	User Centered Design
	UDP	User Datagram Protocol
	UI	User Interface
	UUX	Usability and User Experience
	UX	User Experience
V	VDC	Virtual Design and Construction
	VDM	Virtual Design and Manufacturing
	VE	Virtual Environments
	VC	Virtual Commissioning
	VORD	Viewpoint-Oriented Requirement Definition
	VR	Virtual Reality
W	WYSIWYG	What You See Is What You Get

Appendix A

Curriculum Vitae

Franca Alexandra Rupprecht

Born in Flörsheim am Main, Germany
rupprecht@cs.uni-kl.de

Education

PhD Student, Computer Graphics and HCI Group ***Mar. 2018***

University of Kaiserslautern, Kaiserslautern, Germany

Member of IRTG 2057 (http://www.irtg2057.de/)

Research Topic: Scalable Human-Centered Decision Making Processes in Virtual Environments.

M.Sc. in Industrial Engineering, Specialty: Computer Science ***Oct. 2014***

University of Kaiserslautern, Kaiserslautern, Germany

Focus areas: Banking- and finance management, marketing, software-engineering, information systems

Thesis: Design and implementation of an intuitive user interface for an automotive quality assurance tool ("Virtueller Meisterbock", Volkswagen AG)

B.Sc.in Industrial Engineering, Specialty: Computer Science ***Oct. 2012***

University of Kaiserslautern, Kaiserslautern, Germany

Thesis Determination factors of the capital requirement of a company

Experience

Teaching Assistance ***Nov. 2017 - Present***

University of Kaiserslautern, Kaiserslautern, Germany
Visual Information Analysis Group

Lecture:	Data Visualization

Research Assistant ***Nov. 2012 - Oct. 2014***

University of Kaiserslautern, Kaiserslautern, Germany
Computer Graphics and HCI Group

Topics:	Server-Client communication between diverse (smart) devices
	Cross-platform development of mobile applications
	Development of a library for web based smart-phone application permitting multi-touch gestures

Appendix B

List of Publications

Journal publications

[1] **F. Rupprecht**, G. Kasakow, J. Aurich, B. Hamann, A. Ebert, "Improving Collaboration Efficiency via Diverse Networked Mobile Devices", *Journal of Multimodal User Interfaces*, Springer, 2017.

[2] **F. Rupprecht**, F. Torner, J. Seewig, A. Ebert, "Dependency graph based on user taxonomy and related parameters for more efficient collaborative work", *Applied Mechanics and Materials 869 - Proceedings of the 1st Conference on Physical Modeling for Virtual Manufacturing Systems and Processes*, AMM 869:195-211, 2017.

[3] J. Schwank, **F. Rupprecht**, S. Schöffel, "Orientation-based Interaction on Mobile Devices and Desktops - An Evaluation", *Applied Mechanics and Materials 869 - Proceedings of the 1st Conference on Physical Modeling for Virtual Manufacturing Systems and Processes*, 2017.

Peer-reviewed publications

[4] **F. Rupprecht**, V. Soni, C. Schmidt, B. Ravani, A. Ebert, G. van der Veer, "An Approach for Evaluating Collaborative Software Environments Based on Integration of House of Quality with Multi-Attribute Utility Theory", *The 9th International Congress on Ultra Modern Telecommunications and Control Systems - ICUMT*, 2017.

[5] **F. Rupprecht,** A. Ebert, A. Schneider, B. Hamann, "Virtual Reality Meets Smartwatch: Intuitive, Natural, and Multi-Modal Interaction", *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, (pp. 2884-2890), ACM, 2017.

[6] J. Schwank, **F. Rupprecht**, A. Ebert, "Waggle–Orientation-based Tablet Interaction", *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 2042-2048, ACM, 2017.

[7] **F. Rupprecht**, T. Khan, G. van der Veer, A. Ebert, "Criteria Catalogue for Collaborative Environments", *Proceedings of the 31st International BCS Human Computer Interaction Conference (HCI 2017 - Digital make-believe)*, University of Sunderland, St PeterâĂŹs campus, Sunderland, UK, 3 - 6 July 2017.

[8] **F. Rupprecht**, B. Hamann, C. Weidig, J. Aurich, A. Ebert, "IN2CO - A Visualization Framework for Intuitive Collaboration", *The Eurographics Association EuroVis 2016 Short Paper Track*, Groningen, Netherlands, 2016.

[9] S. Humayoun, **F. Rupprecht**, S. Hess, A. Ebert, "Adding Multi-Touch Gesture Interaction in Mobile Web Apps", *M. Kurosu (Ed.): Human-Computer Interaction*, Part II, HCII 2014, LNCS 8511, pp. 48-57, 2014.

Invited Talks

[10] **F. Rupprecht** and A. Ebert,"Mobile Interaktion & Kollaboration", *Workshop on Interaction with Mobile Systems as part of Smart Ecosystems*, Mensch und Computer Konferenz, 2016.

Accepted for publication

[11] **F. Rupprecht**, B. Heck, B. Hamann, A. Ebert, "Signal-processing Transformation Proceed from Smartwatch into Arm Movement Gestures", submitted at *the conference track "Human Factors and Systems Interaction" of the 9th International Conference on Applied Human Factors and Ergonomics*, AHFE, 2018.

[12] R. Ilsen, **F. Rupprecht**, G. Mert, A. Ebert, J. Aurich, "Impact-Visualization to Evaluate Resource Efficiency of Product-Service Systems", *25th CIRP Life Cycle Engineering (LCE) Conference*, 30 April - 2 May 2018, Copenhagen, Denmark, status: in review.

Submitted

[13] **F. Rupprecht**, G. Kassakow, J. Aurich, A. Ebert, "Enhancement of the Event-driven Production Control to Increase User Acceptance for an Autonomous System", *Journal of Manufacturing Systems*, status: in review.

[14] **F. Rupprecht**, B. Ravani, A. Ebert, "Design of a Collaboration Framework Based on an Enhanced House of Quality Methodology", at *Journal of Research in Engineering Design*, Springer Journals, status: in review.

[15] **F. Rupprecht**, O. Joseph, C. Naranjo Valero, B. Hamann and A. Ebert, "Natural and Multi-Modal Interaction with Large Displays Using a Smartwatch: Speech, Touch, and Non-Touch Gestures", submitted at *The 10th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS, 2018, status: in review.