Karlsruher Institut für Technologie

Karlsruhe Reports in Informatics 2013,3
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Ubiquitäre Systeme (Seminar) WS 2012/13

Mobile und Verteilte Systeme
Ubiquitous Computing

Teil VIII

Herausgeber:
Predrag Jakimovski, Matthias Berning,
Markus Scholz, Martin Alexander Neumann,
Anja Bachmann, Yong Ding, Till Riedel

2013

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Karlsruhe Institute of Technology

Ubiquitäre Systeme (Seminar)
WS 2012/13

Mobile und Verteilte Systeme
Ubiquitous Computing

Teil VIII

Herausgeber

Predrag Jakimovski
Matthias Berning
Markus Scholz
Martin Alexander Neumann
Anja Bachmann
Yong Ding
Till Riedel

Karlsruhe Institute of Technology (KIT)
Fakultät für Informatik
Lehrstuhl für Pervasive Computing Systems (PCS) und TECO

Interner Bericht 2013 - 03

Vorwort

Die Seminarreihe Ubiquitäre Informationssysteme hat eine lange Tradition in der Forschungsgruppe TecO. Ziel der Seminarreihe ist die Aufarbeitung und Diskussion aktueller Forschungsfragen im Bereich Ubiquitous Computing. Seit dem Wintersemester 2003/2004 werden die Seminararbeiten als KIT-Berichte veröffentlicht. Seit das TecO im Wintersemester 2010/2011 Teil der Forschungsgruppe Pervasive Computing Systems wurde, findet das Seminar nun in jedem Semester statt. Dieser Seminarband fasst die Ergebnisse der Arbeiten der Wintersemester 2012/2013 und 2011/2012 zusammen. Die Themenvielfalt der hier zusammengetragenen Aufsätze reicht von Mobile Augmented Reality, Mechanismen zur Software-Updates von Cyber-physical-Systems und Cyber-physical-Systems an sich, sowie Anwendungen von Data Mining in Ubiquitäre Systeme und Analyse von Methoden zum 3D-Körperscan mit Kinect.

Hiermit danken wir den Studierenden für ihren besonderen Einsatz sowohl während des Seminars als auch bei der Fertigstellung dieses Bandes.

Karlsruhe, den 15. April 2013

Predrag Jakimovski
Matthias Berning
Markus Scholz
Martin Alexander Neumann
Anja Bachmann
Yong Ding
Till Riedel

Inhaltsverzeichnis

Software Update Mechanisms for Cyber-Physical Systems

Seminar Ubiquitous Systems - WS 2012/2013

Sebastian Höninger
sebastian.hoeninger@student.kit.edu

Karslruhe Institute of Technology
Institute of Telematics - Pervasive Computing Systems - TecO
Advisor: Martin Alexander Neumann

Abstract. The update process of cyber-physical systems (CPS) and wireless sensor networks (WSN) is accompanied by various challenges. This article provides a list of challenges to be solved and motivates them with two exemplary CPS use cases: the smart grid and automotive electronics with vehicle to vehicle communication. Afterwards the software update mechanisms RemoWare, Deluge and Dynamic TinyOS, HERMES and QARI are compared, each presenting possible solutions for various challenges.

1 Introduction

The structure and the way of interaction with computers will probably change in the future, and so will the tools and requirements for the maintenance of computer systems. From large centralized mainframe systems, followed by personal computers and finally compact devices like smartphones and tablets as well as embedded systems, computer-systems have already come a long way.
The realization of the next vision - ubiquitous computing and cyber-physical systems (CPSs) - has just begun. The way of interaction with computers will probably change to be rather supportive and subconscious, the systems will comprise various interactions between digital and physical environments and new services and applications will be feasible and existing ones can be improved. This trend is also reflected in the 2010 report to the president of the United States [9, p.46ff.], saying that research on CPSs is of high importance and may impact various social challenges.
But the problem of deploying and updating software on a multitude of distributed and potentially inaccessible system nodes arises, aggravated by limited resources and the need to preserve certain quality of service measures throughout the update process [26]. This problem is amplified by the fact, that those nodes might have heterogeneous hardware and software because of specialization or as a result of the system's development life-cycle.

Current research on wireless sensor networks, sensor actuator networks, cyber-physical systems and such is tackling those challenges in theory and practice by developing and deploying distributed systems and trying to create models in order to simulate and analyze their behavior. The ability to update and dynamically modify the software of those nodes was already considered one of the key challenges as early as 1978 [7] and still is today. Implementing those systems in software rather than hardware adds the necessary flexibility to fix errors, add functionality and react to changed requirements. Solving the challenge of easily updating them will likely influence the success and adoption of those systems [8].
Real-time constraints, quality of service and lossy low-bandwidth network connections combined with limited memory, computing and energy resources are main aspects to be considered besides managing the heterogeneity of the nodes concerning both, the hardware and the software.
This work gives an overview of typical challenges for software update mechanisms in cyber-physical systems followed by a comparison of existing implementations.

2 Terminology

The research area of distributed systems has spawned various terms describing their focus, that are sometimes hard to distinguish. On the one hand there is a focus on the computing side. Both pervasive and ubiquitous computing mainly focus on the computing side of human-centric, supportive computer systems "available throughout the physical environment, while making them effectively invisible to the user" [27], which is also called calm computing. Ubiquitous computing stems from the academic area whereas pervasive computing was coined by the industry, but both are much alike. Then there is the term cyber-physical system (CPS), a term coined by Gill around 2006, describing distributed embedded computer systems that monitor and control physical systems [5]. While both, ubiquitous computing systems and CPSs are complex distributed systems with potential user interaction, CPSs primarily combine the virtual and physical world. The nodes of CPSs extend physical systems with a digital monitoring and control component and are often times embedded into physical entities.
On the other hand there is the focus on the communication, in particular on the wireless communication, and the sensing side. Wireless sensor networks (WSNs) are networks of sensor and/or computing nodes connected with wireless communication technology. Speaking of wireless sensor actuator networks (WSANs) means extending this topic by including actuators, and thus moving towards CPSs.
Generally, all those topics are overlapping to some extend and thus may allow for symbiotic co-evolution.

3 Platforms

Wireless sensor (actuator) networks (WSN, WSAN), cyber-physical systems (CPS) and such come in a multitude of flavors as do their nodes' hardware and software capabilities. Depending on the use cases, available resources and node capabilities update system architectures may differ from each other.
This section will depict typical CPS and WSN platform characteristics to be considered while designing a software update mechanisms for them.

3.1 Hardware Restrictions

Often times micro-controllers are used, providing only limited computing and memory resources. This implicates limited possibilities to store different code versions and must be taken into account when making use of computing intensive technology like encryption. Those computing nodes are likely to be powered autonomously, thus also having limited energy resources. This has strong implications on communication and computation, and in some cases even makes energy consumption by memory operations a factor to be considered [26].
The communication part is complicated by mostly relying on wireless communication. Besides the increased energy demand, wireless communication tends to be lossy and thus poses problems when trying to reliably propagate data across the network.
Han et al. [8, p.5] suggests an additional differentiation between hardware platforms that offer a memory management unit (MMU) and thus virtual memory space facilitating the update process and those that don't. The latter hardly allow position independence of code and are "vulnerable to incorrect or malicious memory references" [8, p.5]. Unfortunately, according to Han et al., small microcontrollers often times do not have a MMU. This will probably stay the same for miniaturized nodes in the future, but with modern 32bit SoC systems like those used in smartphones, compact nodes having a MMU may become more common.

3.2 Location of the Update Mechanism

Software update mechanisms may run on different application levels in order to perform an update, and use different strategies of propagation depending on the system. Brown[1, p.6] talks about four basic locations of the update mechanism inside a node's system architecture: application level, middleware level, operating system level and firmware level. For all four of them Brown presents a short overview of update mechanism implementations as of 2006.
Integrating update mechanisms on the firmware level may lead to interoperability problem, as future systems are likely to consist of nodes of different makes. Implementing the mechanism on the operating system level adds a certain degree of hardware independence. The middleware solution is what seems to be the typical location of software update mechanisms if we look at section 6, leaving resource management to the operating system and taking care of the software

version management. Application level software update mechanisms are most easily added to the system as they probably have the least implications on their environment. But the more abstracted the view on the hardware and software becomes, the harder it gets to perform updates which includes monitoring and control of system operation.

4 A Typical CPS

As an introduction to typical challenges and problems when updating distributed systems, this chapter will present typical CPSs and reasons for the necessity of a software update. Thus the environment a software update mechanism is working in will be clarified on an exemplary basis and common pitfalls will become obvious.

4.1 The Advanced Power Grid / Smart Grid as a CPS

The power grid as we know it today is mainly formed by rather large power plants and control units generating electricity and controlling the grid in a rather centralized way. But in recent years there has been a trend towards smaller distributed generation sites. One of the biggest challenges in providing electricity is that production and consumption must be balanced at all times within tight margins, but this is what becomes more complex when integrating distributed sites. The electric power system of tomorrow has to cope with a large amount of distributed electricity generation instead of large scale central generation and additional consumption by new technology like electric vehicles while at the same time providing electricity in an efficient and reliable manner. [23]
To reach that goal, the control system must adapt to the decentralized structure of the system [4]. This can be reached with the development of a distributed control system embedding computing technology into many parts of the system, like the distributed generation sites and the transmission grid but also distributed storage systems (batteries, spinning wheels and other storage solutions) and common home appliances like fridges or washing machines. This power grid of the future is often called smart grid.
The future fridge or washing machine may control their electricity consumption based on current grid load, which might be reflected by dynamic electricity prices that also enable arbitrage profits with storage solutions. This is called demand side management and supports the balancing parts of the generation and transmission side in keeping the demand and supply balanced and avoid grid blackouts. According to Ramchurn et al. [23] a simulation system is an important part of the smart grid to predict its behavior. This may also be used to predict the influence of software update mechanisms with all their consequences. But embedding such technology into various power grid devices also means, that the deployed nodes must be inter-operable to perform their distributed control task. This must be considered when updating some of them. Though software

updates also are a key element to preserve interoperability when the system and its components evolve.
Integrating interconnected digital components into the grid also means, that there might be new hazards for the power grid like security vulnerabilities of the integrated computer systems as presented in [25] by Sridhar et al.. They differentiate between application and infrastructure security, analyzing the interdependencies of controls, communication and computation structures.

4.2 Automotive Electronics and V2V Networks

Modern vehicles already consist of a multitude of micro-controllers and multiple communication networks. To enable applications like Airbags, ABS and ESP, more and more sensors and actuators are being added to the system. With the addition of cameras to support the driver and elaborated entertainment systems, components are added to the mix that consume lots of network bandwidth. All of those components must be able to perform their assigned task, especially the components installed for safety and control reasons. Inflating an airbag too late because of other systems taking their share of the available resources is unacceptable.
Hands-free sets are already common place in modern vehicles, generally connected to the user's mobile. This combined with the trend towards compact mobile computing devices and the user being used to rich, but compared to the lifetime of a car, short-lived contents may lead to vehicle software updates becoming much more frequent than they are today. The same holds true for emerging vehicle-2-vehicle network technology and autonomous driving. In the same way technology evolves and the cultural and media environment changes, the software will have to develop and be updated.
Thus (future) vehicles may be seen as an interesting distributed computing system with properties comparable to CPSs and WSNs with an increasing need for regular software updates performed in a reliable, safe and secure way.

4.3 Reasons for Software Updates

Software must sometimes be updated. As already seen in above use cases, this may be necessary for different reasons. Mukhtar et al. [20] mention the following reasons: the requirements have changed; a bug or a security vulnerability must be fixed; update as part of the application development cycle such as integration of new features or improving existing ones. This includes updates for being interoperable with other updated or added systems and for adapting controlling functionality to new requirements. These are the very basic reasons for software updates commonly known by software developers.
But in the scope of CPS there is one more reason, why parts of a software might need to be added, removed or updated. Schroeder et al. [24] suggest distributing some software components depending on available resources like energy and local demand of a functionality. One can even think of dynamic redistribution

of software components, supported by local update mechanisms.

5 Challenges for Update Mechanisms

Various challenges make their appearance when trying to update CPSs and such. This chapter provides a list of challenges identified by various research groups. An extensive list is provided by Brown et al. [1] which serves as a source for the better part of the following list. This list does not claim to be exhaustive, but offers an overview over a wide range of challenges for a software update mechanism.

Planning Planning and simulating an update before its actual deployment can be used to analyze and optimize energy consumption, computation time etc. and provide necessary information to make trade-offs. One necessary prerequisite to be able to run a simulation is an appropriate model of the distributed system.
Deciding which software components will be deployed on which nodes may also be part of a software update mechanisms, taking quality goals and available resources into account like QARI by Horre et al. does [10].

Size Reduction and Performance Nodes of CPSs and WSNs often times run on limited energy resources, wireless transmission being one of the main energy consumers [12]. And even if energy is not a problem, transferring lots of data over the network may hinder the network in performing its current task. Thus minimizing the amount of transferred data for a software update makes sense. There are different ways of reducing the size of data that needs to be transferred to the nodes, all having advantages and disadvantages. Updates may only include some affected modules of the software, be differential on a more fine-grained code representation or comprise the complete software[3,12]. Different compression algorithms allow for another trade-off between computational complexity and data size, thus influencing the time it needs to distribute and perform the update. Brown et al. [1] note that the update package may as well contain some code supporting the update process like adapting stored data and therefore allow future-proofing and add flexibility.

Static Source Code Analysis Analysis of the static source code is necessary to compute the differences and thus creating the patch to update the software. It is also used to determine parts of the software where dynamic updates do not pose any problems. That way, one can determine what parts of a software can safely be replaced during the execution. [19]

Injection Strategies The injection strategy describes the way, the update is pushed into the network. [1] lists four exemplary strategies:
- Send update to a node which then acts as a base station.
- Send update to a base station.
- Send update to multiple seed nodes.

– Send update to each node individually.

Dissemination / Propagation Protocols The dissemination protocol is used to distribute the update across the network, starting from the base station or seed nodes. According to [1] this may be the "most advanced area of the field", and an energy-aware approach comparable to other WSN protocols might need to be taken. The protocol must take into account the usually lossy and low-bandwidth wireless network connections and provide reliable code dissemination at adequate energy consumption. Brown et al. also mention the possibility of an additional, parallel maintenance network for update control and dissemination.

The challenge is to provide a reliable yet efficient way to propagate the updates. Besides simple flooding there are a lot more possibilities like spanning trees and multi-path propagation as well as optimizations like local packet loss recovery.

Activation Control After successful dissemination, the pending software update must be activated. Primitive approaches are automated activation after a timeout or manual activation. More sophisticated approaches may include rule-based activation and a controlled order of activation within the network to ensure compatibility between individual nodes and reach certain service quality measures. Horre et al. [11] presented the middleware service QARI enabling enforcement and maintenance of service quality goals during software deployment. In case of CPSs the period during which an update process is performed has to be chosen carefully according to the physical system's characteristics.

While activation control is already challenging when focusing on a single node, it becomes even more challenging when considering the interactions and interdependencies between multiple nodes. Updating the software of one node may have effects on other nodes, for example in case of a shared state or when the node is part of a path within the network thus influencing the routing, or in the worst case, being essential for the interconnection of graph partitions.

Node Mobility Albeit more likely in a WSN, like for measuring sea currents, than in a CPS, nodes may be mobile and thus the network topology may change. This influences the code dissemination, planning and monitoring within the network and thus might need to be considered. Node mobility may also be a reason for a changed context which may require node reconfiguration because of changed requirements.

Continuity / Disruption During the update process it is necessary to replace code that is potentially running and has a certain state. Therefore it is beneficial to have an update mechanism that does not require software to be interrupted at all, or only for a short period of time. Update systems avoiding a disruption of software processes while updating are called dynamic update systems. Solutions range from replacing parts of the memory, running software components in parallel routing data to both versions to having two complete software images running at the same time[3]. Some dynamic update systems are presented by Kim et al. in [14] and they have developed

a system fitted to resource constrained computing systems of CPSs. In case a disruption is required, the activation control plays a decisive role to minimize the bad impact of the update process.

Dependability [1] only gives some hints on what dependability means for software update mechanisms. According to Brown et al. two key elements are integrity and compatibility checks of the received software update and monitoring the operation after the activation. Techniques developed in autonomous computing may also be feasible to increase the dependability and thus develop robust software updates according to Brown et al..

Monitoring Monitoring is necessary to assess the status of installed software in the network and to track faulty components. It may also act as an information source for activation control decisions.

Security The software update mechanism must be secured against unauthorized or altered updates. This includes authorization and key-distribution [1,16]. Brown et al. [1] mention secrecy as one aspect of security, though they do not clarify what secrecy is applied to in software update mechanisms. It might be about hiding the update and its contents from third parties, or hiding design decisions of various software components and only provide stable interfaces to limit the range of modifications.

It is obvious that these requirement come at increased energy, memory and computing resource consumption. Therefore there is a challenge to develop optimized algorithms and protocols with smaller footprints while preserving an adequate level of security [16]. There is a controversy about the applicability of asymmetric cryptography because of its higher computational complexity. But Haas et al. [6] were able to show empirically, that using asymmetric cryptography is feasible to exchange keys for use with symmetric cryptography as the energy consumption overhead is not significant compared to the networks energy consumption over a longer period of time. Additionally the special properties of CPSs bring the necessity to review attack vectors and analyze the interdependency between communication, computation and control of physical systems. Sridhar et al. [25] published a paper doing just that for the power grid.

Support for Very Small Nodes As further stated by [1] large WSNs are likely to consist of small nodes with very limited resources. Due to limited memory and high power consumption of writing to memory, incremental writing of updated code into code memory might be necessary.

Version Control To avoid version mismatch problems within and between nodes, there must be mechanisms to keep track of installed, downloaded and to be activated software versions. Therefore software configuration management is an integral part of a software update mechanism.

Version Coexistence Version coexistence was identified by Miedes et al. [19] to be very important for dynamic update mechanisms. It allows a software to work according to both versions' specifications.

Heterogeneity Support CPSs, WSNs and alike may consist of nodes varying in their hardware and installed software. The reason for this may for example be subsequent expansion of the network or specialized nodes for data

transport, data collection and so on. Thus different software versions and variants may have to be managed.

Energy Efficiency The update process must consider the available energy in the whole network and in a single node when figuring out how to perform the update. This also includes routing decisions and compression trade-offs. As energy resources are often limited, energy efficiency needs to be considered in the development of an update mechanism.

Recovery from Faulty Updates An update may fail due to various reasons, like incompatibility with the hardware, interconnectivity problems after the update, version conflicts etc.. Once the connection to a node is lost, maintenance requires physical access which may often times prove difficult to impossible. Therefore an autonomous recovery must take place in order to avoid the loss of a node, or even the whole network. In [2] Brown et al. present a software update recovery mechanism that does not only rely on watchdog timers and exception handlers, but also on a symptomatic approach of loss-of-control, which means that the management connectivity is lost. They also suggest a two phase approach, starting with a trial phase with increased monitoring activity for quick recovery, followed by an operational phase with reduced energy consumption. Recovery needs enough hardware resources to store a working image of the previous software version, or at least a minimal maintainable image.

Rollback Another challenge for update systems noted by Miedes et al. [19] is the ability to rollback an update. It may be used to recover from faulty updates, but also if the system maintainers decide that the previous version should be running again.

5.1 Focus on a Single versus Multiple Nodes

Further dividing the focus of analysis into single node and multi node perspectives on the computing system model helps to compare and understand different solution concepts. The combination of both perspectives allows for a holistic approach on the development of an efficient and effective software update mechanism.

Focusing on the single node, the focus is set to receiving and installing software updates while having the node attend its duties that can't be carried out by another node and avoiding the loss of a node in case of a faulty update. Limited memory, computing and energy resources play a major role in that case. Application code must be replaced and in case of an error, there should be some kind a recovery routine. If there is no recovery to a runnable state, physical access to the node is necessary to recover it, which may be expensive or even impossible.

A multi node perspective takes into account the possibility of allowing downtime of some parts of the system, while the rest of the nodes are able to keep certain quality goals, thus offering more freedom on when to perform an update. In order to do this, a global view on the network performance and service goals may be necessary. Another aspect is the interoperability of the nodes during the update process and the stability of the system. Typical questions are how the system

does perform in case of a faulty update or if one is at risk of loosing parts of the network due to missing links. This view also allows the analysis of software distribution and propagation patterns. Some software components may only run on a subset of nodes, thus optimizing the resource consumption amongst neighboring nodes or making use of nodes specialized for a certain task. The software propagation may be done using different patterns, like flooding, spanning trees or multiple paths.
Table 1 is a try to classify the main focus of the above challenges.

Table 1: Single vs. Multi Node Challenges

Single Node	**Multi Node**	**Both**
– Static Source Code Analysis – Dependability – Recovery from Faulty Updates – Rollback – Support for Small Nodes	– Injection Strategies – Dissemination / Propagation – Node Mobility – Security – Heterogeneity Support	– Planning – Size Reduction and Performance – Activation Control – Continuity / Disruption – Monitoring – Version Control – Version Coexistence – Energy Efficiency

6 Existing Implementations

Various, in most instances partial, software update mechanisms have been developed. The lack of a common solution is due to extensive resources usage and lack of generality according to Taherkordi et al. [26]. The risk of loosing a CPS or WSN while performing an update [2] combined with various shortcomings of existing solutions in code dissemination, memory management, update method, monitoring, recovery and so on lead to the development or refinement of software update mechanisms [26,20,11].
This section will compare some newer or recently refined solutions and compares them based on various dimensions of analysis.

6.1 Dimensions of analysis

In order to compare and classify existing software update mechanisms for CPSs and WSNs I propose the following dimensions:

– Location in Software Stack

– Real Time / Control of Physical Systems Supported
– Single / Multi Node Scope
– Service Quality Control
– Support for Software and Hardware Heterogeneity
– Platform Independence
– Update Granularity
– Dynamic Updates / Interruption

Additional to these dimensions of analysis, if and how the challenges listed in 5 are solved can be used to compare update mechanisms.

6.2 Compared Implementations of Software Update Mechanisms

The presented implementations were chosen because of their actuality, recurring mentions in the literature and because they do not only focus on a single challenge. A first overview over their characteristics is given in table 2, followed by further information about each system.

Table 2: Classification of Implemented Software Update Mechanisms

	RemoWare Middleware	**Deluge and Dynamic TinyOS**	**HERMES**	**QARI**
Location in Software Stack	middleware	os / middleware	os / middleware / application	middleware
Real Time / Control of Physical Systems Supported	yes	no	yes	?
Single / Multi Node Scope	multi / single	single	single	multi
Service Quality Control	no	no	no	yes
Support for Software and Hardware Heterogeneity	multiple applications	node specific images	yes	multiple applications
Platform Independence	OS-independent C code	partially	dynamic linking required	yes

Table 2: Classification of Implemented Software Update Mechanisms

	RemoWare Middleware	**Deluge and Dynamic TinyOS**	**HERMES**	**QARI**
Update Granularity	modular	monolithic, modular	modular	none
Dynamic Updates / Interruption	stateful	?	transparent, stateful	no

RemoWare Middleware The RemoWare middleware adds dynamic reconfiguration to the Remora component model for Contiki. The underlying model is a component model consisting of dynamically and statically linked components as per the developer's choice. Thus it allows a trade-off between flexibility, memory demand and performance. But as there does not seem to be too much of a performance hit, its more of a trade-off between flexibility and the memory demand for linking tables. This and other optimizations like in-place code updates show that RemoWare tries to minimize the necessary memory overhead to be able to run on very memory constricted notes. The components' services are described in a XML-file following the service component architecture (SCA) specification [22] exposing interfaces and events. RemoWare thus allows event-driven communication and thus very loose coupling of software components. If a dynamic component is removed, dynamic invocations are rerouted to a dummy function included in the middleware. In case of updating a component, it's state described in a standardized way can be transferred to the updated software component. If an update fails, so far they only considered restoring the overwritten code leaving other recovery topics for future research. [26]
The update dissemination protocol was not further investigated by Taherkord et al. as there has already been a lot of research in that direction. One new aspect they added is the use of code repositories locally on the nodes and server-based central ones. The local code repository avoids frequently transferring code via network if there are changes in the needed component stack, for example in case of context driven application switches. Server-based repositories provide information about each node's or node-group's software configuration and thus provide information about what can and has to be updated. To minimize the impact of the update process on a system's services, RemoWare tries to run updates when the system is not processing a request. As Contiki does not allow preemptive scheduling, the update process runs as an atomic task, which simplifies keeping the system in a safe state during the update. [26]
In a final section of [26] various update mechanisms like Deluge, FlexCup, OpenCOM and Runes are compared to RemoWare.

Deluge and Dynamic TinyOS TinyOS is a common operating system for low-power wireless devices. An early code dissemination and update mechanism for those nodes is Deluge, developed by Hui et al. around 2005. It has been developed to allow efficient dissemination of large data objects like system images across networks by making all nodes propagate the data to neighboring nodes. It supports node recovery by loading a so called Golden Image, a minimal system image for a network programmable node, upon failure. Heterogeneity within the network is enabled by letting the nodes decide which images they load. Thus each node can pick an appropriate image for its hardware and assigned task. [13]
Thus Deluge allows monolithic updating of network nodes running TinyOS [11]. Since 2005, lots of research has been done by various research groups to extends the capabilities of Deluge and improve its features. MDeluge by Zheng et al. improves the efficiency of data dissemination and the ability to restrict the dissemination to subsets of the network's nodes [28]. The efficiency upon packet loss has been improved by Rateless Deluge, to which authentication of network packages has been added by Law et al. [17] in Sreluge to secure the update mechanism against pollution attacks, a kind of denial of service attack by altering the packets.
The problem with the monolithic approach of Deluge is, that the distribution of complete system images imposes unnecessary resource usage in many cases of software updates despite being run in an environment with very restricted resources. But TinyOS by itself does not allow dynamic linking, thus replacing only parts of the systems is hard. FlexCup [18] is one solution to this problem, adding the ability to replace system and application components in TinyOS. Only differential updates are transmitted to the node, where a new system image is prepared that is afterwards installed by a bootloader that is able to write program memory. Thus a reboot is necessary and the update process is not really dynamic, as stated by Marron et al.. Another approach called Dynamic TinyOS is presented by Munawar et al. [21] that integrates transparently into TinyOS. Instead of removing the component model of TinyOS during compilation, they allow the user to define dynamically replaceable components. The updates can be disseminated with available protocols like Deluge, and are then loaded by a managing system component that takes care of linking and dynamic routing to location dependent code.

HERMES While primarily being developed as a monitoring and debugging system, HERMES offers an interesting approach to software updates. The HERMES system designed by Kothari et al. [15] and implemented for the SOS operating system on an exemplary basis leverages the dynamic linking process present in some node operating systems like SOS, Contiki, Mantis and Dynamic TinyOS. But it could also be integrated into other systems during compile time. The idea is to run as kind of a middleware between various interfaces of functional software units (function, modules etc.) and thus be able to monitor and control the data- and control-flows between them. One use case implemented by Kothari et

al. to demonstrate the possibilities offered by such a system is transparent updates. In order to update parts of a software, the data- and control-flow can be forked to both, the old and the new version. And once the new version shall be activated, the source of response is changed to the new version. Thus the update can be performed in a stateful and transparent way, and the same is possible for rollbacks in case of a failure. [15]
Other challenges for software update mechanisms may also be solved by a system like HERMES. Being able to monitor the system's internal communication potentially also allows for fault detection and optimized activation control over the software update process. The planning, dissemination, memory modification etc. will need to be taken care of by another system.

QARI Middleware QARI is a middleware service for quality-aware software deployment including quality-aware reconfiguration in WSNs. The description of quality goals takes place in a quality-aware deployment specification. It might for example contain a specification for a minimal coverage of 80% and a preferred coverage of 100% for the task of measuring the temperature in a room. Various specifications can be merged into local or a single node's specification trying to cover at least the minimal requirement of all specifications. Though conflict resolution has not yet been addressed by Horre et al. [11]. QARI is able to handle node failure, appearance of new nodes and status changes in deployed components. Node mobility is handled like node failures and appearance of new nodes. The computed deployment pattern takes various parameters into account, like the area covered by the node, network usage, the node's local context or the components already deployed onto it. Horre et al. [10] mention that MiLAN also addresses the topic of optimal sensor-subsets in WSNs to achieve service quality levels. The deployment itself is decentralized and is "delegated to local management entities near the deployment target"[10]. [11]
So far, QARI has been implemented for the LooCI component model on Sun SPOT and Contiki / AVR Raven, offering the possibility to easily add, replace and remove components of the software [11]. The goal of QARI is to allow the management of functionality independent from the update mechanism in use[10]. Thus it may be used in other environments, too.

6.3 Other Implementations

The solutions presented above have been selected to showcase exemplary solutions and approaches to different challenges and aiming towards a functioning system instead of focusing on specific parts of the system. Various authors have published overviews about related publications, some more recent overviews being provided by Horre et al. [10] and Taherkordi et al. [26].

7 Discussion

Looking at the previous chapter 6, it is obvious that *the* software update mechanism for WSNs, CPSs and alike does not exists. Table 3 gives an overview of the challenges tackled by the individual solutions.

Table 3: Addressed Challenges

	RemoWare Middleware	**Deluge and Dynamic TinyOS**	**HERMES**	**QARI**
Planning	● ○ ○	○ ○ ○	○ ○ ○	● ● ○
Size Reduction and Performance	● ● ●	● ○ ○	● ○ ○	○ ○ ○
Static Source Code Analysis	● ● ○	● ○ ○	● ● ○	○ ○ ○
Injection Strategies	● ○ ○	○ ○ ○	○ ○ ○	● ○ ○
Dissemination / Propagation Protocols	● ○ ○	● ● ○	○ ○ ○	○ ○ ○
Activation Control	● ○ ○	○ ○ ○	● ○ ○	● ○ ○
Node Mobility	○ ○ ○	○ ○ ○	○ ○ ○	● ○ ○
Continuity / Disruption	● ● ○	● ○ ○	● ● ○	● ○ ○
Dependability	○ ○ ○	○ ○ ○	● ● ○	○ ○ ○
Monitoring	○ ○ ○	○ ○ ○	● ● ●	● ● ○
Security	○ ○ ○	● ○ ○	○ ○ ○	○ ○ ○
Support for Very Small Nodes	● ● ●	● ● ○	○ ○ ○	○ ○ ○
Version Control	● ● ●	○ ○ ○	● ● ○	○ ○ ○
Version Coexistence	● ○ ○	○ ○ ○	● ● ○	○ ○ ○
Heterogeneity Support	● ● ○	● ● ○	○ ○ ○	● ○ ○
Energy Efficiency	● ● ○	● ● ○	● ○ ○	○ ○ ○
Recovery from Faulty Updates	● ○ ○	● ○ ○	● ○ ○	● ○ ○
Rollback	● ○ ○	● ○ ○	● ○ ○	● ○ ○

While RemoWare has a very holistic approach, dealing with lots of the challenges, the other solutions are more focused on some of them. But if one compares this table with table 2 one can think of combining some of the presented solutions, as they nicely complement each other in many topics. RemoWare is about efficient dissemination and configuration management. Deluge, being the default dissemination mechanism for TinyOS, is used for code dissemination. But while being rather inefficient as is, improved solutions based on Deluge have been presented by various research groups. FlexCup and Dynamic TinyOS are focused on modular updating of systems running with TinyOS. HERMES also is about modular updating, though its main focus is monitoring and debugging of distributed systems. But as it acts like a man in the middle between software components, it allows for transparent dynamic updates, version coexistence and so on. Last but not least, QARI has been presented, aiming to offer quality-aware reconfiguration in WSNs and CPSs. It provides ways of defining and monitoring service quality levels and thus can be used to minimize the impact of performing updates within the system.
Despite each system's different focus, some challenges seem to be hardly considered. Namely those are planning, injection strategies, dissemination, activation control, node mobility, dependability, security, version coexistence and rollback. The following list contains notes concerning some of them.

Dissemination / Propagation Protocols Dissemination has already been researched extensively according to [26]. Thus it has not been examined that much in recent research.

Activation Control Activation control requires knowledge about the system and it's current state. This is peculiarly important in CPSs when controlling physical systems that must be kept in a safe state. Most approaches rather seem to try to make the update process transparent (thus not or only insignificantly interfering with normal operation) instead of figuring out when to perform it. Finding the right moment is very dependent on the specific task performed by the WSNs or CPSs. In order to hit the perfect moment for an update, monitoring of the system's activity (e.g. with HERMES) and knowledge about its environmental influences and current context is necessary.

Node Mobility The most simplistic approach to support node mobility is handling it as loss and discovery as proposed by QUARI. It has also been considered in disseminations protocols as part of various routing protocols. Another aspect is context-based software reconfiguration that might be triggered by node mobility.

Security Despite being essential for safe and secure operation of distributed systems, this aspect of a software update mechanism has hardly been mentioned in the presented solutions. The problem is, that due to limited resources, available security technology may not be efficiently usable in the domain of WSNs and CPSs. Currently researchers discuss about the applicability of asymmetric encryption on resource constrained devices [6], thus still working on the fundamentals of secure communication.

Rollback Performing a rollback to return to a previous software configuration is hard enough, not even talking about resource constraints. Though all solutions do at least offer basic functionality for rollbacks by updating to older version of software components or restoring backup images.

Various solutions and proposals exist for most of the mentioned challenges and the software update mechanisms continue to integrate more and more of them. But it seems to be important to pursue both, integration and focused research, as solutions spreading over various challenges appear to be more superficial concerning the individual challenges. Though once there are applicable software update mechanisms, they can be improved upon, as shown by the example of Deluge with all its successors like MDeluge, Sreluge and so on.
Some of the challenges are tightly correlated and therefore require trade-off decisions. A typical issue is security versus energy efficiency and performance. Security always comes at a cost. It may cause communication overhead and is prone to be computationally intensive. Another problem are recovery and rollbacks on resource-constrained systems, as quite some memory capacity might be needed for those operations.
Other challenges nicely complement each other, like monitoring and activation

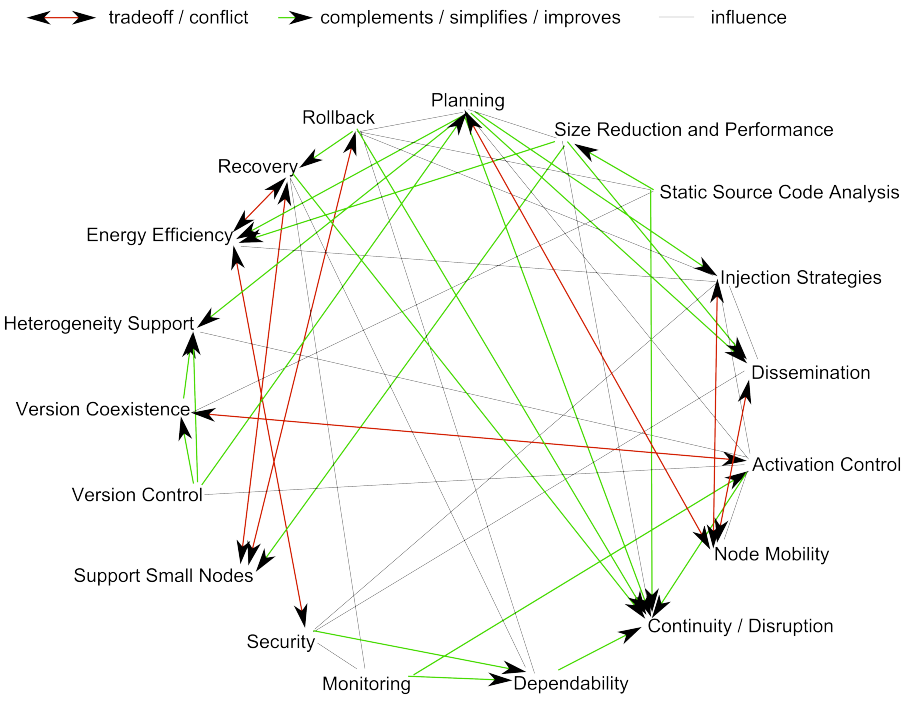


Fig. 1. Interdependencies of Challenges

control or version coexistence and heterogeneity support. And others are completely orthogonal thus having no or only indirect influence on each other like e.g. node mobility and rollback. In figure 1 a multitude of obvious trade-off situations and influencing relations between the challenges presented in section 5 are displayed.

Obviously many of them influence each other, be it in helping to solve other challenges or things to consider. Therefore in order to develop an effective and efficient software update mechanism, most of them have to be tackled. Hot topics seem to be activation control, continuity, planning and energy efficiency. Activation control has connections to many other challenges. Well timed activation of new software versions and components is essential for a good system performance. Starting an update process in the wrong moment or activating incompatible software versions will certainly degrade the systems service quality levels. Continuity goes into the same direction, but also is about service quality during and after the update process. It requires viable solutions for many of the other challenges to be reached. The planning component requires information about the system's state and allows for safe and optimized operation. Last but not least, energy efficiency is a permanent challenge when working with systems with limited energy resources. It is one key factor demanding specialized or optimized solutions for challenges already solved in other environments. Furthermore node mobility complicates the planning process, data injection and dissemination as it potentially leads to unknown or indeterministic node movements.

8 Conclusion

In the domain of WSNs and CPSs and alike, there does not exist a general software update mechanism. Despite being identified as a key challenge as early as 1978, research in this area has gained significant popularity only for about the last 15 years. On the one hand, many solutions from non-resource-constrained environments seem to be suitable for CPSs, but on the other hand efficient operation of networks consisting of very resource constrained nodes requires modified or special solutions. There are some solutions that already solve various challenges to a certain degree, while solutions of other challenges are still pending or need to be integrated into complete software update mechanisms. A driving factor for development certainly is the appearance of pervasive and ubiquitous computing solutions everywhere from the electrical power grid and vehicles to personal devices. And once there are long-lived systems one depends upon, fully functional software update mechanisms being able to handle the various challenges will be a decisive factor of success.

References

1. Brown, S., Sreenan, C.: Updating software in wireless sensor networks: A survey. Dept. of Computer Science, National Univ. of ... (2006), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.100.4510&rep=rep1&type=pdf
2. Brown, S., Sreenan, C.: Software update recovery for wireless sensor networks. Sensor Applications, Experimentation, and ... (2010), http://www.springerlink.com/index/mv5r265282767224.pdf
3. Chiang, M.L., Lu, T.L.: Two-Stage Diff: An Efficient Dynamic Software Update Mechanism for Wireless Sensor Networks. 2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing pp. 294–299 (Oct 2011), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6104540
4. Crow, M., Gill, C., Liu, F., McMillin, B., Niehaus, D., Tauritz, D.: Engineering the advanced power grid: Research challenges and tasks. In: RTAS 2006 Workshop on Research Directions for Security and Networking in Critical Real-Time and Embedded Systems. pp. 4–7 (2006), http://moss.csc.ncsu.edu/~mueller/crtes06/papers/008-final.pdf
5. Gill, H.: High Confidence Software and Systems : Cyber-Physical Systems (2008)
6. Haas, C., Wilke, J.: Evaluating the energy-efficiency of key exchange protocols in wirelesssensor networks. In: Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks - PM2HW2N '12. p. 133. ACM Press, New York, New York, USA (2012), http://dl.acm.org/citation.cfm?doid=2387191.2387210
7. Habermann, N.: Dynamically Modifiable Distributed Systems. In: Distributed Sensor Net Workshop. pp. 111–114 (1978)
8. Han, C.C., Kumar, R., Shea, R., Srivastava, M.: Sensor network software update management: a survey. International Journal of Network Management 15(4), 283–294 (Jul 2005), http://doi.wiley.com/10.1002/nem.574
9. Holdren, J.P., Lander, E., Jackson, S.A., Schmidt, E.: Report to The President and Congress - Designing a Digital Future: Federally Funded Research and Development in Networking and Information Technology. Tech. rep., Executive Office of the President - President's Council of Advisors on Science and Technology (2010)
10. Horré, W., Michiels, S., Joosen, W., Hughes, D.: QARI: Quality Aware Software Deployment for Wireless Sensor Networks. 2010 Seventh International Conference on Information Technology: New Generations pp. 642–647 (2010), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5501659
11. Horré, W., Michiels, S., Joosen, W., Hughes, D.: Advanced Sensor Network Software Deployment using Application-level Quality Goals. Journal of Software 6(4), 528–535 (Apr 2011), http://ojs.academypublisher.com/index.php/jsw/article/view/3296
12. Hu, J., Xue, C.J., Qiu, M., Tseng, W.C., Sha, E.H.M.: Algorithms to Minimize Data Transfer for Code Update on Wireless Sensor Network. Journal of Signal Processing Systems (Sep 2012), http://www.springerlink.com/index/10.1007/s11265-012-0689-z
13. Hui, J.: Deluge 2.0 - TinyOS Network Programming (Feb 2005)
14. Kim, D.K., Kim, W.t., Park, S.m.: DSUENHANCER : A Dynamic Update System for Resource-Constrained Software. In: International Conferences, CA and CES3 2011, Held as Part of the Future Generation Information Technology Conference, FGIT 2011, in Conjunction with GDC 2011, Jeju Island, Korea,

December 8-10, 2011. Proceedings. pp. 195–201. Springer Berlin Heidelberg (2011)
15. Kothari, N., Nagaraja, K., Raghunathan, V., Sultan, F., Chakradhar, S.: HERMES: A Software Architecture for Visibility and Control in Wireless Sensor Network Deployments. 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008) pp. 395–406 (Apr 2008), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4505490
16. Krontiris, I., Dimitriou, T.: Scatter – secure code authentication for efficient reprogramming in wireless sensor networks. International Journal of Sensor Networks 10(1-2/2011), 14–24 (2011)
17. Law, Y., Zhang, Y., Jin, J., Palaniswami, M., Havinga, P.: Secure Rateless Deluge: Pollution-Resistant Reprogramming and Data Dissemination for Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking 2011(1), 685219 (2011), http://jwcn.eurasipjournals.com/content/2011/1/685219
18. Marrón, P.J., Gauger, M., Lachenmann, A., Minder, D., Saukh, O., Rothermel, K.: FlexCup: A flexible and efficient code update mechanism for sensor networks. Wireless Sensor Networks 3868, 212–227 (2006), http://www.springerlink.com/index/766T1M555T558286.pdf
19. Miedes, E., Mu~noz-Escoí, F.D.: Dynamic Software Update (2012)
20. Mukhtar, H., Kim, B.W., Kim, B.S., Joo, S.S.: An efficient remote code update mechanism for Wireless Sensor Networks. In: MILCOM 2009 - 2009 IEEE Military Communications Conference. pp. 1–7. IEEE (Oct 2009), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5379862http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5379862
21. Munawar, W., Alizai, M.H., Landsiedel, O., Wehrle, K.: Dynamic TinyOS: Modular and Transparent Incremental Code-Updates for Sensor Networks. 2010 IEEE International Conference on Communications pp. 1–6 (May 2010), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5501964
22. OASIS: The Service Component Architecture (2007), http://www.oasis-opencsa.org/sca
23. Ramchurn, S.D., Vytelingum, P., Rogers, A., Jennings, N.R.: Putting the 'smarts' into the smart grid. Communications of the ACM 55(4), 86 (Apr 2012), http://dl.acm.org/citation.cfm?id=2133825http://dl.acm.org/citation.cfm?doid=2133806.2133825
24. Schröder-Preikschat, W., Kapitza, R., Kleinöder, J., Felser, M., Karmeier, K., Labella, T.H., Dressler, F.: Robust and Efficient Software Management in Sensor Networks. In: 2007 2nd International Conference on Communication Systems Software and Middleware. pp. 1–6. IEEE (Jan 2007), http://www.ccs-labs.org/bib/pdf/schroeder-preikschat2007robust.pdfhttp://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4268114
25. Sridhar, S., Hahn, A., Govindarasu, M.: Cyber–Physical System Security for the Electric Power Grid. Proceedings of the IEEE 100(1), 210–224 (Jan 2012), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6032699
26. Taherkordi, A., Loiret, F., Rouvoy, R.: Optimizing Sensor Network Reprogramming via In-situ Reconfigurable Components 9(2), 1–37 (2013)
27. Weiser, M.: Some computer science issues in ubiquitous computing. Communications of the ACM 36(7), 75–84 (Jul 1993), http://portal.acm.org/citation.cfm?doid=159544.159617

28. Zh, X., Sarikaya, B.: Code Dissemination in Sensor Networks with MDeluge. In: 2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks. vol. 00, pp. 661–666. IEEE (2006), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4068328

Methoden zur Messung der User Experience in Mobile Augmented Reality Anwendungen

Sergej Werfel
email@sergej-werfel.de

KIT - Campus Süd - TECO
Vincenz-Prießnitz-Str. 1, 76131 Karlsruhe, Germany
http://www.teco.edu

Zusammenfassung Die zunehmende Zahl leistungsstarker mobiler Geräte ermöglicht zahlreiche Einsatzmöglichkeiten für Augmented Reality. Bei der Entwicklung solcher Anwendungen steht der Benutzer und seine Zufriedenheit mehr im Vordergrund. Für die Einschätzung der Güte einer Augmented Reality Anwendung und die bessere Vergleichbarkeit von ihnen werden deshalb Methoden benötigt, die Benutzerzufriedenheit zu messen. Diese Arbeit erklärt und fasst die gängigen Evaluationsmethoden für die Usability von Anwendungen zusammen. Es wird weiterhin aufgezeigt, in wie weit diese speziell für mobile Augmented Reality Systeme geeignet sind und wie sie verwendet werden.

Keywords: User Experience, Usability, Augmented Reality, AR, Evaluation, Nutzerzufriedenheit, Messung

1 Einleitung

Die Verbreitung von mobilen Geräten hat in den letzten Jahren stark zugenommen. Die Unterhaltung und eine angenehme Bedienung gewinnen in diesem Umfeld immer mehr an Bedeutung, was ein Umdenken bei der Gestaltung von Benutzeroberflächen (User Interfaces, UIs) erfordert. Die Bedürfnisse und Wünsche des Benutzers verschieben sich in den Vordergrund. In der Entwicklung spricht man vom User Centered Design, bei dem sich der gesamte Entwicklungsprozess am Benutzer ausrichtet. Um unterschiedliche Interaktionstechniken oder Anwendungen bezüglich der Nutzerzufriedenheit vergleichen zu können, werden Verfahren benötigt, die User Experience (UX) zu messen.

Die aktuell verbreitete Geräte finden sich in unterschiedlichen Ausführungen wieder, wie z.B. als Smartphone oder Tablet-Computer. Zu den Gemeinsamkeiten zählen oft eine Kamera, ein hochauflösender Bildschirm, ein Internetzugang, mit dem Informationen passend zum Kontext des Benutzers abgefragt werden können, und ein leistungsstarker Prozessor.

Die eingesetzten Technologien der mobilen Geräte sind gut genug Bilder mit weiteren Informationen in Echtzeit zu verarbeiten und zu verknüpfen, um sie dem Benutzer in einer nützlichen Art und Weise darzustellen (Abb. 1). Bei

dieser Erweiterung der Realität (Augmented Reality) unterscheidet sich die Interaktion des Benutzers mit dem Gerät von dem gewohnten Umgang mit Computern (Point-and-Click). Sowohl die Beschaffenheit der Geräte (z.B. Touch-Oberflächen oder kleine Bildschirme), die Möglichkeit Kontextinformationen zu nutzen als auch die indirekte Interaktion mit der realen Welt des Benutzers beeinflussen sein Verhalten.

Abbildung 1. AR-App Layar für Android. Hier am Beispiel der Erweiterung von Printmedien um weitere digitale Inhalte. (Quelle: `https://play.google.com/store/apps/details?id=com.layar`)

Viele Evaluationstechniken sind für klassische PCs entwickelt worden und sind dadurch nur bedingt für die Untersuchung der Nutzerzufriedenheit bei mobilen Augmented Reality Anwendungen geeignet. Angepasste oder neue Methoden sind notwendig, die das Messen von Interaktionsparametern wie der Effektivität und der Effizienz erlauben.

Dieser Arbeit beschäftigt sich mit der Messung der User Experience. Bei der Anwendung der vorgestellten Mess- und Evaluationstechniken wird dabei insbesondere auf die erweiterte Realität eingegangen. Der Fokus liegt dabei bei Methoden, die für die Evaluierung von Anwendungen aus dem Bereich Mobile Augmented Reality (MAR) geeignet sind.

Im folgenden werden die relevanten Begriffe dieser Arbeit erläutert sowie der Umfang der Arbeit beschrieben.

1.1 Definition von Augmented Reality

Verbreitete Definition: Der Begriff Augmented Reality (AR) wird in der Literatur für unterschiedliche Konzepte verwendet. An manchen Stellen kann so beispielsweise bereits die Möglichkeit den nächsten Parkplatz mit einem Smartphone abzufragen als erweiterte Realität bezeichnet werden. Es zeichnet sich jedoch stark heraus, dass in den aktuellen Publikationen die Definition von Azuma[1] besonders oft verwendet wird, um Augmented Reality zu beschreiben. Diese Definition eignet sich gut für unterschiedliche Arten von AR und wird zum Zwecke der Stimmigkeit mit der gängigen Literatur auch in dieser Arbeit verwendet.

Ein Anwendung oder ein Konzept gilt im Folgenden als Augmented Reality, wenn es diese drei Eigenschaften erfüllt:

1. Es vereinigt reale und virtuelle Elemente
2. Es ist in Echtzeit interaktiv
3. Es wird in die 3D-Umgebung integriert

Der erste Punkt der Definition schließt die ausschließlich reale oder virtuelle Darstellung der Umwelt aus, auch wenn beim letzten die tatsächliche Umwelt nachempfunden wäre. Der zweite Punkt fordert eine Interaktionsmöglichkeit des Benutzers, was z.B. bei Filmen mit teilweise animierten Objekten nicht möglich ist. Der dritte Punkt verbietet beispielsweise Meldungen die einfach über einem realen Bild angezeigt werden. Es wird erwartet, dass das virtuelle Objekt in seiner Position von der Umwelt abhängig ist. So könnten die virtuellen Objekte z.B. von realen verdeckt oder verschoben werden oder sie könnten an ihnen haften und ihre räumliche Struktur annehmen.

Diese Definition kann beispielsweise an der Anwendung Layar für Android (Abb. 1) erläutert werden. Bei dieser Anwendung werden auf dem Bildschirm des mobilen Geräts zusätzliche Informationen über einem Printmedium (wie einer Zeitschrift) dargestellt. Damit sieht der Benutzer sowohl die Zeitschrift selbst, als auch zusätzliche digitale Informationen auf seinem Gerät. Dadurch ist der erste Punkt der Definition erfüllt. Weiterhin ist diese Anwendung interaktiv. So kann der Benutzer die Seiten umblättern, wodurch sich auch die digitale Information ändert oder auch durch einen Klick auf ein Bild (oder entsprechend eine Berührung des Bildes) das Bild vergrößern, ein Video dazu abspielen oder eine passende Webseite aufrufen lassen. Schließlich wird der digitale Inhalt in die reale 3D-Umgebung eingebunden. Hält man z.B. die Zeitschrift schief, passt sich die Darstellung entsprechend der Lage der Zeitschrift an, sodass der Eindruck entsteht, die digitalen Informationen würden an dem Printmedium haften. Die Erfüllung der drei Definitionspunkte zeigt, dass die Anwendung Layar der Definition von Azuma gerecht wird.

Reality-Virtuality Continuum: Die Begriffe Mixed Reality (MR) und Augmented Virtuality (AV) fallen oft bei der Erläuterung von erweiterter Realität und der Frage, in wie weit die Realität erweitert werden kann, ohne zu einer virtuellen Realität zu werden. Milgram et al [16] bieten eine graphische Einordnung (Abb. 2) dieser Begriffe in ein so genanntes Reality-Virtuality Continuum.

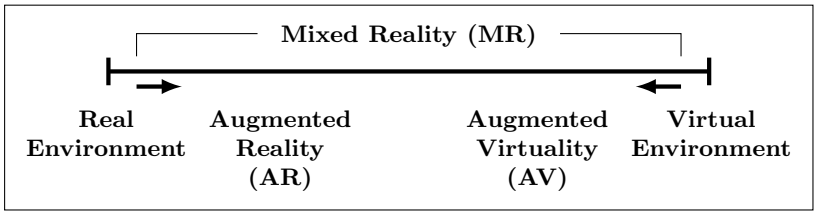


Abbildung 2. Milgrams Realität-Virtualität-Kontinuum nach [16]

Zwischen den Reinformen der realen und der virtuellen Umgebung gliedert sich die Mixed Reality ein. MR entsteht durch die Erweiterung einer Umgebung mit der Elementen der anderen. So handelt es sich bei AR um die Erweiterung der realen Umgebung um virtuelle Elemente und bei AV um die Erweiterung der virtuellen Umgebung um reale Elemente.

1.2 Definition von User Experience

Usability: Ähnlich wie bei Augmented Reality, kann auch die User Experience so definiert werden, wie sie in der Literatur allgemein anerkannt wird. So gilt im Bereich der Mensch-Maschine-Interaktion Jakob Nielsen als ein Pionier. Seine Definition von Usability [17] wird heute in vielen Arbeiten und Büchern zitiert. Er hebt hervor, dass Usability mehrdimensional ist und nicht durch einen einzelnen Wert ausführlich beschrieben werden kann. Folgende Dimensionen sind für die gute Benutzbarkeit eines Systems relevant:

1. *Learnability*: Das System sollte einfach und schnell zu erlernen sein.
2. *Efficiency*: Das System sollte effizient gestaltet sein, dabei soll besonders die Interaktion des Benutzers möglichst effizient sein.
3. *Memorability*: Der Benutzer sollte das System und seine Benutzung möglichst gut merken können, um nach einer Pause schnell wieder effizient arbeiten zu können.
4. *Errors*: Zum Einen sollten Fehler möglichst selten auftreten und zum Anderen sollten sie im Falle ihres Auftretens leicht zu beheben sein.
5. *Satisfaction*: Das System sollte für den Benutzer angenehm zu benutzen sein und somit sein subjektives Wohlbefinden ansprechen.

Nielsen hebt hervor, dass die Wichtigkeit dieser Punkte stark von der Aufgabe des Systems abhängt, das eine gute Usability haben sollte. So ist beispielsweise bei mobilen Geräten der Trend bemerkbar, dass in nur wenigen Sekunden nach dem ersten Start einer neuen Anwendung (App) entschieden wird, dieses Programm weiterhin zu behalten. In einer solchen Umgebung sollte man also einen hohen Wert auf Learnability legen.

Der Begriff User Experience wird heute und auch in dieser Arbeit als Synonym für Usability verwendet. Dazu kann der Kommentar von Wilson [27] zu Nielsens Definition herangezogen, bei dem hervorgehoben wird, dass sich der Begriff User Experience zu einem Nachfolger von Usability entwickelt hat.

DIN EN ISO 9241 ist zugleich eine deutsche, europäische und internationale Norm mit dem deutschen Titel „Ergonomie der Mensch-System-Interaktion“. Sie deckt viele Bereiche ab und ist damit oft der erste Einstiegspunkt bei der Lehre der Mensch-System-Interaktion.

Die Teile 110 („Grundsätze der Dialoggestaltung“) und 11 („Anforderungen an die Gebrauchstauglichkeit“) der Norm beinhalten wichtige Leit- und Grundsätze, die zur Gebrauchstauglichkeit eines Systems beitragen. Sie können als Heuristiken (vergleiche Abschnitt 3.1) oder als eine Definition von User Experience verwendet werden. So definiert Teil 11 der Norm die Gebrauchstauglichkeit als:

1. *Effektivität*: Wie gut wird eine Aufgabe erfüllt
2. *Effizienz*: Wie steht der Aufwand des Benutzer im Verhältnis zu der Lösung
3. *Zufriedenheit*: Wie ist das subjektive Empfinden des Benutzers gegenüber der Software

Für die Evaluation eines Systems können für die einzelnen Punkte der Gebrauchstauglichkeit einzelne Parameter abgeleitet werden, mit denen solche Systeme verglichen werden können. So kann für die Effektivität beispielsweise die Anzahl der korrekt erfüllten Aufgaben und für die Effizienz die durchschnittliche Bearbeitungszeit der Aufgaben verwendet werden.

1.3 Umfang und Struktur der Arbeit

Um Einschränkungen zu vermeiden, wird in dieser Arbeit keine einzelne Definition der Usability bzw. User Experience verwendet. Damit werden grundsätzlich keine Arbeiten ausgeschlossen, sofern sie sich eigenen Angaben nach mit der Gebrauchstauglichkeit beschäftigen.

Nicht beachtet werden hier Messtechniken der User Acceptance (UA). UA wird beispielsweise von Nilsson [18] beschrieben und beschäftigt sich in der Regel mit der Akzeptanz einer untersuchten Technologie oder eines Interaktionskonzeptes durch potenzielle Benutzer. UA bewertet damit allgemeine Konzepte (vergleiche [20]) und nicht konkrete Implementierungen oder Entwürfe, wie es bei der User Experience üblich ist. Die Nutzerakzeptanz wird aus diesem Grund in dieser Arbeit nicht behandelt. Der interessierte Leser wird jedoch darauf verwiesen, dass für die User Acceptance meistens Umfragen mit Fragebögen, wie im Abschnitt 4.2 erläutert, oder Wizard-of-Oz-Experimente, wie am Anhang des Abschnitts 4 skizziert, verwendet werden.

Diese Arbeit fasst gängige Evaluierungstechniken der User Experience zusammen. Sie soll dabei nicht als ein Survey dieser Techniken (vergleiche [7]), sondern als eine Übersicht der Messmethoden der UX sowie ihrer Beschreibungen gesehen werden. Dabei werden sowohl die üblichen UX-Evaluierungstechniken vorgestellt und ihre Verwendung bei der Augmented Reality aufgezeigt, als auch Verfahren, die explizit für AR- oder zumindest Mixed Reality-Anwendungen entwickelt wurden, sofern sie vorhanden sind.

Neben der Einleitung besitzt diese Arbeit vier weitere Abschnitte. Im Abschnitt 2 wird die verwandte Literatur und dabei insbesondere Arbeiten zur

Messung der allgemeinen Benutzbarkeit und der Benutzbarkeit im AR-Kontext vorgestellt. Außerdem wird Literatur vorgestellt, die sich thematisch dieser Arbeit ähnelt.

Der Abschnitt 3 stellt die gängigen Methoden der analytischen Evaluation vor. Zuerst wird dabei auf Untersuchungen eines Systems durch Usability-Experten eingegangen, die meistens in Form von Cognitive Walkthroughs und heuristischen Evaluationen durchgeführt werden. Weiterhin werden hier kognitive und das Verhalten beschreibende Modelle des Menschen vorgestellt, die bei der Analyse seiner Interaktion mit einer allgemeinen (GOMS, HMP) oder einer AR-Anwendung (Codein) helfen sollen.

Im Abschnitt 4 werden empirische Evaluationsmethoden vorgestellt. Als Beobachtungstechniken werden dabei besonders die Protokollierung aufgabenabhängiger Werte und das Thinking Aloud Protocol aufgeführt. Bei den Befragungstechniken werden Umfragen in Form von Fragebögen sowie Interviews vorgestellt. Außerdem wird hier kurz erläutert, wie die so gesammelten Daten ausgewertet werden können.

Zum Schluss der Arbeit findet man eine kurze Zusammenfassung der Ausarbeitung und eine Beschreibung dessen, was bei der Evaluierung der UX von AR-Anwendungen noch fehlt.

2 Verwandte Literatur

2.1 Evaluation der User Experience

Wie bereits im Abschnitt 1.2 beschrieben, gilt Jakob Nielsen als einer der wichtigsten Mitbegründer der User Experience. In seinem bekanntesten Buch [17] beschreibt Nielsen das Vorgehen bei der Entwicklung einer Anwendung unter dem Usability-Fokus. Nielsen führt zahlreiche Überlegungen und Leitsätze ein, die bis heute bei dem Entwurf und für die Evaluation von User Experience von hoher Bedeutung sind. In dieser Arbeit wird mehrfach auf die Veröffentlichung von Nielsen eingegangen[1]. So werden beispielsweise im Abschnitt 3.1 die Heuristiken von Nielsen eingeführt.

Weiterhin gibt es zahlreiche Bücher die den nutzerorientierten Entwicklungsprozess und die Evaluierung der Nutzungstauglichkeit von unterschiedlichen Systemen beschreiben. Als ein gutes Beispiel kann hier *Designing Interactive Systems* [3] von David Benyon genannt werden. Benyon beschreibt den gesamten nutzerzentrierten Entwicklungsprozess eines Systems und geht dabei auch auf die Evaluation der Usability ein. In diesem Buch fasst er viele aktuelle Forschungsergebnisse zusammen, bleibt dabei jedoch allgemein genug, dass die vorgestellten Methoden auch für neue Anwendungen, wie in Mobile Augmented Reality, anwendbar sind. In gesonderten Abschnitten wird außerdem auf unterschiedliche

[1] Für den interessierten Leser wird angemerkt, dass Nielsen auf der Webseite der Nielsen Norman Group Artikel zu aktuellen Usability-Themen veröffentlicht: `http://www.nngroup.com/`

Konzepte und ihre Besonderheiten eingegangen. So wird beispielsweise der Design von Webseiten aber auch der Entwurf von multimodalen und Mixed Reality Anwendungen behandelt.

2.2 Benutzbarkeit in AR

Das ausführliche Survey von Azuma [1] eignet sich sehr gut für die erste Einführung in Augmented Reality. In dieser Arbeit von 1997 geht Azuma auf die technischen Lösungen unterschiedlicher Probleme der erweiterten Realität ein, die heute noch relevant sind.

Im Jahr 2005 veröffentlichte Swan et al [25] ein Survey, in dem er Veröffentlichungen aus den Jahren 1992 bis 2004 untersuchte. Von den 266 AR-Publikationen fand er nur 21, die sich mit nutzerbasierten Experimenten beschäftigten. Im Jahr 2008 erweiterte Dünser [7] die Liste dieser Veröffentlichungen unter anderem um Arbeiten, die von ACM veröffentlich wurden. Die Anzahl der beachteten AR-Paper erhöhte sich hierdurch und durch die Ausweitung um die Jahre 2005-2007 auf 557. Die Anzahl der Veröffentlichungen, die eine Nutzerevaluation enthielten stieg dabei auf 161.

Zum Ende der Erstellung dieser Arbeit wurde das Buch *Human Factors in Augmented Reality Environments* [11] veröffentlicht. Es deckt viele der hier behandelten Themen ab. Das Buch deckt sowohl den praktischen Teil (wie die Richtlinien für den Nutzerschnittstellenentwurf in Kapitel 7) als auch den theoretischen Teil (wie die Kategorisierung des Begriffs *experience* in Kapitel 9) der Entwicklung von AR-Anwendungen. Das Buch bietet sich insbesondere für eine Vertiefung in die Mensch-Maschine-Interaktion unter dem Aspekt der erweiterten Realität an.

Für eine bessere Übersicht übernahm Dünser zum Einen die Klassifikation von Swan, führte zum Anderen auch eine eigene Klassifikation ein. Damit lässt sich sein Survey gut für die Suche nach einzelnen Veröffentlichungen aus unterschiedlichen Verwendungen der UX-Evaluationstechniken verwenden.

In einer Onlineumfrage sammelte Olsson [19] 90 Meinung zu den aktuellen MAR-Anwendungen. Seine Arbeit zeigt das Nutzerverhalten, z.B. dass die Anwendungen meistens aufgrund von Neugier bezüglich AR installiert wurden oder dass sie meistens in Stadtzentren eingesetzt werden. In einer späteren Arbeit [20] präsentiert Olsson eine weitere Umfrage zu den allgemeinen Erwartungen an zukünftige AR-Szenarien. Diese Arbeit beschäftigt sich also überwiegend mit der User Acceptance, deckt aber Wünsche und Erwartungen der Benutzer auf, die zum Beispiel bei der heuristischen Evaluation mitbeachtet werden können.

Die von Nilsson veröffentlichte Dissertation [18] bietet eine ausführliche Einführung und Vertiefung in die benutzerorientierte Entwicklung und Evaluation von AR-Anwendungen. Sie stellt dabei nicht nur zahlreiche Interaktionstheorien vor, sondern behandelt außerdem mehrere Nutzerstudien.

In dieser Seminararbeit werden einige Veröffentlichungen von Olsson zitiert, die er für seine Doktorarbeit [21] verwendet hat. Im gegensatz zu Nilsson setzt Olsson in seiner Arbeit den Schwerpunkt auf MAR-Dienste. Dabei konzentriert er sich auf User Experience sowie die Nutzererwartungen und stellt dabei seine

Untersuchungen zu bereits existierenden MAR-Anwendungen und zu potenziellen MAR-Szenarien vor. Für seine Studien benutzt er für die Datensammlung überwiegend Umfragen und Interviews.

3 Analytische Methoden

Bei den in dieser Arbeit vorgestellten Methoden wird zwischen analytischen und empirischen unterschieden (vgl. Kostaras und Xenos [13]). Die analytischen Methoden setzen sich zum Einen aus Regeln, Standards oder Heuristiken und zum Anderen aus theoretischen Modellen zusammen. Diese Methoden werden in der Regel von Usability-Experten angewendet und finden in früheren Phasen des Entwicklungsprozesses, wie zum Beispiel bei dem Entwurf oder der Evaluierung von Prototypen, ihre Anwendung. Diese analytischen Methoden werden im Folgenden zusammengefasst.

3.1 Untersuchung (Inspection)

Bei einer Inspection geht ein Usability-Experte die Anwendung durch und untersucht dabei die Einhaltung bestimmter Regeln bzw. versucht mögliche Probleme eines Benutzers vorherzusagen. Solche Untersuchungen können bei einer fertigen Anwendung aber auch bei Nutzungsszenarien oder Prototypen (z.B. Papierprototyp) durchgeführt werden. Man unterscheidet grundsätzlich zwischen Cognitive Walkthrough und Heuristiken.

Cognitive Walkthrough ist eine Technik, bei der ein Usability-Experte jeweils ein bestimmtes Szenario einer zukünftigen Anwendung simuliert. Dafür werden gerne Prototypen verwendet, da diese Untersuchung in der Regel in der frühen Phase der Entwicklung durchgeführt wird. Der Experte geht dabei die einzelnen Schritte einer Interaktionskette durch und fragt sich, ob z.B. der nächste Schritt für den Benutzer naheliegend ist oder ob er sich durch etwas irritieren würde.

Da diese Methode etwas über das Nutzerverhalten während eines Szenarios vorhersagt und nicht etwas über konkrete Interaktion mit einer fertige Anwendung, die meistens im Vordergrund steht, wird der Cognitive Walkthorugh kaum im Zusammenhang mit MAR-Anwendungen genannt. Nichtsdestotrotz wird diese Untersuchung empfohlen, da sie in frühen Stadien der Entwicklung Fehler oder Probleme aufdecken kann, die sich sonst erst später bemerkbar machen könnten.

Heuristische Evaluation nutzt wie der Name schon sagt eine Heuristik oder einen Regelsatz für die Evaluierung. Dabei werden einzelne Schritte eines Szenarios oder einzelne Abschnitte einer Anwendung überprüft, ob es die Ansprüche des Regelsatzes erfüllt. Die heuristische Evaluation wird in der Regel von einem Usability-Experten durchgeführt, wobei sich dafür auch potenzielle Benutzer eignen würden. Besonders populär für allgemeine Mensch-Maschine-Schnittstellen sind dabei die Heuristiken von Nielsen oder Shneiderman.

Nielsen definiert zehn Prinzipien, denen ein gutes Mensch-Maschine-System folgen sollte. Die ursprünglich von ihm genannten Kriterien [17] wurden im Laufe der Jahre überarbeitet und befinden sich in der neusten Version unter anderem auf der Homepage der Nielsen Norman Group [2]. Seine zehn Regeln sind besonders für graphische Oberflächen geeignet, können aber aufgrund ihrer Allgemeinheit auch für andere Systeme, wie für AR-Anwendungen, verwendet werden. Nielsen schlägt vor, eine zu testende Anwendung auf folgende Punkte zu untersuchen:

1. Sichtbarkeit des Systemzustandes
2. (konzeptionelle) Übereinstimmung zwischen dem System und der realen Welt
3. Kontrolle durch den Benutzer und seine Freiheit (z.B. Rückgängig machen)
4. Konsistenz und Einhaltung von Standards
5. Fehlervermeidung
6. (Wieder-)Erkennen statt Erinnern (Recognition rather than recall)
7. Flexible und effiziente Nutzung
8. Ästhetischer und minimalistischer Entwurf
9. Helfe Nutzern Fehler zu erkennen, sie zu prüfen und zu korrigieren
10. Zusätzliche Hilfe und Dokumentation

Shneiderman hat ebenfalls Regeln für Interaktive Systeme aufgestellt [24], die auch die *Acht goldene Regeln von Ben Shneiderman* genannt werden[3]. Die zuletzt veröffentlichten Regeln lauten:

1. Strebe nach Konsistenz
2. Biete häufigen Benutzern Abkürzungen an (shortcuts)
3. Biete informative Rückmeldungen an
4. Entwerfe Dialoge mit ersichtlichem Ende
5. Biete einfache Fehlerbehandlung
6. Biete einfache Rücksetzmöglichkeiten
7. Erlaube dem Benutzer laufende Aktionen zu kontrollieren und neue zu starten
8. Halte die Belastung des Kurzzeitgedächtnisses gering

Die Regeln von Shneiderman sind denen von Nielsen ähnlich. So wir beispielsweise *Fehlervermeidung* und *Helfe Nutzern Fehler zu erkennen, sie zu prüfen und zu korrigieren* von Nielsen unter *Biete einfache Rücksetzmöglichkeiten* bei Shneidman zusammengefasst.

Der Einsatzzweck bestimmt grundsätzlich, welche Heuristik benutzt wird. Dabei sind die von Shneiderman und Nielsen besonders beliebt. Im Allgemeinen kann jedes Unternehmen oder jeder Entwickler eigene Regeln aufstellen oder für die Evaluierung allgemein anerkannte Richtlinien verwenden. So können unter Anderem auch die Grundsätze der Dialoggestaltung nach dem ISO 9241 als eine Heuristik verwendet werden. Zu diesen Grundsätzen zählen:

[2] `http://www.nngroup.com/articles/ten-usability-heuristics/`

[3] In aktueller Fassung online zusammengefasst unter `http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html`

1. Aufgabenangemessenheit
2. Selbstbeschreibungsfähigkeit
3. Erwartungskonformität
4. Lernförderlichkeit
5. Steuerbarkeit
6. Fehlertoleranz
7. Individualisierbarkeit

Wie man sieht, decken sich auch diese Grundsätze mit den Heuristiken von Nielsen und Shneiderman. Die Gemeinsamkeiten der Autoren können dafür als Zeichen gesehen werden, dass die genannten Grundsätze hohe Relevanz für die Praxis haben. Aus diesem Grund können beide Heuristiken gut für die Evaluation von Interaktionstechniken verwendet werden.

An dieser Stelle wird weiterhin darauf hingewiesen, dass auf Kosten der Übersichtlichkeit und der praktischen Anwendbarkeit die oben beschriebenen Regelsätze nur die wichtigsten Grundsätze nennen und sie teilweise unter einzelnen Begriffen zusammengefasst sind. So kann, als ein etwas größeres Beispiel, die Liste von Bruce Tognazzini[4] genannt werden, die 16 unterschiedliche Aspekte beschreibt.

Allgemein anerkannte Heuristiken für AR-Anwendungen konnten sich bis jetzt noch nicht etablieren. Dennoch kann die heuristische Evalaution für Augmented Reality verwendet werden. Dazu bieten sich allgemeine Heuristiken, wie die von Nielsen und Shneiderman, aber auch Tipps und Empfehlungen für den Entwurf von AR-Anwendungen an. So stellt Olsson [21] nicht nur sechs Hinweise auf, die man beim Entwurf von MAR-Anwendungen beachten sollte, sondern nennt auch elf Implikationen für die Interaktionen in solchen Anwendungen. Die meisten dieser Implikationen könnten in entsprechende Regeln bzw. Heuristiken umgewandelt werden (z.B. *Erlaubt die Anwendung eine effiziente Steuerung der Realitäten?*).

Anzumerken ist, dass Nilsson in ihrer Arbeit [18] hervor hebt, dass in den meisten Usability-Richtlinien (wie von Nielsen [17] und Shneiderman [24]) der Kontext des Benutzer nicht beachtet wird oder sie sich nur mit graphischen Benutzerschnittstellen (Graphical User Interface, GUI) beschäftigen. Damit eignen sie sich nur teilweise für die Evaluierung von Augmented Reality-Anwendungen. Für die Benutzung dieser Richtlinien sollte dieser Kritikpunkt also beachtet und diese Heuristiken um die entsprechenden Punkte erweitert werden.

3.2 Modelle

In der Usability-Forschung werden oft Modelle des Menschen verwendet, um sein Verhalten vorhersagen zu können und so Interaktionsmethoden zu evaluieren bevor sie implementiert werden. Im Folgenden werden zwei der bekanntesten Modelle für allgemeine Mensch-Maschine-Interaktion und ein Modell, dass für die

[4] Englische Version unter: `http://www.asktog.com/basics/firstPrinciples.html` Die deutsche Version unter: `http://meiert.com/de/publications/translations/asktog.com/firstprinciples/`

erweiterte Realität besser geeignet zu sein scheint, vorgestellt. Die vorgestellten Modelle und viele weitere versuchen in erster Linie die Dauer der Benutzeraktionen vorherzusagen. Es existieren auch weitere Modelle, die beispielsweise beachten, welche Informationen im Kurzzeitgedächtnis abgelegt sind. Diese werden jedoch bei der Usability-Evaluation eher selten verwendet.

Die Modelle, wie die hier vorgestellten, werden in der Regel nach empirischen Beobachtungen aufgestellt. Auch die Werte, die für die Vorhersage von menschlichen Aktionen durch diese Modelle, wurden empirisch ermittelt. Es ist allgemein bekannt, dass Menschen sich in auch in annähernd gleichen Situation unterschiedlich Verhalten können. Besonders die unbewussten Vorgänge können so das menschliche Verhalten stark beeinflussen. Aus diesem Grund muss beachtet werden, dass die Vorhersagen dieser Modelle nur eine Näherung darstellen. Besonders für GOMS und HMP gibt es inzwischen zahlreiche Erweiterungen, die kleine Änderungen, andere Grundwerte oder weitere Anpassungen vorschlagen. Der interessierte Leser ist gerne dazu eingeladen weitere Modelle für seine Entwicklungen zu betrachten.

GOMS steht für Goals, Operators, Methods und Selection Rules und ist ein Modell der informationellen Verarbeitung des Menschens. Das von Card et al [4] vorgestellte Konzept unterteilt eine Gesamtaufgabe in mehrere Ziele. Die Operatoren beschreiben einzelne Aktionen, die der Benutzer ausführen kann, um die Ziele zu erreichen. Kombinationen von diesen Operatoren und Zielen werden Methoden genannt und beschreiben einzelne Wege zu Erfüllung größerer Teilaufgaben. Da es in der Regel unterschiedliche Methoden gibt, eine Aufgabe zu erfüllen, geben die Selektionsregeln Auskunft darüber, welche Methoden gewählt werden.

GOMS erlaubt es komplexere Interaktionen des Benutzers mit einem System zu erklären. So ist es möglich auch längere Interaktion eines Benutzers mit einer AR-Anwendung oder einem Prototyp vorherzusagen. Von GOMS gibt es inzwischen viele unterschiedliche Versionen wie GOMSL oder CPM-GOMS. Eine Version, die sich gut für AR-Interaktion nutzen ließe, heißt Codein und wird weiter unten erläutert.

Der Human-Model-Processor (HMP) wurde 1986 ebenfalls von Card et al [5] vorgestellt. Dabei wird der Mensch als ein technisches System modelliert. So stellt eine Speicherhierarchie das Gedächtnis des Menschen dar (siehe Abbildung 3). Card et al unterscheiden zwischen dem Langzeitgedächtnis, dem darin liegenden Arbeitsgedächtnis und den zwei Gedächtnissen für visuelle und auditive Bilder, die wiederum im Arbeitsgedächtnis eingeordnet werden. Für die Verarbeitung von Informationen hat der HMP drei Prozessoren. Der kognitive Prozessor verarbeitet die Information im Arbeitsgedächtnis, der perzeptuelle Prozessor die ankommende Information der Sinne und der motorische Prozessor steuert die Bewegungen, die im Arbeitsgedächtnis aktiviert werden.

Card et al haben mit unterschiedlichen Experimenten die Parameter der Gedächtnisse und der Prozessoren ermittelt. Die empirisch ermittelten Werte

geben Auskunft über die Halbwertszeiten der Informationen in den einzelnen Gedächtnissen sowie ihre Speicherkapazität und die Taktdauer der Prozessoren. Mit diesen Werten und einigen Verarbeitungsregeln ist es möglich eine Zeit anzugeben, in der der Benutzer eine bestimmte Aufgabe erfüllen kann.

Da die kognitive Leistung des Menschen von seinem aktuellen Befinden und sonstigen Belastungen abhängt, sind die Werte, die mit HMP ermittelt werden nur ungefähre Angaben. Sie eignen sich dennoch besonders für den Vergleich unterschiedlicher Interaktionsmethoden, unter anderem weil sie verglichen mit anderen Modellen besonders genau und dennoch einfach zu bestimmen sind.

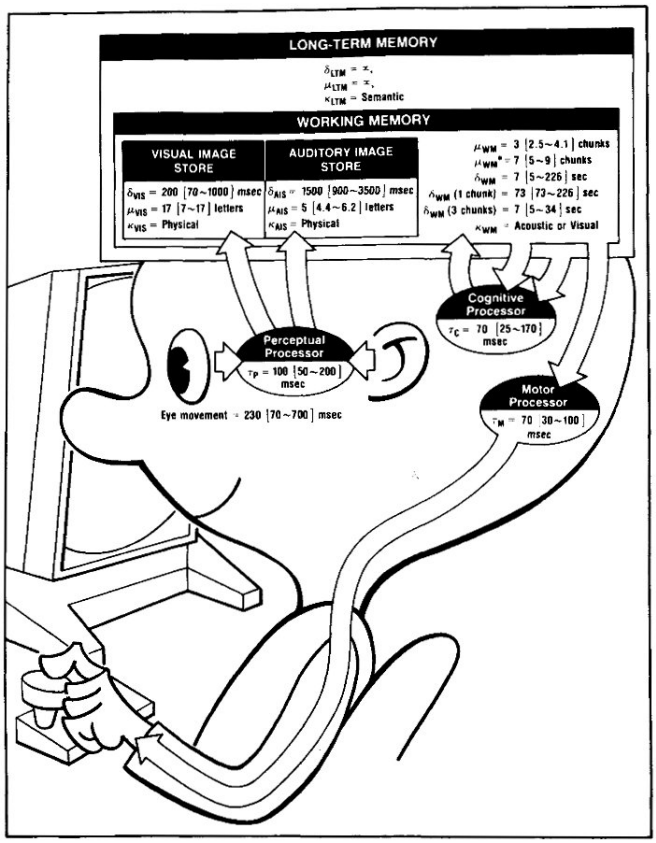


Abbildung 3. Human Modell Processor (aus [5])

Codein steht für COgnitive Description and Evaluation of INteraction und ist ein von Christou et al [6] vorgestelltes Modell für die Evaluation von rea-

litätsbasierten Interaktionstechniken (RBI) [5]. Codein erweitert GOMS und erlaubt unter anderem die Evaluation der Interaktion mit realen Objekten.

Codein zeichnet sich dadurch aus, dass es eine graphische Notation benutzt. Die zwei wichtigsten Vorteile gegenüber GOMS und ähnlichen Modellen sind die Unterstützung von parallelen Aktionen und die Hervorhebung der Information, die zur Ausführung der Aufgabe benötigt wird. Codein modelliert hierarchische Wissenszustände und unterscheidet zwischen prozeduralen und deklarativen Wissenseinheiten. So weiß der Modellierer nach der Erstellung eines gültigen Codein-Modell welche Informationen der Benutzter für die Ausführung der Aufgabe wissen sollte (z.B. deklarative, dass die Bewegung der Maus mit der des Zeigers gekoppelt sind, oder prozedurale, wie man die Maus dazu bewegt).

Die Entwickler dieses Modells nennen zwei Nachteile von Codein. Zum Einen kann die graphische Repräsentation von verschachtelten Wissenszuständen groß und etwas unübersichtlich werden und zum Anderen gibt es aktuell keine Werkzeugunterstützung für Codein, wobei die Autoren versuchen möchten dies zu beheben.

Bis jetzt konnte keine Veröffentlichungen gefunden werden, die Codein für Augmented Reality oder speziell für MAR-Anwendungen bzw. Untersuchungen zur MAR-Interaktion verwenden, obwohl die Erstveröffentlichung davon in 2007 stattgefunden hat. Nichtsdestotrotz scheint es eine vielversprechende Möglichkeit der Modellierung menschlichen Verhaltens zu sein. Außerdem ist bis jetzt keine Alternative bekannt, die für den Augmented Reality Bereich besonders geeignet zu sein scheint.

4 Empirische Methoden

Bei der empirischen Evaluation kann im allgemeinen zwischen zwei unterschiedlichen Experimenten unterschieden werden. Liegt bereits eine Anwendung oder ein Prototyp vor, kann ein normales Experiment mit Benutzern oder benutzerähnlichen Teilnehmern durchgeführt werden. Dabei wird ein (halb-)fertiges System direkt evaluiert. Der Benutzer verwendet also direkt das Entwicklungsergebnis.

Bevor eine Anwendung entwickelt wird, wird gerne eine Untersuchung gemacht, ob die neue Anwendung den Benutzern gefällt oder ob das gewählte Konzept einen Benutzer sinnvoll bei der Aufgabenerfüllung unterstützt. Besonders für neue und innovative Interaktionsmethoden, wie bei AR-Anwendungen, wird deswegen oft ein sog. Wizard-Of-Oz-Experiment durchgeführt. Dabei glaubt der Teilnehmer mit einer fertigen Anwendung zu interagieren. In Wirklichkeit wird

[5] Den Begriff Realitätsbasierte Interaktion(RBI) führte einer der Mitautoren in einer weiteren Veröffentlichung [12] ein. Jacob et al bezeichnen mit RBI die Erweiterung der üblichen Interaktionstechniken, wie Konsole oder WIMP (windows, icon, menu, pointing device), um die Interaktion mit realen und virtuellen Objekten, die von einem System interpretiert wird. Damit kann auch die Interaktion im AR-Kontext als eine Realitätsbasierte Interaktion bezeichnet werden.

die Interaktion von einem Menschen interpretiert und die Anwendung manuell in den entsprechenden Zustand gebracht.

Die in den Abschnitten 4.1 und 4.2 vorgestellten Beobachtungs- und Befragungstechniken können in der Regel bei beiden Experimenttypen verwendet werden. Der Abschnitt 4.3 beschreibt anschließend kurz, wie die Daten ausgewertet werden können, die durch die vorher vorgestellten Techniken erhoben wurden.

4.1 Beobachtungstechniken

Bei den Beobachtungstechniken handelt es sich um Methoden, mit denen Daten während der Experimentdurchführung erhoben werden. Die wichtigsten Techniken sind die Messungen aufgabenabhängiger Werte, wie die Anzahl der bearbeiteter Aufgaben oder gemachter Fehler, Messung der Biosignale des Probanden, wie die Atemfrequenz oder den Stresslevel, sowie das subjektivere Thinking Aloud Protocol, dass die Gedanken des Probanden nutzt.

Aufgabenabhängige Werte sind bei der Evaluierung von Anwendungen sehr beliebt. Die ermittelten Zahlenwerte erlauben den genauen Vergleich zweier Systeme. Einige Beispiele für solche Werte sind: die Anzahl der erfüllten oder nicht erfüllten Aufgaben, die Anzahl der gemachten Fehler, die Dauer der einzelnen Aufgabenerfüllungen, die Dauer der Suche nach der nächsten Aktion usw. Diese Werte können in der Regel auch kombiniert werden. So kann die durchschnittliche Aufgabenbearbeitungszeit aller Benutzer einer Anwendung oder die Geschwindigkeit der Aktionen (Schritte pro Minute) berechnet werden.

Obwohl diese Werte innerhalb einzelner Veröffentlichungen einen sinnvollen Vergleich erlauben, sind sie meistens nur bedingt dazu geeignet eine eigene Anwendung mit fremden Publikationen zu vergleichen, da sich Grundwerte oder Rahmenbedingungen, wie der Schwierigkeitsgrad einzelner Aufgaben, unterscheiden können.

Zu den aufgabenabhängigen Werten zählen z.B. die Versuchsdauer oder die Fehlerrate. Rohs et al [23] nutzen diese Werte für den Vergleich dreier Navigationsmethoden mit mobilen Geräten auf einer Karte.

Bei dem Vergleich von klassischen mit AR-Anleitungen (über ein Head Mounted Display und monitorbasiert) bestimmten Tang et al [26] die Aufgabenleistung (task performance) der Probanden. Die Leistung setzte sich dabei zusammen aus der Zeit, die für die Aufgabenerfüllung benötigt wurde, und der Genauigkeit, die sich als die Fehleranzahl messen lies. Tang et al beachteten dabei nicht nur die gemessenen Werte sondern führten auch eine statistische Analyse durch (vgl. Abschnitt 4.3), um weitere Zusammenhänge zu bestimmen.

Messung der Biosignale des Probanden kann ebenfalls den Beobachtungstechniken zugerechnet werden. Hierbei versuchen die Versuchsleiter über die Biosignale Aufschlüsse über den inneren Zustand des Probanden zu bekommen.

Aufgrund der Einfachheit ihrer Messung sind die Atemfrequenz und der Puls beliebte Messgrößen. Dabei bedeuten eine höhere Atemfrequenz und ein höherer Puls in der Regel eine innere Aufregung des Probanden. Durch den Vergleich der Messwerte können zum Beispiel zwei Interaktionstechniken darauf verglichen werden, welche von ihnen einen höheren Stresslevel verursacht. Die u.a. aus der Atemfrequenz und dem Puls ermittelte Aufregung kann dabei als ein Indikator für den Stresslevel bzw. für die geistige Belastung benutzt werden. Beispielsweise verwenden Jarvis et al [9] viele der in diesem Abschnitt genannten Biosignale, um die Belastung eines Autofahrers durch zusätzliche Aufgaben zu messen.

Grundsätzlich gibt es sehr viele unterschiedliche Vorgehen, wie und welche Biosignale man misst. Im Folgenden werden einige Beispiele für die messbaren Biosignale genannt:

- Atemfrequenz, Puls und Hautleitwert, Messung der Aufregung
- EEG (Elektroenzephalogramm), Messung der Gehirnaktivität oder einzelner Areale
- Augenaktivität (Elektrookulogramm, Video-Okulographie u. ä.), Messung der Aufmerksamkeit oder Konzentration (Focus of attention)
- Video-Überwachung der Gestik, Mimik oder Körperhaltung, Messung der Stimmung

Bei der Wahl eines Biosignals oder seiner Messung muss besondere Aufmerksamkeit der Aussagestärke der gemessenen Werte geschenkt werden. So können die meisten gemessenen Augenbewegungen gut beschreiben, was der Proband besonders lange angeschaut hat und damit welche graphischen Elemente gut und welche weniger gut platziert sind. Andere Messtechniken wie das EEG oder die Videoüberwachung der Mimik befindet sich noch selbst im Forschungsstadium. Eine Entscheidung, welche Anwendung mehr Spaß bereitet, kann mit ihnen nur wenig allgemeingültig getroffen werden.

Im Bereich der AR-Forschung werden Biosignale aktuell noch wenig eingesetzt. Für den Vergleich der Tauglichkeit von Interkationstechniken in Stresssituationen können z.B. die Atemfrequenz oder der Puls benutzt werden. Die Augenaktivität kann dagegen dafür genutzt werden zu untersuchen, wohin der Proband während des Experiments schaut. So kann es für manche Anwendungen beispielsweise interessant sein, welchen Anteil der Zeit der Proband auf das mobile Gerät und welchen Anteil er in seine reale Umwelt schaut.

Thinking Aloud Protocol Im Gegenteil zu den oben beschriebenen direkt beobachtbaren Werten bietet das Thinking Aloud Protocol (TAP) die Gelegenheit die Gedanken des Probanden besser nachzuvollziehen. TAP besteht darin, dass der Benutzer während der Benutzung der Anwendung versucht möglichst viel zu kommentieren. Insbesondere wird er darum gebeten seine Entscheidungen und Fragen bzw. Unklarheiten mitzuteilen. Das Gesagte wird entweder von einer Kamera oder Mikrophon aufgezeichnet oder von einem Protokollanten mitgeschrieben. Das Protokollieren von Hand hat den Vorteil, dass der Protokollant in manchen Situationen vorgegebene Fragen stellen kann, um den Probanden an das *laute Denken* zu erinnern.

Das Thinking Aloud Protocol wurde bereits 1982 von Lewis in einem Technical Report von IBM erwähnt[6]. In einem späteren Buch [14] beschreiben Lewis und Rieman den aufgabenzentrierten Entwurf von Benutzerschnittstellen. Dabei erläutern sie auch ausführlich das TAP und gehen nicht nur auf das Verfahren ein, sondern heben gleich hervor, worauf man bei der Durchführung achten sollte. Zum Beispiel ist es sehr wichtig dem Probanden zu verdeutlichen, dass nicht er sondern das Systems evaluiert wird.

Im Bereich der Augmented Reality wird der TAP gerne verwendet, um die ersten Probleme einer Anwendung zu finden. Da es jedoch nur einige Schwierigkeiten der Benutzer in der Interaktion mit dem entwickelten System aufdeckt und keine Vergleichswerte bietet, wird das Thinking Aloud Protocol meistens nur als eine verwendete Methode genannt und die Ergebnisse nicht weiter ausgeführt. Als Beispiel können die Arbeiten von Balabuer et al [2] und Liarokapis et al [15] herangezogen, wo keine konkreten Ergebnisse des TAP genannt werden, es jedoch benutzt wurde.

4.2 Befragungstechniken

Im Gegensatz zu den meisten Beobachtungstechniken werden die Befragungen meistens nach einem Experiment durchgeführt. Dabei wird zwischen Umfragen, bei denen der Proband Fragebögen ausfüllt, und Interviews, bei denen die Fragen von den Experimentatoren oder einem Interviewer gestellt werden, unterschieden.

Umfragen Umfragen bilden bei der Evaluation von AR-Anwendungen das am häufigsten genutzte Datenerhebungsmittel. In der Regel wird nach einem durchgeführten Experiment (z.B. mit der fertigen Anwendung oder einem Mock-Up) dem Benutzer ein Fragebogen mit Fragen zu der Anwendung vorgelegt.

Sehr beliebt sind die Likert-Skalen, bei denen der Befragte in Abstufungen zwischen zwei Präferenzen wählen kann. Dabei kann zwischen zwei Aspekten (z.B. bequemere Handhabung mit System A oder B) oder zwischen zwei Ausprägungen eines Aspekts (z.B. die farbliche Gestaltung war gut oder schlecht) gewichtet werden. Der Vorteil von Likert-Skalen ist, dass aggregierende Aussagen möglich sind (z.B. *Die Durchschnittliche Bewertung der Textlesbarkeit lag bei 4,7 (wobei 1 schlecht und 5 gut entspricht)*).

Bei dem Entwurf der Skalen sollten mehrere Punkte beachtet werden. So können viele Ausprägungen (z.B. 100) den Benutzer mangels der Genauigkeit der eigenen Einschätzung überfordern oder zu wenige (z. B. 3) zu sehr einschränken. Die gängige Anzahl liegt dabei etwa zwischen 5 und 7.

Auch die Entscheidung eine gerade Anzahl an Ausprägungen zu wählen, sollte abhängig vom Experiment getroffen werden, da dem Befragten so die Möglichkeit der neutralen Antwort vorenthalten und eine Präferenz erzwungen wird. Neben

[6] Ursprüngliche jedoch nicht mehr verfügbare Quelle: Lewis, C. H. (1982). Using the "Thinking Aloud" Method In Cognitive Interface Design (Technical report RC-9265). IBM.

der neutralen Antwort kann dem Benutzer auch die Möglichkeit der Vorenthaltung eingeräumt werden, zum Beispiel in Form eines anzukreuzenden Kästchens.

Neben den Abwägungen des Befragten bei den Likert-Skalen können bei den Fragebögen weitere Informationen abgefragt werden. So kann der Benutzer zum Beispiel bei spezifischen Fragen (wie *Welches der Folgenden Technologien haben Sie bereits genutzt: A, B, C, D*) oder Fragen mit Freitextantworten (wie *Wie würden Sie das System einsetzen?*) Informationen über sich angeben.

Eine kurze Einführung zu den Fragebögen und den Skalen bieten beispielsweise Nielsen [17] und Benyon [3].

Der NASA-TLX (Task Load Index) wurde für die Luft- und Raumfahrt entwickelt und ist bei der Usability-Evaluation beliebt. Der Fokus dieser Umfrage liegt bei der Messung der Belastung des Probanden. Weil sie sehr allgemein gehalten ist und keine anwendungsspezifischen Fragen enthält, bietet sie sich für unterschiedliche Anwendungsgebiete, wie graphische Benutzeroberflächen, Sprachsteuerung aber auch für Mixed und Augmented Reality, an.

Der Task Load Index wird in zwei Schritten ausgeführt. Im ersten Teil bewertet der Proband die von ihm verspürte Anforderung während des Experiments. Dabei bewertet er die folgenden sechs Punkte auf einer Skala mit 20 Ausprägungen:

- Die geistige Anforderung war... (niedrig bis hoch)
- Die körperliche Anforderung war... (niedrig bis hoch)
- Die zeitliche Anforderung war... (niedrig bis hoch)
- Die erbrachte Leistung war... (gut bis schlecht)
- Die aufzubringende Anstrengung war... (niedrig bis hoch)
- Die Frustration war... (niedrig bis hoch)

Im zweiten Teil bewertet der Benutzer alle 15 Kombinationen der zuvor bewerteten Punkte. So entscheidet er beispielsweise, ob für ihn die geistige oder die körperliche Anforderung wichtiger war. Auf diese Weise wird eine Gewichtung der Parameter bestimmt, mit der die Gesamtbewertung berechnet werden kann. Neben der Vielzahl von Anwendungsgebiete ist ein weiterer Vorteil des Task Load Index das Ergebnis in Form eines einzelnen Wertes. Die so erstellte Bewertung kann für die Vergleichbarkeit zwischen unterschiedlichen Systemen oder Probanden genutzt werden.

Auf der Homepage der TLX-Gruppe [7] wird alles nötige für die Durchführung einer TLX-Umfrage angeboten. So gibt es beispielsweise eine Version für die Durchführung mit Stift und Papier aber auch eine digitale Version für die bequeme Auswertung am PC. Für weitere Vertiefung sind auf der Webseite außerdem einige Publikationen verlinkt.

Grundsätzlich und auch bei AR-Anwendungen bieten sich Umfragen gut für die allgemeine Einschätzung der Nutzerzufriedenheit an. Auch für den Vergleich von zwei unterschiedlichen Anwendungen sind Umfragen gut geeignet. Standardisierte Umfragen wie der NASA-TLX bieten außerdem einen ungefähren Vergleich mit fremden Forschungsergebnissen. Auch eine Kombination aus NASA-

[7] `http://human-factors.arc.nasa.gov/groups/TLX/index.html`

TLX für die Belastung des Nutzers und eigenen Fragen für die Bewertung einzelner Anwendungsabschnitte ist denkbar.

In der Forschung werden für die Bewertung allgemeiner Ideen oder Szenarien gerne Online-Umfragen (wie [19]) genutzt, da man so mehr Teilnehmer ansprechen kann. Es gibt allgemein nur sehr wenige weitere Evaluationsmethoden, die über eine so große Entfernung angewendet werden können, wie es bei Online-Umfragen der Fall ist.

Für die Evaluierung können auch weitere standardisierte Fragebögen verwendet werden. So verwenden beispielsweise Rohs et al [23] den Evaluationsfragebogen für Benutzerschnittstellen aus dem ISO 9241-9 Standard.

Den NASA-TLX verwenden Tang et al [26] für die Messung der mentalen Belastung (mental workload) der Probanden. Sie zeigen damit das die Verwendung von AR-Anleitungen eine geringere kognitive Belastung für die Probanden darstellt, als die klassische Anleitung in Papierform.

Interviews stellen eine weitere Befragungstechnik dar. Der Proband antwortet dabei auf die Fragen eines Interviewers. Bei einem Interview werden die Fragen oft vorher vorbereitet, sodass der Interviewer einen Fragebogen abarbeitet. Das Interview bietet sich besonders für kleine Probandengruppen an. Es erlaubt dem Probanden bei Unklarheiten den Interviewer zu fragen und andererseits auch dem Interviewer besonders die Freitextantworten nachzuvollziehen und zu interpretieren. Aufgrund des höheren Aufwands werden Interviews dennoch seltener als einfache Fragebögen verwendet.

Bei Interviews wird in der Regel zwischen strukturierten, semi-strukturierten und unstrukturierten unterschieden. Bei einem strukturierten Interview sind die Fragen und der Ablauf vor der Befragung festgelegt. Oft werden auch die möglichen Antworten vorgegeben, sodass ein solches Interview einer begleiteten Umfrage gleicht. Dadurch, dass der Befragte in seinen Antworten eingeschränkt wird, wird durch den festen Ablauf eines strukturierten Interviews die Auswertung erleichtert. Bei semi-strukturierten Interviews hat der Interviewer nur einige Stichpunkte oder Leitfragen, die dem Dialog einen Rahmen geben. Im Laufe der Unterhaltung können dann die Aussagen des Befragten vertieft oder einzelne Themenabschnitte gewechselt werden. Beim unstrukturierten Interviews werden in der Regel keine expliziten Fragen vorbereitet. Ein solches Interview kann mit einer einfachen Frage, wie *Wie hat Ihnen die Anwendung gefallen?*, beginnen und nimmt seinen Verlauf, wie eine gewöhnliche Unterhaltung. Die Auswertung unstrukturierter Interviews gestaltet sich in der Regel schwieriger als die strukturierter Interviews.

Für die Datensicherung können die Antworten des Probanden von dem Interviewer mitgeschrieben werden. Als Absicherung, oder um das Interview freier zu gestalten, kann es auch als eine Audio- oder Video-Aufnahme gesichert werden.

Die oben vorgestellte Beobachtungstechnik Thinking Aloud Protocol kann den Interviews zugeordnet werden, da auch hier eine Art Dialog zwischen dem Probanden und dem Experimentator stattfindet (vgl. [3]). Dabei findet es meistens als ein unstrukturiertes Interview statt. In manchen Fällen, kann es auch

den semi-strukturierten Interviews zugeordnet werden, da einige Fragen (z. B.: *Was denken Sie, was diese Meldung bedeutet?*) zuvor vorbereitet werden können. Es soll jedoch beachtet werden, dass beim TAP der Interviewer den Probanden nicht anleiten sollte, was durch manche (eventuell vorher vorbestimmte) Fragen geschehen könnte.

Im Augmented Reality Bereich werden oft kurze Interviews nach den Experimenten verwendet, um eine erste grobe Einschätzung der Probanden über das getestete System zu bekommen. Ein Beispiel hierfür ist der Experiment von Henrysson et al [10] zur Kollaboration mit einer MAR-Anwendung. Ein weiteres Beispiel bietet Nilsson [18] an. Sie verwendet unter anderem ein semi-strukturiertes Interview, um die Meinungen der Probanden zu dem getesteten System einzuholen. Die vorbereiteten Fragen können im Anhang ihrer Arbeit nachgelesen werden.

4.3 Auswertungsmethoden

Viele der hier vorgestellten empirischen Methoden sind Techniken zur Sammlung von Daten. Für eine ausführliche Evaluierung wird oft neben den so erhaltenen Werten auch eine statistische Analyse benötigt. Eine solche Analyse ist in der Regel nur wenig davon abhängig ist, welche Art von Anwendung untersucht wird. Deswegen sind die gängigen Methoden zur statistischen Evaluation von Computeranwendungen für die Evaluation von MAR-Anwendungen gleich geeignet. Explizite Auswertungsmethoden für AR- oder MAR-Systeme sind daher nicht bekannt.

Die in dieser Arbeit bereits erwähnte Veröffentlichung von Tang et al [26] kann als ein Beispiel für die statistische Auswertung benutzt werden. Die Autoren benutzen hier die Varianzanalyse ANOVA. Sie (ANOVA) und einfache Durchschnittsberechnungen sind aufgrund ihrer Einfachheit bei statistischen Auswertungen sehr beliebt. Es gibt eine Vielzahl weiterer Werte, die für die Analyse berechnet bzw. genutzt werden können. Zum Beispiel verwenden Pribeanu und Balog [22] den χ^2-Wert für den Vergleich von Modellen zur wahrgenommenen Qualität von AR-Plattformen. Auf eine ausführliche Nennung und Beschreibung von möglichen statistischen Verfahren für die Auswertung der Messdaten wird im folgenden verzichtet, da sie unabhängig von der Verwendung für die AR-, MAR- oder UX-Evaluierungen sind.

5 Diskussion

5.1 Zusammenfassung

Im Rahmen dieser Arbeit wurden unterschiedliche Evaluierungstechniken vorgestellt. Im Bereich der analytischen Evaluation wurde dabei insbesondere auf die Inspections und zwei kognitive Modelle eingegangen. Bei den empirischen Methoden wurde zwischen Beobachtungs- und Befragungstechniken unterschieden.

Bezogen auf Augmented Reality kann hervorgehoben werden, dass die empirischen Methoden beliebter zu sein scheinen, als die analytischen. Ein wichtiger

Grund dafür ist mit Sicherheit, dass sich die empirischen Methoden besser für den Vergleich mit anderen Anwendungen eignen. Die analytischen Methoden werden dagegen oft am Anfang der Entwicklung verwendet, um das Konzept eines zukünftigen Systems zu verifizieren.

Im Rahmen der Recherche konnte keine Verwendung der hier vorgestellten Modelle GOMS und Human-Model-Processor in der Augmented Reality Forschung nachgewiesen werden. Dies könnte ein Zeichen für die noch überwiegend praktische Ausrichtung des jungen Forschungsbereichs sein. Das von Christou [6] vorgestellte Modell Codein erlaubt es zwar auch die AR-Interaktion besser zu modellieren, es ist jedoch komplexer als die verbreiteten Modelle der Mensch-Maschine-Interaktion. Da Codein noch nicht so alt ist, wie GOMS oder HMP, muss es sich in den nächsten Jahren zeigen, in wie weit dieses Modell von anderen Forschern verwendet wird. Bis jetzt kann jedoch keine häufige Verwendung nachgewiesen werden.

Weiterhin zeigt es sich, dass die Untersuchungen, wie Cognitive Walkthrough und die heuristische Evaluation, sowie einige Beobachtungstechniken, wie das Thinking Aloud Protocol, wenig eingesetzt werden. Ein Grund hierfür kann sein, dass diese Methoden eher für die Verbesserung von Anwendungen genutzt werden. Sie bieten aber wenig Raum für die Vergleichbarkeit, die in den wissenschaftlichen Veröffentlichungen oft gefordert wird.

Umfragen oder die Beobachtung von Werten, wie der Fehleranzahl, sind die beliebtesten Mittel bei der Evaluation von AR-Anwendungen. Die verwendeten Fragebögen unterscheiden sich jedoch in Abhängigkeit von den evaluierten Anwendungen. Zwar werden auch NASA-TLX oder ISO-standardisierte Fragebögen verwendet, sie bilden jedoch nur einen kleinen Anteil bei den veröffentlichten Umfragen. Auch die unterschiedlichen beobachteten oder gemessenen Werte unterscheiden sich stark von den verwendeten Aufgaben der Experimentleiter. Damit vermitteln solche Werte und die Umfragen einen guten Eindruck über die Anwendung oder die stattgefundene Interaktion der Probanden mit den Anwendungen. Sie erlauben jedoch in der Regel nur wenig Raum für den Vergleich mit anderen oder eigenen Systemen.

5.2 Ausblick

Grundsätzlich lässt sich zusammenfassen, dass bis auf einige Entwurfsrichtlinien, die zur heuristischen Evaluation verwendet werden könnten, und der hier vorgestellten jedoch weniger verbreiteten Codein-Notation, es nur sehr wenige Evaluationsmethoden gibt, die sich auf Augmented Reality oder sogar auf Mobile Augmented Reality konzentrieren. In einigen Veröffentlichungen zu der Evaluation von Augmented Reality Anwendungen, findet man Aufforderungen Evaluationsmethoden zu entwickeln, die besonders gut für AR geeignet sind. Leider wird dies nur wenig beachtet, was unter Umständen daran liegen könnte, dass solche Methoden zugleich speziell (AR oder MAR) und auf der anderen Seite gleichzeitig möglichst allgemein sein sollten, um viele der Ausprägungen von Augmented Reality abzudecken.

Um beispielsweise geeignete AR-Heuristiken aufzustellen wird weitere Grundlagenforschung benötigt, da die AR-Forschung noch relativ jung ist. Einen Beispiel geben Gabbard und Swan [8], die die Darstellung von Text auf unterschiedlichen Hintergrundtexturen untersucht haben. Ihre Ergebnisse könnten Beispielsweise in Form einer Regel für die heuristische Evaluation formuliert werden oder zumindest als die Frage, ob alle eingeblendeten Texte lesbar auf unterschiedlichen Hintergründen dargestellt werden.

Literatur

1. Azuma, R.T.: A Survey of Augmented Reality. In: Presence, vol. 6, pp. 355-385. (1997)
2. Balaguer, A., Lorés, J., Junyent, E., Ferré, G.1: Scenario based design of augmented reality systems applied to cultural heritage. In: Proceedings of the PH-CHI (2001)
3. Benyon, D.: Designing interactive systems: a comprehensive guide to HCI and interaction design. Addison-Wesley (2010)
4. Card, S.K., Moran, T.P., Newell, A.: The psychology of human-computer interaction. In: Handbook of pereption and human performance ch. 45, pp 1-35 (1986)
5. Card, S.K., Moran, T.P., Newell, A.: The Model Human Processor: An Engineering Model of Human Performance. In: Handbook of pereption and human performance ch. 45, pp 1-35 (1986)
6. Christou, G., Ritter, F. E., Jacob, R. J. K.: Codein — A New Notation for GOMS to Handle Evaluations of Reality-Based Interaction Style Interfaces. In: International Journal of Human-Computer Interaction, 28(3), pp. 189–201 (2012)
7. Dünser, A., Raphaël, G., Billinghurst, M.: A Survey of Evaluation Techniques Used in Augmented Reality Studies. At: HIT Lab NZ (2008)
8. Gabbard, J. L., Swan, J. E.: Usability engineering for augmented reality: employing user-based studies to inform design. In: IEEE transactions on visualization and computer graphics (2007)
9. Jarvis, J., Putze, F., Heger, D., Schultz, T.: Multimodal Person Independent Recognition of Workload Related Biosignal Patterns. In: ICMI '11 Proceedings of the 13th international conference on multimodal interfaces, pp. 205-208 (2011)
10. Henrysson, A., Billinghurst, M., Ollila, M.: Face to face collaborative AR on mobile phones. In: ISMAR'05, pp. 80–89 (2005)
11. Huang, W., Alem, L., Livingston, M. A.: Human Factors in Augmented Reality Environments. Springer New York (2013)
12. Jacob, R. J. K., Shaer, O., Girouard, A., Hirshfield, L. M., Horn, M. S., Solovey, E. T., Zigelbaum, J.: Reality-Based Interaction: A Framework for Post-WIMP Interfaces. At: CHI 2008, (2008)
13. Kostaras, N., Xenos, M.: Usability evaluation of Augmented Reality systems. In: Intelligent Decision Technologies vol. 6, pp. 139-149 (2012)
14. Lewis, C., Rieman, J.: Task-Centered User Interface Design - A Practical Introduction. http://hcibib.org/tcuid/chap-5.html
15. Liarokapis, F., Anderson, E. F.: Using Augmented Reality as a Medium to Assist Teaching in Higher Education. In: Eurographics 2010-Education Papers (2010)
16. Milgram, P., Takemura, H., Utsumi, A., Kishino, F.: Augmented Reality: A class of displays on the reality-virtuality continuum. In: SPIE vol. 2351, pp. 282-292. (1994)
17. Nielsen, J.: Usability Engineering. Morgan Kaufmann (1993)

18. Nilsson, S.: Augmentation in the Wild: User Centered Development and Evaluation of Augmented Reality Applications. (2010)
19. Olsson, T., Salo, M.: Online User Survey on Current Mobile Augmented Reality Applications. In: IEEE International Symposium on Mixed and Augmented Reality, pp. 75-84. (2011)
20. Olsson, T., Kärkkäinen, T., Lagerstam, E., Ventä-Olkkonen, L.: User evaluation of mobile augmented reality scenarios. In: Journal of Ambient Intelligence and Smart Environments 4, pp. 29-47. (2012)
21. Olsson, T.: User Expectations and Experiences of Mobile Augmented Reality Services. (2012)
22. Pribeanu, C., Balog, A.: Towards a hierarchical model for the perceived quality of an augmented reality platform. At: COST Workshop 2011, pp. 13–18 (2011)
23. Rohs, M., Schöning, J., Raubal, M., Essl, G., Krüger, A.: Map Navigation with Mobile Devices: Virtual versus Physical Movement with and without Visual Context. In: ICMI '07, pp. 146-153 (2007)
24. Shneiderman, Ben: Designing the user interface : strategies for effective human-computer interaction. Addison-Wesley (1987)
25. Swan, J. E., Gabbard, J. L.: Survey of User-Based Experimentation in Augmented Reality. (2005)
26. Tang, A., Owen, C., Biocca, F., Mou, W.: Comparative Effectiveness of Augmented Reality in Object Assembly. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 73–80 (2003)
27. Wilson, C.: User Experience Re-Mastered - Your Guide to Getting the Right Design. Morgan Kaufmann (2010)

Process Data Mining in Ubiquitous Systems: A Survey

Stefan Tomov

Karlsruhe Institut of Technology (KIT), TecO,
Vincenz-Prießnitz-Str. 1, 76131 Karlsruhe, Germany
stefan.tomov@student.kit.edu
http://www.teco.edu

Abstract. As in the future computing will be more and more ubiquitous, processes supported by information technology will become ubiquitous as well. These processes need to be monitored in order to be transparent. The objective of this survey is to provide an overview of different methods creating usable knowledge out of process-generated data in the context of ubiquitous process monitoring (UPM). To obtain the overview, these methods are analyzed based on several criteria that are crucial for todays monitoring demands. The result gives a categorized overview with respect to the requirements often present in UPM. Furthermore it will help others to choose an appropriate method for a given problem in the area of ubiquitous process monitoring.

Keywords: Process Monitoring, Ubiquitous Computing, Ubiquitous Systems, Data Stream Mining

1 Introduction

This chapter serves as an introduction to the topic. A motivation for the following work is given in the first section of the chapter. It states the practical relevance of the study. Second, the objectives of this work are given in order to depict its intended purpose.

1.1 Motivation

In general a set of tasks and their inter-relationships like their order are considered as a process. This interpretation of what a process is indicates that many situations in daily and business life can be seen as a process. For example imagine a sequence of tasks for processing an incoming order triggered by a customer. This sequence is a process that consists of a structured set of tasks to finish this order. Other examples for processes include machining processes, chemical or biological processes. Task sequences in many other areas can be considered as processes, too.

Involved entities are often badly influenced by misfunctioning, faults, abnormal behavior or other exceptions happening inside those processes. Imagine an incident while processing an incoming order. When staying undetected the customer is unsatisfied, which is costly for the company. In this example both involved entities are badly influenced by a misfunctioning of the process. For another exmaple consider a process in a nuclear plant that is responsible for regulating the heat, steam-valves and state of the metals and alloys in use. An abnormal behavior or misfunctioning of the heat development or steam-valves that is recognized slightly too late might result in tremendous nuclear damage and costs. In order to detect and appropriately react to the above mentioned problems the processes can be monitored. A comprehensive monitoring solution should be capable of detecting deviations of performance indicators with respect to application-specific requirements like high detection accuracy. However, having all the information required for monitoring the predefined indicators at hand is not the common case. It is demanded that the required information are gathered by entities like resources or person that are involved inside the processes as these parts are controllable. Processes that are not well supported by IT systems, which automatically generate information, require an active creation of information triggered by humans. This can lead to an overload for the person creating such information. So, in many process areas it is hard to automatically monitor the happenings.

Though, new technologies in the area of sensing devices allow a non-invasive generation of events on a big scale. For example creating information without active human interaction through body sensor networks [1]. Sensors are cheap and small and allow collecting diverse data, which can be used to infer the context or other monitoring relevant connections of the sensed object [2]. Actually a set of sensors is carried by most of us in daily life. It is embedded in our smartphones and includes sensors like acceleration, light, proximity, touch and several more. Hence, high potential for creating information in daily life that is usable by process monitoring is implied. This opens the door for monitoring of ubiquitous processes without manual information retrieval. So, a bunch of new opportunities to apply process monitoring is given. In order to create usable knowledge based on the available data, different techniques can be applied that correlate and manage the gathered information. Many proposals for selecting an appropriate technique for process monitoring are present. However, they are tailored for specific use cases and their respective requirements. Generally these requirements can not be transfered or generalized to be applied to ubiquitous processes because they are too specialized. So, practitioners are faced with the problem of selecting an appropriate technique for knowledge extraction. For a single use case it is required to find a set of requirements and respectively a technique for monitoring. These steps can be very time-consuming. In addition the results of these steps would look similar for a wide range of applications. In order to avoid redundant work and save time and efforts when implementing a monitoring solution, the definition of two things is required.

First, a set of requirements for process monitoring that are common to many monitoring applications. Second, techniques that are compatible with those requirements. These might be used in the future to support people applying ubiquitous process monitoring with similar requirements as stated here.

1.2 Objectives of this Work

This paper has two goals. The first is an elaboration of important aspects and implied requirements. This elaboration adheres to ubiquitous process monitoring. Hence these requirements are abstract and general. The requirement's set has the objective to be applicable to a wide range of scenarios. It is important to mention that this set of requirements is not intended to be used for a final decision about what technique should be used for a specific application. Instead it is considered as a baseline for further requirements for respective use cases.
The second goal is to provide an overview of monitoring techniques. This overview is correlated with the predetermined set of requirements. So it can be used by practitioners for guidance in making a decision about what method to use for an application in ubiquitous process monitoring. The objectives can be summarized as follows:

1. Providing requirements to process monitoring techniques that are applicable to a wide range of ubiquitous processes
2. Evaluation and discussion of usability for different approaches in process monitoring

Whereas there is a clear **non-objective**. This evaluation should not be used for final decision-making, as different specificities like extreme high accuracy are demanded by different domains.

1.3 Structure of the Paper

The rest of the paper is structured as follows. Next, some basics about processes and process monitoring are presented. In the third chapter requirements with respect to process monitoring in ubiquitous systems are elaborated. The fourth chapter introduces and evaluates approaches to monitor processes based on the requirements determined previously. Afterwards a comprehensive overview of the evaluation is provided. The results are discussed and conclusions are drawn in the last chapter.

2 Basics about Processes and Process Monitoring Approaches

First, this section is intended to introduce more basic knowledge about process monitoring and related aspects of ubiquitous computing. A finer definition of a process and process monitoring as well as of ubiquitous computing and sensor technologies are given in order to better understand the elaboration of requirements and subsequent evaluation of monitoring methods. Second, an overview of different approaches for creation of monitoring-relevant information is presented.

2.1 Basics

Process and Process Monitoring in General As already mentioned, a process consists of a set of tasks and inter-relationships. Its intended purpose is to reach an objective that is achievable through execution of the task sequence. There exist a variety of process modeling languages to model a task flow, like [3], [4]. Generalizing the definition of a workflow process, at least two aspects of processes need to be present in a process model. The first is the functional aspect describing the single tasks and composed tasks of a process. Secondly a control flow aspect need to be considered. It describes the flow in which the tasks are executed including branches and joints. Depending on the use case several more aspects are taken into account. For example an operational aspect, which is often employed in the area of business processes and includes related programs or resources.
To monitor a defined process, instrumented points for information creation are determined with respect to available sensor technology and the application-specific monitoring demands. For example each start and end of a task could be sensed in order to create monitoring relevant information for control flow monitoring. So, it need to be defined clearly what to monitor, at which points in the process sensors need to be placed in order to create required information, where these information came from and how to use them for monitoring.
An example of an abstract process and these relationships is illustrated in figure 1.

In this paper process monitoring approaches for ubiquitous computing are evaluated. So, before continuing with this evaluation a short introduction about the topic is provided. To keep it short this part is tailored to the connection of process monitoring and ubiquitous systems.

Ubiquitous Computing This paper is focused on processes in ubiquitous systems, so a short introduction to this topic is given.
'Ubiquitous computing is the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user (M.Weiser 1993).'

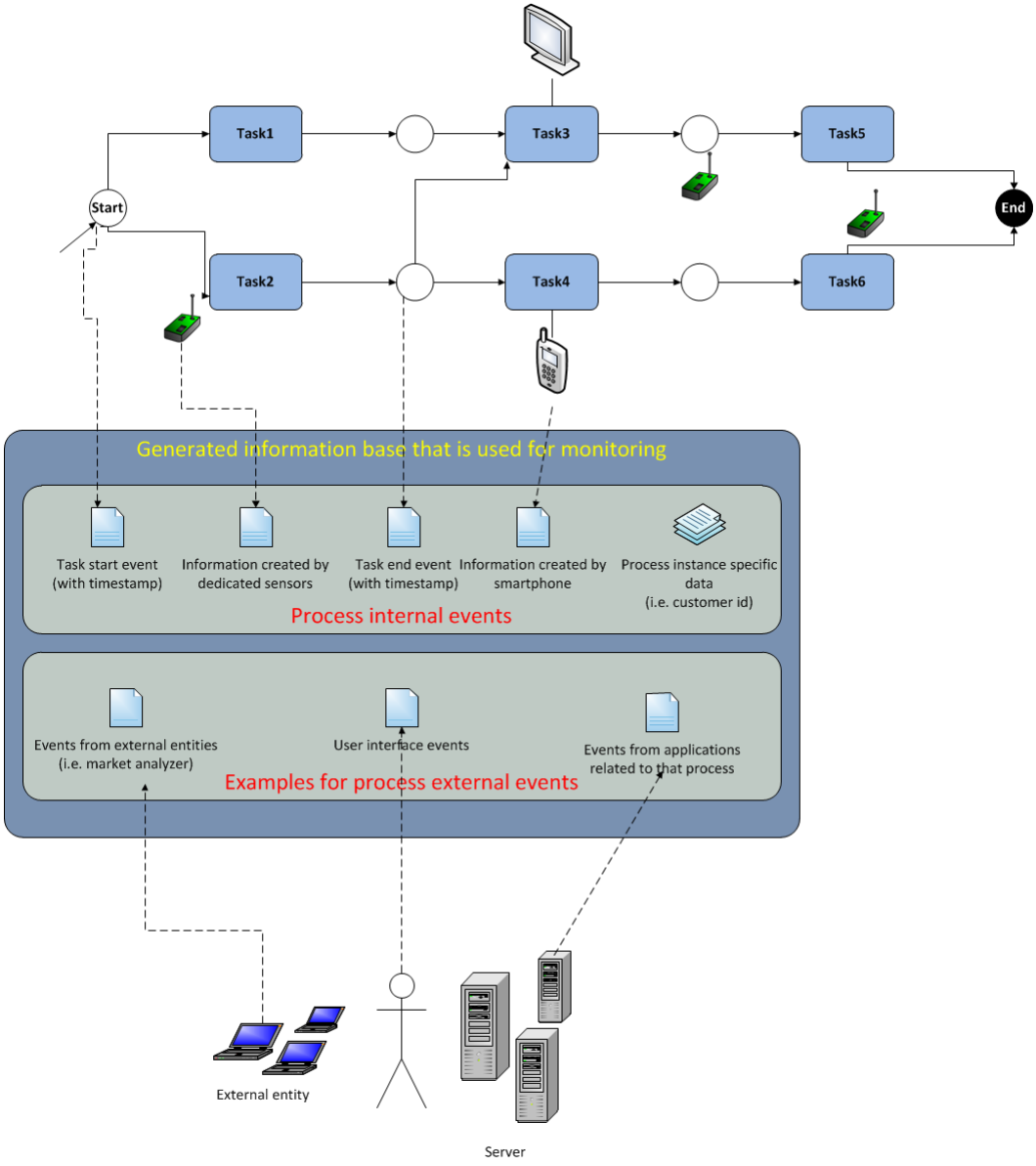


Fig. 1. Abstract process with instrumented process points. Circles are process states, round-cornered rectangles are tasks/activities and arrows indicate the control flow. Several sensing devices and other information creating devices are pictured to state the connection between processes, sensed information and process monitoring.

Adhering to Mark Weiser, the 'father' of ubiquitous computing, it might enhance daily life in a similar way as the emergence of printing did [5]. It can be seen as the third era of computing following the eras of PCs and mainframes. In order to give a short view on ubiquitous computing some important properties are given:

- Ubiquitous systems are calm and invisible systems. This is, interacting persons are not interfered in an annoying fashion by an ubiquitous system [5].
- Often context sensitive while staying non-invasive and calm. Context-sensitivity is the ability of a system to (even proactively) react on situations given by the context without explicit need for human interaction. The ability to do so without the need to invasively sense the required information is called non-invasiveness [5].
- Exhibit high dynamism and volatility as in many use cases new devices join and leave in an unpredictable fashion [6].
- High volume of data are produced and managed in an efficient way.

There exist more properties about ubiquitous systems. However, a short overview is given by the properties stated above. Furthermore, they are chosen so that they are used to infer requirements for ubiquitous process monitoring (UPM).

2.2 Considered Process Monitoring Approaches

This section is intended to give a short overview of techniques and methods for creating monitoring relevant data. This data creation methods are based on available process data. There exists a wide range of approaches. For the purpose of this paper two categories are considered:

Monitoring based on machine learning: Target concepts like an alert level describing the correctness of a process are learned given a set of training examples. The trained models are used for monitoring. There exist approaches based on machine learning in different fields. For example in machining process monitoring artificial neural networks (ANN) are applied to predict process variables [7]. Other methods like support vector machines (SVM) or direct comparison of learned models with alert models exhibit their usability for monitoring as well [8] [9].

Monitoring based on event processing: Uncorrelated information generated by sensors and devices are considered as events. Those events are correlated by an application program to check for predefined patterns and create higher level information like an alert. Event processing approaches are already in wide use in the area of business process monitoring [10] [11]. Recently applications of event processing for monitoring of machining processes were implemented. Another area in which event processing is successfully applied for monitoring is industrial supply chain management [12]. However, the underlying concept is very general and therefore can be used in a wide range of applications [13].

3 Elaboration of Requirements for Ubiquitous Process Monitoring

In this section the first objective - elaboration of requirements - is achieved. At first an explanation about the methodology to acquire the requirements is presented. Afterwards goals and implied requirements of ubiquitous process monitoring are pictured.
In order to define proper requirements for ubiquitous process monitoring two sources of information were tapped:
The first source: UPM will be conducted in many domains like logistics, machining processes, social processes and many more. A big overlap of requirements for the single domains need to be covered by the requirements defined here. Therefore, it is assumed that constraints for monitoring valid in a wide range of approaches from different domains are valid for UPM as well. To find an overlap many approaches are analyzed with respect to their requirements. Then the intersection of the found requirements is determined and considered as a valid set of requirements for UPM.
The second source: Based on the previously defined properties of ubiquitous systems further requirements for UPM are extracted. For each of them the validity and whether it is reasonable or not for UPM is justified.

3.1 Definition and Justification of Requirements

The requirements are defined and explained in this section. To accomplish this two things are done. First, general objectives of UPM are stated. Second, the requirements are defined with respect to the objectives and based on the above mentioned sources.

General objectives of UPM Process monitoring generally aims at measuring and estimating process variables in order to get insight into the processes. When considering *ubiquitous* process monitoring several subgoals are worth to be mentioned. The subgoals are stated based on a ubiquitous example: monitoring of manufacturing processes in order to optimize them and prevent harming situations. In such processes new production lines are assembled for new products. With each production line a set of sensors and respective information are available. Hence, one goal of monitoring in this context is given by the need to flexibly correlate data tuples given by different sets of available sensors. Changing sets of information sources are present in other applications as well, like monitoring of group activities [14]. Additionally, one more goal is implied by these changes. It is given by the demand to flexibly adapt to new monitoring needs. For example assembling a new production line often implies new situations of interest like a definition of a new alert. So, another goal is to adapt to new patterns that need to be monitored. The next important objective of process monitoring in general is given by the ability to detect situations in real-time. This reduces the time and efforts for corrective actions and prevents from undetected aftereffects [15]. To optimize operations in a manufacturing process several other goals need

to be achieved by a monitoring method. They mainly include a high degree of automation regarding the detection, reaction and inclusion of new patterns and alerts. In most application, particularly in manufacturing processes monitoring an emphasis is put on high detection accuracy [16] and reliability in order to prevent costly issues caused by undetected misfunctionings.

Generally Applied Requirements The requirements stated here are justified by the fact that they apply to a wide range of domains.

Degree of automation (R1) It is highly demanded to maintain a high degree of automation regarding the monitoring of processes. The automation of the process itself hinges on the degree of automation of the monitoring [17]. In particular detecting situations of interest and triggering regulating reactions are in the focus of automation. An important aspect is the abstraction level of presented monitoring results. In complex processes it is advantageous to get high level insight rather than a mass of low level information in order to react faster and with less errors. Hence, automating the provision of high-level information and appropriate reactions are desired to be the next step. Imagine a business process for customer support. Reducing the detection and reaction time of incidents can highly increase the customer satisfaction. This is feasible through automation of process monitoring. These advances can be transfered to many other areas and so they are applicable to UPM as well.
A high degree of automation has another advantage. Remind that calmness is a fundamental property of ubiquitous systems. Automating process monitoring in a manner that allows non-invasive information generation, detection and even reactions leads to more calmness. For example consider the monitoring of humans health through body sensor networks. A high degree of automation in this scenario implies that the monitoring can be conducted while the human is doing different tasks. The reason is that the technology is 'calm' and does not require continuous focus of the monitoring subject.
To justify its validity for UPM, some approaches from different areas that tackle this requirement are given. For example a monitoring system is proposed that allows users to configure dashboard on a high abstraction level. Automatic reactions, like a notification email can be defined, too [18]. Continuous monitoring is another area that focuses on a high degree of automation in order to increase cost-effectiveness and risk response [19]. Other attempts around business processes are made to include the automated generation of high level information and respective responses for pervasive RFID sensor networks [20].

High accuracy (R2) The requirement of high accuracy is a very general one. It is considered in nearly all process monitoring applications and is taken as a requirement for UPM as a high detection accuracy is often demanded. In order to justify its importance an example is given that describes problems following a low detection accuracy. Afterwards approaches are depicted, which strongly investigate these requirements.

Consider the simple case of a process milling aerospace alloys, as given in [21]. Small breakages of the tooth milling the alloys imply high damage rates at the milled surface. Hence, a monitoring solution for monitoring both the surface and tooth are needed. Particularly, a high detection accuracy is required as undetected surface anomalies in aerospaces can lead to damages during the flight. This example is generalized to the statement that undetected situations of interest lead to high costs or damages.
Other applications that put a strong emphasis on detection accuracy are given in the area of machining process monitoring [17]. Especially statistical methods are used in applications where a high accuracy is demanded. There exist several such methods that mainly aim at achieving a high accuracy. Examples include PCA based approaches [22] and others like the Shewhart-chart [23].

Requirements Based on Properties of Ubiquitous Systems The requirements given in this section are related to properties of ubiquitous systems. The objective is that the requirements chosen here preserve the stated properties of ubiquitous computing. This is mandatory, because a monitoring solution should not change the most important aspects of the system on which a detection method work.

Real-time monitoring (R3) Todays advances in technology lead to an increased velocity at which processes are performed. The fact that monitored processes generate information as they are enacted, an increase in velocity also leads to an increase in the information amount. It is huge and often unstructured, making it hard to get insight as required into the data. This circumstance is called *IT-blindness*. So, in order to get fast insight into this mass of data and prevent IT-blindness it is suggested that the monitoring solutions need to work in *(near) real-time* and provide high a enough abstraction level [24].
For example in health monitoring, business process monitoring or monitoring machines like air planes real-time insight into process situations can lead to tremendous advantages. Particularly, in health monitoring or business process monitoring, getting fast insight leads to an increase of health respectively revenue.
Regarding the above mentioned properties of ubiquitous computing this requirement supports the achievement of managing and correlating huge amount of data in an efficient and quick manner.

High flexibility regarding removal or addition of monitoring needs (R4) As mentioned earlier, another important aspect is the ability to remove or add new monitoring needs in a flexible and highly automated way. In general, ubiquitous systems expose a high degree of dynamism, see 2.1. This includes the demand to quickly change the monitoring needs without stopping or even restarting the complete monitoring system.

As an example imagine a business process that is enacted in the environment of supply chain support. Introducing new products lead to monitoring a new distribution line of this product. Hence, respective needs like answering the question 'Is product x distributed to location z in time?', are desired to be detected and notified automatically. Adhering to this use-case it is undesirable to stop or restart the monitoring method just to include a new pattern/concept as this can lead to undetected situations.
So, in order to not hinder but even support the high dynamism of ubiquitous systems it is required to flexibly include or remove new situations of interest into the monitoring.

High flexibility regarding the addition or removal of information sources (R5)
Ubiquitous systems are highly dynamic. Notably in the context of adding and removing information sources, like sensors. This is because such systems pervasively sense their environment. However, a particular environment is often not under the control of the entity that conducts the monitoring. This leads to the demand of flexible inclusion or removal of information sources in order to appropriately handle volatile environments.
An illustrative example is given by monitoring social interaction processes including group activities [14]. A changing set of information sources is often exposed by group activities as new individuals might join or leave, which is hard to control. So, wrong detection results are produced by a monitoring approach that is not capable of handling such missing or additional values. But there exist approaches that explicitly try to optimize the feature selection based on a volatile set of underlying information sources in order to prevent this issue.
As ubiquitous systems are dynamic with respect to their information generating sensors and devices, misfunctioning and wrong detections can follow. This problem is tackled through the requirement of a flexible inclusion or removal concerning information sources.

4 Evaluation of Process Monitoring Approaches

The objective of this chapter is to provide a survey of process monitoring methods from different categories. The categories under consideration are *Machine Learning* and *Complex Event Processing*. These approaches are correlated to the requirements for UPM as given in 3.1. The overview is not focused on presenting single approaches. Rather it is intended to provide results of analysis of several approaches for comparison with the previously defined requirements. The provision of a usability evaluation of the two categories, one of the two major objectives, is achieved by the subsequent evaluation.

4.1 Evaluation of Machine Learning Approaches

Machine learning methods are often applied for process monitoring. In particular classification and function approximation are popular methods to infer information on a high abstraction level [17]. Process monitoring is an area in which they are applied. For example notifications are triggered when an abnormality occurred. This is detected by a classifier and used in areas like system administration to find abnormal process, system or network behavior [8]. Another example is the learning of a target function like the surface finish of a machined part in a machining process, which is approximated by artificial neural networks and subsequently used for monitoring when exceeding a threshold [17].
The general steps that need to be conducted when using machine learning for monitoring are stated now. It is important to be aware of them as the ability of machine learning to achieve some of the requirements is influenced by this procedure. Using a machine learning approach like SVMs or ANNs generally requires two major steps. These steps are aligned with the popular KDD process, however they are a shortened version of this process [25].
The first step includes the data understanding and preprocessing. The training of a statistical or mathematical model based on a set of training data is also included in the first step. These parts are crucial for achieving a high quality of the monitoring solution. Data understanding and respective preprocessing, like removing attributes that have less meaning, is required. The performance of the monitoring can be increased heavily by these two actions. The training of the model itself is used to learn from training data. Regarding to monitoring, an example is the learning of sensor value correlations that infer an alert. However, data understanding and preprocessing often require a lot of time and efforts and a complete automation of these steps cannot be accomplished.
The second step includes the classification. The previously trained model is used to classify unknown data samples. An easy example is the use of a model trained with a training sequence, to classify process situations as normal or abnormal. Once a properly trained model exists, the classification task can be conducted with high performance regarding the runtime.

Degree of Automation in Machine Learning Approaches (R1) The overall degree of automation depends on two aspects. They are mapped to the two-step procedure explained above.

The degree of automation is influenced in a different way by each step. The effects on each step are considered following. After this the overall impact of machine learning approaches on the degree of automation is stated.
The first step of data understanding and preprocessing hinders the efficiency of overall automation. That is indebted to several things. A training set is required that reflects situations of interest that should be learned. Using inappropriate data will lead to little generalization and hence to bad results. So the data need to be selected and the correct classes need to be labeled. Subsequently a deep understanding of the data is required in order to choose an appropriate setting, i.e. a learning method and good parameters. The acquisition of data and its understanding are tasks that lower the overall degree of automation. However, there exist approaches which support the automation and decrease the time required for these steps [26].
The second step is the actual use of the trained model. This step runs completely automated, meaning that data tuples are fed into the model and the results are computed automatically. So finally the degree of automation is quite high as the approaches, when in use, compute their results in an automated manner. Adhering to this a reasonable degree of automation of machine learning for monitoring is achieved for the actual usage of the model. Additionally training and data understanding can be supported by some techniques which lowers the implications of non-automation [27], [28].

Accuracy of Machine Learning Approaches (R2) The accuracy of machine learning approaches is influenced by many things. There are two main influencing factors. The first one is a proper choice of training data reflecting the desired ground-truth. An appropriate content and amount of training data is required to reach the demanded ground-truth. The second influencing factor is given by the choice of the learning model. With these influences in mind, a high accuracy can be achieved by machine learning approaches generally [17] [22] [23]. However, in ubiquitous systems other influencing factors occur. They mainly include the demand for low energy consumption and noisy data. In order to get a statement for the accuracy a new evaluation with respect to these two constraints is required. The demand for low energy consumption implies lower accuracy [14]. Adhering to the example of recognizing group activities in social interaction processes, sensors were distributed through attaching them to coffee mugs. However, in spite of having a low energy consumption, a detection accuracy up to 96.2 percentage was achieved. This high accuracy was reached by determination of the optimal trade-off situation between energy consumption, computation capacities and available information [14]. There exist other distributed recognition approaches that are optimized with respect to energy consumption and at the same time provide a high accuracy [29] [30].
The problem of noisy data is constantly present when using sensor networks for information retrieval. For example the sensors are correlated, uncalibrated and the data are measured with low precision or a sensor may even fail, leading to noisy data [30]. A drop in accuracy and precision can be compensated by handling such noises.

There exist probabilistic inference approaches capable of handling these issues and still maintain a high accuracy and precision, as explained in [30]. Hence, it is inferred that basic machine learning approaches will struggle with the advanced requirements of low energy consumption and very noisy data. However a high detection accuracy is achieved by methods adapted to this needs. So, the outcome of the evaluation concerning the accuracy for UPM applications is, that a high accuracy is achievable with appropriate methods in use.

Real-time Capabilities (R3) The two phases of machine learning explained above 4.1 could be used for the evaluation of this requirement. The first step is given with the training and data understanding, the second step is the actual application of the previously trained model. For this evaluation, only the first phase is considered.
It is assumed that there exist enough time for the training and data understanding, so that there is no need for real-time capabilities concerning these steps.
The runtime of the first phase mainly depends on the methods in use. Traditional concept learning approaches can be too slow for real-time applications, which is the reason why achieving real-time for this phase is an issue. However, this problem was adopted to push down the runtime to an acceptable degree. As a consequence, approaches were introduced for which real-time capabilities were included. That is successfully accomplished in many different areas.
Successful real-time model usage include approaches in the areas of recognition of emotions [31] or detection of specific surfaces using AdaBoost [32]. Another method that exposes potential for achievement of real-time is given with a modification of neural networks [33].
Putting all this together it can not be implied that machine learning in general is applicable for real-time UPM. However, potential to achieve this is given by extensions for different approaches. These extensions are used to provide real-time capabilities in recognizing concepts of concern or approximating functions. So as a result of this evaluation it is reasoned that this requirement needs special attention when conducting UPM.

High Flexibility regarding new Monitoring Needs (R4) While training a model a predefined target function like the rate of correct classified data is optimized. However, the definition of that target function makes assumptions. For example for classification these include the respective classes to predict, for function approximation the degree of correspondence to the original function. Once the model is trained it can be used for respective monitoring tasks. But the point is, that a trained model can be used for trained classes/functions only. If a new monitoring need is present it is hard to include this one into the already trained model. An obvious solution to this problem is to train a separate model incorporating these new classes or functions. The drawback of this method is that a new model need to be trained and data understanding need to be accomplished which is costly in terms of time and efforts. Only limited research has been conducted around this issue. As a result, the achievement of this requirement is feasible but

difficult and therefore needs special attention when using machine learning for UPM.

High Flexibility regarding new Information Sources (R5) Typical machine learning methods suffer from this requirement. The issue arises from the fact that traditional machine learning methods are trained and used based on a set of predetermined attributes (for example classifiers or function estimators). These attributes are static meaning that each data tuple should contain the same attributes. If missing values are present, they need to be substituted during preprocessing. However, as ubiquitous systems expose a high volatility, new information sources join or leave in an unpredictable manner. An illustrating example for this circumstance is given with group activity recognition where a data tuple consists of attributes representing the sensed values of the group members. New people can join or leave the group without having control over this. Missed or additional attributes are implied by this join or removal. Hence, the problem is given by the need to handle data tuples that contain more or less attributes than the training data used for building the model.
Basic approaches are not capable of solving this issue. However, many methods were developed tackling the problem of changing data foundations recently, like concept-drifting [34] or mining of time-changing streams [35], [36]. Hence, there is potential that the requirement of flexible inclusion of new monitoring needs can be achieved by machine learning methods. Nevertheless special attention is required to handle this flexible inclusion.

4.2 Evaluation of Approaches Based on Complex Event Processing

Complex Event Processing (CEP) is an emerging technology in general and in particular for the use case of process monitoring [13]. The idea behind CEP is to correlate events from a stream of data. Adhering to ubiquitous computing the sources for that data can be sensors. The correlation is based on a set of predefined patterns or rules. For example an abstract pattern can be defined in the following way: Event a *'followed by'* Event b *'AND'* a.value - b.value $\geq$ c $\rightarrow$ Event d. A CEP engine then checks for an occurrence of this pattern in the data stream. If found, it creates an event d which in turn is used for further reasoning. In this way many abstraction levels regarding the created information can be included. This means, the implied event d could also be used for indicating an alert or abnormalities inside a process. However, it could also be an event from an abstraction layer above the events in the rule body and so can be used for structuring the information according to needs of the respective use-case.

Degree of Automation (R1) CEP for monitoring is not fully automated, rather semi-automated. Semi-automated means that the part of the actual rule/pattern checking and subsequent reasoning of new knowledge is done automatically. It also includes automatic reactions on detected patterns.

This is easily accomplished with CEP as a new event which indicates a specific reaction on the actual alert can be produced [20]. Whereas the definition of patterns is not automated.
In my eyes there is a high potential for automating the pattern definition as well. This could be done with the support of machine learning approaches. For instance new rules could be learned and manifested based on the observable data stream.
All together CEP is appropriate for achieving a high degree of automation in UPM systems.

Accuracy of CEP Approaches (R2) Under the assumption that all monitoring needs like alerts and abnormalities are manifested in rule or pattern definitions the detection accuracy is very high. This is due to the fact that correlation engines used by CEP check for those patterns and detect them, if present, in the investigated data stream. These detections are guaranteed and hence the accuracy is very high.
Problems with accuracy are present if monitoring needs are not defined as patterns, which is a known issue of CEP [37]. In this case the engine will not detect anything concerning this pattern. So zero accuracy can be the result if respective patterns are not defined. Hence, in order to maintain a high accuracy it is very important to define all the required patterns or even let them be learned automatically by machine learning methods. There exist alternatives to solve these problems. For example events are structured on different abstraction levels and other dimensions like the intended objective of a pattern. This will lead to less missing patterns hence to higher accuracy in detection [38], [39].
As a result, CEP provides very high accuracy under the assumption that no patterns and rules are forgotten. Efforts have taken place with the objective to decrease the amount of missing patterns through automatic learning and/or better structuring. So I think for the future the assumption of a set of patterns rich enough for achieving high accuracy can be matched.

Real-time Capabilities (R3) CEP is a technology that is prone for real-time computation [13], [40], [10]. In general the efficiency of detection primarily depends on the complexity of the respective pattern [41]. As an example a CEP engine called ETALIS is considered [41]. For simple patterns a correlation throughput of 37437 detections per second was observed. Whereas very complex patterns can be detected at a rate of nearly 4000/second. Most of the other CEP engines are slower but they still achieve rates high enough for many real-time applications. So in my eyes CEP engines are capable of achieving real-time operation. In addition due to their extreme high throughput the demand of correlating tremendous amounts of data can be achieved. This is one of the major properties of ubiquitous systems stated above in 2.1.

High Flexibility regarding new Monitoring Needs (R4) A monitoring need like detection of an abnormality, misfunctioning or alert can be easily expressed as an event pattern. For illustration an example is given: The processing of an inbound order as given in the motivation of this paper (see 1.1). Consider four events: inbound_order, order_confirmation, payment_aborted and user_logged_out. Now the company wants to monitor missed revenues. This need can be defined in form of a rule as follows.
inbound_order *'followed by'* order_confirmation *'followed by'* payment_aborted *'followed by'* user_logged_out → missed_revenue
In this way a new monitoring need, namely missed_revenue, is defined. In addition this rule need to be incorporated into the pattern hierarchy. After this is done, the CEP engine correlates the newly defined monitoring need when scanning the data stream. Using this methodology, new monitoring demands are incorporated without the need to stop or restart the system. Hence, this requirement is achieved by CEP.

High Flexibility regarding new Information Sources (R5) A CEP engine does not make any assumptions about the source of an event. Instead it considers the data stream containing all the events as a so-called event cloud [38], [13]. Two things are abstracted by the event cloud: the first are the format and the actual source of an event. The second is the fact that events are fed into the system in an unorganized fashion. That means, they are not ordered with respect to their creation-time, causality or responsibility when they arrive [38]. This implies, that in theory, the CEP correlation engine is not affected by unpredictable removal or inclusion of new information sources.
However, it is important to mention that respective adapters and parsers are required for new sources. This is caused by the need of the events in the event cloud to be in a jointly agreed format. Otherwise the CEP engine is not capable of interpreting the events.
Under the assumption that appropriate adapters and parsers are present, CEP approaches for monitoring are capable of flexibly include or remove new information sources.

4.3 Final Discussion and Evaluation

This section contains the discussion of machine learning and CEP approaches for UPM. In particular the discussion is comprised of an overall evaluation of the two categories. The summarized evaluation is accomplished by comparing the evaluation results of the categories with respect to the requirements given in section 3.1 and is depicted in the figure 4.3. It gives an overview of the evaluations for machine learning and CEP approaches regarding the above mentioned requirements. In the following a comparison for each requirement connecting both areas is given. With this summary, the second major objective of the survey from 1.2 is considered.
The first requirement is about achieving a high level of automation (R1).

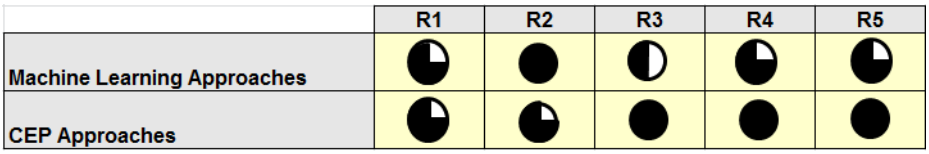


Fig. 2. An overview of the evaluation for both categories, namely machine learning and CEP is pictured. The degree to which a circle is filled black corresponds to the degree to which a respective requirement for UPM is achieved.

This is achieved to an acceptable degree by both approaches, machine learning and CEP. With a view to the future, for both categories further approaches will be developed with the objective to provide full automation. For machine learning approaches this means to increase the automation level of the training and understanding phase. Whereas for CEP approaches automating the pattern definitions is required. Both the existing achievement of automation and improvements that need to be done to achieve full automation are weighted equally for machine learning and CEP. This justifies the result implying the same high degree of automation, as indicated in the figure by a three-quarter filled circle.
A high detection accuracy as stated by the second requirement (R2) is also accomplished to an acceptable extent by both. Machine learning has slight advantages compared to CEP regarding the accuracy. This is due to the fact that for CEP approaches it cannot be assumed that all patterns of interest are included. The consequence is, that forgotten patterns unavoidably lead to an accuracy loss. Compared to this, accuracy in machine learning does not depend on patterns but on the method in use, which seems to have a more certain accuracy outcome. This is the reason, why accuracy is weighted higher for machine learning.
Considering the *real-time capabilities* (R3) of both categories, CEP engines provide better results, as explained above. There exist machine learning methods that can process in real-time, too. However, compared to machine learning, CEP engines are able to correlate tremendous amounts of data in real-time which can not be processed by machine learning methods in a comparable time.
The last two requirements are about *flexibility concerning new monitoring needs* (R4) and inclusion/removal of information sources (R5). In my opinion CEP approaches outperform machine learning. CEP engines provide an easy and intuitive mean to achieve these two requirements, as explained above in 4.2. Compared to this, traditional machine learning methods suffer from the ability to fulfill them. Special extensions and more investigations are required in order to do so. This leads to a higher degree of achievement for CEP approaches concerning the requirements R4 and R5.

5 Conclusion

A survey of different approaches for ubiquitous process monitoring was investigated in this paper. The requirements for UPM were elaborated and subsequently a selection of approaches was analyzed and categorized. These categories were evaluated with respect to the requirements. It turned out that monitoring based on CEP approaches achieved the highest degree of requirements.
Compared to other survey paper in the domain of process monitoring this work is clearly differentiated. The contributions are two-fold. First, an elaboration of generally applicable requirements for ubiquitous process monitoring is given. In other research articles requirements for either ubiquitous systems or process monitoring are presented. However, combining both to requirements for UPM is missing.
Second, an evaluation of two categories of process monitoring applied to ubiquitous systems is presented. Most surveys around process monitoring are concerned with very specific domains compared to the very general field of ubiquitous systems.
This survey is focused on traditional properties whereby the fields of machine learning and CEP are rapidly evolving. Meanwhile the degree to which the requirements are achieved might have changed. Furthermore the results were not underpinned by experiments.
In my eyes, a deeper analysis of machine learning and CEP need to be conducted with the objective to evaluate their applicability for UPM. Additionally for proving the results of the analysis experiments could be performed. The vision of these research efforts is to provide a quality criterion for practitioners in the field of UPM. They should be supported by this criterion in terms of understanding pros and cons of different approaches and choosing an appropriate one.

References

[1] Ng, J.W., Lo, B.P., Wells, O., Sloman, M., Peters, N., Darzi, A., Toumazou, C., Yang, G.Z.: Ubiquitous monitoring environment for wearable and implantable sensors (ubimon). http://ubimon.doc.ic.ac.uk/bsn/public/UbiMonPapers/Ubiquitous_Monitoring_Environment_for_Wearable_and_Implantable_Sensors_%28UbiMon%29.pdf (2004)

[2] Schmidt, A., Beigl, M., Gellersen, H.W.: There is more to context than location (1998)

[3] van der Aalst, W.: The Application of Petri Nets to Workflow Management. DOI: 10.1142/S0218126698000043 (1998)

[4] Axway, BizAgi, Associates, B.S., Scheer, I., Corp., I., International, M., Solutions, M.D., Group, O.M., Oracle, AG, S., AG, S., Software, T., Unisys: Business Process Model and Notation (BPMN) (2011) http://www.omg.org/spec/BPMN/2.0/PDF/.

[5] Weiser, M.: The Computer for the 21st Century (1999)

[6] Kindberg, T., Fox, A.: System software for ubiquitous computing (2002)

[7] Monostori, L.: A step towards intelligent manufacturing: Modelling and monitoring of manufacturing processes through artificial neural networks (1993)

[8] Chiarini, M., Couch, A.: Machine learning for the system administrator. http://people.seas.harvard.edu/~chiarini/docs/lisa07_ml.pdf (2007)

[9] Ganti, V., Gehrke, J., Ramaluishnant, R.: A framework for measuring changes in data characteristics (1999)

[10] Burger, F., Debicki, P., F.Koetter: Vergleich von Complex Event Processing-Ansaetzen für Business Activity Monitoring. http://elib.uni-stuttgart.de/opus/volltexte/2010/5258/pdf/FACH_0112.pdf (2010)

[11] Grauer, M., Karadgi, S., Metz, D., Schaefer, W.: Online Monitoring and Control of Enterprise Processes in Manufacturing Based on an Event-Driven Architecture. In: Lecture Notes in Business Information Processing. Volume 66., Springer (2010) 671–682 http://dx.doi.org/10.1007/978-3-642-20511-8_61.

[12] Ku, T., Zhu, Y., Hu, K.: A Novel Complex Event Mining Network for Monitoring RFID-Enable Application. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4756911 (2008)

[13] Opher Etzion, P.N.: Event Processing in Action. Manning (2010)

[14] Gordon, D., Hanne, J.H., Berchtold, M., Miyaki, T., Beigl, M.: Recognizing group activities using wearable sensors. In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Volume 104. (2012) 350–361

[15] IQMS: Manufacturing process monitoring. http://www.iqms.com/manufacturing-software/process_monitoring.html (2013)

[16] Lee, D., Hwang, I., Valente, C., Oliveira, J., Dornfeld, D.: Precision manufacturing process monitoring with acoustic emission. In: International Journal of Machine Tools & Manufacture. Volume 46. (2005) 176–188

[17] Liang, S.Y., Hecker, R.L., Landers, R.G.: Machining process monitoring and control: The state-of-the-art. In: Journal of Manufacturing Science and Engineering. Volume 126. (2004) 297–311

[18] Guerlain, S., Bullemer, P.: User-initiated notification: A concept for aiding the monitoring activities of process control operators. http://pro.sagepub.com/content/40/4/283.short (1996)

[19] Schultz, E.E.: Continuous monitoring: What it is, why it is needed, and how to use it. http://www.sans.org/reading_room/analysts_program/analyst-tripwire-schultz.pdf (2011)

[20] Kim, K., Oh, K., Rosales, P., Jung, J.Y.: A ubiquitous process coordination system for rfid/usn events. http://www.iieom.org/ieom2011/pdfs/IEOM104.pdf (2011)
[21] Marinescu, I., A.Axinte, D.: An automated monitoring solution for avoiding an increased number of surface anomalies during milling of aerospace alloys. In: International Journal of Machine Tools & Manufacture. Volume 51. (2011) 349 – 357
[22] Bakshi, B.R.: Multiscale pca with application to multivariate statistical process monitoring. In: AIChE Journa. Volume 44 No. 7. (1998) 1596–1610
[23] Box, G., Kramer, T.: Statistical process monitoring and feedback adjustment a discuss. In: TECHNOMETRIC. Volume 34 No.3., American Statistical Association and TECHNOMETRICS, AUGUST 1992, VOL. 34, NO. 3 the American Society for Quality Control (1992)
[24] Luckham, D.: The Beginnings of IT Insight: Business Activity Monitoring. http://www.ebizq.net/topics/cep/features/4689.html (June 2004)
[25] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The kdd process for extracting useful knowledge from volumes of data. http://shawndra.pbworks.com/f/The%20KDD%20process%20for%20extracting%20useful%20knowledge%20from%20volumes%20of%20data.pdf (1996)
[26] Vesanto, J., Hollmen, J.: An automated report generation tool for the data understanding phase. citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.4094&rep=rep1&type=pdf (2004)
[27] Lee, K., Cremer, M.: Automatic labeling of training data for singing voice detection in musical audio. https://ccrma.stanford.edu/~kglee/pubs/klee-sppra09-final.pdf (2009)
[28] Yan, R., Yang, J., Hauptmann, A.: Automatically labeling video data using multi-class active learning. In: Proceedings of the Ninth IEEE International Conference on Computer Vision. Volume 2. (2003) NaN
[29] Wittenburg, G., Dziengel, N., Wartenburger, C., Schiller, J.: A system for distributed event detection in wireless sensor networks. In: Proceedings of IPSN'10. (2003)
[30] Paskin, M.A., Guestrin, C.E.: Robust probabilistic inference in distributed systems. http://ai.stanford.edu/~paskin/pubs/PaskinGuestrin2004a.pdf (2004)
[31] Bailenson, J.N., et al.: Real-time classification of evoked emotions using facial feature tracking and physiological responses. In: International Journal of Human-Computer Studies. Volume 66. (2008) 303–317
[32] Holmes, Q.A.: Textural analysis and real-time classification of sea-ice types using digital sar data. In: Geoscience and Remote Sensing. Volume 22. (1984) 113–120
[33] Carpenter, G.A., Grossberg, S., Reynolds, J.H.: Artmap: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. In: Neural Networks. Volume 4. (1991) 565–588
[34] Zliobaite, I.: Learning under concept drift: an overview. http://arxiv.org/pdf/1010.4784.pdf (2010)
[35] Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: KDD '01 Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. Volume NaN. (2001) 97–106
[36] Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: Framework for clustering evolving data streams. http://www.vldb.org/conf/2003/papers/S04P02.pdf (2003)
[37] von Ammon, R., Silberbauer, C., Wolff, C.: Domain Specific Reference Models for Event Patterns for Faster Developing of Business Activity Monitoring Applications. (2007)

[38] Luckham, D.: The Power of Events. An Introduction to Complex Event Processing in Distributed Enterprise Systems. Addison-Wesley (2007)
[39] Paschke, A.: Design Patterns for Complex Event Processing. (2008) http://arxiv.org/ftp/arxiv/papers/0806/0806.1100.pdf.
[40] Apama: The Apama BAM Architecture - CEP-Enabled Business Activity Monitoring. Technical report, Progress Software (2008) http://www.psdn.progress.com/progress_software/worldwide_sites/be/docs/apama_bam_architecture.pdf.
[41] Anicic, D., Rudolph, S., Fodor, P., Stojanovic, N.: Real-Time Complex Event Recognition and Reasoning - A Logic Programming Approach. Applied Artificial Intelligence **26** (2012) 6–57 http://www.tandfonline.com/doi/abs/10.1080/08839514.2012.636616.

Approaches for the 3D Reconstruction of Human Bodies Using Kinect

Author: Heike Adel
Supervisor: Markus Scholz

TECO, Karlsruhe Institute of Technology (KIT)

Abstract. This term paper presents and compares four state-of-the-art approaches to reconstruct 3D models of human bodies. The persons are captured using the Kinect device. Its sensor is monocular and its scans are rather noisy, especially when the person stands some meters away from the camera. It can be observed that all presented papers solve the same challenges: Firstly, the bodies need to be separated from the background. Secondly, the noise and low resolution of the scanned data might be improved. Thirdly, the authors show methods to register scans from different angles to each other or to a template. After this preprocessing, they apply a mathematical model to obtain a 3D shape of the body.
The term paper shows that depending on the task another approach might be optimal: If the body should be applied to different poses, the SCAPE model is the best choice. If the bodies should wear clothes, one of the other approaches should be applied. For instance, if time is a limited factor, the method that uses three Kinects is recommendable. On the other hand, if it is more important to reduce measurement errors, the super-resolution approach fits the best.

1 Introduction

This term paper describes the challenges and possibilities of building a 3D body model from Kinect scans. As motivation, the reader might imagine the following scenario: An increasing number of people purchase their clothes on the Internet. However, they are not able to try out the dresses. If they had a 3D model of their own body, they could easily see whether the clothes fit to them or not [1]. Unfortunately, traditional devices which scan the whole body are quite expensive. They cost about $35,000 to $500,000. Therefore, investigations are presented that use the Microsoft Kinect [2]. The Kinect is much more cost-effective, can be used in everyone's living room and is very compact [3,4]. According to [1,3], Microsoft's scanning system is as easy to use as a normal video camera, whereas traditional scanning devices often require expert knowledge.
Realistic 3D models of objects (especially of human beings) are essential for many applications such as design, games, fitness, animation or virtual reality [1,3,4]. Nevertheless, building a complete model of a particular person is quite challenging. For example, some body parts might be occluded [5]. Furthermore, a person

usually moves during scanning. Even if he or she is asked to stand still, there will typically be some variations in his or her shape. Hence, if several scans are taken, the alignment between them is not straight-forward [4].

2 Suitability of the Kinect for 3D Body Reconstruction

2.1 The Kinect System

The Kinect was developed by Microsoft and Prime Sense. The device is rather inexpensive (costs about $150 [1,6]), compact, easy to use and widely available [1,3,4]. While traditional scanning systems are based on multiple calibrated cameras [4], the Kinect is a monocular device (i.e. it contains only one camera). It captures image silhouettes and range data [4] without using markers. Figure 1 shows that the device consists of an RGB camera, an infrared camera, and an infrared projector [4,7].

Fig. 1. the Kinect device [8]

The infrared projector and the infrared camera capture the depth using a method called Light Coding [9]. The projector creates a pattern of points in the scene. Those are captured by the infrared camera. A reference pattern of points, captured at a known distance from the sensor, is stored in the memory of the device. The captured points of the current scene are compared to this reference pattern. Objects, whose distances differ from the reference, will change the location of the laser points. Figure 2 illustrates this.

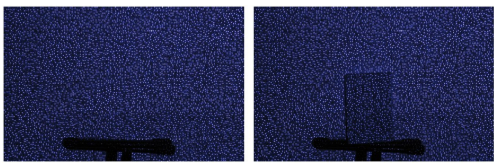


Fig. 2. Changes in the point pattern due to an object [9]

The depth is calculated using equation 1 which is derived from the similarities of triangles [7].

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{f \cdot b} d} \tag{1}$$

Z_k denotes the depth of the object, Z_0 the reference depth. f is the focal length of the infrared camera, b represents the distance between the camera and the projector, and d is the change of the laser points due to the objects (disparity). The variables and triangles are illustrated in figure 3.

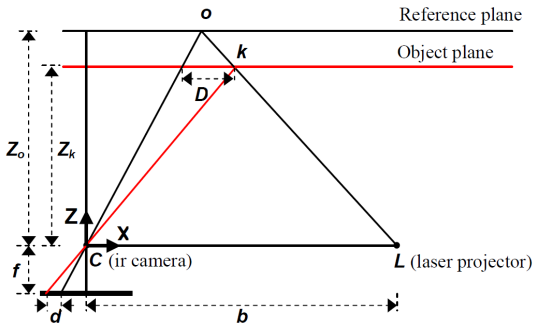


Fig. 3. Depth calculation process [7]

The captured values are stored in a depth map that contains a 11 bit representation of the depth value for each pixel [10]. Hence, the values are discretized into $2^{11} - 1 = 2047$ levels (the 2048^{th} level is used for those pixels for which no depth could be calculated). However, the levels are not uniformly distributed. Instead, the density increasees with the proximity to the device. The depth information is captured at 30 frames per second at a framerate of 640×480 pixel [4,11].
Using the same framerate for the RGB camera also leads to a resolution of 640×480 pixel. However, if using a framerate of 15 Hz, a resolution of 1280×1024 can be obtained [10].
Orginally, the Kinect was designed for object detection and modern user interfaces, not for capturing high-quality 3D data. Therefore, the resolutions of both the RGB and the depth data are rather low and the noise level is quite high [3]. In addition, the field of view of the sensor is rather small (approximately 57 degrees horizontal and 43 degrees vertical [11]). Hence, the captured person needs to stand several meters away from the Kinect so that his or her entire body can be scanned. The optimal distance to the Kinect camera is about 1.2 meters to 3.5 meters. Due to these challenges, the Kinect seems to be inappropriate for the task of scanning full human body models [3]. Nevertheless, it has several advantages, especially its low price, availability and easy use. Hence, there are

investigations about modeling human bodies from Kinect data. They overcome the challenges in different ways and achieve quite impressive results.

2.2 Challenges of the Reconstruction of Bodies

The following list provides an overview of the challenges which all presented methods need to overcome due to the limited sensors of the Kinect.

1. Data Capturing and Segmentation of the Body from the Background:
   A person is captured using the Kinect system. To obtain enough information for the reconstruction of the entire body, several scans from different viewpoints need to be taken. This requires several design choices: How many Kinects should be used, in which distance should the Kinects be placed, how many captures should be obtained. Once the Kinect data is available, the body needs to be separated from the background. Unfortunately, it is very hard to overcome segmentation errors in the following steps.
2. Low Resolution and Noisy Data:
   To obtain more useful data, the resolution and the noise of the scans might be improved.
3. Non-Rigid Registration:
   Even if the scanned person is asked to remain still, movement will occur between different scans [1,3,4]. This complicates the combination of scans from various viewpoints. The procedure of aligning point clouds to each other is called registration [6]. A movement that changes the measurements of the body parts in the scans is called non-rigid deformation. Hence, the alignment is referred to as non-rigid registration.
   In the presented papers, different registration techniques are investigated to align the different scans to each other. A possible difficulty that can occur during registration is the loop closure problem. This challenge arises when the same part of the body is detected twice but assigned to different locations. It can occur if errors between successive scans accumulate over all scans. [6,12] An examplifying picture can be found in section 6 (picture 15).
4. Apply Mathematical Models to Obtain 3D Bodies:
   The information obtained by the Kinect scans needs to be processed with a mathematical model to build a 3D model of the human body.

In the following sections, two possible methods to build a human body from a 3D point clouds are described and then, each approach is presented following the same scheme: First, the capturing and segmentation process is explained. Then, it is described how the system meets the challenge of low resolution and noise, and the challenge of non-rigid movements. Finally, the steps to obtain the 3D body model are presented.

3 Mathematical Models to Build 3D bodies

3.1 The SCAPE Model

The SCAPE model was introduced by Anguelov et al. in the year 2005 [5]. The name SCAPE is derived from the phrase "Shape Completion and Animation of PEople". One of its characteristics is the separation of body modifications due to different poses and different body shapes. Hence, two models are developed: one for the non-rigid surface deformation as a function of the pose, and one for variations based on the body shape. Due to this, poses captured on a large person can be applied to a small person and the other way around. Another advantage is that movements during scanning can be handled [4].
SCAPE is a data-driven method, i.e no prior knowledge about body shape or movement is needed. Furthermore, it can be used to apply body models to people that do not appear in the training data or even to other non-human creatures.
The input of the SCAPE model is a full-body mesh of triangles obtained by a scanning device and several preprocessing steps. Since the method is data-driven, several scans of several persons in different poses are required as training data to learn the model parameters.

Preprocessing steps The scanned data needs to be preprocessed. Figure 4 illustrates those preprocessing steps for SCAPE.

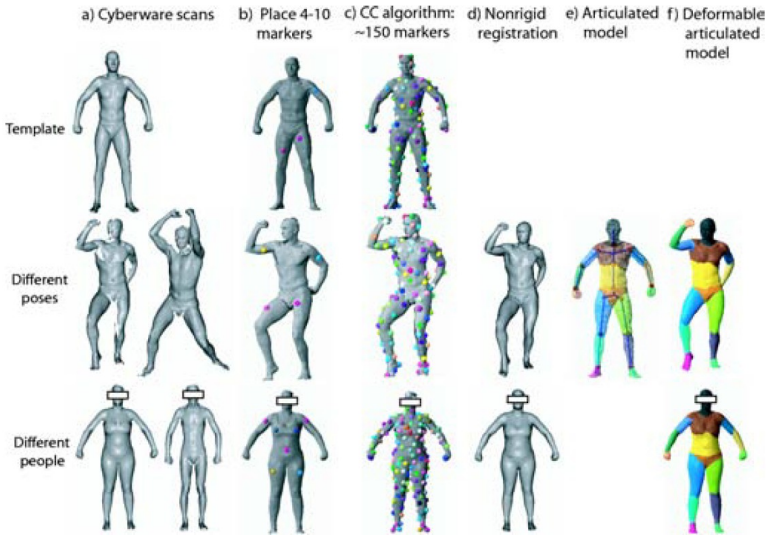


Fig. 4. Preprocessing steps for SCAPE [5]

– One of the captured meshes is determined to be a so-called template mesh. It serves as reference mesh for all the other scans. Therefore, possible holes need to be filled.
– The template mesh is assigned to every other mesh (referred to as instance meshes) using markers. A small number of markers is placed manually. Additional markers are obtained using the Correlated Correspondence Algorithm which replicates the markers over the surface while optimizing a probabilistic model over all point-to-point correspondences [13]. Especially, neighboring points in one mesh are mapped to neighboring points in the other mesh. The results are shown in figure 5.

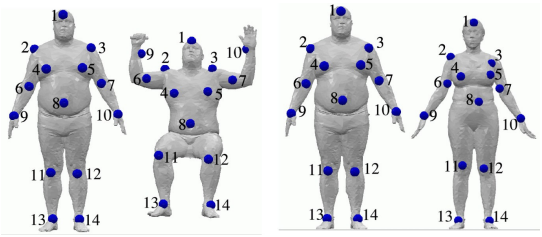


Fig. 5. Results of the Correlated Correspondence Algorithm [13]

– The markers serve as input to a non-rigid registration algorithm that estimates transformations and results in a set of meshes whose common shape approximates the scanned person.

The preprocessing steps generate a set of vertices and triangles for each mesh. They form the input for the mathematical models of SCAPE. In the remainder of this section, the two different models are described.

Pose model To decouple the computation of the pose model from the computation of the body shape model, it is assumed that the body shape does not differ across pose variations. Therefore, the same person is scanned in different poses.
The pose model separates the rigid and non-rigid deformations. The term *rigid deformation* denotes a movement that does not change the length of the body dimensions in the scans. Hence, it can be modelled using a simple rotation. *Non-rigid deformations*, however, change the measurements of the body parts in the scans. To model them, a more general transformation is needed. In the following explanations, i denotes the number of the pose. The rigid deformation is modelled with rotation matrices $R^i_{l[k]}$. All triangles k of a particular body region $l[k]$ share the same rotation. The non-rigid deformation is modelled with transformation matrices Q^i_k.
The transformations are applied to the edges of the triangle. Each triangle of

the template mesh is aligned to the corresponding triangle of each instance mesh that represents a possible pose i. Hence, a triangle edge $x_{k,j}$ from the template ($j = 1, 2$: the two edges adjacent to a certain vertex of the triangle) is transformed to pose i using the following equation:

$$v^i_{k,j} = R^i_{l[k]} \cdot Q^i_k \cdot x_{k,j} \tag{2}$$

While the matrices $R^i_{l[k]}$ are generated during the preprocessing steps, the values of Q^i_k need to be estimated. This is realized by solving the following equation for each instance mesh:

$$argmin_{Q^i_1 \ldots Q^i_P} \sum_k \sum_{j=1,2} ||R^i_k \cdot Q^i_k \cdot x^i_{k,j} - v^i_{k,j}||^2 + \omega_s \sum_{k_1,k_2 adj} I(l_{k_1} = l_{k_2}) \cdot ||Q^i_{k_1} - Q^i_{k_2}||^2 \tag{3}$$

The first summand is the distance between the transformed edge and the observed edge in the instance mesh. The second summand is a smoothing term between neighboring triangles and consists of $\omega_s = 0.001\rho$ and an indicator function $I(\cdot)$.The value of the variable ρ depends on the resolution of the template mesh.
To achieve consistency across triangles, another constraint for the transformations is applied: They should minimize the least square error of the transformed template edge $y^i_{k,j}$ and the edge of the instance mesh $v^i_{k,j}$:

$$argmin_{y_1 \ldots y_M} \sum_k \sum_{j=1,2} ||v^i_{k,j} - y^i_{k,j}||^2 \tag{4}$$

The resulting model is able to represent poses that do not appear in the training data set. An example is shown in figure 6.

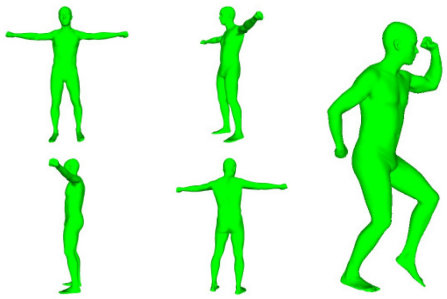


Fig. 6. Recovered shape in new pose
The four shapes and poses on the left are recovered from the scans. The body on the right is the recovered shape in a new pose [4].

Body shape model The training data for the body shape model contain scans of 37 people in similar poses and 8 publicly available models from the CAESAR data set [14]. The shape variation is represented by linear transformation matrices S_k^i. It is assumed that triangle p_k in mesh i (the mesh of person i) is obtained by first applying the non-rigid pose deformation Q_k^i, then the body shape deformation S_k^i and finally the rotation of the body part $R_{l[k]}^i$. Hence, equation 5 shows the computations needed to transform an edge of the template mesh $x_{k,j}$ into an edge $v_{k,j}^i$ of the mesh i:

$$v_{k,j}^i = R_{l[k]}^i \cdot S_k^i \cdot Q_k^i \cdot x_{k,j} \tag{5}$$

The matrices S_k^i are learned from the training data set similar to equation 3:

$$argmin_{S_1^i \ldots S_P^i} \sum_k \sum_{j=1,2} ||R_k^i \cdot S_k^i \cdot Q_k^i \cdot x_{k,j}^i - v_{k,j}^i||^2 + \omega_s \sum_{k_1,k_2 adj} I(l_{k_1} = l_{k_2}) \cdot ||S_{k_1}^i - S_{k_2}^i||^2 \tag{6}$$

To simplify the computation of the shape deformations, the authors use Principal Component Analysis (PCA [15]). It projects the values of S_k^i into a linear subspace with less parameters. The authors argue that PCA can be used because body shape variation is consistent and not too strong. Hence, only few information will be lost when transforming the matrices into the linear subspace.

Limitations of the Model SCAPE focuses on representing muscle deformations resulting from body motion. Changes resulting from muscle activity or other factors are not covered. Furthermore, it is not possible to model correlations between body shape and muscle deformation due to the decoupling of the two models. For example, muscular people mostly have greater muscle deformation than others. In addition, it is assumed that the pose model is learned from scans of the same person. If there are different people in different poses, the pose model and the body shape model need to be trained iteratively. Moreover, SCAPE is not able to represent individual details like faces, hairstyles or dresses [1,6].

3.2 Poisson Mesh Reconstruction

This method creates a smoothed surface in form of a single mesh for a given set of points. Its fundamental idea is an indicator function that has value one inside the surface and value zero outside. Hence, its gradient is always zero except for points near the surface. The given points for which a surface should be constructed can be regarded as samples of this gradient. Therefore, the indicator function is chosen whose gradient best approximates the points [16].
Since this method has been designed for noisy point clouds as input, it is appropriate to this task. The result of the reconstruction is a smooth and hole-filled mesh with much detail. Even occluded or misaligned parts are improved.

4 Application of the SCAPE Model on Kinect Scans

In the original paper about SCAPE [5], the scans are obtained using a Cyberware WBX whole-body scanner. Thus, the resolution is far better than the one provided by the Kinect [17]. Nevertheless, Weiss et al. (2011) apply the model to low resolution data with noise from a single Kinect sensor and achieve quite good results [4]. The following paragraph describes the SCAPE model used for Kinect data.

Data capturing and body segmentation A person is captured four times from different angles: facing the camera, in profile, facing away from the camera and rotated 45 degrees between frontal and profile. At every scan, the person poses differently. Therefore, the same pose is never captured from multiple views. The combination of the scans leads to a good recovery of the shape. To segment the body from the surrounding environment, the depth map of the background is subtracted from the depth map of the person in front of the background. After this, a morphological opening operation is applied to remove small isolated false positives. This is a filter operation that can delete small artefacts in a picture. It consists of the morphological operation erosion that deletes isolated pixels, and the operation dilation that augments pixels to greater areas. Hence, isolated pixels are removed but the connected body parts are not separated.

Low Resolution and Noisy Data The method of [4] does not include additional steps to improve resolutions or noise. This is a significant difference to the other approaches presented in this term paper.

Non-rigid registration The authors do not describe further registration steps to those mentioned in the original paper about SCAPE [5].

Steps to obtain a 3D body model The captured data are used to learn the pose deformation model. The shape deformation model, on the other hand, is estimated using an aligned database of several thousand bodies. The resulting SCAPE model for one body contains 48 pose parameters per scan (192 at all since four scans are captured) and 60 shape parameters.
The parameter vector (denoted by θ) is chosen by fitting the 3D model to the scanned silhouettes:
On the one hand, the average depth error of all pixels is taken into account. It is calculated as the difference between the observed depth D_x of pixel x and the depth of the model silhouette when projecting it onto the screen $D_{x,t}(\theta)$:

$$E_d(\theta; U) = \frac{1}{|U|} \sum_{(x,t)\in U} \rho(D_{x,t}(\theta) - D_x) \tag{7}$$

t denotes the triangle corresponding to pixel x and set U consists of all pixel-triangle pairs. ρ should be a robust error-function.

On the other hand, the distance from the model silhouette $S(\theta)$ to the image silhouette T is computed using the Euclidean distance, and the distance from the image silhouette T to the model silhouette $S(\theta)$ is considered using the current relationships between the 2D and 3D points. This leads to the following equation:

$$E_s(S(\theta), T) = E_{uni}(S(\theta), T) + E_{uni}(T, S(\theta)) \tag{8}$$

Minimizing the distance from the image to the model ensures that all image measurements are considered in the model. Minimizing the distance vice versa ensures that visible body parts are explained by image evidence.
The computation of the model-to-silhouette distance is illustrated in figure 7.

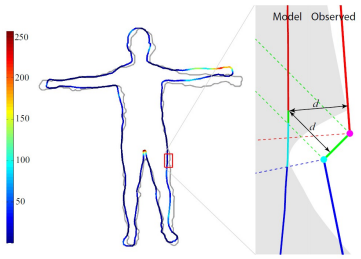


Fig. 7. Illustration of the distance when fitting the 3D model to the silhouette
left part: grey: observed silhouette, blue: silhouette of the 3D model
right part: the colors denote which part of the model should be mapped to which part of the observed silhouette, the grey shadow shows the effect of the applied changes in model parameters [4]

Finally, the combination of equation 7 and 8 leads to equation 9. Furthermore, a pose prior $E_{pose}(\theta)$ is added to penalize scan-to-scan variation of the body parts. Thus, it favors similar poses and helps to predict the location of possibly occluded limbs.

$$E_i(\theta; U_f(\theta_i)) = \sum_f E_d(\theta; U_f(\theta_i)) + \alpha \cdot \sum_f E_s(S_f(\theta), T_f) + \beta E_{pose}(\theta) \tag{9}$$

To estimate θ, iteratively the correspondences $U_f(\theta_i)$ are computed for every scan f and new model parameters θ_{i+1} are estimated by minimizing equation 9. The resulting body model is represented as a triangulated 3D mesh.

4.1 Results

For evaluation, four persons are scanned and their SCAPE bodies are estimated. To achieve measurements from the SCAPE bodies, a linear function from shape parameters to measurements is learned on the CAESAR dataset [14]. For comparison, manual measurements are taken. To compare the system to an application of the SCAPE model on data of higher quality, additional scans are captured

using a high-resolution Vitus laser scanner and SCAPE models are developed. The results are presented in figure 8. It is shown that the results of scans from the Kinect are only slightly worse than those of scans from the Vitus laser scanner.

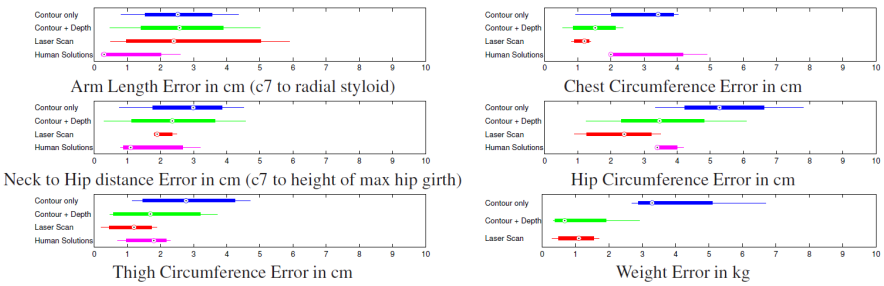


Fig. 8. SCAPE approach: Errors in the reconstructed bodies
contour only: the objective function only regards the distance from the model silhouette to the image silhouette and vice versa (Kinect data)
contour + depth: the objective function also takes the mean error of the depth difference into account (Kinect data)
laser scan: measurements of the SCAPE models of the Vitus laser scanner
human solution: measurement error between hand measurement and measurements calculated from the laser scans by a commercial scan measurement system (Human Solutions Anthroscan)

As mentioned in the preceding description, the resulting SCAPE model for one body contains about 200 parameters. Due to this, the process of building the model for one body takes a computation time of 65 minutes.

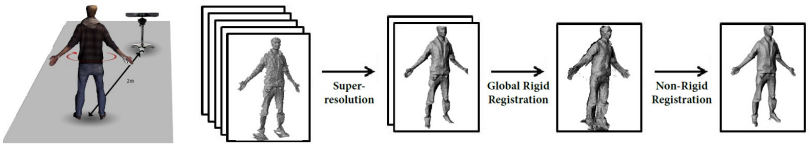
5 Reconstruction after Super-Resolution

In 2012, Cui et al. propose the use a single Kinect sensor to capture human beings [3]. They extend traditional approaches to overcome the second and third challenge (see 2.2) and obtain better results in combination with the Kinect.

5.1 Preprocessing and Mathematic Modeling

Figure 9 provides an overview of the different preprocessing steps before calculating the mathematical model.

Data capturing and body segmentation During the capturing process, the person stands two meters in front of a Kinect in a 'T'-pose and turns around. Every 0.5 seconds, a set of ten depth maps and ten color maps is captured. A



(a) Working setup (b) Processing pipeline

Fig. 9. Outline of the preprocessing steps
[3]

depth map contains the depth value and a color map stores the color (RGB value) of each pixel. The authors do not described how the body is segmented from the background.

Low Resolution and Noisy Data The challenge of low resolution and noise is met with a super-resolution algorithm. It aims at improving resolution and reducing noise while preserving important shape details. The following list provides the steps of the algorithm:

- All depth maps D_1 to D_{10} and color maps C_1 to C_{10} of one capturing set are aligned to the middle scan of the set using the optical flow (the estimated change between two scans)
- A new depth map H_l is obtained by combining the given depth maps and also regarding the color maps:

$$min_{H_l}\varepsilon_{data}(D_1, ..., D_{10}, C_1, ..., C_{10}, H_l) + \gamma\varepsilon_{reg}(H_l) \tag{10}$$

  ε_{data} measures the agreement of H_l with the given depth maps and maximizes the quality of the optical flow alignment. The use of the color maps in addition to the depth maps in the term ε_{data} leads to a smoother surface and preserves important shape details. The difference of the resulting depth map and the given depth maps is minimized. ε_{reg} smoothes the noisy depth data.

The new depth map contains less noise and a higher resolution (by a factor two in both X and Y dimensions). This is illustrated in figure 10.

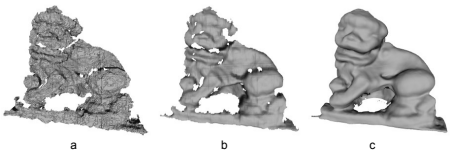


Fig. 10. a: raw data, b: super-resolution without color constraint, c: super-resolution with color constraint

Finally, the new depth map is converted into a 3D point cloud. Hence, the super-resolution step provides one 3D point cloud for each 0.5 seconds as output.

Non-rigid registration The challenge of movements between two subsequent scans is solved in global rigid and non-rigid alignment steps. A probabilistic approach is applied because this usually achieves satisfying results in the presence of noise. Each point cloud Y_f is considered to be generated by a Gaussian Mixture Model (GMM [18]). There is one Gaussian per point in the cloud. The Gaussian Mixture Model provides the probability that a point x is generated by the point cloud Y_f. The registration consists of two steps:

- A transformation $M_f^{i(n)}$ is defined for each point $y_{f,n}$ (n : point index, f: scan index). It denotes the motion of the point in the scan from its original position. To ensure that rigid parts stay connected after the transformation, neighboring rigid parts are detected and joined together with the same label $i(n)$.
- The labels $i(n)$ and the transformations are obtained: The parameters are chosen so that the distance of the alignments is minimized (equation 11). M denotes the transformation set and L the labels for all points.

$$argmin_{M,L} E_{data}(M, L) + \lambda E_{reg}(M, L) \tag{11}$$

  E_{data} is the distance between aligned points in the scans and E_{reg} is a smoothing constraint for the labels and neighboring transformations. For minimization, an iterative procedure similar to Expectation Maximization is used. The expectation step estimates the Gaussian Mixture probabilities based on the parameters from the previous iteration (or initialization), while the maximization step calculates new parameters which minimize equation 11. These two steps are repeated until convergence.

Due to the adaption of the GMM parameters, the input point clouds are aligned to each other. Hence, the output of this step is a single 3D point cloud containing the information of all scans. Especially in the arm and hand regions, the alignments are quite accurate.
Figure 11 compares the non-rigid registration results on raw Kinect scans with those on super-resolved scans. Additionally, a comparison to a traditional registration algorithm (ICP, see section 7) is shown.

Steps to obtain a 3D body model After the registration step, Poisson mesh reconstruction is applied to each view.

5.2 Results

The quality of the 3D bodies is measured in comparison to a laser-scanned ground truth model. For evaluation, eight persons are scanned and their 3D bodies are generated. Table 1 provides the average results.

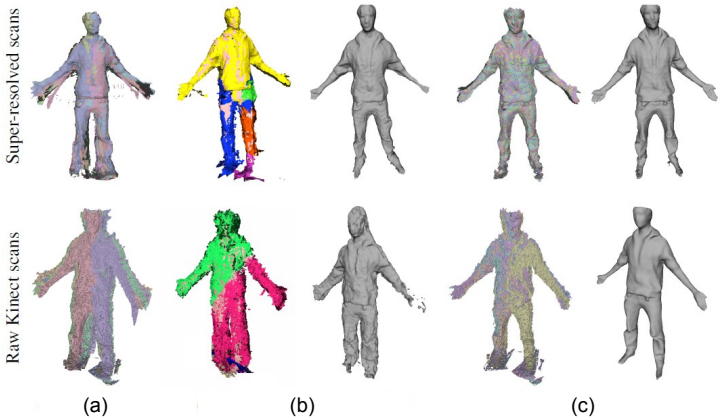


Fig. 11. Comparing global non-rigid registration for super-resolution data (top row) and raw Kinect data (bottom row)
(a): rigid alignment, (b): using a traditional algorithm (ICP),
(c): using the algorithm from this paper [3]

Table 1. Super-resolution approach: Resulting quality

error in neck-to-hip distance	2.1 cm
error in shoulder width	1.0 cm
error in arm length	2.3 cm
error in waist circumference	3.2 cm
error in hip circumference	2.6 cm
error in leg length	3.1 cm

The super-resolution step needs a computation time of 28 seconds, while the rigid part of the alignment lasts 110 seconds and the non-rigid part 620 seconds. This is the most time consuming step since the Poisson reconstruction only needs 68 seconds. All steps together require a time of 826 seconds (13.8 minutes). Finally, the authors claim that the system is able to capture impressive 3D human shapes with detailed geometry, such as face structure and clothes.

6 Reconstruction from Multiple Kinects

Tong et al. present a method for body scanning employing multiple Kinects in 2012 [1].

6.1 Preprocessing and Mathematic Modeling

Figure 12 provides an overview of the different steps used in this approach.

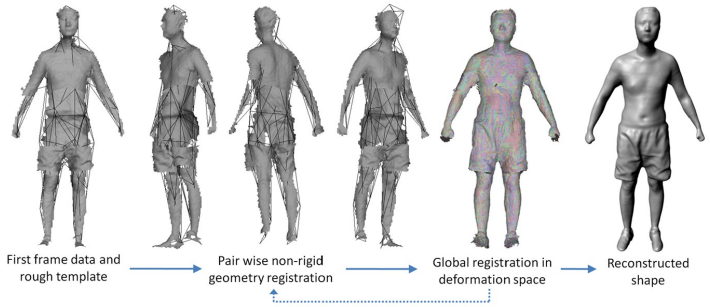


Fig. 12. Overview of the reconstruction algorithm [1]

Data capturing and body segmentation As described before, the distance between a single Kinect that captures the whole body and the person should be two or three meters. However, because of the low resolution (especially several meters away from the sensor), not much geometry can be stored in the depth map. Therefore, the authors use three Kinects. Each Kinect scans a different part of the body. This allows the person to stand in a distance of only one meter in front of the Kinects which leads to a higher quality of the data. However, if the Kinects scan the body simultanously and the scanned parts overlap each other, multiple lasers will cause interference problems. A possible solution would be the scanning at different times but this would reduce the frame rate. Hence, the authors use two Kinects to scan the upper and lower part of the front of the body without overlapping regions and the third Kinect to scan the missing part of the body from behind. This is illustrated in figure 13.

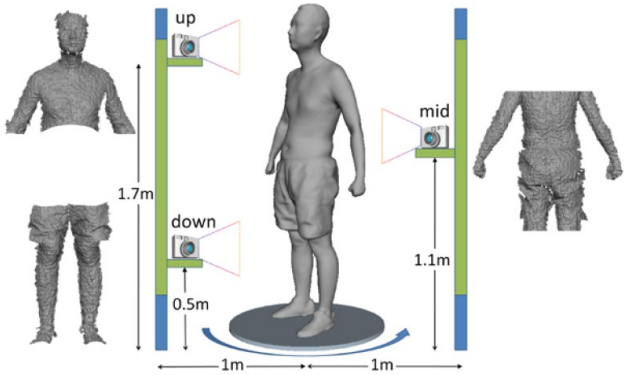


Fig. 13. The setup of the scanning system [1]

During the scan, the person stands on a turnable that is rotated 360 degrees. A framerate of 15 Hz is used which provides RGB images of rather high resolution. The data from the three sensors are synchronized and calibrated automatically using Multi-camera self-calibration. For this process, a laser pointer is moved through the room creating a set of points. Their projections are detected in the camera images and used for calibration of the sensors.[1] To segment the body from the background, a depth and color threshold method is applied. Finally, the mesh is simplified to $\frac{1}{10}$ of all vertices to reduce computation time and space.

Low Resolution and Noisy Data Noise is reduced by applying the Laplacian smoothing method [19]. The vertices of the mesh X are moved in the direction of the Laplacian as shown in equation 12.

$$\frac{\partial X}{\partial t} = \lambda L(X) \tag{12}$$

λ defines the speed of the movement. The Laplacian L can be approximated in many different ways, for instance, as shown in equation 13 with x_k representing the vertices of the mesh and $N(i)$ the set of m neighbors of vertex i.

$$L(x_i) = \sum_{j \in N(i)} \frac{1}{m}(x_j - x_i) \tag{13}$$

Advantages of this method are the perpetuation of the number of vertices and the connectivity of the mesh.

Non-rigid registration The challenge of non-rigid registration is met by a two-phase method. It consists of the following two steps that are repeated until convergence:

- pairwise alignments between the scans:
  From the first scan, a template is constructed. To reduce computation cost, it is simplified to about 50-60 nodes. Due to noise and influence of clothes, it might not be very accurate and could not be used to align the other scans by geometry fitting. Instead, it is deformed to match the other scans. This deformation, which consists of a rotation and a translation, is applied pairwisely in a cycle. Figure 14 illustrate this pairwise alignment.
  The rotations and transformations are computed by solving the following function:
  $$min_{(R_i^k, t_i^k)}(E_{reg} + w_{rot}E_{rot} + w_{con}E_{con}) \tag{14}$$
  E_{reg} ensures the smoothness of neighboring deformations:
  $E_{reg} = \sum_k \sum_{j \in N(k)} ||R_i^k(v_i^j - v_i^k) + v_i^k + t_i^k - (v_i^j + t_i^j)||$ (where v_i^j and v_i^k are neighboring nodes). E_{rot} guarantees that the transformation is a rotation:
  $E_{rot} = \sum_k (c_1 \cdot c_2)^2 + (c_1 \cdot c_3)^2 + (c_2 \cdot c_3)^2 + (c_1 \cdot c_1 - 1)^2 + (c_2 \cdot c_2 - 1)^2$

[1] http://cmp.felk.cvut.cz/~svoboda/SelfCal/Publ/selfcal.pdf

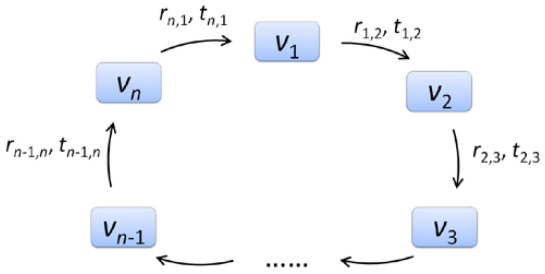


Fig. 14. Pairwise alignment
v_i denotes the set of nodes of the template deformed to match scan i [1]

with c_i representing a column of the 3×3 vector R. E_{con} matches feature correspondences in the color maps which are obtained using optical flow: $E_{con} = \sum_l ||\overline{v}_k^{l1} - v_{k+1}^{l2}||$ with v_{k+1}^{l2} and v_k^{l1} corresponding feature points in two successive scans and $\overline{v}_k^{l1}$ the deformation result of v_k^{l1}.
As figure 15 shows, the results of pairwise registration by themselves are not satifactory: There are loop closure problems (see 2.2) in the head area because the first and last scan do not match. Furthermore, occlusions in the scanned data lead to misalignments in the arm area. Therefore, another step called global alignment is applied.

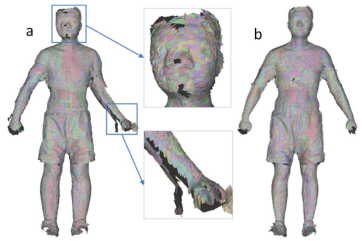


Fig. 15. Registration results
a: after pairwise registration; b: after global registration

– global alignment:
This step aims at minimizing the errors described above. It changes the deformations $f_{i,j}$ obtained by the previous step to deformations $\widehat{f}_{i,j}$ regarding two conditions. First, the cycle should be consistent. That means, that the chain of all deformations of the cycle should lead to the identity:

$$\forall i : \widehat{f}_{1,2} \cdot \widehat{f}_{2,3} \cdot ... \cdot \widehat{f}_{n-1,n} \cdot \widehat{f}_{n,1} = I \tag{15}$$

Second, it is assumed that $f_{i,j}$ is mostly correct. Thus, the change made in this step should be minimized:

$$min \sum w_{i,j}^2 ||\widehat{f}_{i,j} - f_{i,j}||^2 \tag{16}$$

$w_{i,j}$ denotes a confidence measure of the deformation $f_{i,j}$. The authors use $w_{i,j} = \frac{1}{Dist(f_{i,j}(M_i), M_j))}$ with $Dist$ being the average distance of corresponding point pairs in the two meshes.
This extension solves the loop closure problem and handles misalignments due to occlusions.

Steps to obtain a 3D body model Finally, a whole-body model is obtained by applying the Poisson surface reconstruction method [16]. An example of a reconstructed model is shown in figure 16.

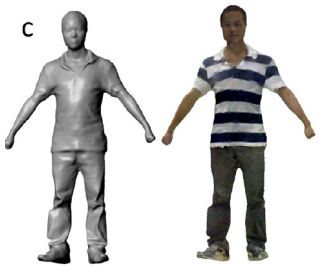


Fig. 16. the reconstructed model [1]

6.2 Results

The authors compare measurements calculated by the models with measurements of the corresponding real persons. For evaluation, six people are scanned and their 3D bodies are computed. Table 2 shows the average results.

Table 2. Multiple Kinect approach: Resulting quality

error in neck-to-hip distance	2.5 cm
error in shoulder width	1.5 cm
error in arm length	3.0 cm
error in waist circumference	6.2 cm
error in hip circumference	3.8 cm
error in leg length	2.1 cm

This approach is capable of modeling personalized details such as faces, clothes and hairstyles.

The time required to construct a human model is about 6 minutes: The data preprocessing takes 1.6 minutes, the registration 3.8 minutes and the reconstruction 0.5 minutes.

7 Reconstruction for Use of 3D Animated Avatars

Charpentier presents a way of generating 3D animated avatars that are copies of real persons in his dissertation (2011) [6]. He uses the Kinect as a 3D scanner.

7.1 Preprocessing and Mathematic Modeling

Data capturing and body segmentation During the capturing process, the person stands on a turnable chair that is rotated four times by 90 degrees. At each rotation step, the subject is captured four times from different heights. The author does not mention which heights are chosen and from what distance the person is captured.
The method of separating person and background is not mentioned.

Low Resolution and Noisy Data To reduce noise, the moving least square method [20] is applied. It approximates (and therefore smoothes) the surface in the scan with a polynomial that minimizes the weighted squared distance of the polynomial and the scan.

Non-rigid registration The author applies the Iterative Closest Point (ICP) algorithm [21,22] because of its popularity and simplicity. It consists of three steps to align one point cloud (source) to another (target) using a linear transformation:

- For each point of the source, the nearest neighbor in the target is computed.
- A matrix is estimated that transforms all points of the source to their neighbors in the target using a mean square function.
- The transformation is applied and the error is computed.

These steps are iterated until the error reaches a certain threshold or until a predefined number of iterations. Since this algorithm will find a local minimum of the error, its result is only good if the transformation is not too considerable. For better results, a prior transformation is estimated between manually placed markers on the source and target point clouds. Then, ICP is used to improve this transformation.

Steps to obtain a 3D body model To obtain a triangle mesh, Poisson surface reconstruction is used.

7.2 Results

Charpentier does not provide any error or time measurements. The author concludes that the method suffers from inferior quality of the point cloud (especially at the hand area) and from clothes which occlude limbs. Figure 17 examplifies this.

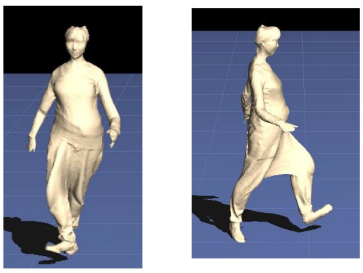


Fig. 17. Avatar approach: The reconstructed model [6]

8 Comparison of the Models

All presented papers share the challenge of coping with low resolution and noisy Kinect data, and with movements between two successive scans. Another similarity of all approaches is that they scan the person several times from different angles. However, the number of Kinects and the distance of the person to the device changes. If a single Kinect is used, the distance between the person and the scanning system needs to be about three meters to capture the entire body in the field of view. On the other hand, using three Kinects allows the person to be scanned in a distance of one meter. Due to the non-uniformly distributed resolution levels of depth (see section 2.2), this leads to better depth information. In addition, the color maps contain more information. Picture 18 illustrates this.

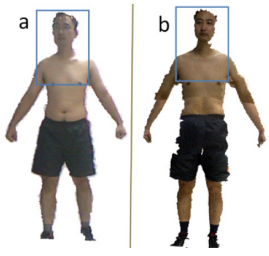


Fig. 18. Data obtained using one Kinect (a) and three Kinect (b) [1]

The multiple Kinect approach [1] is the only one presented in this term paper that uses 3 Kinects. Furthermore, a lower framerate is used for capturing: As mentioned in section 2.1, the lower framerate of 15 Hz leads to a resolution of 1280×1024, while the higher framerate of 30 Hz only results in a resolution of 640×480. Nevertheless, the error results are not superior to the results of the super-resolution approach [3] (apart from the error of the leg length). This might be due to the reduction of the body mesh to $\frac{1}{10}$ of all nodes.
Another difference is whether the person turns around himself / herself or is rotated on a turnable. The turnable reduces the movement of the scanned person between two successive scans and consequently, simplifies the registration step. Thus, Tong et al. (multiple Kinect approach) can apply a pairwise registration and Charpentier (avatar approach) [6] can use the ICP method that is rather simple but would not be applicable for global alignment with movements.
According to [4], the standard measurements to evaluate a body model are arm length and chest circumference. However, the evaluation metrics differ among the papers. Some compare the model results to manual measurements, others use a laser scanned ground truth model of high quality, while Charpentier does not provide measurements at all. Hence, the comparability is limited. Nevertheless, the super-resolution approach performs better than the multiple Kinect approach in almost every measurement. Weiss et al. (SCAPE approach) measure further body parts. Regarding the comparable body parts, the method using SCAPE performs similar to the other approaches in average. The largest difference in errors can be detected in the waist circumference. The results of the multiple Kinect approach are worse than the measurements of the other methods. According to Tong et al., this is caused by the clothes of the subjects. Thus, although dresses can be modelled, they worsen the quality of the result.
In [5], the authors describe the advantages of SCAPE: The model copes well with shoulder deformations, bulging of the biceps and twisting of the spine. It has acceptable performance regarding elbows and knees. However, the greatest disadvantage is its limitation to naked human bodies. To model clothes or other details such as hairstyles or faces, one of the other models has to be applied. On the other hand, the SCAPE model is quite appropriate if the recovered shape should be applied to a new pose that has not been captured. This is possible because of the separation of the body shape model and the pose model. The other approaches lack this ability.
Some of the approaches apply a template or pose prior. On the one hand, this simplifies the registration step because not every scan needs to be aligned to all the others. On the other hand, the template or prior needs to be obtained first. This can be time consuming or requires prior knowledge that might not always be available.
The number of parameters of the models used to build a 3D model of the human's shape varies. Therefore, the computation time of the methods is different, too. The SCAPE model requires the longest time (65 minutes). Two of the papers show that the registration step is the most time consuming. Thus, to reduce computation time significantly, one should start improving this step. For instance,

Tong et al. (multiple Kinect approach) reduce the time necessary for registration by limiting the nodes that are aligned to 50-60. Their time for registration is still about 63.3% of the total computation time but it is about 69% superior to the registration time of Cui et al (super-resolution approach). On the other hand, the quality of the results is only about 17% worse in average. Hence, one might find it worth to reduce computation time like this. Nevertheless, the error in waist circumference of Tong et al. is almost twice as high as the error of Cui et al.

9 Conclusion

In this term paper, four state-of-the-art approaches to reconstruct 3D models of human bodies, which have been scanned with the Kinect, are presented. The Kinect sensor is monocular and its scans are rather noisy, especially when the person stands some meters away from the camera in order to be captured entirely. It can be observed that all presented approaches need to solve the same challenges. All the authors present the capturing process, improve the noise and low resolution of the scanned data and register scans made from different angles to each other or to a template. After this preprocessing, they apply their particular mathematical model to obtain a 3D shape.
If the model should be applicable to new poses that have not been scanned, the SCAPE model should be used. Its greatest disadvantage is its lack of modeling clothes and personal details of the body. To take such attributes into account, another model should be applied. If computation time is limited, the multiple Kinect approach should be used because it takes only about six minutes. Instead, if the quality of the obtained body shape should be as good as possible, the super-resolution approach or the SCAPE method should be chosen. However, the multiple Kinect approach is not much worse regarding the measurement results. Nevertheless, the error difference in waist circumference is quite significant. Thus, if the modelled persons should wear clothes, the super-resolution approach should be chosen although it needs twice as long as the multiple Kinect approach. This might not be so important since both methods are not applicable for realtime applications. If, on the other hand, no clothes are required and time is limited, the multiple Kinect technique can be recommended.

9.1 Perspective: Using the Kinect to Track a Person

To track a person in realtime (e.g. as ground truth for another scanning device), none of the presented approaches seems to be suitable since their computation time is too high. Nevertheless, for obtaining the position of the person at a given time and maybe recognize his or her current pose (to estimate what he or she is doing), it might not be important to obtain a 3D body model of high resolution. Hence, one might consider an approach similar to the multiple Kinect method which simplifies the mesh to reduce computation time. If measurements are not that important, the mesh can be simplified even further (by reducing

nodes, especially when the person is far away from the sensor and details are not visible). As a result, the preprocessing and registration time can be reduced. Additionally, as applied in many other online applications, the 3D body could be created very imprecisely at the beginning and more information could be added over time. A step like this might be necessary in this scenario since all presented methods in this term paper perform offline computations. To sum up, the multiple Kinect approach is recommendable as initial point for an online tracking algorithm using the Kinect.

References

1. Tong, J., Zhou, J., Liu, L., Pan, Z., Yan, H.: Scanning 3d full human bodies using kinects. Visualization and Computer Graphics, IEEE Transactions on **18**(4) (2012) 643–650
2. `www.xbox.com/KINECT`
3. Cui, Y., Chang, W., Nöll, T., Stricker, D.: Kinect avatar: Fully automatic body capture using a single kinect. In: ACCV. (2012)
4. Weiss, A., Hirshberg, D., Black, M.J.: Home 3d body scans from noisy image and range data. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 1951–1958
5. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM Transactions on Graphics (TOG). Volume 24., ACM (2005) 408–416
6. Charpentier, G.: Accurate 3d rigged avatar generation with a kinect. (2011)
7. Khoshelham, K.: Accuracy analysis of kinect depth data. In: ISPRS workshop laser scanning. Volume 38. (2011) 1
8. `http://campar.in.tum.de/twiki/pub/Chair/TeachingSs11Kinect/2011-DSensors_LabCourse_Kinect.pdf`
9. Stenzel, H.: Motion capturing unter verwendung der microsoft kinect. (2011)
10. `http://openkinect.org`
11. Pheatt, C., Ballester, J.: Using the xbox kinect sensor for positional data acquisition. (2011)
12. Schlüter, D.: Erkennung des loop-closure-problems mit hilfe von bildvergleichen (2010) Bachelorarbeit Universität Bielefeld.
13. Anguelov, D., Srinivasan, P., Pang, H.C., Koller, D., Thrun, S., Davis, J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. Advances in neural information processing systems **17** (2005) 33–40
14. Allen, B., Curless, B., Popović, Z.: The space of human body shapes: reconstruction and parameterization from range scans. In: ACM Transactions on Graphics (TOG). Volume 22., ACM (2003) 587–594
15. Daultrey, S.: Principal components analysis. Geo Abstracts Limited Norwich (1976)
16. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. (2006)
17. `http://www.cyberware.com/products/scanners/wbx.html`
18. Reynolds, D.: Gaussian mixture models. Encyclopedia of Biometric Recognition (2008) 12–17
19. Bray, N.: Notes on mesh smoothing. `http://mgarland.org/class/geom04/material` (2004)

20. Breitkopf, P., Naceur, H., Rassineux, A., Villon, P.: Moving least squares response surface approximation: Formulation and metal forming applications. Computers & structures **83**(17) (2005) 1411–1428
21. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Transactions on pattern analysis and machine intelligence **14**(2) (1992) 239–256
22. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International journal of computer vision **13**(2) (1994) 119–152
23. `http://openni.org/`
24. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE (2011) 1–4

Regelungstheorie für Rechensysteme

Lixin Su

Karlsruhe Institute of Technology
Institute of Telematics - Pervasive Computing Systems
Betreuer: Yong Ding
lixin.su@student.kit.edu

Abstract. Diese Seminararbeit behandelt die Regelungstheorie für Rechensysteme (Control Theory of Computing Systems). Im ersten Teil werden jeweils die Grundlagen des Rechensystems und der Regelungstheorie erklärt. Danach werden die Anforderungen des Regelungssystems sowie ein Anwendungsbeispiel mit IBM Lotus Domino Server vorgestellt. Als nächstes wird der Aufbau des Regelkreises analysiert. Der zweite Teil dieser Arbeit behandelt die aktuellen Anwendungen der Regelungstheorie für Rechensysteme. Zuerst werden die Anwendungen der Regelungstheorie für Rechensysteme zusammengefasst und des Weiteren werden als Schwerpunkt die PID-Regler erklärt. Schließlich werden die Vor- und Nachteile mit/ohne Regelungssystem diskutiert und ein Fazit gegeben.

1 Einleitung

1.1 Motivation

Heutzutage spielen die Rechensysteme eine sehr wichtige Rolle in unserem Leben. Fast alle Bereiche werden dadurch beeinflusst, wie z.B. Internet, Kommunikationstechnik, Verkehr ebenso wie unser Lebensmittel. Die Rechensysteme sind ein Schlüssel zur Verbesserung der Produktivität und Sicherheit, des Wohlbefindens und der Gesundheit von der Herstellung, Landwirtschaft, Kommunikation, Energie bis hin zur Medizin. In der Zukunft werden Rechensysteme weiterhin ein mächtiges Werkzeug für die Entwicklung neuer Bereiche und revolutionärer Technologien sein. Allerdings sind die Rechensysteme mit 50 Jahren noch eine sehr neue Sache für die Geschichte der Menschheit und es gibt noch viele Mängel, so dass die Menschen immer einen großen Anreiz haben, neue Geräte und Funktionen von Rechensystem zu erfinden und verbessern. Man kann sich vorstellen, dass die Rechensysteme in der Zukunft eine wichtige Rolle spielen werden.

Regelungstheorie für Rechensysteme sind die Anwendung der Regelungstheorie in den Rechensystemen(p. 11) [1]. Sie werden in vielen Bereichen wie z.B. Datennetze, Betriebssysteme, Middleware, Multimedia und dynamisches Power-Management verwendet(p. 11) [1]. Das Konzept der Regelungstheorie für Rechensysteme basiert auf einer Output-Input-Beziehung, die durch das Rechensystem gesteuert wird. Ein

Regler kann eingesetzt werden, damit die Rechensysteme stabil und wirkungsvoll arbeiten. Nach der Einsetzung des Reglers können die CPU-Auslastung, Speicher und Ressourcennutzung der Rechensysteme verbessert werden.

1.2 Rechensystem

Ein Rechensystem ist ein System von miteinander verbundenen Computern, zentralen Datenbanksystemen und diversen Peripherie-Geräten wie z.B. Drucker, die zusammen genutzt werden [10]. Jeder Computer kann nicht nur unabhängig arbeiten, sondern auch mit anderen Computern oder externen Geräten kommunizieren [10].

Die Rechensysteme werden nach der Anforderung von Menschen aufgebaut. Sie funktionieren nach dem Wunsch, der mittels eines Inputs in die Rechensysteme eingegeben werden [7]. Nach der Verarbeitung kann man einen Output als Antwort von den Rechensystemen bekommen.

1.3 Regelungstheorie für Rechensysteme

Ein Rechensystem hat das Problem, dass es die Aufgabe nicht selbst kontrollieren kann(p. 13ff) [1]. Falls beispielsweise plötzlich viele große Aufgaben kommen, arbeitet das Rechensystem sehr langsam und es gibt eine Betriebsunterbrechung. In diesem Fall ist eine Regelungstheorie für Rechensysteme sehr sinnvoll, weil das Rechensystem durch einen voreingestellten Regler automatisch gesteuert wird.

„Die Regelungstheorie ist ein Teilgebiet der angewandten Mathematik“ [11]. Sie betrachtet dynamische Systeme, deren Verhalten durch sogenannte Eingangsgrößen, die von Menschen gegeben hat, beeinflusst werden kann[3]. Typische Fragestellungen in der Regelungstheorie betreffen die Analyse eines gegebenen Systems sowie dessen gezielte Beeinflussung durch Vorgabe geeigneter Inputs. Die praktischen Fragen lauten beispielsweise:

- Ist das Regelsystem stabil(p. 8) [1]?
- Bleiben alle Systemvariablen in den kontrollierenden Bereichen(p. 8) [1]?
- Ist es möglich, einen gegebenen Zielzustand zu erreichen (p. 8) [1]?

Um die erste Frage „Ist das Regelsystem stabil?“ zu erklären, sollte man Mitkopplung und Gegenkopplung im Regelsystem kennenlernen. Bei der Mitkopplung kommt eine Rückführung der Ausgangsgröße im Zusammenspiel mit verstärkenden Elementen des Systems zum Tragen(p. 5 ff) [1]. Mitkopplung findet man oft bei Wachstumsprozessen. Bei der Gegenkopplung handelt es sich um eine Rückführung des Ausgangssignals mit negativem Vorzeichen. Diese negative Rückführung wirkt der äußeren Anregung entgegen und führt zu einer sich verringernden Zustandsänderung. Damit ein Regelungssystem stabil ist, wird eine Gegenkopplung angewendet, weil eine Gegenkopplung die Störungen im Regelungssystem verkleinern kann(p. 5 ff) [1]. Die Mitkopplung kann jedoch die Störungen im

Regelungssystem vergrößern, deswegen kann man kein stabiles Regelungssystem bekommen.

Um die zweite Frage „Bleiben alle Systemvariablen in den kontrollierenden Bereichen“ zu erklären, sollte man die Art des Regelkreises kennenlernen. Es gibt den **offenen Regelkreis** (Open-Loop) und den **geschlossenen Regelkreis** (Closed-Loop) (Fig. 1). Im Gegensatz zur geschlossenen Regelkreis fehlt beim offenen Regelkreis die Rückkopplung des Outputs auf den Eingang, wodurch kann der offene Regelkreis nicht auf Outputs reagieren (p. 10) [1]. Der geschlossene Regelkreis ist ein Vorgang in Systemen, in denen Wechselwirkung stattfindet, und bei der einen prinzipiell veränderlichen Größe in der Regel automatisch konstant oder annähernd konstant gehalten wird(p. 10) [1]. In dieser Arbeit soll der geschlossene Regelkreis als Schwerpunkt weiter behandelt werden.

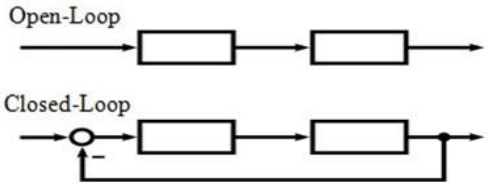


Fig. 1. Open-Loop vs. Closed-Loop

Die dritte Frage ist „Ist es möglich, einen gegebenen Zielzustand zu erreichen?“ Der offene Regelkreis und der geschlossene Regelkreis haben das gleiche Ziel: Eine Anlage zu automatisieren und den gegebenen Zielzustand zu erreichen. In einem offenen Regelkreis wird der Input durch Regler verändertet, damit das Rechensystem den Zielzustand erreichen kann. Durch diese Rückkopplungen im geschlossenen Regelkreis kann das Rechensystem stabil angepasst werden(p. 10) [1]. Somit kann der Output des Regelsystems die Anforderungen des Benutzers erfüllen.

Im welchem Bereich der Rechensysteme wird Regelungstheorie angewendet?

Um ein Rechensystem stabil zu halten, wird die Regelungstheorie angewendet. „Seit 1990 wird an der Anwendung der Regelungstheorie für verschiedene Rechensysteme geforscht, insbesondere in den Bereichen **Betriebssysteme für Datennetzwerke**, **Middleware, Multimedia und Dynamisches Power-Management**“ (p. 11) [1].

„Im Bereich der Betriebssysteme für Datennetzwerke wird die Regelungstheorie für Rechensysteme angewendet, um Probleme des Flow-Managements zu lösen, z. B. Rate Allocating Server, der den Fluss der Warteschlangen von Paketen regulieren kann“(p. 11 f) [1].

„Middleware ist der neueste Bereich, wo die Regelungstheorie für Rechensysteme angewendet worden ist. Middleware sind Software-Systeme, die die Entwicklung von robusten und Anwendungen auf Unternehmensebene erleichtern“ (p. 12) [1]. Es gibt

drei weitere Teilbereiche von Middleware: Anwendungsserver (z.B. Apache HTTP Server), Datenbank-Management-Systeme (z.B. IBM Universal Database Server) und E-Mail-Server (z.B. IBM Lotus Domino Server) (p. 12) [1].

Management von **Multimediaströmen** ist auch ein Schwerpunkt für die Anwendung der Regelungstheorie für Rechensysteme. „Die Herausforderung dabei ist, dass Endbenutzer-Performance auf den Empfang eines regelmäßigen Flusses von korrelierten Datenströmen angewiesen ist. Die zugrunde liegenden Systeme sind die Basis." (p. 12) [1]. Eine Lösung hierfür ist, Prioritäten in Übereinstimmung mit dem gewünschten Dienst zu regulieren(p. 12) [1].

„Dynamisches Power-Management ist der letzte Bereich für die Anwendung der Regelungstheorie für Rechensysteme" (p. 13) [1]. Die zentrale Frage des dynamischen Power-Managements ist, wie die Energie innerhalb von Rechensystemen verwaltet wird, damit die Kosten reduziert werden(p. 13) [1].

2 Anforderungen an Rechensysteme

Es gibt mehrere Anforderungen an die Rechensysteme, wobei die verschiedenen Teile der Rechensysteme berücksichtigt werden soll, z. B. CPU-Auslastung und Speicher-Verbesserung(p. 9) [1]. In dieser Arbeit wird diskutiert, welche Anforderungen für ein Rechensystem bzgl. der Anwendung eines Reglers wichtig sind. Jede Anforderung wird basierend auf einem Rechensystem erklärt, damit dies besser nachvollzogen werden kann.

2.1 CPU-Auslastung im IBM Lotus Domino Server

CPU-Auslastung ist ein wichtiger Indikator, ob das Rechensystem stabil und effizient ist (p. 8f) [1]. Als Beispiel wird hier die Anforderung im Rechensystem IBM Lotus Domino Server erklärt, weil durch das Einstellen der MaxUsers Parameter im IBM Lotus Domino Server die CPU des Rechensystems ausgenutzt werden kann.

Die IBM Lotus Domino Server besteht aus drei Teilen: Notes Clients, Notes Server und Server Log (p. 13) [1]. Normalerweise gibt es mehrere Notes Clients in einem Rechensystem und sie können durch Remote Procedure Calls (RPCs) mit dem Notes Server kommunizieren [8]. Der Vorgang wird durch Server Log gespeichert(p. 13) [1]. Der wichtigste Punkt für CPU-Auslastung ist die RIS (Nummer der RPCs), wenn die RIS zu hoch ist, werden die CPUs übermäßig verwendet, dann können sie nicht vernünftig genutzt werden.

Um das Problem zu lösen, hat IBM Lotus Domino Server eine Stellgröße MaxUsers eingesetzt(p. 13) [1]. Die Stellgröße begrenzt die maximale RIS und die übrigen RPCs werden etwas später verarbeitet, damit die CPU immer in einer stabilen

Situation bleibt (p. 14) [1]. In **Fig. 2** sieht man eine Notiz der CPU eines Rechensystems. Die horizontale Achse stellt die Zeit und die vertikale Achse die Performance der CPU dar. 300 Sekunden lang ist die CPU mit einem Regler in einer stabilen Situation. Wenn das Rechensystem in einer stabilen Situation bleibt, dann kann das Rechensystem auch eine bessere Leistung erbringen. Nach 300 Sekunden ist die CPU ohne Regler in einer instabilen Situation, wodurch das Rechensystem nicht selbst kontrollieren wird(p. 13) [1]. Wenn der Wert der CPU 1 ist, arbeitet das Rechensystem sehr langsam, wenn der Wert der CPU 0 ist, wird Zeit verschwendet.

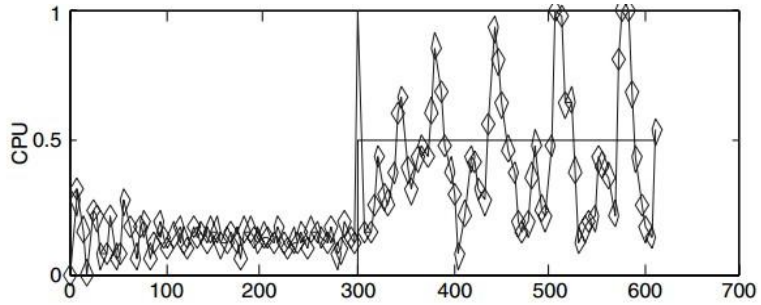


Fig. 2. CPU-Auslastung (p. 9) [1]

2.2 Speicher-Auslastung im Apache HTTP Server

„Der Apache HTTP Server ist ein quelloffenes und freies Produkt der Apache Software Fundation und der meistbenutzte Webserver im Internet. Der Webserver ist ein wesentlicher Bestandteil der Verteilung von Informationen und im elektronischen Geschäftsverkehr“ [9]. In einer typischen Konfiguration interagieren Endbenutzer mit Clients, die wiederum Hypertext Transfer Protocol (HTTP) Nachrichten an einen oder mehrere Webserver senden(p. 16) [1]. Es gibt zwei wichtige Anforderungen für den Apache HTTP Server, nämlich die CPU- und die Speicher-Auslastung.

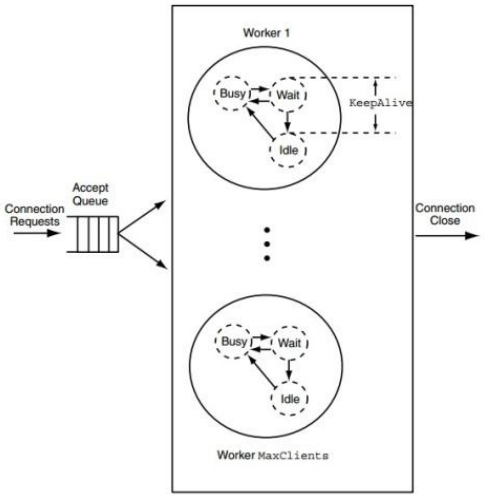


Fig. 3. Apache Architektur und Session-Fluss (p. 17) [1]

In **Fig.3.** sind die Struktur und ein Session-Fluss eines Apache HTTP Server dargestellt. Der Apache HTTP Server wird wie ein Pool von Arbeitern aufgebaut(p. 17) [1]. Anfragen werden in einer Warteschlange platziert, wo sie auf einen verfügbaren Arbeiter warten müssen. Ein Arbeiter ist verfügbar, wenn er sich im "idle" Zustand befindet, während der Arbeiter eine Anfrage verarbeitet, ist er im "busy" Zustand(p. 17) [1]. Das weit verbreitete Protokoll HTTP sorgt für dauerhafte Verbindungen, d. h. der Arbeiter schließt die Verbindung nicht, nachdem die Anfrage verarbeitet wurde. Stattdessen tritt der Arbeiter in den Zustand "warten" ein und die Verbindung bleibt offen, sodass nachfolgende Anfragen aus dem gleichen Client effizienter verarbeitet werden können (p. 17) [1]. Das Problem ist, dass während der Arbeiter im Wartezustand ist, er andere Anfragen von anderen Clients nicht verbinden und verarbeiten kann.

Um das Problem zu lösen, werden im Apache HTTP Server zwei Parameter eingesetzt:

- ein Parameter ist MaxClients, er begrenzt die Größe des Pools von Arbeitern, dadurch wird die Verarbeitungsgeschwindigkeit des Servers eingeschränkt (p. 18) [1]. Deswegen kann der Apache HTTP Server mit einem höheren MaxClients Wert mehrere Anfragen gleichzeitig verarbeiten. Wenn der MaxClients Wert jedoch zu groß ist, wird der Verbrauch der CPU und Speicher-Ressourcen übermäßig belegt, wodurch die Leistung des Rechensystems schlecht wird.

- der andere Parameter ist KeepAlive, er steuert die maximale Zeit, die ein Arbeiter im Wartezustand bleibt(p. 18) [1]. Wenn KeepAlive zu groß ist,

werden CPU und Speicher nicht ausgenutzt, da sich die neue Kunden nicht mit dem Server verbinden können.

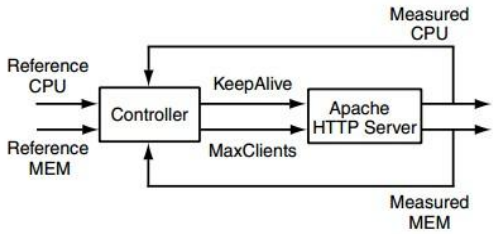


Fig. 4. Regelkreis des Apache HTTP Servers (p. 18) [1]

Das Problem beim Apache HTTP Server ist, wie man CPU und Speicher-Nutzung verbessern kann. **Fig.4.** zeigt einen Regelkreis des Apache HTTP Servers. Der Administrator gibt die gewünschten Werte für CPU und Speichernutzung an, damit kann der Regler die beiden Referenz-Eingänge sowie gemessene CPU und Speicher Nutzungen verwenden, um den passenden Wert von KeepAlive und MaxClients einzustellen(p. 18) [1]. Falls der Wartezustand groß ist, wird MaxClients groß und KeepAlive kleiner eingesetzt, damit die Ressourcen effizient ausgenutzt werden(p. 18) [1].

3 Anwendungsmöglichkeiten der Regelungstheorie für Rechensysteme

3.1 Zusammenfassung der Anwendungsmöglichkeiten

In diesem Kapitel werden einige Anwendungsmöglichkeiten der Regelungstheorie für Rechensysteme vorgestellt, darunter IBM Lotus Domino Server, Queueing Systems, Apache HTTP Server, Random Early Detection of Router Overloads, Load Balancing und Caching with Differentiated Service.

IBM Lotus Domino Server ist ein spezifischer E-Mail-Server, mit dem die Endbenutzer durch Clients mit dem Lotus Server kommunizieren können. Die Interaktion zwischen Client und Server verläuft in Form von Remote Procedure Calls (RPCs) (p. 13) [1]. Der eingesetzte Parameter im IBM Lotus Domino Server heißt MaxUsers, mit dem die Nummer der RPCs (RIS) gesteuert werden kann (p. 13) [1].

Queueing Systeme werden häufig eingesetzt, um die Performance von Computersystemen zu modellieren. Die Aufträge oder Kunden kommen an und werden in der Warteschlange oder im Puffer platziert, wo sie für die weitere Verarbeitung durch den Server ausgewählt werden(p. 15) [1]. Solche Arbeitsanforderungen werden außerdem durch einen eingesetzten Regler nach der Wichtigkeit der Kundenbedürfnisse unterschieden, um bessere Performance für das Rechensystem oder bessere Services für den Kunden zu schaffen (p. 15) [1].

Apache HTTP Server ist der meistbenutzte Webserver, mit dem die Endbenutzer sich durch Clients mit dem Apache HTTP Server verbunden werden können(p. 16 ff) [1]. Nach der Verbindung kann der Endbenutzer die gewünschten Informationen vom Apache HTTP Server bekommen(p. 16 ff) [1]. Die Parameter, die im Apache HTTP Server eingesetzt werden, heißen MaxClients und KeepAlive. Der MaxClients Parameter begrenzt die Verarbeitungsgeschwindigkeit des Servers und KeepAlive begrenzt die maximale Wartezeit des Arbeiters.

Random Early Detection of Router Overloads
„Ein zentrales Element des Internet ist TCP, welches End-to-End-Kommunikation über Netzknoten ermöglicht“ (p. 19) [1]. Eine Möglichkeit ist, dass man die Pakete beim Routen zwischen Endpunkten in der Reihenfolge vertauscht. Die Router haben eine bestimmte Puffergröße, wenn dieser Puffer voll ist, werden die überflüssigen Pakete verworfen, damit kein Stau entsteht. Dann ist normalerweise das Netzwerk bereits überlastet, deswegen sollte die Puffergröße kontrolliert werden. Um dies zu tun, wird die Random Early Detection (RED) als Maßnahmen im Router eingesetzt(p. 19) [1]. Die RED kann alle Pakete nach Wichtigkeit der Kundenbedürfnisse unterscheiden und unnötige Pakete automatisch verwerfen [6].

Load Balancing ist eine der am häufigsten verwendeten Techniken für den Aufbau hochleistungsfähiger Rechensysteme(p. 20) [1]. Damit werden umfangreiche Berechnungen oder große Mengen von Anfragen auf mehrere parallel arbeitende Systeme verteilt. Eine einfache Load Balancing Methode findet sich zum Beispiel auf Rechnern mit mehreren Prozessoren. Jeder Prozess kann auf einem eigenen Prozessor ausgeführt werden. Die Art der Verteilung der Prozesse auf Prozessoren kann dabei einen großen Einfluss auf die Gesamtperformance des Systems haben [5]. Im Load Balancing System werden mehrere Clients durch einen Router mit verschiedenen Servern verbunden, die beste Situation für das Load Balancing System ist, wenn jeder Server in einer Balance Situation arbeiten kann, deshalb wird die Aufgabe durch die eingesetzte Parameter je nach Routing-Gewicht unterschieden und dann dem passenden Server zugeteilt[5].

Caching with Differentiated Service wird häufig verwendet, um die Leistung von Rechensystemen zu verbessern. Caching bezeichnet einen schnellen Puffer-Speicher, der Zugriffe auf ein langsames Hintergrundmedium zu vermeiden hilft(p. 22) [1]. Daten, die bereits einmal beschafft wurden, verbleiben im Cache, so dass sie bei späterem Bedarf schneller zur Verfügung stehen (p. 22) [1]. Allerdings können die Daten nicht immer im Puffer-Speicher gespeichert werden, man muss die Daten regelmäßig löschen. Im Caching with Differentiated Service werden die alle Server in viele kleine Platten unterteilt und die Web-Server Daten werden durch die eingesetzten Parameter nach Verwendungshäufigkeit unterschieden und dann im verschiedenen Platten gespeichert(p. 22) [1].

Die Ähnlichkeit der verschiedenen Anwendungsmöglichkeiten ist, dass alle auf dem geschlossenen Regelkreis basieren. Der Unterschied ist, dass ein solcher Regelkreis die verschiedenen Anforderungen des Rechensystems verändert**.**

3.2 IBM Lotus Domino Server als Schwerpunkt

In diesem Abschnitt wird der IBM Lotus Domino Server als Schwerpunkt erklärt, weil er ein typisches Beispiel ist, um die Anwendung der Regelungstheorie für Rechensysteme besser zu erläutern.

Die heutige Corporate IT benutzt typischerweise einen beträchtlichen Teil ihres Budgets für E-Mail-Dienste[8]. Der IBM Lotus Domino Server ist eine zuverlässige, skalierbare und sichere Plattform für Social Business Anwendungen[8]. Die Plattform hilft bei der Beschleunigung von Geschäftsoperationen, bei der Verbesserung der Entscheidungsfindung und der Erweiterung der Produktivität, wie in **Fig. 5** zu sehen ist, stellt Lotus Notes eine Client-Server-Anwendung dar(p. 13) [1]. Notes Clients beim IBM Lotus Domino Server interagieren mit den Endbenutzern durch E-Mail und andere Anwendungen(p. 13) [1]. Der Server verwendet eine Datenbankabstraktion, um eine Rechenumgebung für Anwendungen zur Verfügung zu stellen.

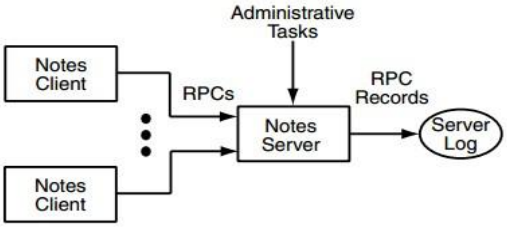


Fig. 5. Lotus Notes: eine Client-Server-Anwendung (p. 13) [1]

Notes Clients werden mit einer E-Mail-Datenbank verbunden, indem sie eine E-Mail öffnen, wodurch sie eine Ansicht der Elemente in der Datenbank erhalten(p. 13) [1]. Dann kann man diese Daten lesen, ändern, löschen und in der Datenbank einfügen. Die Interaktion zwischen dem Notes Client und dem Server wird mit Hilfe von Remote Procedure Calls (RPCs) realisiert(p. 13f) [1]. Wenn eine Arbeit fertig ist, wird der Verlauf des Prozesses im Server Log gespeichert, z. B. wenn man die Daten verändert hat, wird dann der Zeitpunkt der letzten Änderung gespeichert. Der Schwerpunkt des Administrators für den IBM Lotus Domino Server ist, wie sich der Vorgang des Systems bei verschiedenen Nummern der RPCs verändert.

Der Begriff RIS steht für die Anzahl der RPCs in einem Server, wenn sie zu groß ist, werden die CPU, Speicher und andere Ressourcen übermäßig verwendet, wodurch die Leistung sehr schlecht wird. Um RIS zu beschränken, wird ein Grenzwert für den MaxUsers Parameter angegeben, wodurch kann eine dynamische RIS über eine

Konsolen-Schnittstelle eingestellt werden (p. 14) [1]. MaxUsers beschränkt die Anzahl der gleichzeitigen Client-Verbindungen zum IBM Lotus Domino Server. Es ist wichtig anzumerken, dass die Anzahl der Verbindungen nicht das gleiche wie RIS ist(p. 14) [1]. Die erste Ursache für die Anzahl der Verbindungen nicht gleich wie RIS ist, dass der verbundene Benutzer in einer Ruhezeit bleibt und keinen RPC an den Server schickt. Die zweite Ursache dafür ist, dass der Benutzer eine Email an eine andere Person geschrieben hat, aber für diese Email gibt es keine RPCs im Server.

Fig. 6 zeigt, wie das Regelungssystem für den IBM Lotus Domino Server funktioniert. Das Target System ist der IBM Lotus Domino Server in Kombination mit einem Sensor (Server-Log), um RPC-Statistiken zu erhalten. Der Administrator legt einen Referenz-Eingang für RIS(p. 14) [1]. Die Regelabweichung ist die Differenz zwischen diesem Referenzeingang und RIS, daraus werden die Controller-Einstellungen für MaxUsers berechnet(p. 14) [1].

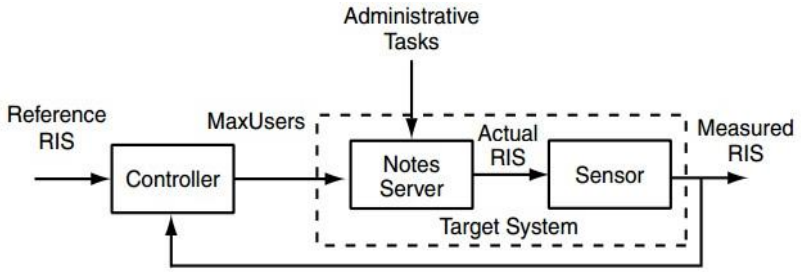


Fig. 6. Regelungssystem für IBM Lotus Domino Server (p. 14) [1]

4 Aufbau von Regelkreisen

Der Aufbau eines Regelkreises ist in **Fig. 7** dargestellt. Ein Regelkreis besteht aus einer Regelstrecke, einem Umwandler, einem Regler, und einer negativen Rückkopplung der Regelgröße Output(p. 5) [1]. Am Anfang gebe ich einen Input in das Rechensystem und bekomme dann einen Output, die Differenz zwischen den beiden ist die Regelabweichung und wird vom Regler zu einer Stellgröße verarbeitet. Normalerweise kann man den Output nicht direkt benutzen, sondern er muss durch einen Umwandler in eine verwertbare Regelgröße umgewandelt werden.

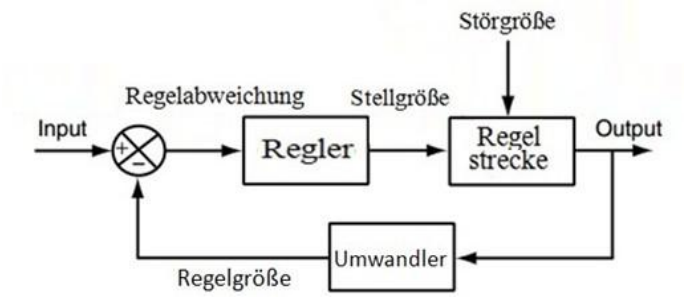


Fig. 7. Aufbau eines Regelkreises (p. 5) [1]

In diesem Regelkreis **(Fig.7.)** gibt es zwei wichtige Teile: Die Regelstrecke mit einem auf einem Umwandler begründetes Rechensystem, wodurch die Daten bearbeitet werden, der andere Teil ist ein Regler, der eine Stellgröße liefert, um den Regelkreis besser zu verarbeiten(p. 5) [1].

Normalerweise verhält sich die Regelstrecke träge. Dies ist zum einen durch speichernde Eigenschaften verursacht und zum anderen durch schwache Ankopplung der Stellgröße an die Regelgröße, wodurch entsteht eine verzögerte Reaktion der Regelstrecke mit charakteristischen Zeitkonstanten auf äußere Einflüsse(p. 2) [4]. Aufgabe der Regelungstechnik ist es nun, für ein spezielles System einen geeigneten Regelalgorithmus zu finden. Dieser Algorithmus soll in der Praxis dafür sorgen, dass der gewünschte Sollwert möglichst genau, möglichst schnell und möglichst stabil eingestellt wird (p. 2) [4].

Eine Regelstrecke ist ein wichtiges Element von Regelungssystemen(p. 35) [1]. **Fig. 8** zeigt die Beziehung zwischen Input und Output einer Regelstrecke. Die Stellgröße wird in der Regelstrecke durch das eingesetzte Rechensystem nach dem Konfigurationsparameter verarbeitet, um den Output zu steuern. Die Störungen sind unkontrollierte Faktoren und können nicht entfernt werden(p. 35) [1].

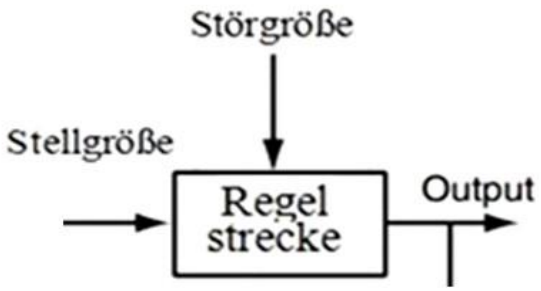


Fig. 8. Regelstrecke mit Input und Output (p. 35) [1]

„Ein Umwandler ist ein Element von Regelungssystemen und wird immer in einem geschlossenen Regelkreis von Output nach Input verwendet“ (p. 6) [1]. Fig. 9 zeigt die Beziehung zwischen dem Output des Rechensystems und der Regelgröße eines Umwandlers. Normalerweise kann man den Output in einem Regelungssystem nicht direkt benutzen, sondern er wird durch einen Umwandler auf einen erkennbaren Wert umwandelt(p. 6) [1].

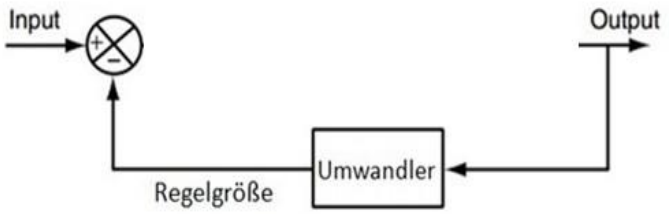


Fig. 9. Umwandler mit einer negativen Rückkopplung der Regelgröße (p. 5) [1]

Ein **Regler** ist ein anderes wichtiges Element eines Regelkreises, mit dem der Input verarbeitet wird(p. 5) [1]. **Fig. 10** zeigt die Beziehung zwischen der Regelabweichung und der Stellgröße eines Reglers. Die Regelabweichung kann man nicht entfernen, deswegen muss man die Regelabweichung möglichst minimieren (p. 6) [1].

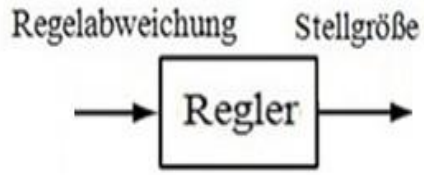


Fig. 10. Regler (p. 5) [1]

5 Die aktuellen Anwendungen der Regelungstheorie für Rechensysteme

5.1 Zusammenfassung

Um die Leistung zu optimieren, sind in vielen Rechensystemen Regler eingesetzt worden. Aufgabe eines Reglers in einem technischen Prozess ist es, eine oder mehrere physikalische Größen auf ein vorgegebenes Niveau zu bringen und dabei die Störeinflüsse zu reduzieren[12]. Um diese Aufgabe zu erfüllen, vergleicht ein Regler innerhalb eines Regelkreises ständig den Sollwert und den Istwert der Regelgröße und berechnet die Differenz[12]. Diese Differenz bewirkt, dass die Regelstrecke so

beeinflusst wird, dass die Regelabweichung im eingeschwungenen Zustand minimiert wird [12].

Viele Reglerans ätze k önnen f ür Rechensysteme angewendet werden, wie z. B. P-Regler, PID-Regler, LQR-Regler und Fuzzy-Regler. In Kapitel 3 wurde IBM Lotus Domino Server als Beispiel dargestellt. Daher ist schon bekannt, dass dieses System mit einem Regler bessere Leistung liefern kann. Allerdings gibt es auch die Regelabweichung im Regelkreis[12]. Um diese zu minimieren, sind verschiedene Regler im Regelkreis angewendet worden.

- **P-Regler:** Ein Proportional-(P-)Regler liefert eine kontinuierliche Stellgr öße, die zur Regelabweichung proportional ist(p. 5) [4]. Der Sinn des P-Reglers f ür Rechensysteme ist, dass man durch einen maximalen oder minimalen Wert der Stellgr öße die Regelabweichung in der S ättigung beobachten kann (p. 5) [4].

- **PID-Regler:** Der PID-Regler (=proportional–integral–derivative Regler) besteht aus Anteilen des P-Reglers, des I-Reglers und des D-Reglers (p. 320) [1]. Der PID-Regler ist der universellste der klassischen Regler und vereinigt die guten Eigenschaften der P-, I- und D-Regler, deswegen kommt er in den meisten Anwendungen zum Einsatz. [13]

- **LQR-Regler:** Der LQR-Regler wird in Bezug auf die relativen Kosten der Steuerungsfehler und des Kontrollaufwands angegeben(p. 358) [1]. Dies wird durch zwei Eingabeparameter gemessen, n ämlich die Matrizen Q und R: Q steht f ür die Kosten der einzelnen (und kombinierten) Zustandsvariablen, die von ihrem Betriebspunkt abweichen, R gibt die Kosten des Kontrollaufwands an(p. 359) [1]. Die einzelnen Schritte im LQR Design lassen sich wie folgt zusammenfassen: Zuerst werden die gewichteten Matrizen Q und R gew ählt, danach wird die R ückkopplungsverst ärkung K berechnet, die R ückkopplungsverst ärkung K ist eine Matrix, man kann durch Software MATLAB die Werte von K berechnet wird(p. 359) [1]. Anschlie ßend kann man die Leistung des Regelungssystems basierend auf dem geschlossenen Regelkreis vorhersagen oder eine Computersimulation laufen lassen, um die Leistung des geschlossenen Regelkreises zu verifizieren. Dann werden Q und R neu bestimmt und die vorherigen Schritte werden wiederholt, wenn die Leistung nicht zufriedenstellend war(p. 359) [1].

- **Fuzzy-Regler:** Fuzzy-Regler k önnen in Verbindung im Rechensystem als Regler angewendet werden. „Das Prinzip des Fuzzy-Reglers besteht darin, scharfe Eingangssignale zu erfassen und mit Hilfe von linguistischen Begriffen aus dem Expertenwissen über Zugeh örigkeitsfunktionen und logischen Wenn-Dann-Operationen zu bewerten und den Übergang von linguistischen Variablen zu scharfen Stellgr ößen zu bilden“ [2]. Durch die

Eingangssignale kann der Fuzzy-Regler mit mehr oder weniger empirischer Methodik optimal an einen nichtlinearen Prozess mit mehreren Ein- und Ausgangsgrößen modelliert werden, ohne dass das mathematische Modell des Prozesses vorliegt [2]. Die typische Anwendung des Fuzzy-Reglers ist wegen der fehlenden dynamischen Eigenschaften der Regelbasis bevorzugt ein Verfahren zur Steuerung eines technischen Prozesses. Deshalb ist die Anpassung der Fuzzy- Regler mit dem Expertenwissen von einem bekannten Prozess ohne mathematisches Modell auch relativ unproblematisch [2].

Der P-Regler und PID-Regler sind die meistbenutzten Regler, weil sie leicht zu bedienen sind und eine große Anpassungsfähigkeit haben (p. 293 f) [1].

5.2 Proportional-(P-)Regler für Rechensysteme

In **Fig.11.** ist ein P-Regler im Regelkreis. Um diese Abbildung besser zu erklären, werden die einzelnen Komponenten jeweils vorgestellt.

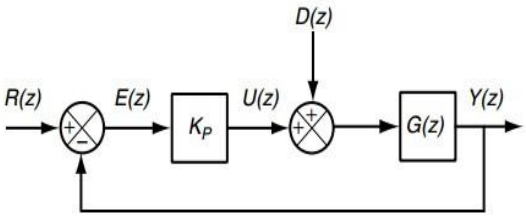


Fig.11. P- Regler (p.248)[1]

In dieser Abbildung wird der Input durch R(z), Regelabweichung durch E(z), die gewählte Verstärkung durch Kp**,** die Stellgröße durch U(z), die Störgröße durch D(z), die Übertragungsfunktion von Rechensystem durch G(z) und der Output durch Y(z) bezeichnet.

Das erste Element ist die Übertragungsfunktion des Rechensystems in diesem Regelkreis. Das Verhältnis zwischen den Einheiten von Input und Output wird durch die Übertragungsfunktion des Rechensystems (**Fig. 12**) beschrieben.

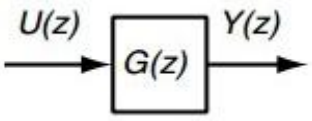


Fig. 12. Übertragungsfunktion von G(z) (p. 246) [1]

Die Übertragungsfunktion von G(z) lautet:

G(z) = Y(z) / U(z) (p. 246) [1].

Das zweite Element ist die Störgröße D(z). Die Wirkung der Störung wird durch Zugabe von D(z) auf den Wert modelliert, der vom Administrator festgelegt wird. (p. 247) [1]. In **Fig. 13** ist ein Regelkreis mit der Störgröße D(z), wodurch wird die Stellgröße U(z) verändert(p. 247) [1].

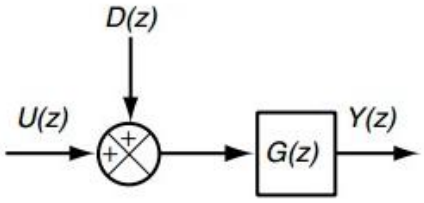


Fig. 13. Regelkreis mit D(z) (p. 247) [1]

Das dritte Element im Regelkreis mit P-Regler ist Kp **(Fig. 14)**. Kp ist im Idealfall frequenzunabhängig(p. 246) [1]. Das Problem des P-Reglers liegt darin, dass er stets eine endliche Regelabweichung benötigt, um reagieren zu können(p. 248) [1]. Man könnte nun Kp möglichst groß wählen, um die dauerhafte Regelabweichung entsprechend klein zu halten.

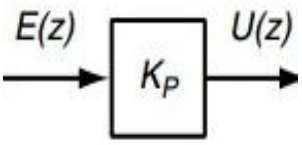


Fig. 14. Regelkreis mit Kp (p. 246) [1]

Der Wert von Kp wird durch den Wert U(z) und E(z) berechnet(p. 248) [1]. Die Übertragungsfunktion von Kp lautet:

Kp = U(z) / E(z) (p. 248) [1]

Zuerst messen wir explizit den gewünschten Output, dies wird im geschlossenen Regelkreis weiter den Input R(z) verändern(p. 248) [1]. Die Regelabweichung E(z) kann durch den Output Y(z) und den Input R(z) berechnet werden(p. 248) [1]. Die Übertragungsfunktion von E(z) lautet:

E(z) = R(z) -Y(z) (p. 248) [1].

Danach berechnen wir die Einstellungen der Stellgröße U(z), basierend auf G(z) und Y(z) (p. 248) [1].

Anwendungsbeispiel von IBM Lotus Domino Server

Nun wird eine Simulation (**Fig. 16)** des P-Reglers in einem Regelkreis, IBM Lotus Domino Server, erklärt. In diesem Regelkreis (**Fig. 15)** wird der Input als Reference RIS, der Output als Measured RIS geschrieben(p. 251) [1]. Die Regelabweichung ist die Differenz zwischen Reference RIS und Measured RIS. Es gibt auch Kp und eine Störgröße in diesem Regelkreis(p. 251) [1]. Die Stellgröße ist MaxUsers und das Rechensystem in diesem Regelkreis ist Notes Server (p. 251) [1].

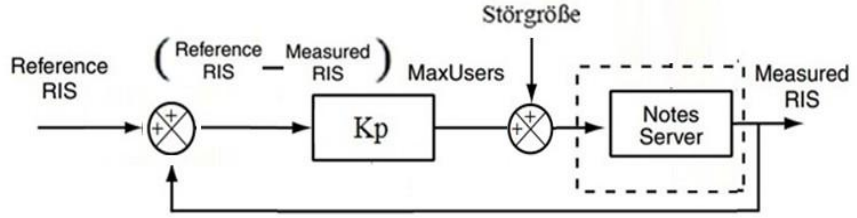


Fig. 15. Regelkreis IBM Lotus Domino Server (p. 246) [1]

In der Simulation wird die Reference RIS durch R(z), die Regelabweichung durch E(z), die gewählte Verstärkung durch Kp, die Stellgröße (MaxUsers) durch U(z), die Störgröße durch D(z), die Übertragungsfunktion des Rechensystems (Notes Server) durch G(z) und die Measured RIS mit Y(z) bezeichnet (p. 251) [1].

Durch die **Übergangsfunktion** kann ein Rechensystem Input nach Output wandeln(p. 89) [1], z.B. G(z) = Y(z) / U(z) ist eine Übergangsfunktion. Im **Zeitbereich** liegt Output zeitbezogen vor(p. 158) [1], wodurch wird im Form z.B. y(1) = 0.43y(0) + 0.47 u(0) geschrieben(p. 158) [1].
Die gegebenen Informationen sind: G(z) = Y(z) / U(z) = 0.47 / (z - 0.43), Y(0) = 0, D(z) = 20 für z >= 5, Kp = 2, R(z) = 10. Was ist Y(z)?

(1) Die Regelabweichung E(z) wird zuerst berechnet, also E(0) = R(0) - Y(0) = 10 - 0 = 10;

(2) Dann kann man den Steuereingang U(z) = Kp * E(z) = 2 * 10 = 20, also hat der P-Regler eine gewählte Verstärkung von **Kp = 2**;

(3) Schließlich kann man den Output Y(1) = 0.43Y(0) + 0.47U(0) = 9.4 berechnen.

Y(1) ist die Measured RIS des IBM Lotus Domino Servers nach der ersten Regelung(p. 246) [1]. Y(z) läuft weiter, bis das Rechensystem stabil ist.

Im **Zeitbereich** kann man durch y(z) am jeden Zeitpunkte berechnen.

(4) y(0) = 0;

(5) y(1) = 0.43y(0) + 0.47 u(0) = 9.4;

(6) y(2) = (0.43)y(1) + (0.47)u(1) = (0.43)(9.4) + (0.47)(2 * 0) = 4.0.

und weiter bis z = 5. Dann haben wir y(5) = 6.4 mit e(5) = 3.6 und u(5) = 7.2.

(7) y(6) = (0.43)y(5) + (0.47)q(5) = (0.43)(6.4) + (0.47)(7.2 + 20) = 15.5
mit q(z) = u(z) + d(z);

und weiter bis z = 10. Wir bekommen mit Y(10) = 12.7 ein relativ stabiles Ergebnis.

(8) Da Y(10) = 12.7 > Y(5) = 6.4 ist, konvergiert die Measured RIS des IBM Lotus Domino Servers um z = 10(p. 246) [1].

Die Simulation wird in **Fig. 16** dargestellt.

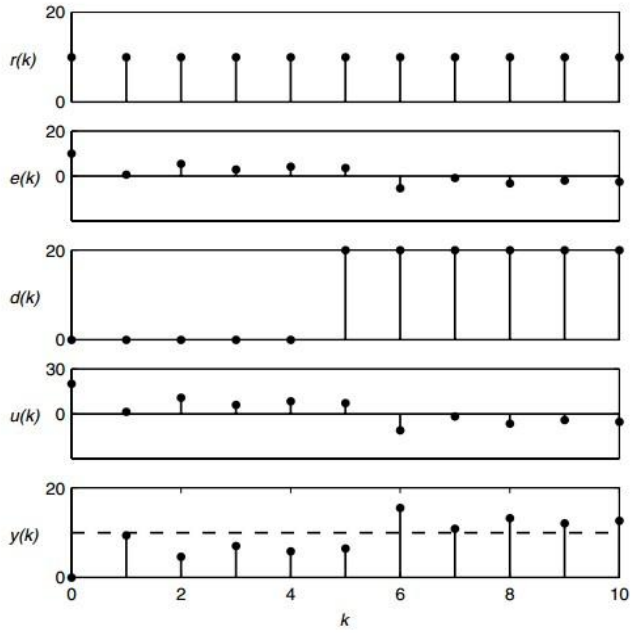


Fig. 16. Simulation (p. 249) [1]

5.3 PID Regler für Rechensysteme

5.3.1 PI-Regler

Der PI-Regler **Fig. 17** (proportional–integral Controller) besteht aus einem P-Glied **Kp** und einem I-Glied **Ki**. (p. 301) [1].

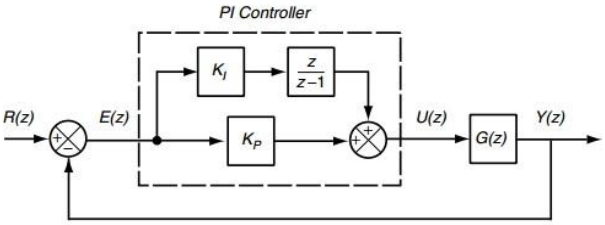


Fig. 17. PI-Regler (p. 302) [1]

Für P-Regler ist der Steuereingang proportional zur Regelabweichung, während bei der Integral-Regelung die Veränderung des Steuereingangs proportional zur Regelabweichung ist:

Beim P-Regler: „Up(z) = Kp * E(z)“(p. 302) [1]
Beim I-Regler: „Ui(z) = Ui(z-1) + Ki * E(z)“ (p. 302) [1]

Der PI-Regler besteht aus einem P-Regler und einem I-Regler,

„Beim PI-Regler: U(z) = Up(z) + Ui(z)
= Kp * E(z) + U(z-1) + Ki * E(z)“ (p. 302) [1]

Da U(z) - U(z-1) = Up(z) + Ui(z) - Up(z-1) - Ui(z-1)
= Kp * E(z) - Kp * E(z-1) + Ki * E(z) ist,

Die Übertragungsfunktion lautet:

„U(z) = U(z-1) + (Kp + Ki) * E(z) - Kp * E(z-1)“ (p. 302) [1]

Die Übertragungsfunktion kann direkt in die Reihenstruktur überführt werden und lautet für den idealen Regler:

„ $\frac{U(z)}{E(z)} = \frac{(Kp+Ki)z-Kp}{z-1} = Kp + \frac{Ki*z}{z-1}$ “ (p. 302) [1]

5.3.2 PD-Regler

Der PD-Regler **Fig. 18.** (proportional–derivative Controller) besteht aus der Kombination eines P-Gliedes Kp mit einem D-Glied Kd(p. 320) [1]. Der Differentiator arbeitet vorwiegend bei hohen Frequenzen und bei einer Phasenvoreilung von +90 °(p. 320) [1]. Diese Eigenschaft führt dazu, dass eine zusätzliche Phasenreserve von 90 ° zur Verfügung steht und deshalb eine größere Verstärkung auch bei hohen Frequenzen möglich ist, wodurch reagiert der Regler schneller und kann Oszillationen beim Einschwingvorgang effektiv unterdrücken (p. 320) [1].

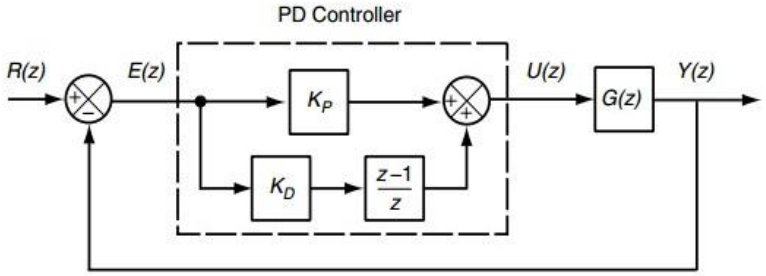


Fig. 18. PD-Regler (p. 316) [1]

Die Übertragungsfunktion lautet:

„U(z) = Kp* E(z) - Kd *(E(z) -E(z-1))“ (p. 320) [1]

Die Übertragungsfunktion kann direkt in die Reihenstruktur überführt werden und lautet für den idealen Regler:

„$\frac{U(z)}{E(z)} = Kp + \frac{Kd*(z-1)}{z} = \frac{(Kp+Kd)*z-Kd}{z}$“ (p. 320) [1]

Der PD-Regler ist ein sehr schneller Regler, denn er fügt im Gegensatz zum PI-Regler keinen zusätzlichen Pol durch Integration in den offenen Regelkreis ein(p. 315ff) [1]. Selbstverständlich ist auch die Verzögerung mit kleiner Zeitkonstante im Regelkreis nicht vernachlässigbar (p. 315 ff) [1].

5.3.3 PID-Regler

In Fig. 19 ist ein PID-Regler dargestellt.

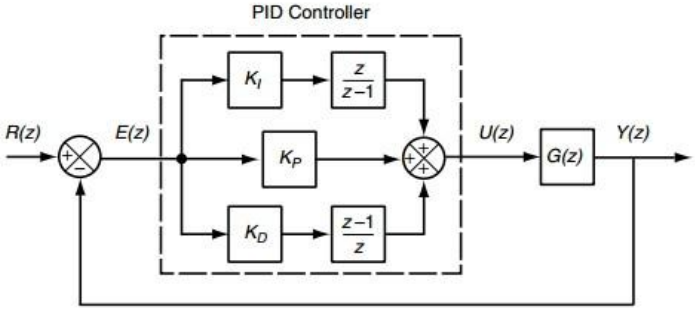


Fig. 19. PID-Regler (p. 321) [1]

Der PID-Regler besteht aus einem P-Regler, einem I-Regler und einem D-Regler,

Beim P-Regler: Up(z) = Kp * E(z)
Beim I- Regler: Ui(z) = Ui(z-1) + Ki * E(z)
Beim D-Regler: Ud(z) = Kd(E(z) - E(z-1)) (p. 320) [1]

Übertragungsfunktion des idealen PID-Reglers:
Beim PID-Regler: U(z) = Up(z) + Ui(z) + Ud(z) =
„Kp * e(k) + Ki *$\sum_{i=0}^{k-1} e(i)$ + Kd * (e(k) -e(k-1))“ (p. 320) [1]

Übertragungsfunktion des idealen PID-Reglers:

$$\text{„}\frac{U(z)}{E(z)} = Kp + \frac{Ki*z}{z-1} + \frac{Kd*(z-1)}{z}\text{“ (p. 320) [1]}$$

Anwendungsbeispiel von IBM Lotus Domino Server

Nun wird eine Simulation **(Fig. 21)** des PID-Reglers in einem Regelkreis, IBM Lotus Domino Server, erklärt. In diesem Regelkreis **(Fig. 20)** wird der Input als Reference RIS durch r(k), der Output als Measured RIS durch y(k), die drei gewählte Verstärkung durch Ki, Kp und Kd, die Regelabweichung ist die Differenz zwischen Reference RIS und Measured RIS durch $u_p(k)$, $u_i(k)$ und $u_d(k)$, die Stellgröße durch u(k), die Störgröße durch d(k) und die Übertragungsfunktion des Rechensystems (Notes Server) durch g(k) bezeichnet (p. 251) [1].

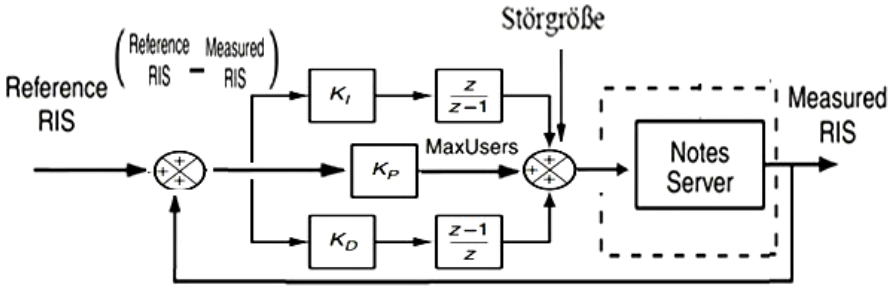


Fig. 20. Regelkreis IBM Lotus Domino Server (p. 246) [1]

Die Simulation wird in **Fig. 21.** dargestellt.

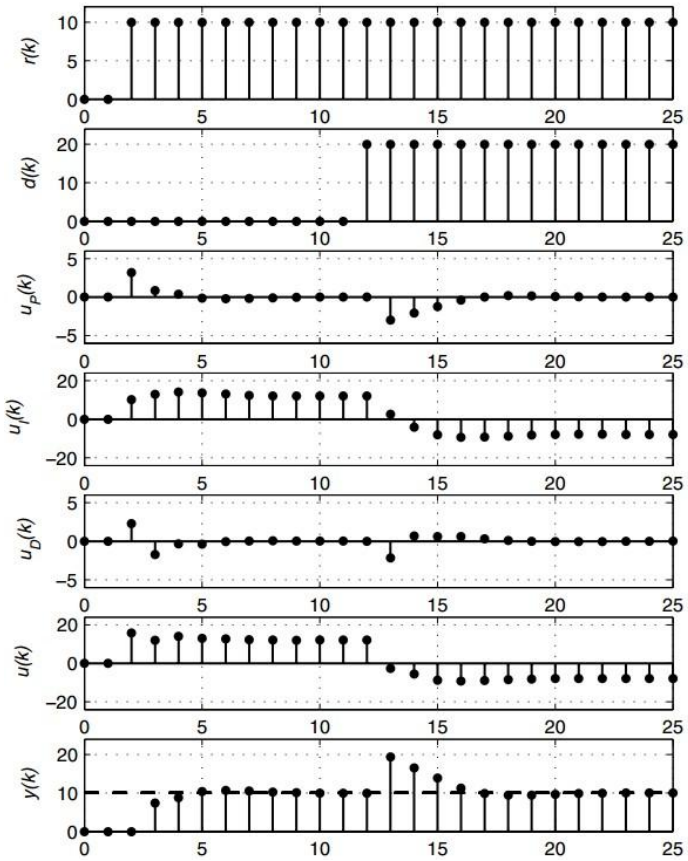


Fig. 21. Simulation (p. 325) [1]

In dieser Simulation **(Fig. 21)** mit PID-Regler vom IBM Lotus Domino Server haben die Werte von g(k), r(k), d(k) und Kp, Ki und Kd gegeben(p. 323) [1]. Man kann dann damit die $u_p(k)$, $u_i(k)$ $u_d(k)$ und u(k) berechnen. Schließlich kann man den Output y(k) im Zeitbereich berechnen.

PID-Regler haben einen hohen Bekanntheitsgrad in der Anwendung empirischer Regler-Einstellungen bei Regelkreisen mit unbekannten Regelstrecken und geringen dynamischen Anforderungen[4]. Aber der PID-Regler hat auch einen Nachteil, dass der PID-Regler ein langsamer Regler als P-Regler ist, weil er ein I-Glied enthält [4].

6 Vor- und Nachteile der Regelungstheorie in Rechensystemen

Ein wichtiger Indikator zur Abschätzung der Güte eines Reglers ist, ob das Regelungssystem stabil ist(p. 8) [1]. Die Stabilität ist die erste Anforderung der

Regelungstheorie für Regelungssysteme, weil ein instabiles System nicht für ein Vorhaben von entscheidender Bedeutung verwendet werden kann. Ein System ist stabil, wenn es für jeden begrenzten Input nur begrenzten Output liefern kann. Deswegen wird im Regler die Stabilität des Regelungssystems immer geprüft (p. 8) [1]. Der zweite wichtige Punkt für ein System ist, wie man die Störgröße minimieren kann. Um diese Störgröße einzuschränken, werden verschiedene Regler benutzt.

Der Vorteil von **P-Reglern** ist, dass sie die Verstärkung nach Anforderung verändern kann, und die Verstärkung kann theoretisch unendlich hoch eingestellt werden, wobei es zu einem aperiodischen Einschwingen der Regelgröße kommt, man kann durch die Anwendung des P-Reglers die Performance des Rechensystems besser zu beobachten(p. 261) [1].

PID-Regler besteht aus einem P-Regler, I-Regler und D-Regler (p. 320) [1]. Der Vorteil eines PID-Reglers ist, dass er eine gute Anpassungsfähigkeit hat. Er verhindert bei konstantem Sollwert eine bleibende Regelabweichung bei Führungs- und Störgrößensprung und kann Verzögerungen der Regelstrecke kompensieren (p. 293) [1]. Allerdings hat der PID-Regler noch einen Nachteil, dass er ein langsamer Regler ist, weil er einen I-Regler enthält(p. 320) [1]. Der I-Regler wird immer eine lange Zeit geschwungen, nach man eine Integrationszeit wählt [4]. Durch die Vergleichung der Abbilder **Fig. 16.** und **Fig. 21** erfahren wir, dass das Rechensystem(IBM Lotus Domino Server) mit PID-Reglern eine stabilere Performance als das Rechensystem mit P-Reglern liefern kann[4].

Der Vorteil von **LQR-Reglern** ist, dass die Signalanteile im überlagerten Zustands-Regelkreis gering sind, weil keine Differenzierung beim PI-Regler vorkommt (p. 358) [1]. Dies bietet den Vorteil, dass keine Schätzgrößen in die eigentliche Regelung eingehen, und die Nachteile sind jedoch, dass die Zustandsgrößen meist nicht zur Verfügung stehen(p. 363) [1]. Je nach Anforderung hinsichtlich Regelabweichung und Störunterdrückung am Ausgang der Regelstrecke kann der LQR-Regler im Vergleich zu einem konventionellen PID-Standardregler unterlegen sein (p. 363) [1].

Fuzzy-Regler sind gut geeignet für komplexe Rechensysteme mit nichtlinearem Verhalten, da man mit ihnen schnell, realistisch, problembezogen und aussagekräftig modellieren kann [2]. Ein weiterer Vorteil ggü. mathematischen Beschreibungsverfahren ist die Möglichkeit, mit Fuzzy-Reglern die Beschreibung und das Verhalten des Systems linguistisch ausdrücken zu können [2]. Des Weiteren kann auch die Definition der Fuzzy-Mengen verändert werden, wenn die Regelung in einem Bereich nicht optimal ist, oder es können Regeln ergänzt oder verändert werden. Ein gravierender Nachteil von Fuzzy-Reglern ist allerdings, dass sie nicht selbständig lernfähig sind[2].

7 Fazit

Das Ergebnis dieser Untersuchung ist, wie ein Rechensystem mit Regler funktioniert. Mit diesen Verfahren und Methoden kann ein Rechensystem mit Regler nach und nach auf den richtigen und stabilen Zielzustand kommen.

Ein besonderer Vorteil des Rechensystems mit Regler ist, dass die Veränderungen im Verhalten nicht mehr von Hand eingestellt werden müssen, was nur einem Experten möglich ist, oder in regelmäßigen Updates aufgespielt wird. Man sollte nur einen Zielzustand einsetzen, dann kann durch das Rückkopplungs-Verfahren das Regelungssystem selbst den Zielzustand erreichen. In dieser Arbeit wurde die Modellierung des Regelungssystems analysiert. In der Modellierung wurde die Regelstrecke, der Regler, der Umwandler und die negative Rückkopplung der Regelgröße Output vorgestellt. Das fünfte Kapitel behandelte die aktuellen Regelungssysteme. Am Anfang wurde eine Zusammenfassung über die bekannten Regler gegeben und als Schwerpunkt wurden der P-Regler und PID-Regler jeweils erklärt. Der PID-Regler ist ein P-Regler, ein I-Regler und ein D-Regler vereint(p. 293) [1]. Der PID-Regler hat den Vorteil, dass er in vielen Bereichen funktioniert; der Nachteil ist, dass es ein langsamer Regler ist (p. 293 ff) [1].

Literatur

1. Hellerstein, J. L. (2004). Feedback Control of Computing Systems. New Jersey: John Wiley & Sons.

2. Kruse, R. (1996). Fuzzy-Systeme. Braunschweig: Springer.

3. Mittelstaedt, H. (2002). Die Regelungstheorie . Naturwissenschaften , 246.

4. Physikalisches Praktikum. (2012). PID - REGLER. Retrieved 5.1.2013, from http://www.physik.uni-augsburg.de/~sausemar/FP14/FP14.pdf

5. Radermacher, R. (1996). Eine Ausführungsumgebung mit integrierter Lastverteilung für verteilte und parallele Systeme. München: Herbert Utz Verlag.

6. Random early detection gateways for congestion avoidance. (1993). Networking .

7. R.Kaiser. (2006). Koexistenz unterschiedlicher Zeitanforderugen in einem gemeinsamen Rechensystem. Berlin: Springer.

8. J. Powers, D. Dumouchel.(2009). IBM Lotus Domino Server.Load V8. Retrieved 6.1.2013, from https://www.ibm.com/developerworks/cn/lotus/domino8-serverload/

9. Software Foundation: Apache.(2012). Foundation Project. Retrieved 8.1.2013, from http://www.apache.org/

10. S.G.Ziavras.(2012). COMPUTER SYSTEMS. Retrieved 3.1.2013, from:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.4490&rep=rep1&type=pdf

11. Schlacher, M. Z. (2000). Anwendungen der nichtlinearen Regelungstheorie in der Mechatronik. *Automatisierungstechnik* , S. 103.

12. Wissensportal.(2010)Regler.Retrieved3.1.2013, from:http://de.inforapid.org/index.php?search=Regler

13. Regelungstechnik.(2012) Der Regler. Retrieved 8.1.2013, from:https://www.xplore-dna.net/mod/page/view.php?id=88

Cyber-physische Systeme

Jian Gong

Karlsruhe Institute of Technology
Institute of Telematics - Pervasive Computing Systems
Betreuer: Dr. Till Riedel
jian.gong2@student.kit.edu

Zusammenfassung:
Aktuelle Studien sagen voraus, dass sogenannte Cyber-physische Systeme (CPS) in bisher kaum vorstellbarer Art unser alltägliches Leben verändern. Diese Arbeit erläutert grundlegend den Begriff Cyber-physisches Systeme. Weiter geht sie darauf ein, welche Vorteile und Herausforderungen mit CPS verbunden sind? Dazu stellt sie Grundbausteine und relevante Technologien für CPS sowie Anwendungsbeispiele vor. Resultierend dient die Arbeit als einführende Grundlage für das Verständnis Cyber-physischer Systeme.

1 Einleitung

Es gibt heute schon etwa 98 Prozent der Mikroprozessoren, die in Alltagsgegenständen und Geräten eingebettet. Diese besitzen Sensoren und Aktoren, wodurch mit der Außenwelt verbunden sind(p.5)[1]. Die Entwicklungstendenz verdeutlicht, dass sie unaufhörlich steigend miteinander und über das Internet vernetzt werden. Unsere Welt lässt sich in zwei Teile unterscheiden, und zwar die reale Welt und die virtuelle Welt. Die reale Welt bezieht sich auf die physikalische Welt; die virtuelle Welt bezieht sich auf den Cyberspace. Die physikalische Welt verschmilzt Welt, wie in Abbildung 1 anhand einer „augmented reality“ Anwendung dargestellt, mit der virtuellen Welt und daraus entsteht ein cyber-physisches System(p.5) [1].

Abbildung1: physikalische und virtuelle Welt verschmelzen[7]

Die reale und virtuelle Welt wurden in klassischen Computersystemen immer strikt getrennt [9] und es wird der modellierende und abstrahierende Charakter eines IT-System herausgestellt. Modernem Steuerungssysteme entsprechen dieser Sichtweise nur sehr limitiert. Die z. B. in einer modernen mobilen Anwendung genutzten Systeme, interagieren oft stark mit der physikalischen Welt - durch Sensoren und Aktoren. Cyber-physische Systeme (CPS) sind solche Systeme, die „ihre physische Umgebung durch Sensoren erkennen, diese Informationen verarbeiten und andererseits die physische Umwelt auch koordiniert beeinflussen können“ [9].

1.1 Definition "Cyber-physisches System":

Cyber-physisches Systeme (engl.: cyber-physical systems) „*adressieren die enge Verbindung eingebetteter Systeme zur Überwachung und Steuerung physikalischer Vorgänge mittels Sensoren und Aktuatoren über Kommunikationseinrichtungenmit den globalen digitalen Netzen (dem Cyberspace)“*(p.17)[2].

Es existieren viele verwandte Systembegriffe bzw. Forschungsthemen. Sie haben - bis zu einem gewissen Ausmaß - ähnliche Grundlagen oder Forschungsausrichtungen. Die folgenden Begriffe zeigen beispielsweise eine spezifische Sichtweise, die das Potenzial dieser Systeme unterstreichen(p.25)[2]:

- „Ubiquitous Computing (Ubiquität)
- Organic Computing (emergente Anpassung)
- Pervasive Computing (Durchdringung)
- Self-X-System“ (p.25)[2] (Autonomie)

Ein cyber-physisches System ist „*ein System mit eingebetteter Software, erfasste Daten auswerten und speichern und aktiv oder reaktiv mit der physikalischen*

sowie der digitalen Welt interagieren, über digitale Kommunikationseinrichtungen untereinander sowie in globalen Netzen verbunden sind, weltweit verfügbare Daten und Dienste nutzen und über eine Reihe dedizierter, multimodaler Mensch-Maschine- Schnittstellen verfügen"(p.13)[1].

1.2 Ausprägungen des Systems

Im Laufe der Weiterentwicklung von eigebetteten Systems gibt es verschiedene Ausbaustufen. Diese Entwicklungsversionen sind in der Komplexitätssteigerung ähnlich der Entwicklung des menschlichen Gebrauchs von Werkzeugen - von einem Stein bis hin zur vernetzten Küchenzeile. „Diese Versionen reichen von einfachen Anbindungen bis hin zu globalen Netzwerken mit vielfältig eingebetteten Systemen" (p.23)[2]. Diese Stufen von Systemen können wie folgt in Kategorien eingeteilt.

- Lokale, isolierte System: Dabei handelt es sich nur um eine Funktion oder einfache Nutzungsschnittstelle (p.23)[2].

- Multifunktionale Systeme (unvernetzte Systeme): Dabei handelt es sich um mehrere Steuergeräte, die oft komplexe Nutzungsschnittstellen haben und funktional abhängig sind(p.23)[2].

- Lose vernetzte Systeme: Hier handelt es sich um mehrere Steuergeräte, die nicht nur komplexe Nutzungsschnittstelle haben, sondern auch lose nach außen vernetzen können. Die Vernetzung ist allerdings nur eingeschränkt bzw. nicht stabil(p.23)[2].

- Netzwerke von funktional eng gekoppelten Systemen sind „mehrere Netzwerke von Steuergeräten mit komplexen Nutzungsschnittstellen". Im Gegensatz zu Lose vernetzte Systeme sind diese Systeme „eng untereinander und nach außen vernetzt". Die Vernetzung ist auch stärker als Lose vernetztes System(p.23)[2].

- Systeme von Systemen: beziehen sich auf global vernetzte Systeme wie z.B. „Internet der Dinge, cyber-physisches System" (p.23)[2]. CPS ist ein „gro-

ßes“ Gesamtsystem, es bestehen meist aus vielen „kleinen“ Einzelsystemen. Solche Einzelsystemen sind auch eine vollständige intelligente System, also sie haben alle entsprechende Eigenschaft von Systeme, z.B. Self-Awareness (sehe Kapital 2.2.2) [16]. Solche Einzelsysteme sind miteinander vernetzten und können sich untereinander koordinieren. Aufgrund der Zusammensetzung des CPSs aus eine Vielzahl von autonomen Einzelsystemen werden CPS deshalb auch als "Systems of Systems" bezeichnet[9].

Durch diese Entwicklung können wir zusammenfassen, dass die Systeme steigende offener, komplexer, autonomer und intelligenter sind. Cyber-physische Systeme entsprechen dieser Entwicklungstendenz.

2 Grundbausteine des cyber-physischen Systems

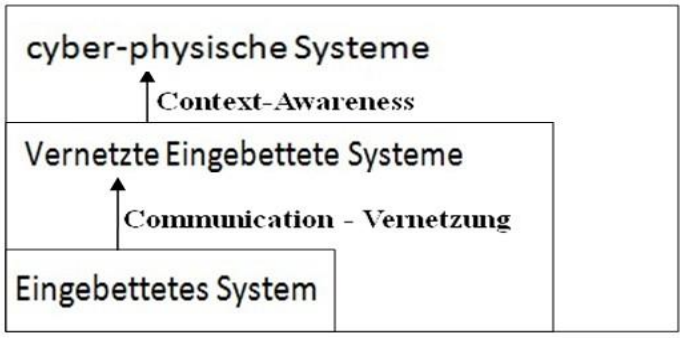


Abbildung 2: Evolution vom eingebetteten System zum CPS(p.21)[5]

2.1 Eingebettetes System

Wie Abbildung auf Basis der vorherigen Definition illustriert, sind eingebettete Systeme eine wesentliche Grundlage für CPS. Die Entwicklung der CPS basiert auf eingebetteten Systemen:

„Eingebettetes System bezeichnet einen elektronischen Rechner oder auch Computer, der in einen technischen Kontext eingebettet ist. Dabei übernimmt der Rechner entweder Überwachungs-, Steuerungs- oder Regelfunktionen oder ist für

eine Form der Daten- bzw. Signalverarbeitung zuständig, beispielsweise beim der Verschlüsselung bzw. Entschlüsselung, Codierung bzw. Decodierung oder Filterung" [11].

Es gibt Messfühler und Aktoren in den Systemen. Die Systeme Wahrnehmen Informationen aus ihrer Umgebung durch diese Messfühler, bearbeiten diese und „steuern über Aktoren ihrerseits unmittelbar physikalische Vorgänge"(p.18) [2]. Eingebettete Systeme sind unsichtbar für den Benutzer. Allerdings kann der Nutzer kann den Aufgaben durch eine einfache Bedienfunktion in Anwendungsbereichen ausführen [11].

Eingebettete Systeme sind die intelligenten Kontrollzentralen in unseren alltäglich technischen Geräte, beispielsweise in „Waschmaschinen, Flugzeugen, Kühlschränken, Fernsehern, DVD-Playern, Mobiltelefonen usw."[11]. Ohne eingebettete Systeme müssten wir auf viele Bequemlichkeiten in unserem Alltag verzichten. Beispielsweise würde der ICE der Deutschen Bahn nicht mehr funktionieren, da er ohne eingebettete Steuersysteme nicht fahren könnte(p.7)[2].

2.2 Unterschiede zwischen CPS und eingebetteten Systemen

Eingebettete Systeme beziehen sich eher auf ein einzelnes Gerät. Im Unterschied dazu bestehen CPS meist aus eine Vielzahl von Einzelsystemen, die sich miteinander koordinieren und verbinden können. Oft sind diese Einzelsystemen von CPS selbst eingebettete Systeme[9](sehe Kapital 1,3). In das cyber-physischen System können die verschiedenen Systeme integriert und vernetzt werden. Aber nur im Fall von komplexen Gesamtsystemen handeln eingebettete Systeme sich dabei um eine Vernetzung einer Vielzahl autonomer Systemen(p.9)[2] (z. B. im Fahrzeug oder Flugzeug). Darüber hinaus beziehen sich generell eingebettete Systeme besonders nur an einige Aufgaben, teilweise sogar nur eine Aufgabe. Eine Waschmaschine muss z. B. nur waschen können. Die verfügbare Ressourcen für Eingebettete Systeme meist eingeschränkt. Die Software auf dem eingebetteten System ist in der Regel vorübergehend unverändert[11].

Zwei wichtige Unterschiede zwischen eingebetteten Systemen und CPS ist der Grad der Vernetzung sowie der Umgebungs-reaktivität (Context-Awareness).

2.2.1 Vernetzung

Wie soeben dargestellt, ist die Vernetzung das dominierende Element bei CPS(p.29)[2]. Kommunikation ist eine wichtig Merkmal von CPS (sehe Kapital3, 1). Das heißt, dass die bestehenden Komponenten des CPSs untereinander kommunizieren und Informationen tauschen und damit die richtige Entscheidungen treffen können. Nur im Fall von komplexen Gesamtsystemen bezieht ein eingebettetes System sich auf eine Vernetzung von ansonsten eingebetteten Systemen(p.9)[2]. Ein Navigationssystem welches z. B. „neues Kartenmaterial mit anderen Geräten tauscht oder die Position des eigenen Fahrzeugs weitergibt, wäre ein CPS“[4]. Vernetzung ist eine sehr wichtige Eigenschaft, die für CPS erforderlich ist. Sie ist auch die Voraussetzung, die Kommunikation zwischen verschiedenen Komponenten in der Gesamtsystem zu ermöglichen. Durch die offene und sogar globale Vernetzung können CPS die relevante Umgebung beobachten, um richtige Vorhersagen und Entscheidungen zu treffen. Ein Teil der Energiegewinnung in Smart Grid(sehe Kapital 5) aus Sonne. Es ist bekannt, dass sie abhängig von der Wette ist. Durch die Erfassung von Wetterdaten der globalen Vernetzung, kann man die Durchführungsmöglichkeit in der Energiegewinnung flexible anpassen(p.86)[5].

2.2.2 Context-Awareness

Die einzige unveränderte Sache in unserer Lebensumgebung ist Veränderung. Wie kann sich das cyber-physische System der sich kontinuierlich ändernden Umgebung anpassen und sich selbst weiter regulieren? Diese Frage wird durch „Context-awareness“ beantwortet.

Wir können uns vorstellen, wenn wir in einem Einkaufszentrum sind, brauchen wir nichts auf unserem Handy eingeben, sondern unser intelligentes Smartphone teilt uns schon automatisch die aktuellen Sonderangebote mit. Wenn wir vor einer bekannten Statue stehen, kann uns unser Smartphone automatisch Informationen zum historischen Hintergrund mitteilen. Wenn wir ein Medikament kaufen möchten, können wir uns schon ohne Beratung (und sogar mit einer Videopräsentation) über die ärztliche Verordnung informieren[15]. Solche intelligenten Funktionen sind nicht weit von uns weg. Sie durchdringen sukzessiv unser Leben. Durch solche Anwendungsbeispiele können wir Context-awareness zuerst erkennen.

„Context-awareness bezeichnet das Verhalten von Anwendungsprogrammen, die Informationen über ihren „Kontext“, also ihre Umgebung, benutzen, um ihr Verhalten darauf abzustimmen. Das System basiert auf Informationen, welche durch unterschiedliche Sensoren und Aktoren zur Verfügung gestellt werden“ [12].

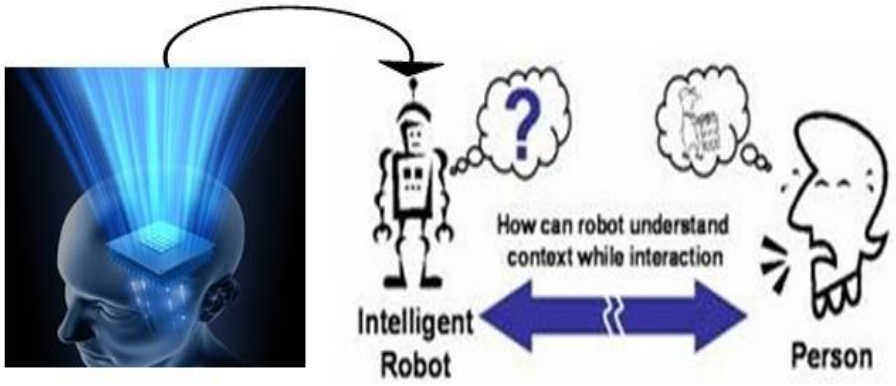


Abbildung 4: Context-awareness zwischen Menschen und Robot[3]

Durch die Eigenschaft der Context-awareness ist der Nutzwert des Systems erhöht, also kann sich das System durch intelligente Sensoren und Aktoren an stetig wechselnde, unvorhergesehene Situationen anpassen und besitzt Selbständigkeit(p.141f)[5]. Damit kann man feststellen, dass Context-awareness auch eine wichtige Eigenschaft von CPSs ist. Ohne Context-awareness wäre ein cyber-physisches System nicht mehr „intelligent“.

3 Cyber-physische Systemarchitektur

Bezüglich einer generalisierbaren Architektur unterscheiden wir zwischen grundsätzlich zwischen logische Struktur und physikalische Struktur.

3.1 Logische Struktur

Eines der grundlegenden Merkmale der heutigen CPS ist die Existenz eines Communicationsnetzes zur Vermittlung zwischen und unter Computer- und phy-

sikalischen Einheiten. Die Vernetzungen können als Communicationskanäle betrachtet werden. Dies setzt voraus, dass umfassende Vernetzungen im Einsatzkontext des CPS zugänglich sind[16]. Zusammen mit der Communication bilden Computation und Control(Steuerung) die drei grundlegenden Merkmale des cyber-physischen Systems. Sie verbinden durch Informationen den physischen Raum und den Cyberspace eng miteinander, um eine enge Kopplung zu bilden[16]. Diese logische Struktur wird in folgender Abbildung dargestellt:

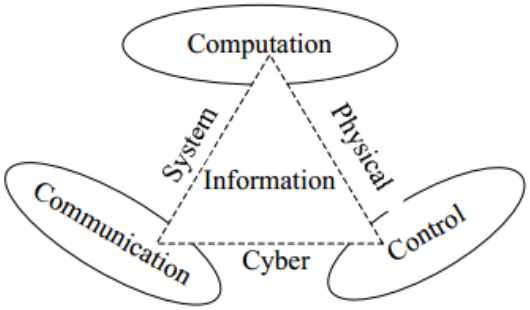


Abbildung 5: logische Struktur der CPS[16]

Bei der logischen Struktur der CPSs gibt es eine Beziehung zwischen physischem Space und Cyberspace und dadurch entsteht ein informationszentriertes System. Im Cyberspace sind Communication, Control und Computation diskret miteinander, aber im Gegensatz dazu auch eng miteinander verbunden. Durch Sensoren und andere relevante Geräte übergeben die Entitäten im physischen Space die Informationen an den Entscheider. Der Entscheider bewertet diese und trifft aufgrund bestimmter Regelungen eine Entscheidung. Dadurch werden Entscheidungen vom Cyberspace in den physischen Space übergeben, und dies ermöglicht die Kontrolle durch den physischen Space. Die Elemente in Cyberspace und physischem Space sind voneinander abhängig und beeinflussen sich gegenseitig[16].

3.2 Physikalische Struktur

In Verbindung mit Merkmalen des CPSs und erwarteten Funktionen wurde eine physikalische Struktur des CPS konstruiert. Diese Struktur lässt sich in drei Teile aufteilen, und zwar Steuerschicht, Netzwerkschicht und physikalische Schicht[16].

- Steuerschicht: Menschen und CPS-Einheit dienen als Steuerungseinheiten. Sie bestimmen die semantische Kontrolle anhand von zu erreichenden Zielen. Darüber hinaus wird es in die Regelungswerk eingefügt, sodass sie als Entscheidungsregel genutzt werden.
- Netzwerkschicht: Die Netzwerkschicht in der Mitte der Abbildung ist der Kanal für Informationsübertragung. Die Entscheidungen oder Befehle der Steuerschicht sowie die abstrakten Informationen aus physikalischer Schicht werden durch die Netzwerkschicht gleichzeitig übertragen, sodass die Interaktionen zwischen physikalischer Schicht und Steuerschicht ermöglicht werden.
- physikalische Schicht: Die physikalische Schicht ist die unterste Schicht der Struktur, die aus Sensoren und Aktoren besteht. Nach der Ankunft der oberen Anweisungen durch die Netzwerkschicht an die Aktoren werden die Anweisungen der Aktoren durch die Entitätensteuerung ausgeführt. Sensoren erfassen die Informationen aus der physikalischen Schicht und entscheiden anhand der Regelungen, ob die Informationen in die Steuerschicht übergeben werden sollen. Die Beziehungen zwischen den drei Schichten werden wie folgt abgebildet[16]:

In der physikalischen Struktur des CPS entsteht eine Feedback-Schleife durch CPS-Einheit, Mensch, Vernetzungen sowie Sensoren und Aktoren. Die oberen Steuerregelungen werden von Menschen und CPS-Einheit gestaltet. Zuerst erfassen Sensoren Informationen aus relevanten Untersuchungsgebieten (z. B. Temperatur, Verkehrssituation, Wetterdaten usw.), vergleichen diese dann mit vorher definierten Regelungen nach Abstrakten, um eine vorläufige Entscheidung zu treffen. Wenn die Ergebnisse sich innerhalb der erlaubten Begrenzungen befinden, führen wir keine Aktion aus. Ansonsten werden die Ergebnisse über die Vernetzungen der CPS-Einheit übergeben. Nach der Bewertung anhand der oberen semantischen Regelungen und Menschen werden die Anweisungen wieder an Aktoren übergeben, sodass sie die Entitäten kontrollieren, um den physischer Space zu verändern und um erwartete Ziele nach den Anforderungen der Nutzer zu erreichen[16].

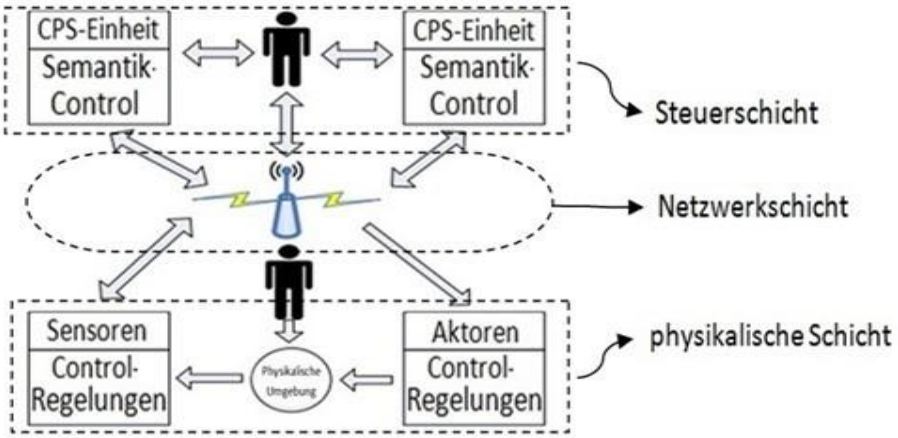


Abbildung 6: physikalische Struktur von CPSs[16]

In der logischen und physikalischen Struktur der CPSs werden der Computing-Prozess und der physikalische Prozess durch Vernetzung eng miteinander verbunden. Die Mensch-Maschine-Schnittstelle ermöglicht die Interaktion zwischen Menschen und Maschinen. Die Vernetzungen der CPS haben die Eigenschaften der Self-Adaption, Self-Organisation, Self-Adjustment und Self-Control[16]. CPS-Einheit, Sensoren und Aktoren haben auch die Eigenschaft der Self-Awareness (sehe Kapital 2.2.2) unter relevante Steuerregelungen. Dadurch haben CPSs auch die Eigenschaften Self-Awareness, Self-Judgement, Self-Adjustment und Self-Control. Bezüglich solch spezifischer Eigenschaften kann man sagen, dass das cyber-physische System ein intelligentes System ist.

3.3 Schnittstellen "Cyber-physisches System"

3.3.1 Mensch-Maschine-Schnittstelle

Eine Mensch-Maschine-Schnittstelle bildet - wie der Name schon sagt – „die logische Trennlinie zwischen den zwei Systemen Mensch und Maschine“[6]. Eine Durchführung eine Aufgabe wird durch Kommunikation der beiden Systempartner über jene Schnittstelle erledigt. Generell ist „eine Schnittstelle eine Einrichtung mit Übersetzungs-und Vermittlungsfunktion zwischen gekoppelten Systemen“[8]vgl.[6]. Die Schnittstellen hat die Fähigkeit, Komplexität verbergen zu

können. Der Benutzer einer Maschine, der Mensch-Schnittstelle vertritt, weiß normalerweise keine Details über ihre technische Realisierung[8]. Also muss der Benutzer nicht wissen, wie ein Geldautomat funktioniert. Wichtig ist, durch seine drucken mit Tastatur einige Funktionen wählen zu können, um Geld ein- und auszuzahlen.

Mensch-Maschine-Schnittstellen sind uns nicht fremd. Sie haben uns schon viele Vorteile mitgebracht. Z.B. erleichtert sie bestimmte Arbeite und Aufgaben von Benutzer nur durch entsprechende einfache Bedienungen[8].

3.3.2 "System of Systems"

Das nächste Merkmal des CPS ist "System of Systems"**,** (in 1.Kapitel schon kurz erwähnt). CPS ist ein „großes“ Gesamtsystem, es bestehen meist aus vielen „kleinen“ Einzelsystemen. Solche Einzelsystemen sind auch eine vollständige intelligente System, also sie haben alle entsprechende Eigenschaft von Systeme, z.B. Self-Awareness(sehe Kapital 2.2.2)[16]. Solche Einzelsysteme sind miteinander vernetzten und können sich untereinander koordinieren. Aufgrund der Zusammensetzung des Gesamtsystems aus mehreren Einzelsystemen werden CPS daher gelegentlich auch als "Systems of Systems" bezeichnet[9]. Dienste und andere CPS-Einzelsystem werden beim Einsatz dynamisch genutzt und verbunden. Das gilt nicht nur innerhalb des kontrollierten Bereichs, sondern auch außerhalb des Bereichs der CPSs(p.23)[5]. Diese kooperieren also mit anderen Systemen oder Teilsystemen, die z. B. noch nicht identifiziert sind. Beispielsweise vernetzt sich ein Fahrzeug in der vernetzten Mobilität dynamisch mit einer ortsfesten Infrastruktur und auch mittlerweile mit anderen Fahrzeugen und dadurch entsteht ein cyber-physisches System(p.21)[1].

3.4 Relevante Technologien für CPSs

CPSs sind zurzeit ein neues und populäres Forschungsthema. Ihre Techniken basieren meist auf anderen bereits vorhandenen Techniken. Die relevanten Techniken umfassen „Wireless-Netzwerktechnologie, Sensornetzwerktechnologie (Sensornetz), Embedded Software Technologies, intelligente Regeltechnik, künstliche Intelligenz, Kybernetik (Steuerungstechnik), Kommunikationstechnik, Verteilte

Technologien usw.“[16]. CPSs integrieren alle wichtigen Eigenschaften aus solchen Techniken, um ein integriertes System von Berechnung, Kommunikation und Steuerung in der nächsten Generation aufzubauen[16]. Die relevanten Technologien der CPSs sind in folgender Abbildung dargestellt.

Hier werden nur die wichtigsten Technologien detailliert dargestellt.

Abbildung 7: relevante Technologie für CPSs[16]

1）Ad-hoc-Netzwerktechnologie.

Ohne solch umfassende Vernetzung können die Komponenten von CPSs nicht miteinander kommunizieren und somit funktioniert das System nicht. Im sogenannten Internet der Dinge ist jede physikalische Entität adressierbar, damit sie identifizierbar ist und extern gesteuert werden kann.

2) Sensor-Aktor-Netze

CPS sind komplexe intelligente Systeme. Die Wahrnehmung der Veränderung der physikalischen Umgebung ist ihr wichtiges Merkmal. Diese Funktionen werden durch ein Sensornetz ermöglicht.

Ein Sensornetz ist ein „Rechnernetz von Sensorknoten, d. h. winzigen bis relativ großen per Funk kommunizierenden Computern, die entweder in einem infrastrukturbasierten oder in einem sich selbst organisierenden Ad-hoc-Netz zusammenarbeiten, um ihre Umgebung mittels Sensoren abzufragen und die Information weiterzuleiten“[17].

3）Die eingebettete Technologie wurde bereits im vorigen Kapitel erklärt. Sie ist eine wichtige Technologie für CPS. Außerdem basiert die Entwicklung des CPS auf eingebetteten Systemen.

4) Intelligente Regeltechnik unterstützt „Control“(Steuerung) des Systems. Und die künstliche Intelligenz ermöglicht „Context- Awareness“.

4 Herausforderungen

Um die CPSs im Alltag durchzuführen und ihre Vorteile für die ganze Gesellschaft zu realisieren, müssen wir zuerst alle technischen, theoretischen und anderen vielfältigen Herausforderungen erkennen und überwinden.
Es handelt sich um die „gesellschaftliche Akzeptanz, Zielkonflikte und erforderliche Garantien von erweiterter Betriebs- und IT-Sicherheit, Verlässlichkeit und Schutz der Privatsphäre“(p.27)[5].

4.1 Technische Herausforderungen

Grundsätzlich geht es um Technologien, die zu entwickelnde Techniken und Verfahren und zur Realisierung von CPS-Eigenschaft benötigt werden(p.127)[5]. Nur wenn diese erforderliche Technologie erfüllt wird, können die speziellen Fähigkeiten der CPSs beherrscht und benutzt werden (z. B. mehr Intelligenz und Sicherheit im Straßenverkehr). Die erforderliche Technologie für CPSs sind Erkennung der Umgebungssituation, intelligente Sensoren, Echtzeitregelung, vernetzte Kommunikation usw. Solche Technologien beziehen sich auf verschiedene Systemebenen und Architekturen, z. B. „Mensch-Maschine-System, Gesamtsystem, Schnittstellen, Vernetzung, Software und Hardware, Sicherheitstechniken und Kommunikationstechnik“(p.127ff)[5].

X-Awareness bezeichnet die richtige „Wahrnehmung und Interpretation“ von Umgebungssituation und Einsatzkontext(p.141)[5]. Sie ist eine wichtige Eigenschaft der CPSs. Aber die heutigen vorhandenen Technologien sind noch nicht genug fähig für die erforderliche X-Awareness der CPS. Die erste Herausforderung ist, wie man Sensortechnologien verbessern kann. Denn das CPS wahrnehmen seine Umgebung nur durch Sensoren, die ähnlich wie seine „Hand“ und daher sehr wichtig ist. Danach liegt Herausforderung in einer Verbesserung der Erkennung komplexer Situationen und der dazu entstehende Echtzeitanforderung.

Denn es gibt nicht nur einfach und bekannte Situationen, sonder auch fremde und komplexe Situationen, damit entsteht eine Vielzahl von Datenmengen. Die intelligente CPS sollte natürlich in der Lage sein, die großen Datenmengen in Echtzeit effizient zu bearbeiten sowie die unbekannte Situation anzupassen(p.141)[5].

Die jeweilige physikalische Entität in CPSs ermöglicht Interaktionen miteinander durch das Netz. Wegen des momentanen Fehlens der vorhandenen Technologie und Ausstattungen, sowie niedrigen Breitbands und relativ großer Zeitverzögerungen (Network-Induced Delay) kann die Service-Qualität nicht zu 100% gewährleistet werden. Das vorhandene Internet kann die Durchführung der CPSs, welche strenge Anforderungen, nämlich Echtzeit haben, nicht ermöglichen. Hier wirft sich die Frage auf, wie die Zeitverzögerung entsteht Bereits in Kapitel 3.1.1 Logische Struktur wurde erklärt, dass ein Netz zwischen der Computing und der physikalischen Entität existiert und die Kommunikationsmöglichkeit zur Verfügung stellt. Wegen des niedrigen Breitbands kann bei der Interaktion zwischen Kontroller und physikalischem System deswegen eine Netzwerkverzögerung (Network-induced delay) entstehen[16].

Das Netz benötigt eine große Bandbreite und sollte die Eigenschaften der Self-Organisation und Self-Adaptation haben. Aufgrund der höheren Anforderungen, nämlich Echtzeit, muss das Netz die Fähigkeit des Self-Adjustment im Fall von Stau besitzen. Darüber hinaus muss eine gewisse QoS (Quality of Service) gewährleistet werden. Das Netz liefert eine einheitliche Standardzeit für das gesamte System. Die Komponenten der Systeme - ebenso wie die Menschen - sollen Anweisungen oder Befehle anhand dieser Standardzeit erteilen und ausführen[16].

Die Forschung für das eingebettete System, die auf der neuen nächsten Generation des Internet (Next Generation Network) basiert, hat bereits angefangen. Es ist ein offenes und erweiterbares Netz. Dieses Next Generation Network kann im Smart Grid (Stromversorgung) angewendet werden. Durch diese Anwendung kann die Herausforderung der Abhängigkeit von einem separaten Netzwerk überwunden werden.

Die Geschwindigkeit und Medien der Kommunikation spielen eine wichtige Rolle für die Echtzeitfähigkeit und Reaktionsfähigkeit von CPSs(p.142)[5]. Nur eine

kontinuierliche Kontextaktion und kooperatives Handeln können die nötige X-Awareness, sowie die weiteren erforderlichen Eigenschaften ermöglichen. Wenn die Bearbeitung der Information und Reagieren zu lange dauert, wird es Echtzeitfähigkeit fehlen und zur Zeitverzögerung (Network-Induced Delay) führen. Bei der Anwendung bestehen auch Herausforderungen, die mit den heutigen vorhandenen Technologien nicht vollständig geschaffen können(p.142)[5].

4.2 Wissenschaftliche Herausforderungen

Die wissenschaftliche Herausforderung ist zuerst, wie ein geeignetes System gestaltet wird, die alle notwendigen Eigenschaften für CPSs umfasst(p.26)[2]. Darüber hinaus ist eine weitere wissenschaftliche Herausforderung die „Erschließung der verschiedenen Anwendungsgebiete anhand der speziellen Eigenschaften der CPSs“(p.26)[2].

Das Design der Mensch-Maschine-Interaktion(sehe Kapital 3.2.1) bezieht sich auf der Erkennung der Verhaltens, der Wünsche und Ziele der Nutzer. Hier liegt die Herausforderung in der Vorhersage menschlichen Verhaltens und der entsprechende Gestaltung der Mensch-Maschine-Interaktion(p.142)[5]. Die Gestaltung geeigneter Mensch-Maschine-Interaktionen ist deswegen besonders wichtig, weil sie dem Nutzer eine „situationsgerechte Koordination und Steuerung von CPSs ermöglichen können“(p.142)[5].

Die Umsetzung neuer, dynamischer Systemmodelle stellt Anforderungen an die Systemarchitekturen. CPS bestehen aus verschiedenen Komponenten, die aus unterschiedlichen Fachgebieten stammen könnten, z.B. Elektronik, physikalischen Systemen usw. CPS ist ein Gesamtsystem, deshalb ist ein neuer „ganzheitlicher Systemische Sicht“ erforderlich(p.24)[1]. Dieses bezieht sich auf die interdisziplinäre Verknüpfung von Fachgebieten der Physik, der Informatik und des Maschinenbaus usw. Beispielsweise ist die Anwendung von ABS (Anti-Blockier-Systemen) abhängig von der interdisziplinären Verbindung von verschiedenen Technik und Methode(p.24)[1]. Ohne solche unterschiedliche untereinander vernetzte Wissenschaftszweige können die Spezialität und Funktionen des CPS nicht ermöglicht werden. Die vorhandenen relevanten Wissenschaftskenntnisse benötigen auch Innovationen und stetige Entwicklungen. Die Integration der stetig ent-

wickelten interdisziplinären Wissenschaftskenntnisse ist nicht einfach. Außerdem wissen wir nicht genau, ob die neu entwickelte Systemarchitektur zugänglich ist.

Die Einsatzwelt verändert sich kontinuierlich. Hier stellt sich die Frage, ob sich das System an eine sich ändernde Umgebung anpassen kann. Die Fähigkeiten zur Selbstorganisation und zur Adaption sind besonders bedeutend für die Koordination der Systeme(p.136)[5]. Da das endliche Ziel des CSP ist, dass es nach die richtige Wahrnehmung seiner Umgebung und Bearbeitung der Informationen selbst richtige Entscheidung treffen kann. Aber die zentrale Frage ist, wie das System mit ausschließlich stochastischem, sehr selten auftretendem Verhalten umgehen kann, um die präzisen Realzeitanforderungen zu erfüllen (p.142)[5].

Nach der Überwindung der technischen Herausforderungen liegt die Herausforderungen in Zukunft in die Realisierung neuer erforderlich Funktionen durch Adaptation (p.24)[1]. Die vorhandenen Anwendungsmodelle können in einer komplexen Umgebung kein absolut sicheres Arbeiten gewährleisten. Unser Ausbildungsniveau und die Entwicklungsprozesse sind bisher nur eingeschränkt, cyberphysische Systeme zu schaffen.

4.3 Wirtschaftliche Herausforderungen

Die traditionelle Systementwicklung, Industrie und Dienstleistung, die schon sehr reif sind und Erfolge in den Entwicklungsprozessen aufweisen können, werden durch CPS-Trends, Systemwandel und Dynamik der Innovationen vollständig oder teilweise „zerstört“ und vor große wirtschaftliche Herausforderungen gestellt(p.22)[2]. Der Systemwandel und die offene Vernetzung von CPSs erfordern, dass beteiligte Betrieb neue Geschäftsmodelle sowie Organisations- und Kooperationsformen entwickeln müssen(p.175f)[5]. Das bedeutet auch, dass alle alten traditionellen Systeme grundlegend verändert werden müssen. Dies bezieht sich auch auf die Architektur der Wertschöpfungsnetze und Anwendungsprozesse. Zusätzlich muss man die Kosten berücksichtigen. Diese Innovation bringt nicht nur Vorteile mit sich, sondern auch Risiken: fehlende Gestaltung, Gefährdung der sozialen Sicherheit sowie Verständnis, Akzeptanz und die Unterstützung in der Bevölkerung(p.26f)[1].

4.4 Sicherheitsherausforderungen

Hierzu kommen Fragen der funktionalen Sicherheit. Unsere Erwartung ist, dass durch die Systeme und deren Einsatz grundsätzlich keine Lebensgefahr entsteht. Aber aufgrund der noch fehlenden Technik und nicht perfekter Anwendungsgelegenheiten bringen CPS möglicherweise einige Gefahren mit sich. Deshalb muss man die Sicherheit und Zuverlässigkeit der Systeme berücksichtigen(p.143f)[5]. Bereits bei kleinen Fehlern könnten Menschen zu Schaden kommen oder Materialien zerstört werden.

Der Begriff Sicherheitsproblem bezieht sich auch auf Datensicherheit(p.146)[5], d. h. aufgrund der offenen Vernetzung könnten Datenverluste entstehen. Die Anforderung von Zuverlässigkeit spielt eine entscheidend Rolle für Sicherheitssysteme in sicherheitskritischen Bereichen z. B. in Kernkraftwerken(p.22)[2]. Sicherheitssysteme werden meist spezielle in Kernkraftwerken eingesetzt. Solches System hat eine Notabschaltung, die in kritischen Situationen durchgeführt werden soll, Hier nehmen Sicherheit mit zuverlässiger Autonomie eine Schlüsselrolle ein(p.22)[2]. Das Sicherheitsproblem ist der wichtigste Aspekt, sowohl für Wissenschaftler und Techniker als auch für Regierung und Gesellschaft. Es beeinflusst auch die Akzeptanz der Bevölkerung und Durchführungsgrad der CPS.

Nur wenn alle soeben definierten Herausforderungen bewältigt, klar identifiziert und bearbeitet werden können, wird diese komplexe Technik schrittweise gelingen.

5 Anwendungen von CPSs

In Bezug auf Kosteneinsparung, Energiesparen und Umweltfreundlichkeit sowie höheren Komfort und Assistenz für Mobilität werden CPS für die Zukunft in folgenden Anwendungsgebieten untersucht und überprüft(p.20)[1].

5.1 Mobilität – Cyber-physisches System für vernetzte Mobilität

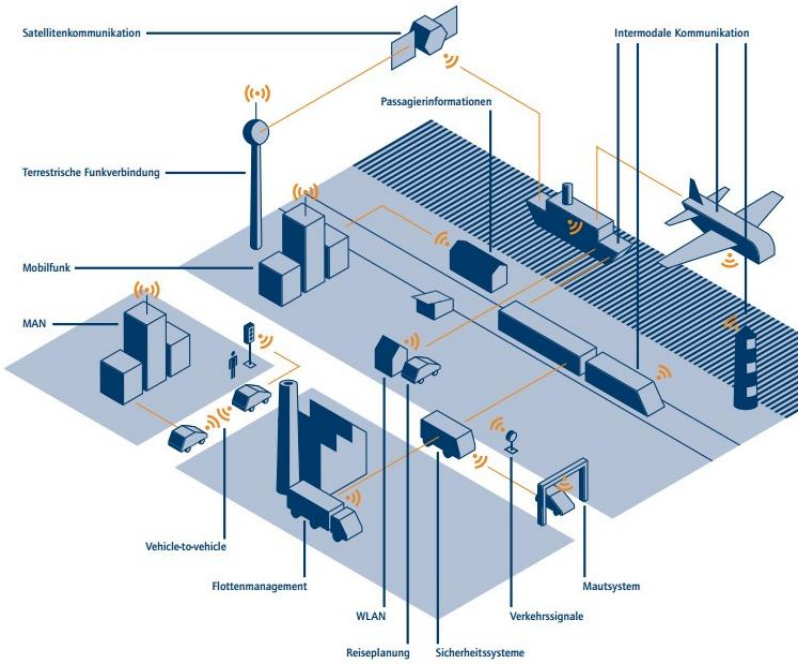


Abbildung 8: Vernetzte Mobilität durch verteiltes Verkehrsmanagement (p.21)[1]

Eine umfassende Vernetzung ist die Voraussetzung für die Durchführung des Mobilitätsystems(p.29)[2]. Da nicht nur einzelne Fahrzeuge und Verkehrsteilnehmer, sonder auch die gesamte Verkehrsinfrastruktur untereinander vernetzt werden(p.21f)[1]. Dadurch können die enthaltenden Komponenten in den Systemen echtzeitlich benötigter Verkehrsinformationen, Störungsinformation oder Wetterdaten, usw. austauschen. Die Vernetzung in einem cyber-physischen System bringt einige Vorteile mit sich: Vermeidung von Unfällen, energieeffizientes Arbeiten und Verringerung von CO_2-Emissionen (p.21)[1]. Solche Vorteilen werden durch ein Beispiel erklärtet: Erkennt ein Navigationssystem, dass eine 300 m entfernte Ampel bald auf Rot schalten wird, kann das Auto bremsen und somit Energie sparen und Unfall vermeiden(p.41)[2]. Dabei werden CPS durch Sensoren die entsprechenden Umgebungsinformationen, wie z.B. Verkehrsstau, Wetterdaten und relevante Zustandsänderungen wahrnehmen und auf diese reagieren. Durch die vielfältige Eigenschaften und Fähigkeit des CPS wird verteiltes Ver-

kehrsmanagement ermöglicht. CPS dienen Hier als Verkehrsassistenz für Verkehrsplanung und Steuerung(p.20f)[1].

Elektromobilität, also Elektrofahrzeuge, sind nicht mehr eine neue Begriff in unserem Alltag, sondern werden schon durch einige umweltfreundliche und innovative Unternehmen untersucht und produziert(p.63)[2]. Cyber-physische Systeme spielen auch große Rolle im Elektromobilitätsbereich, weil „sie die Grundlage für das Energie-, Batterie-, und Lademanagement liefern“(p.21)[1]. Außerdem bieten sie ein intelligentes Stromnetz(„Smart Grid“, sehe nächste Teil), wodurch die Elektrofahrzeuge einfach Strom tanken können. Der größte Vorteil von Elektrofahrzeuge ist emissionsfrei(p.63)[2].

5.2 Energie: Cyber-physisches System für das Smart Grid

Abbildung9: Smart Grid[14]

Smart Grid ist ein Intelligent und moderne Stromnetz[10]. Dieses System versorgt nicht nur herkömmlich Energie, sondern auch neu Energie, die aus Sonne und

Wind erzeugt werden. Aber solche Neu Energie sind abhängig von der Wetter und Tageszeit, und sind nicht immer kontinuierlich verfügbar(p.20)[1]. Wegen der Volatilität erfordert hier eine Steuerung und technische Unterstützung. CPS spielt hier eine wichtige Rolle.

Die Sensoren von CPS prüfen die Lauffähigkeit des Stromnetzes und liefern eine rapide Reparatur bei einer Störung, um Energieverlust zu vermeiden und kontinuierlich Energiegewinnung zu unterstützen[13]. CPS ermöglicht die Kommunikation zwischen verschiedene Komponente des Systems. Es gibt zwei Kanal für Kommunikation: Normalerweise werden Strom nur in eine Richtung von Kraftwerk gezogen, stattdessen können bestimmte Geräte auch Energie zurück ins Netz speisen[13]. Also es gibt ein Speicher. Der Speicher kann Energie aufnehmen, wenn erzeugt Energie mehr als Verbrachte Energie im Netz wird. Umgekehrt kann der Speicher die Energie auch zurückspeisen. Es gibt noch eine intelligent Stromzähler. Dadurch misst und überwacht der Nutzer den Stromverbrauch, um den Verbrauch zu minimieren[13] vgl. (p.52)[5]. Durch die Steuerung des CPS können die Auslastung von herkömmlich Energie immer herunterfahren werden, wenn neue Energie mehr Strom produzieren[13]. Dadurch ermöglicht CPS stabile und intelligente Energieversorgung.

6 Zusammenfassung und Ausblick

Cyber-physische Systeme haben viele intelligente Funktionen und Eigenschaften. Solche Funktionen sind nicht weit von uns weg. Sie durchdringen sukzessiv unser Leben. Das heißt, dass CPSs kein mehr nur eine „Denkidee“ sind, sonder durch Entwicklung der Technologie und Überwindung der Herausforderungen ermöglicht werden. Sie können uns in Alltag in vielfältiger Weise assistieren und neue Möglichkeiten bieten.

Eingebettete Systeme werden in Zukunft durch CPS ersetzt, da diese den Nutzern mehr Bequemlichkeit und Möglichkeiten bieten. Der Wandel ist sinnvoll. Allgemein werden die CPS benutzerfreundlicher und leistungsfähiger als eingebettete Systeme und daher auf lange Sicht den Markt übernehmen.

Durch die Anwendungsbeispiele in Kapital 5 wissen wir, dass sie in Zukunft den Verkehr durch Koordination sicherer machen und den CO_2-Ausstoß durch geringeren Treibstoffverbrauch reduzieren werden. Darüber hinaus spielt CPSs in Smart Grid eine große Rolle, die herkömmliche verfügbare Energie aus konventionellen Kraftwerken z.B. Kohle und Gas schrittweise durch nachhaltige und erneuerbare neue Energie ersetzt werden. Der Wandel ist sinnvoll und umweltfreundlich.

Cyber-physische Systeme werden in Zukunft immer wichtiger Rolle in der Welt spielen und in bisher kaum vorstellbarer Art und Beiträge für unser alltägliches Leben leisten.

Literaturverzeichnis

1.acatech (Hrsg.): *Cyber-Physical Systems.Innovationsmotro für Mobilität, Gesundheit, Energie und Produktion(acatech POSITION)*, Heidelberg u.a.: Springer Verlage 2011.

2.Broy, M. (Hrsg.). In M. Broy: *CYBER-PHYSICAL SYSTEMS-Innovation Durch Softwareintensive Eingebettete Systeme*.München: Springer Verlag 2010.

3.*Context-Awareness.* (30. 12 2012). Abgerufen am 03. 01 2013 von http://sclab.yonsei.ac.kr/team/team_page.php?language=eng&team_id=5

4.Edward A. Lee. Electrical engineering and computer sciences. Technical report, Berkeley, 2008.

5.Geisberger, Eva/Broy, Manfred: *agendaCPS- Integrierte Forschungsagenda Cyber-Physical Systems(acatech STUDIE),*Heidelberg u.a.:Springer Verlag 2012.

6.Halbach, W.R.*: Interfaces : Medien- und kommunikationstheoretische Elemente einer Interface-Theorie, Fink, München 1994*

7.*Handspiel.Magazin*. (14. 02 2011). Abgerufen am 22. 01 2013 von http://www.handspiel.net/magazin/2011/02/14/erweiterte-realitat-wenn-physische-und-virtuelle-welt-verschmelzen/

8.*IT-Trends*. (15. 08 2012). Abgerufen am 29. 12 2012 von http://130.75.63.115/upload/lv/wisem0708/SeminarIT-

Trends/html/ms/#_Toc186989113

9.Klie, D. T. (2011). *Technische Fakultät*. Abgerufen am 01 2013 von http://www12.informatik.uni-erlangen.de/edu/cps/SS11/

10.*NIST-Smart Grid Interoperability Standards Roadmap*. Abgerufen am 02 2013 von *http://www.williams-pyro.com/content/file/Smart%20Grid%20PDF.pdf*

11.*nuinno Fachbegriffe*. (2013). Abgerufen am 06.03.2013 von http://www.nuinno.de/service/fachbegriffe/#embedded systems

12.Rosemann, M., & Recker, J. *(2006). "Context-aware process design: Exploring the extrinsic drivers for process flexibility". In T. Latour & M. Petit.* 18th international conference on advanced information systems *engineering. proceedings of workshops and doctoral consortium*. Luxembourg: Namur University Press. pp. 149–158.

13.*Spiegel Online*. Abgerufen am 01 2013 von http://www.spiegel.de/wirtschaft/unternehmen/bild-667878-42650.html

14.“The Smart Grid: An Introduction,” *2008, a publication sponsored by the U.S. DOE's Office of Electricity Delivery and Energy Reliability.*

15.Wang, L. M. (09 2011). *CNKI*. Abgerufen am 01 2013 von http://www.cnki.net/kcms/detail/11.2560.tp.20110908.1645.001.html

16.Wang, Z. (2011). Cyber Physical Systems-网络物理系统. *Mini-micro Systems* (小型微型计算机系统), S.8.

17.W. Dargie and C. Poellabauer, *"Fundamentals of wireless sensor networks: theory and practice", John Wiley and Sons, 2010 ISBN 978-0-470-99765-9 Sensornetz*.

18.*weltweite-Vernetzung* (15. 10 2012). Abgerufen am 30. 12 2012 von http://www.itler.net/2010/12/facebook-grafik-zeigt-weltweite-vernetzung/

Vergleich von Quality of Context und Anomalieerkennung im Bereich der Aktivitätserkennung

Seminar: Ubiquitäre Systeme WS 2011/2012

Antim Mironov
Betreuer: Markus Scholz

Telecooperation Office (TecO)
Karlsruher Institut für Technologie (KIT)

Zusammenfassung Im Rahmen dieser Arbeit werden die Ansätze von Quality of Context und Anomalieerkennung (*Outlierdetection*) im Bereich der Aktivitätserkennung erläutert und gegenübergestellt. Es werden Beispiele für die Realisierung dieser Ansätze dargestellt und die Hauptprobleme identifiziert. Am Ende wird ein kombiniertes System vorgeschlagen, das mit dem Problem einer hohen False-Alarm-Rate bei der Erkennung von Anomalien umgeht.

1 Einleitung

Die Vorstellung von Weiser [Wei91] über die ubiquitären Technologien (*ubiquitous computing*), die ihre Umgebung wahrnehmen und ihr Verhalten entsprechend daran anpassen, ist heutzutage nicht mehr nur eine Vision, sondern realisierbare Möglichkeit. Diese Technologien umfassen Geräte, die ihren Kontext verstehen können (*context aware computing*). Der Kontext erfasst Informationen über eine Situation, die sich auf die Interaktion zwischen Benutzern, Anwendungen und ihrer Umgebung bezieht [DAS01]. Die Handlung eines Akteurs bzw. eines Benutzers macht ein wichtiges Teil dieses Kontextes aus. Deshalb ist die Aktivitätserkennung als Prozess, der diese Information zur Verfügung stellt, so wichtig und wird aktiv untersucht [CK11].

Ein intelligentes Haus, das mit Sensoren ausgerüstet ist und sich an die Aktivitäten seiner Bewohner anpassen kann (z. B. Anpassung der Innenraumtemperatur), ist ein Beispiel dafür, wie die Ideen von Weiser angewendet werden können. In diesem Fall sind die Handlungen gewöhnliche Aktivitäten des Alltags wie z. B. das Haus verlassen, Essen vorbereiten, Fernsehen, Schlafen etc. Es gibt Situationen, in denen man an ungewöhnlichen Aktivitäten bzw. selten auftretenden unvorhersehbaren Ereignissen interessiert ist, die eine hochwertige Information für den Kontext liefern können. Dies kann z. B. der Fall sein, wenn ein Bewohner eines intelligenten Hauses einen Unfall erlebt und dies eine schnelle Reaktion voraussetzt wie z. B. Notfall signalisieren [SLP11]. In diesem Fall müssen Techniken zur Erkennung von Anomalien eingesetzt werden.

Soweit die Handlung eines Akteurs wichtige Informationen zum Kontext liefert und wichtige Entscheidungen auf Basis dieser Informationen getroffen werden können, besteht Interesse daran zu wissen, wie hoch die Qualität dieser Informationen ist, d. h. bis zu welchem Grad die Kontextinformationen der Realität entsprechen [VRL+09]. Es kann sein, dass sich die Nutzung von qualitätsniedriger Information über die Aktivitäten nicht lohnt, weil dies zu schlechten oder sogar falschen Entscheidungen führen könnte. Die Ansätze von Quality of Context (QoC) ermöglichen die Einschätzung der Qualität von Kontextinformationen.

Ziel dieser Seminararbeit ist, den Zusammenhang zwischen der Aktivitätserkennung und dem QoC bzw. der Anomalieerkennung zu erläutern sowie auch die Methoden dieser Ansätze in Bezug auf Gemeinsamkeiten und Unterschiede gegenüberzustellen. Es werden die Kernprobleme dargestellt und die möglichen Lösungen diskutiert.

In Kapitel 2 dieser Seminararbeit wird der Prozess der Aktivitätserkennung dargestellt. Es werden die einzelnen Schritte mit ihren Besonderheiten beschrieben. In Kapitel 3 wird der Begriff Quality of Context erläutert und die Methoden dieses Ansatzes mit konkretem Beispiel vorgestellt. Kapitel 4 befasst sich mit den Methoden der Anomalieerkennung und ihren Vor- und Nachteilen. In Kapitel 5 werden die Ansätze von QoC und die Erkennung von Anomalien gegenübergestellt. Die Hauptprobleme dieser Ansätze werden identifiziert. In Kapitel 6 wird ein kombiniertes System zur Erkennung von normalen und abnormalen Aktivitäten vorgeschlagen. Bei der Konzipierung des Systems wird angestrebt, die Nachteile der etablierten Systeme zur Anomalieerkennung zu vermeiden. Im abschließenden Kapitel 7 werden die wichtigsten Schlussfolgerungen des Seminars zusammengefasst.

2 Grundlagen der Aktivitätserkennung

In diesem Kapitel werden die Hauptschritte des Prozesses der Aktivitätserkennung und die damit verbundenen Ansätze erläutert. Es werden auch verschiedene Metriken für die Evaluation von Systemen zur Aktivitätserkennung vorgestellt.

2.1 Prozess der Aktivitätserkennung

Aktivitätserkennung ist der Prozess der Überwachung und der Analyse des Verhaltens eines Akteurs und seiner aktuellen Umgebung mit dem Ziel, auf die laufenden Aktivitäten dieses Akteurs zu schließen [CK11]. Abbildung 1 stellt die Hauptschritte dieses Prozesses dar, die im Folgenden ausführlich erläutert werden.

Im ersten Schritt der **Datenerhebung** werden die Daten über die Handlung des Akteurs und seine Umgebung erfasst. Abhängig vom Anwendungsbereich des Aktivitätserkennungssystems und den zu erkennenden Aktivitäten werden zwei typische Ansätze unterschieden [CK11]. Der erste basiert auf

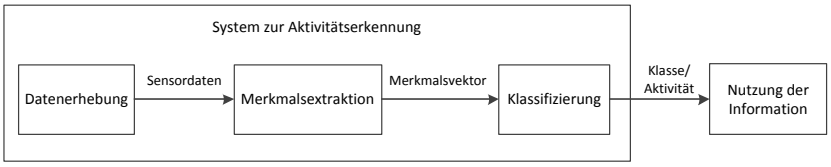


Abbildung 1. Prozess der Aktivitätserkennung (auf Basis der Beschreibung in [BI04]).

Videoaufnahmen des Akteurs und seiner Umgebung. Typische Anwendungsbereiche dieses Ansatzes sind: Mensch-Maschine-Interaktion, Roboter-Lernen, Videoüberwachung und andere. Der zweite Ansatz basiert auf Sensoren, die am Körper des Akteurs befestigt (*wearable computing*) oder in seiner Umgebung bzw. in den von ihm alltäglich benutzten Gegenständen integriert sind. Die am häufigsten verwendeten Sensoren für Aktivitätserkennung im Wearable Computing sind Beschleunigungssensoren, Gyroskope und Magnetometer [CK11]. Die Genauigkeit der Erkennung hängt von der Anzahl der Sensoren und von deren Befestigungsstelle am Körper ab. Grundsätzlich steigt der Erkennungsgrad mit der Anzahl der verwendeten Sensoren [BI04].

Bei den beiden Ansätzen ist eine passende Abtastfrequenz zu wählen, mit der die Daten erhoben werden. Nach [BKV+97] werden die normalen täglichen menschlichen Aktivitäten wie z. B. Laufen, Gehen oder Springen mit einer Frequenz durchgeführt, die in der Regel 20 Hz nicht überschreitet. Das heißt, die Abtastfrequenz muss höher als 40 Hz sein.

Der zweite Schritt ist die **Merkmalsextraktion**. Die Sensordaten stellen ein digitales Signal dar, das sich mit der Zeit verändert. Aus diesem Signal werden bedeutungstragende Informationen, die sog. Merkmale (*features*), herausgenommen, normiert und in einem Merkmalsvektor zusammengefasst [Sch10, Kapitel 5]. Es werden solche Merkmale berechnet, die eine möglichst eindeutige Identifikation und Abtrennung der einzelnen Aktivitätsklassen voneinander ermöglichen. Solche Merkmale sind z. B. der Durchschnittswert, die Standardabweichung, das Frequenzspektrum des Signals u. a. [BI04]. Eine wichtige Rolle spielt hier auch die Anzahl der Sensorwerte, für die schrittweise ein Merkmalsvektor berechnet wird. Die Anzahl der Werte zusammen mit der Abtastrate definieren ein Zeitintervall. Der Merkmalsvektor repräsentiert die Aktivität im Rahmen dieses Intervalls. Nach Bao et al. [BI04] weisen die menschlichen Aktivitäten häufig eine Periode auf. Die Anzahl der Sensorwerte bzw. das Zeitintervall muss an die Periodenlänge angepasst werden. Eine Überlappung von zwei konsekutiven Zeitintervallen bei der Berechnung der Merkmale ist möglich. Bevor die Werte der berechneten Merkmale im Merkmalsvektor erfasst werden, müssen diese Werte normiert werden, in der Regel auf dem Intervall von 0 bis 1. So haben die einzelnen Merkmale einen ausgeglichenen Einfluss bei der Klassifizierung der Merkmalsvektoren, die die Aktivitäten repräsentieren.

Nach der Merkmalsextraktion folgt die **Klassifizierung**. In diesem Schritt wird der entsprechende Merkmalsvektor durch einen Klassifikator gemäß dem

verwendeten Algorithmus einer Klasse bzw. einer Aktivität zugeordnet. Der Klassifikator stellt ein Modell dar, welches Merkmalsvektoren auf die Aktivitäten abbildet.

Die Modelle der einzelnen Aktivitäten werden durch ein sogenanntes Training des Klassifikators erstellt. In dieser Trainingsphase werden die Parameter des Klassifikators optimiert, bis die Aktivitäten so gut wie möglich modelliert sind. Dafür wird ein bestimmter Algorithmus auf einem Trainingsdatensatz angewendet [Sch10, Kapitel 5]. Der Trainingsdatensatz stellt nur einen Teil eines gesamten Datensatzes dar. Der Gesamtdatensatz umfasst die Sensordaten aller bekannten Aktivitäten. Sie werden in der Regel anhand von Experimenten mit Testpersonen, die die entsprechenden Aktivitäten ausgeübt haben, im Voraus gesammelt. Der Testdatensatz stellt den andren Teil des Gesamtdatensatzes dar. Er wird für die Evaluation des Klassifikators bzw. des Erkennungssystem benutzt (siehe Abschnitt 2.2).

Es gibt Situationen, in denen nicht alle Aktivitäten des Akteurs von Interesse sind. Das können z. B. Übergangsaktivitäten sein, die zwischen den interessierenden Aktivitäten stattfinden. Für solche Aktivitäten wird häufig eine NULL-Klasse definiert. In diesem Fall wird der Klassifikator in der Trainingsphase auch bezüglich dieser Klasse optimiert.

Die Algorithmen zur Klassifizierung lassen sich grundsätzlich in zwei Gruppen unterteilen: Algorithmen, die auf probabilistischen und stochastischen Maschinenlernverfahren basieren, und Algorithmen, die auf logische Modellierung und Deutung (*logical modeling and reasoning*) basieren [CK11]. Die zweiten werden im Rahmen dieser Seminararbeit nicht weiter diskutiert.

Abhängig davon, ob der Trainingsdatensatz annotiert ist oder nicht, werden überwachte und nicht überwachte Maschinenlernverfahren unterschieden. Bei einem annotierten Datensatz ist bekannt, welche Handlung des Akteurs welche Sensordaten produziert hat und wie lange diese Handlung gedauert hat. Damit sind die Aktivitätsklassen bekannt. Bei den überwachten Verfahren wird der Klassifikator so optimiert, dass die vordefinierten Aktivitätsklassen aus dem annotierten Trainingsdatensatz so gut wie möglich modelliert werden. Typische Algorithmen bzw. Modelle der überwachten Verfahren sind Hidden Markov Models (HMMs), dynamische und naive Bayes'sche Netze, Entscheidungsbäume, der Nearest-Neighbour-Algorithmus und Support Vector Machines (SVMs). Die nicht überwachten Verfahren versuchen, selber die möglichen Aktivitätsklassen aus einem nicht annotierten Trainingsdatensatz zu identifizieren, z. B. durch Anwendung von Methoden für Clustering oder Dichten-Schätzung. Beim Clustering werden die Daten (Merkmalsvektoren) mit ähnlichen Eigenschaften zusammen gruppiert. Bei der Dichten-Schätzung werden die Eigenschaften einer unbekannten Dichtefunktion, die die jeweiligen Aktivitätsklassen beschreibt, eingeschätzt. (vgl. [CK11])

Die letzte Phase stellt die **Nutzung** der Informationen über die erkannten Aktivitäten des Akteurs und/oder seine Handlungsumgebung dar. Diese Ergebnisse können benutzt werden, um eine Entscheidung zu treffen, die z. B. unmittelbar eine Reaktion eines übergeordneten Aktuatorsystems auslöst. Auf Basis

dieser Informationen können durch Fusion mit anderen Informationen auch neue Kontextinformation auf einer höheren Ebene erzeugt werden.

2.2 Evaluation eines Systems zur Aktivitätserkennung

Bevor ein System zur Erkennung von Aktivitäten unter reellen Bedingungen eingesetzt wird, muss bekannt sein, wie gut die Leistung dieses Systems bzw. des Klassifikators bezüglich der Erkennung ist. Verschiedene Systeme müssen in Bezug auf ihre Leistung auch verglichen werden können. Dies verlangt eine Evaluation des Erkennungssystems.

Um ein Erkennungssystem zu evaluieren, wird ein Experiment (Testphase) durchgeführt, in dem das System Aktivitäten aus einem Testdatensatz klassifizieren muss. Dieser Datensatz enthält Instanzen (Merkmalsvektoren) von bekannten Aktivitäten, für die das System trainiert wurde, die aber während der Lernphase nicht verwendet worden sind [BI04]. Die Ergebnisse dieser Testphase werden in der Regel in einer sog. Verwechslungsmatrix (*confusion matrix*) erfasst. Aus der Matrix werden danach verschiedene Metriken, die das Erkennungssystem charakterisieren, abgeleitet. Abbildung 2 zeit die Struktur dieser Matrix für den Fall mit vier Klassen. Anhand der Verwechslungsmatrix werden die erkannten Aktivitäten der jeweiligen Klassen den tatsächlichen Aktivitäten gegenübergestellt. Die Diagonale enthält die Anzahl der korrekt erkannten Aktivitäten (**TP**, *true positives*) für jede Klasse. Die anderen Elemente enthalten die Anzahl der falsch erkannten Aktivitäten, also diejenigen, die mit einer anderen Aktivität verwechselt wurden.

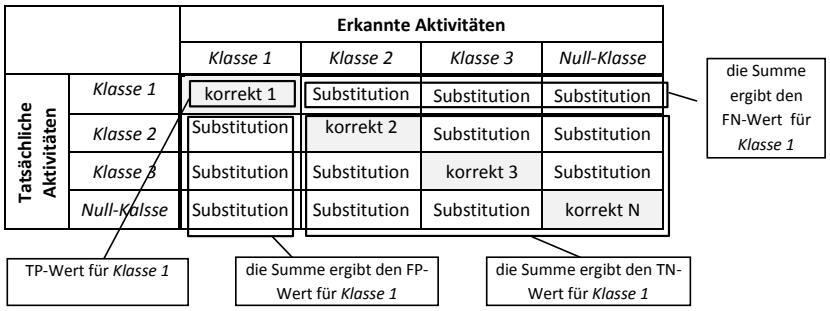


Abbildung 2. Struktur einer Verwechslungsmatrix. (vgl. [VRL+09])

Eine häufig angewendete Metrik ist die **Genauigkeit** (*accuracy*) eines Erkennungssystems. Sie wird berechnet, indem die Anzahl aller korrekt erkannten Aktivitäten durch die Anzahl aller Aktivitäten aus dem Testdatensatz dividiert wird:

$$accuracy = \frac{\sum_{i=1}^{C} TP_i}{N} \tag{1}$$

[VRL+09]. C ist die Anzahl der Klassen und N ist die Anzahl aller Aktivitäten im Testdatensatz. Nachteil dieser Metrik ist, dass die Häufigkeit der einzelnen Aktivitäten im Testdatensatz nicht berücksichtigt wird. Häufig ist der Testdatensatz in Bezug auf die Aktivitäten der jeweiligen Klassen nicht balanciert. Dies resultiert in eine Metrik, die diese Klassifikatoren höher bewertet, die die häufig auftretenden Aktivitäten besser erkennen [KAE11].

Für jede Klasse kann der **FN**-Wert (*false negatives*), der **FP**-Wert (*false positives*) und der **TN**-Wert (*true negatives*) bestimmt werden (siehe Abb. 2). Der **FN**-Wert wird ermittelt, indem alle Elemente, außer dem TP-Wert, der entsprechenden Zeile summiert werden. Bezüglich der *Klasse 1* kennzeichnet dieser Wert die Anzahl der Aktivitäten dieser Klasse, die als eine Aktivität einer anderen Klasse falsch erkannt wurden. Der **FP**-Wert kann durch Summierung der Elemente der entsprechenden Spalte, ausschließlich des TP-Werts, berechnet werden. Hinsichtlich der *Klasse 1* ist das die Anzahl der Aktivitäten, die als eine Aktivität der *Klasse 1* falsch erkannt wurden. Der **TN**-Wert wird durch Summierung der verbleibenden Elemente der Matrix berechnet. In Bezug auf die *Klasse 1* gibt der Wert die Anzahl der Aktivitäten an, die nicht der *Klasse 1* gehören und auch als eine Aktivität erkannt werden, die nicht der *Klasse 1* angehört. (vgl. [VRL+09])

Weiterhin können für jede Klasse die Metriken **Präzision** (*precision*), **Sensitivität** (*sensitivity, recall*) und **Spezifizität** (*specificity*) berechnet werden. Sie sind in [VRL+09] wie folgt definiert:

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$specificity = \frac{TN}{TN + FP} \tag{4}$$

Die Ergebnisse für die einzelnen Klassen lassen sich auch aggregieren, damit man eine einzige Metrik für das Erkennungssystem erhält, die auch den Vergleich zwischen unterschiedlichen Systemen erleichtert [KAE11]. Die Präzision und die Sensitivität lassen sich durch die Metrik **F-Measure** folgendermaßen aggregieren:

$$F - Measure = 2\frac{precision \cdot sensitivity}{precision + sensitivity} \tag{5}$$

[KAE11].

Ein anderes Kriterium, das ein Erkennungssystem charakterisieren kann, ist die **Art der Fehler**, die während der Klassifizierung auftreten können. Solche Fehler sind: *insertions* für erkannte Aktivitäten, die aber tatsächlich nicht stattgefunden haben; *deletions* für Aktivitäten, die nicht erkannt wurden, aber stattgefunden haben; *merges* für eine Folge von unterschiedlichen Aktivitäten, die aber als nur eine Aktivität erkannt wurde; *fragmentations*, wenn eine einzige Aktivität als Folge von verschiedenen Aktivitäten erkannt wurde;

overfill und *underfill* beschreiben, wie gut die Dauer einer Aktivität erkannt wurde. Diese Metriken sind insbesondere geeignet für die Evaluation von Online-Erkennungssystemen. (vgl. [VRL+09])

Eine andere Metrik ist die **ROC**-Kurve (*Receiver Operation Characteristic curve*). Sie stellt eine Grafik dar, die das Verhältnis zwischen der True-Positive-Rate (Sensitivität) und der False-Positive-Rate (False-Alarm-Rate, $FP-Rate = 1-specificity$) bei unterschiedlichen Erkennungsschwellenwerten darstellt. Die Kurve untersucht den Trade-off zwischen hoher Sensitivität und hoher Präzision [VRL+09]. Diese Metrik wird häufig für die Bewertung von Klassifikatoren verwendet, die nur zwischen zwei Klassen unterscheiden sollen, wie die Klassifikatoren für Anomalieerkennung (Unterscheidung zwischen normalen und abnormalen Aktivitäten) [YYP08,SLP11].

Andere mögliche Kriterien zur Evaluierung eines Erkennungssystems sind der Energieverbrauch und die Verzögerung, die Zeit zwischen dem Stattfinden einer Aktivität und ihrer Erkennung [VRL+09].

Bei allen vorgestellten Metriken sind die folgenden Annahmen zu berücksichtigen: Alle Aktivitäten aller Klassen sind gleich wichtig in Bezug auf ihre Erkennung; in jedem Zeitpunkt findet nur eine Aktivität statt [KAE11].

3 Quality of Context

In diesem Kapitel werden der Ansatz von QoC und die QoC-Parameter erläutert. Es wird der Zusammenhang zwischen QoC und Aktivitätserkennung beschrieben, indem die QoC-Parameter und die Evaluationsmetriken der Aktivitätserkennung in Verbindung gebracht werden. Es wird auch ein Beispiel vorgestellt, wie QoC im Bereich der Aktivitätserkennung realisiert werden kann.

3.1 Grundlegende Definitionen

Nach Dey et al. [DAS01] ist Kontext jede Information, die zur Charakterisierung der Situation einer Entität (eine Person, ein Ort oder ein Objekt) benutzt werden kann. Diese Entität ist relevant für die Interaktion zwischen einem Benutzer und einer Anwendung, einschließlich des Benutzers und der Anwendung selbst. Beispiele für Kontext sind: Ortsangaben, die Identität, die Aktivitäten von Personen oder Gruppen von Personen etc. Diese Kontextinformation hat eine Qualität, die variieren kann [BKS03]. Im Rahmen der Aktivitätserkennung kann diese Qualität durch Ausfall von Sensoren, inkorrekte Sensordaten oder falsche Klassifizierung beeinflusst werden [VRL+09]. Quality of Context stellt jede Information über die Qualität von Kontextinformationen dar. Dies bezieht sich auf die Kontextinformation und nicht auf den Prozess oder die Hardware, die die Information liefert [BKS03]. Der Ansatz von QoC definiert Parameter. Sie identifizieren, bis zu welchem Grad die Kontextinformation mit der Realität übereinstimmt [BKS03,VRL+09].

Die wichtigsten QoC Parameter, die von Villalonga et al. in [VRL+09] identifiziert wurden, sind:

- **Genauigkeit**: Sie beschreibt, wie nah die Kontextinformation an der Realität liegt.
- **Wahrscheinlichkeit der Korrektheit**: Sie gibt eine Wahrscheinlichkeit dafür, dass die Kontextinformation korrekt ist.
- **Auflösung**: Dieser Parameter bezieht sich auf die Granularität der Information.
- **Zeit der Messung**: Sie beschreibt den Zeitpunkt der letzten Erfassung der Kontextinformation.
- **Verzögerungszeit**: Sie erfasst die Verzögerung zwischen dem Auftritt der Situation in der Realität und der Verfügbarkeit der entsprechenden Kontextinformation.
- **Herkunft**: Der Parameter identifiziert die Quelle der Kontextinformation, z. B. einen Sensor oder ein anderes System (Context Provider).
- **Zeitlicher Bereich**: Er definiert eine Zeitspanne, in der die Kontextinformation gültig ist.
- **Räumlicher Bereich**: Er definiert einen physischen oder virtuellen Bereich, in dem die Kontextinformation gültig ist.

Bei der Realisierung von QoC werden nur die Parameter benutzt, die für den konkreten Anwendungsfall relevant sind.

3.2 QoC und Aktivitätserkennung

QoC gibt Informationen über die Qualität von Kontextinformationen anhand der QoC-Parameter. Im Fall der Aktivitätserkennung müssen die QoC-Parameter also Informationen darüber geben, inwieweit eine erkannte Aktivität mit der realen Aktivität übereinstimmt. Dies kann realisiert werden, indem die QoC-Parameter mit den Evaluationsmetriken der Aktivitätserkennung (siehe Abschnitt 2.2) in Verbindung gebracht werden [VRL+09].

Die Metrik Genauigkeit entspricht dem Genauigkeit-Parameter von QoC und kann für dessen Quantifizierung einbezogen werden. Da dieser Parameter einer der relevantesten für die Aktivitätserkennung ist, ist es sinnvoll, nicht die durchschnittliche Genauigkeitsmetrik über allen Aktivitätsklassen zu nutzen, sondern die Genauigkeit für jede einzelne Klasse getrennt. Auf diese Weise wird Informationsverlust vermieden, der durch die Nutzung des Durchschnittswerts entsteht. (vgl. [VRL+09])

Die Metriken *insertions, deletions, merges, fragmentations, overfill* und *underfill*, die die Art der Fehler von Online-Erkennungssystemen erfassen, haben kein Äquivalent zu den oben erwähnten QoC-Parametern. Da diese Metriken wichtige Information enthalten, ist es relevant, für Online-Erkennungssysteme die QoC-Parameter durch diese Metriken zu erweitern. (vgl. [VRL+09])

Der Parameter über die Zeitverzögerung kann eindeutig mit der Metrik, die die Verzögerung zwischen dem Aktivitätsauftritt und deren Erkennung bewertet, verbunden werden [VRL+09].

Die Evaluationsmetrik, die den Energieverbrauch eines Erkennungssystems berücksichtigt, hat auch kein Äquivalent zu den QoC-Parametern. Die Informa-

tion über den Energieverbrauch ist aber insbesondere wichtig für Erkennungssysteme, bei denen einen Trade-off zwischen Genauigkeit der Erkennung und dem Energieverbrauch möglich ist. Soweit dies keine direkte Information über die Qualität der Kontextinformation gibt, ist es irrelevant, diese Metrik in die QoC-Parameter direkt zu übernehmen. Nach Villalonga et al. [VRL+09] ist es hier sinnvoll, ein neues Konzept zu definieren – Cost of Context (CoC) – das den Ressourcenverbrauch bei der Ermittlung des Kontextes berücksichtigt.

In [VRL+09] gibt es Vorschläge, wie QoC in einem System für Aktivitätserkennung modelliert werden kann. Es werden drei Vorgehensweisen unterschieden: empirische Offline-Modellierung, analytische Modellierung und empirische Online-Modellierung.

- **Empirische Offline-Modellierung**: Bei diesem Ansatz wird der Zusammenhang zwischen den Inputparametern des Erkennungssystems und QoC des Outputs (hier die Aktivitätsklasse) empirisch während der Trainings- und Testphase des Systems festgestellt. Die Parameter des Erkennungssystems bestimmen die Qualität der Erkennung. Solche Parameter sind z. B. die Genauigkeit und Zuverlässigkeit der Sensoren, die Abtastfrequenz bei der Datenerhebung, die verwendeten Merkmale und das Klassifizierungsverfahren. Bei dieser Modellierung werden also die Evaluationsmetriken des Systems ermittelt und in QoC übernommen. Nachteil dieses Vorgehens ist, dass QoC nur eine statische Information (vor allem aus Durchschnittswerten) über die Qualität darstellt [VRL+09].
- **Analytische Modellierung**: Bei der analytischen Modellierung wird eine mathematische Funktion verwendet, die auf Basis des QoC des Inputs bzw. der Inputparameter des Erkennungssystems das QoC des Outputs bestimmt. Eine solche Funktion ist für jeden Schritt des Erkennungsprozesses definierbar [VRL+09].
- **Empirische Online-Modellierung**: Die Hauptidee dieses Verfahrens ist, dass das Erkennungssystem annotierte Referenzsensordaten für jede Aktivitätsklasse speichert. Wenn ein Parameter des Systems geändert wird, werden die gespeicherten Referenzinstanzen als Input für das System benutzt. Das QoC wird festgestellt, indem die erkannte Aktivität aus diesem Input mit der realen Aktivität gegenübergestellt wird. Die reale Aktivität wird in diesem Fall durch die entsprechende Referenzinstanz definiert. Vorteil dieses Modellierungsvorgehens ist die Möglichkeit, Systemparameter dynamisch zu ändern, z. B. durch Hinzufügen/Wegnahme von Merkmalen oder zusätzlichen Klassifikatoren [VRL+09]. Ein Nachteil ist der zusätzliche Bedarf an Speicherkapazität [VRL+09].

3.3 Implementierung von QoC

In diesem Abschnitt wird ein Erkennungssystem vorgestellt, das QoC nach ähnlichem Ansatz der in [VRL+09] vorgeschlagenen analytischen Modellierung implementiert. Das System ist schematisch in Abbildung 3 dargestellt und

wurde im Rahmen des Projektes *AwareOffice* entwickelt. Das Erkennungssystem wird von einem kontextgewahren Stift, der die Aktivitäten Schreiben, Spielen, Liegen erkennen kann, benutzt. Das Ziel der Anwendung von QoC in diesem Fall ist die Verbesserung der Qualität von Entscheidungen, die von einem übergeordneten System getroffen werden und auf dem Ergebnis dieses Erkennungssystems basieren. (vgl. [BDR+07])

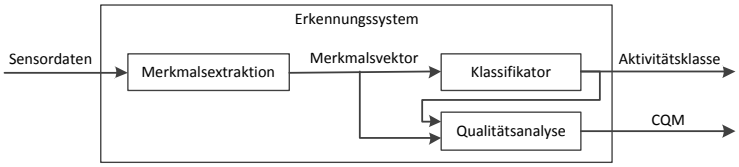


Abbildung 3. Erkennungssystem mit QoC. (vgl. [BDR+07])

Das Modul *Qualitätsanalyse* ist für die Analyse der Qualität bei der Klassifizierung zuständig. Es verwendet als Eingabe denselben Merkmalsvektor, den der Klassifikator klassifiziert, und zusätzlich dazu die erkannte Klasse für diesen Merkmalsvektor. Das Modul stellt ein FIS (Fuzzy Inference System) dar und liefert als Ergebnis einen kontinuierlichen reellen Wert zwischen 0 (falsche Erkennung) und 1 (korrekte Erkennung). Dieser Wert stellt eine Qualitätsmetrik über den Kontext (*Context Quality Measure, CQM*) dar.
Das Modul *Qualitätsanalyse* speichert anhand der Fuzzy-Logik Vorkenntnisse darüber, ob eine vorherige Klassifizierung korrekt oder falsch war. Die Erfassung dieser Information verlangt wieder eine Trainingsphase, wie die Trainingsphase für den Klassifikator. Danach erfolgt auch eine Testphase, wobei eine statistische Analyse durchgeführt wird, um die Wahrscheinlichkeiten für die Trennung der falschen und korrekten Klassifizierungen zu bestimmen. Es wird ein Schwellenwert (S) für die CQM-Metrik ermittelt, wonach entschieden wird, ob eine Klassifizierung falsch oder korrekt ist. Das System wird so optimiert, dass die Wahrscheinlichkeit, dass CQM größer als S ist, wenn die Klassifizierung korrekt ist, maximiert wird (analog für CQM kleiner als S bei falscher Klassifizierung) und die Wahrscheinlichkeit, dass bei korrekter Klassifizierung CQM kleiner als S ist, minimiert wird (analog für CQM größer als S bei falscher Klassifizierung). (vgl. [BDR+07])

Durch Anwendung dieses Erkennungssystems konnten bis zu 33 % der falschen Erkennungen gefiltert werden. Somit konnten dem übergeordneten System mit hoher Wahrscheinlichkeit nur korrekte Kontextinformationen zur Verfügung gestellt werden. Vorteil dieses Erkennungssystems ist die Unabhängigkeit der Qualitätsanalyse vom Algorithmus für die Klassifizierung. Ein Nachteil ist, dass bei einem großen Trainingsdatensatz die Wahrscheinlichkeiten für die Trennung der korrekten und falschen Klassifizierungen schlechter sein

können. Die Wahl des Trainingssatzes ist also kritisch für die Leistungsfähigkeit des Moduls für Qualitätsanalyse. (vgl. [BDR+07])

4 Erkennung von Anomalien

In diesem Kapitel wird der Ansatz der Anomalieerkennung erläutert, wobei unterschiedliche Techniken vorgestellt werden. Es werden auch zwei Beispiele dargestellt, die einen Teil dieser Techniken veranschaulichen.

4.1 Grundlegende Definitionen

Anomalien sind Muster in einer Datensammlung, die nicht zu wohl definierten Mustern für Normalhandlungen konform sind [CBK09]. In Bezug auf die Aktivitätserkennung sind die Anomalien abnormale Aktivitäten bzw. Ereignisse, die selten und grundsätzlich unerwartet auftreten [YYP08]. In vielen Anwendungsbereichen der Aktivitätserkennung, wie Sicherheitsüberwachung oder Gesundheitsversorgung für ältere Menschen, stellen die Anomalien eine kritische Situation dar, deren Identifizierung hochwertige Information liefert.

Grundsätzlich ist die genaue Definition von Anomalien domänenspezifisch und häufig auch kontextspezifisch. Zum Beispiel wird im Bereich der Medizin eine kleine Abweichung der Körpertemperatur eines Patienten als Anomalie gekennzeichnet, im Bereich des Börsenhandels aber wird eine schwache Fluktuation der Aktienpreise in der Regel als normal wahrgenommen. (vgl. [CBK09])

Ein anderes Merkmal der Anomalieerkennung ist die Tatsache, dass in vielen Anwendungsbereichen die normalen Aktivitäten sich ständig ändern und weiterentwickeln. Eine Umwandlung der selten auftretenden abnormalen Aktivitäten in normale Aktivitäten ist nicht ausgeschlossen. Dies kann die Trennung der abnormalen von normalen Aktivitäten erschweren. (vgl. [CBK09])

Anomalieerkennung hat auch einen Bezug zu Rauschunterdrückung aus dem Gesamtdatensatz. Häufig enthält der Datensatz Datenpunkte, Rauschen genannt, die für die weitere Analyse nicht von Interesse sind und in der Regel ausgefiltert werden. Solche Datenpunkte sind z. B. die Ausreißer. Es ist nicht ausgeschlossen, dass das Rauschen und die Anomalien ähnliche Eigenschaften haben, was deren Trennung bzw. die Rauschunterdrückung verhindert. (vgl. [CBK09])

Auch ein wichtiger Aspekt der Erkennung von Anomalien ist das Problem, dass es häufig keine annotierten Dateninstanzen im Gesamtdatensatz gibt, die die abnormalen Aktivitäten repräsentieren. Sogar wenn eine Erhebung von solchen Instanzen möglich ist, ist sie in der Regel kostenintensiver und aufwendiger als die Datenerhebung für normale Aktivitäten. Diese Tatsache erschwert den Entwurf und insbesondere die Validierung eines Systems zur Anomalieerkennung. (vgl. [CBK09])

All diese Aspekte der Erkennung von Anomalien führen dazu, dass keine generelle Lösung des Problems existiert, sondern eher eine domänenspezifische Lösung [CBK09].

4.2 Vorgehensweise bei der Anomalieerkennung

Abhängig davon, inwieweit Dateninstanzen für abnormale Aktivitäten im Gesamtdatensatz vorhanden sind, gibt es grundsätzlich drei Vorgehensweisen für die Realisierung eines Systems zur Anomalieerkennung durch Anwendung von Algorithmen für überwachte Anomalieerkennung, semiüberwachte Anomalieerkennung oder nicht überwachte Anomalieerkennung [CBK09].

- **Überwachte Anomalieerkennung**: Bei dieser Vorgehensweise wird ein Modell für normale und abnormale Aktivitätsklassen erstellt. Dies verlangt verfügbare annotierte Dateninstanzen sowohl für normale als auch für abnormale Aktivitäten. Nachteile dieses Verfahrens sind die relativ kleine Anzahl von Instanzen für Anomalien im Trainingsdatensatz im Vergleich zu den Instanzen für normale Aktivitäten und die relativ kostenintensive Erhebung von Dateninstanzen für abnormale Aktivitäten [CBK09].
  Im Rahmen dieser Vorgehensweise können die gewöhnlichen Algorithmen aus der Aktivitätserkennung eingesetzt werden: SVMs, Bayes'sche Netze, Entscheidungsbäume, HMMs und andere. Dieser Ansatz bietet die Möglichkeit, eine hohe Erkennungsrate und eine niedrige False-Alarme-Rate für normale bzw. abnormale Aktivitäten zu erreichen, wenn ausreichend Dateninstanzen für Anomalien im Trainingsdatensatz zur Verfügung stehen.
- **Semiüberwachte Anomalieerkennung**: Im Vergleich zum ersten Verfahren wird hier angenommen, dass der Trainingsdatensatz nur annotierte Instanzen für normale Aktivitäten enthält.
  In diesem Fall werden häufig die sog. One-Class-Klassifikatoren verwendet, die nur zwischen zwei Klassen unterscheiden können: die Klasse der normalen Aktivitäten und die Klasse der Anomalien. Die Idee dieser Technik ist, dass ein Modell für die Klasse der normalen Aktivitäten erstellt und alles außerhalb dieser Klasse dann als Anomalie kennzeichnet wird. Für die Realisierung werden Algorithmen bzw. Modelle wie SVM, HMM, neuronale Netze und Entscheidungsbäume verwendet [CBK09]. Diese Methode ist intuitiv und lässt sich einfach realisieren. Das Erkennungssystem hat in der Regel eine hohe Erkennungsrate für die normalen Aktivitäten, aber gleichzeitig auch eine relativ hohe False-Alarm-Rate bezüglich der Anomalien, d. h. es werden häufig normale Aktivitäten als Anomalien erkannt [YYP08,SLP11].
  Eine andere Möglichkeit ist die Verwendung von Klassifikatoren, die alle normalen Aktivitäten modellieren, wobei zwischen allen einzelnen Aktivitätsklassen unterschieden werden kann. In diesem Fall wird eine Aktivität als Anomalie gekennzeichnet, wenn sie keiner der bekannten Aktivitätsklassen zugeordnet werden kann. Häufig angewendete Algorithmen bzw. Modelle sind: SVMs, HMMs, Entscheidungsbäume, neuronale Netze und Bayes'sche Netze. (vgl. [CBK09])
- **Nicht überwachte Anomalieerkennung**: Diese Vorgehensweise verlangt keinen annotierten Trainingsdatensatz, also kein a-priori-Wissen über die normalen und abnormalen Aktivitäten. So kann der langwierige und teure Prozess der Annotierung des Datensatzes vermieden werden. Es werden

Verfahren für nicht überwachtes Lernen angewendet, die auf der Annahme beruhen, dass die Instanzen der normalen Aktivitäten in dem Testdatensatz grundsätzlich viel häufiger als die Instanzen der abnormalen Aktivitäten auftreten. Trifft diese Annahme nicht zu, dann weist das Erkennungssystem für die Anomalien eine hohe False-Alarm-Rate auf. (vgl. [CBK09])
Eine häufig angewendete Technik für nicht überwachte Anomalieerkennung basiert auf dem k-Nearest-Neighbour-Algorithmus (k-NN). Es sind grundsätzlich zwei Techniken zu unterscheiden. Die erste Technik verwendet Distanzmetrik, definiert für zwei Dateninstanzen bzw. Merkmalsvektoren. Um zu bestimmen, ob eine Dateninstanz eine Anomalie ist, wird die Distanz zu den k nahe liegenden Dateninstanzen ermittelt. Dabei wird angenommen, dass die Instanzen der Anomalien deutlich entfernt von ihren Nachbarn liegen. In der Regel wird ein Schwellenwert für die berechnete Distanz benutzt, um diese Entscheidung zu treffen. Die zweite Technik verwendet Verdichtungsverhältnis mit der Annahme, dass die Dateninstanzen von Anomalien in einer Umgebung mit niedriger Dichtheit liegen, während die Instanzen der normalen Aktivitäten in einer Nachbarschaft mit hoher Dichtheit liegen. Diese Techniken zeigen eine schlechte Erkennungsleistung, falls die Instanzen der normalen Aktivitäten nicht eine ausreichend hohe Anzahl von Nachbarn haben bzw. die Instanzen der Anomalien eine zu hohe Anzahl von Nachbarn haben. (vgl. [CBK09])
Eine andere Möglichkeit für Anomalieerkennung stellt der Einsatz von Clustering Verfahren dar, wobei ähnliche Dateninstanzen zusammen gruppiert werden. Es sind drei Typen von Clustering-Verfahren nach den Annahmen, die sie machen, zu unterscheiden. Der erste Typ macht die Annahme, dass die Dateninstanzen bzw. die Merkmalsvektoren der normalen Aktivitäten zu einem Cluster gehören, während die Instanzen von Anomalien zu keinem Cluster gehören. Der zweite Typ nimmt an, dass die normalen Instanzen nah an dem nächsten Cluster-Schwerpunkt (*cluster centroid*) liegen, während die Instanzen der Anomalien weit entfernt von ihrem nächsten Cluster-Schwerpunkt liegen. Nachteil der ersten beiden Verfahren ist, dass die Anomalien, die selbst Cluster bilden, nicht als solche erkannt werden. Das dritte Verfahren überwindet diesen Nachteil mit der Annahme, dass die Instanzen der normalen Aktivitäten zu großen und dichten Clustern gehören, während die Instanzen der Anomalien zu relativ kleinen Clustern gehören, deren Punkte nicht dicht nebeneinander liegen. Für die Cluster-Größe und die Cluster-Dichte werden entsprechend Schwellenwerte definiert. Die Clustering Verfahren sind effektiv, soweit die Instanzen der Anomalien keine großen Cluster unter sich bilden. (vgl. [CBK09])
Sowohl k-NN-Verfahren als auch Clustering-Verfahren können auch für die Realisierung von semiüberwachter Anomalieerkennung verwendet werden.

Unabhängig davon, welche Methode für die Implementierung eines Systems zur Anomalieerkennung verwendet wird, werden für die Testphase immer Dateninstanzen für Anomalien benötigt, sonst wäre die Evaluation des Systems nicht vollständig.

4.3 Systeme zur Anomalieerkennung

Im Folgenden werden zwei Systeme zur Erkennung von Anomalien jeweils mit ihren Vor- und Nachteilen vorgestellt.

Das erste System ist von Shin et al. [SLP11] entworfen worden und überwacht das Verhalten von älteren Menschen mit dem Ziel, ihnen ein unabhängiges Leben zu ermöglichen und sie zu unterstützen. Das System soll Anomalien erkennen und diese an eine Zentrale, die für die Pflege der älteren Personen verantwortlich ist, übermitteln. Die Anomalien in diesem Anwendungsfall sind Unfälle, wie z. B. Stürze, Schwäche, Epilepsieanfälle etc.

Die Daten über die Aktivitäten der Menschen werden anhand IR-Bewegungssensoren, die im Haus der entsprechenden beobachteten Person eingebaut sind, gesammelt. Um das Verhalten der Personen anhand der Sensordaten zu erfassen, werden drei Merkmale berechnet: Aktivitätsniveau, Mobilitätsniveau und Non-Response-Intervall. Das System basiert auf der Vorgehensweise der semiüberwachten Anomalieerkennung, in dem für die Erkennung ein One-Class-Klassifikator eingesetzt wird. Es wird der Support Vektor Data Description-Algorithmus (SVDD) verwendet. Eine schematische Beschreibung des Systems ist in Abbildung 4 dargestellt. Die Idee des SVDD-Algorithmus ist, eine möglichst kleine Sphäre zu finden, die alle Instanzen der verfügbaren normalen Aktivitäten umfasst. So wird die Wahrscheinlichkeit minimiert, dass eine Anomalie als normales Verhalten erkannt wird, was in diesem Anwendungsfall kritisch wäre. (vgl. [SLP11])

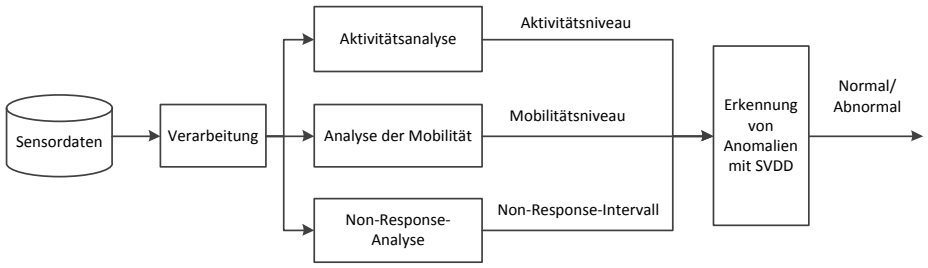


Abbildung 4. System zur Erkennung von Anomalien durch Anwendung des One-Class-SVDD-Klassifikators. (vgl. [SLP11])

Der Datensatz enthielt sehr wenige Instanzen für Anomalien, um das System richtig validieren zu können. Deswegen wurden zwei unterschiedliche Validierungen durchgeführt, einmal mit realen Daten und einmal mit künstlich generierten Daten. Das System erreicht während der Testphase der ersten Validierung eine Präzision (*true positive rate*) bezüglich der normalen Aktivitäten von 90,5 %, wobei die Sensitivität 74,2 % und die Spezifizität 85,8 % betragen. Bei der zweiten Validierung betragen die Metriken entsprechend 95,8 %, 86,0 % und 85,5 %.

Die niedrige Sensitivität lässt sich durch Anpassung der Parameter des SVDD-Klassifikators erhöhen, was aber in eine Reduktion der Spezifizität resultiert. (vgl. [SLP11])

Das System wurde auch für vier Monate bei acht Testpersonen unter reellen Bedingungen eingesetzt. Es wurden insgesamt 71 Anomalien gemeldet, von denen 58 FNs (falsch als Anomalien erkannt) und 13 TNs (korrekt als Anomalien erkannt) waren. Das System weist also eine hohe False-Negative-Rate (*false alarm*) auf. Die hohe Anzahl von FNs ist einerseits durch Sensorausfall und ungewöhnliches Verhalten der Testpersonen (aber keine Anomalien) und andererseits durch die Eigenschaften des SVDD-Algorithmus begründet [SLP11].

Das zweite System ist von Yin et al. [YYP08] entworfen worden und identifiziert abnormale Aktivitäten auf Basis von Daten, die durch am Körper getragene Sensoren erhoben werden. Hier handelt es sich wieder um ein System, das Techniken der semiüberwachten Anomalieerkennung mit One-Class-Klassifikator einsetzt. Der Klassifikator verwendet den SVM-Algorithmus. Der Trainingsdatensatz enthält wieder nur Instanzen für normale Aktivitäten wie Sitzen, Gehen, Rennen und Treppensteigen. Alles, was von diesen Aktivitäten nicht erfasst ist, wird als abnormale Aktivität gekennzeichnet wie z. B. Ausrutschen auf dem Boden oder Herunterfallen. Um für die Evaluation des Systems Dateninstanzen für Anomalien zu beschaffen, sollten die Testpersonen, die am Experiment beteiligt waren, künstlich abnormale Aktivitäten simulieren.

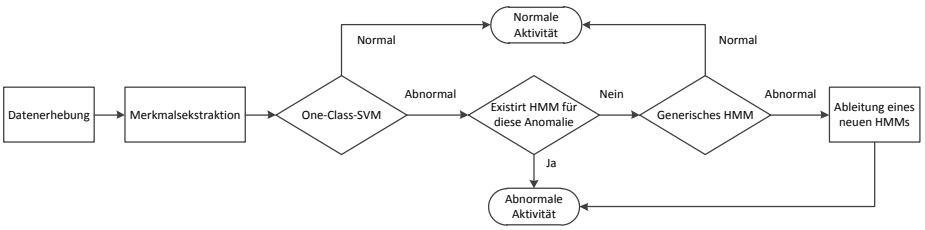


Abbildung 5. Der Prozess der Erkennung von abnormalen Aktivitäten beim zweiten System. (auf Basis der Beschreibung in [YYP08])

Im Unterschied zum ersten System zur Anomalieerkennung verwendet das zweite System mehr als einen Klassifikator. Der Prozess der Erkennung von abnormalen Aktivitäten ist in Abbildung 5 schematisch dargestellt. Der SVM-Klassifikator modelliert die Klasse der normalen Aktivitäten, sodass mit einer hohen Wahrscheinlichkeit eine normale Aktivität als normal erkannt wird, d. h. die TP-Rate (Präzision) ist hoch. Gleichzeitig werden aber auch viele der normalen Aktivitäten als abnormal gekennzeichnet, d. h. der Klassifikator weist eine hohe False-Alarm-Rate wie bei dem ersten System auf. Um dieses Problem zu lösen, werden zusätzliche HMMs, die die abnormalen Aktivitäten modellieren, einge-

setzt. Diese HMMs werden dynamisch aus einem generischen HMM abgeleitet, wenn eine abnormale Aktivität zum ersten Mal auftritt. Das generische HMM modelliert wieder nur die normalen Aktivitäten. Für die Generierung der neuen HMMs werden in [YYP08] zwei Techniken vorgeschlagen: *Kernel Nonlinear Regression* (KNLR) und *Maximum Likelihood Linear Regression* (MLLR).

Abbildung 6 stellt anhand einer ROC-Kurve einen Vergleich von verschiedenen Ansätzen zur Erkennung von Anomalien für das zweite System dar. Wenn nur der SVM-Klassifikator ohne die zusätzlichen HMMs benutzt wird, weist das System eine hohe False-Alarme-Rate von 40 % auf, wenn die Erkennungsrate (hier bezüglich der abnormalen Aktivitäten) bei 90 % liegt. Die Ansätze, bei denen die zusätzlichen HMMs eingesetzt werden, weisen für dieselbe Erkennungsrate eine deutlich niedrigere False-Alarm-Rate auf. Der Ansatz, der die KNLR-Technik verwendet, liefert die besten Ergebnisse. (vgl. [YYP08])

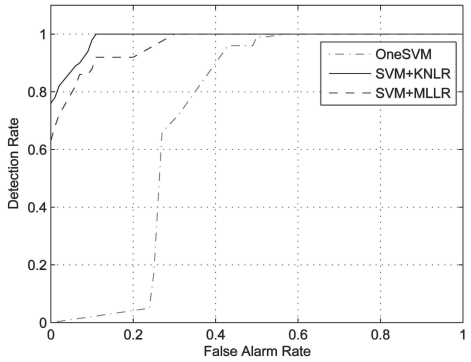


Abbildung 6. Vergleich der Erkennungsrate und False-Alarm-Rate mit den verschiedenen Erkennungstechniken. (vgl. [YYP08])

Im zweiten System wird das Problem der hohen False-Alarm-Rate gelöst, was aber in eine hohe Komplexität resultiert. Ein anderer Nachteil ist der potenziell unbeschränkte Zuwachs an HMMs für die Anomalien, insbesondere falls abnormale Aktivitäten zu normalen Aktivitäten wandern [YYP08].

5 Diskussion

In den Kapiteln 3 und 4 wurden die Ansätze von Quality of Context (QoC) und Anomalieerkennung mit entsprechenden Beispielen für die Realisierung dieser Ansätze in Bezug auf die Aktivitätserkennung vorgestellt. Grundsätzlich haben diese Ansätze unterschiedliche Ziele und verwenden auch unterschiedliche Methoden.

Ziel von QoC ist es, Informationen darüber zu geben, bis zu welchem Grad eine erkannte Aktivität (die Kontextinformation) mit der tatsächlich

eingetretenen Aktivität (die Realität) übereinstimmt. QoC stellt ein Konzept dar, das Kontextparameter definiert, mit denen die Kontextinformation bewertet werden kann, um Informationen mit geringer Qualität herauszufiltern. Es existieren aber keine Standardtechniken, die weit verbreitet eingesetzt werden. QoC definiert selbst auch keine Methoden für die Berechnung der QoC-Parameter und deren Anwendung, also wie QoC zu implementieren ist. Nur [VRL+09] gibt einen Vorschlag, wie die QoC-Parameter mit den Evaluationsmetriken im Rahmen der Aktivitätserkennung in Verbindung gesetzt werden können und wie QoC in der Aktivitätserkennung modelliert werden kann. Von allen definierten QoC-Parametern werden die Parameter *Genauigkeit* und *Wahrscheinlichkeit der Korrektheit* als die bedeutendsten Parameter identifiziert [BKS03,BDR+07,VRL+09].

Der Ansatz der Anomalieerkennung definiert konkrete Techniken zur Erkennung von Anomalien bzw. von abnormalen Aktivitäten: überwachte, semiüberwachte und nicht überwachte Methoden für Anomalieerkennung. Die Methoden werden in unterschiedlichen Anwendungsgebieten erfolgreich eingesetzt wie z. B. Bildverarbeitung, Gesundheitswesen, Textverarbeitung, Betrugserkennung bei Kreditkarten (*fraud detection*) etc. [CBK09]. Die Wahl des konkreten Verfahrens ist grundsätzlich davon anhängig, ob im Datensatz Dateninstanzen für die Anomalien zur Verfügung stehen. Die am meisten verbreiteten Techniken sind der Einsatz von One-Class-Klassifikatoren wie One-Class-SVM und die Anwendung von Clustering-Verfahren. Nachteil der Systeme, die auf diesen Techniken basieren, ist häufig die relativ hohe False-Alarm-Rate bezüglich der Anomalien, was grundsätzlich auf das Problem der fehlenden Instanzen für Anomalien zurückzuführen ist.

Obwohl die beiden Ansätze verschiedene Zielsetzungen haben, lassen sie sich auch miteinander kombinieren. Eine niedrige Qualität bei der Erkennung einer Aktivität kann z. B. als Signal für den Eintritt einer unbekannten Aktivität bzw. einer Anomalie interpretiert werden. So können anhand eines Klassifikators mit CoQ zur Erkennung von normalen Aktivitäten auch Anomalien identifiziert werden. Eine andere Möglichkeit wäre, einen One-Class-Klassifikator, der z. B. SVM anwendet, mit einem System zur Qualitätsanalyse, das in [BDR+07] vorgeschlagen wird (siehe Abschnitt 3.3), miteinander zu kombinieren. Auf diese Weise soll die Leistung des Gesamtsystems bezüglich der Erkennung von Anomalien insgesamt verbessert werden. Die Qualitätsanalyse soll die schlechten Erkennungen des One-Class-Klassifikators identifizieren und somit zur Senkung einer eventuell hohen False-Alarm-Rate beitragen.

Sowohl die Systeme zur Anomalieerkennung als auch die Systeme mit QoC weisen eine relativ hohe Komplexität bei der Implementierung auf, was sich auf die Systeme selbst und die angewendeten Algorithmen bezieht. Bei der Anomalieerkennung ist dies der Fall, wenn ein System mit einer hohen Erkennungsrate und gleichzeitig einer niedrigen False-Alarm-Rate implementiert werden soll (siehe Abschnitt 4.3).

6 Konzept zur Erkennung von normalen und abnormalen Aktivitäten

Im Folgenden wird ein Konzept für ein System zur Erkennung von sowohl normalen als auch abnormalen Aktivitäten vorgeschlagen. Das System stellt ein Ensemble dar und kombiniert zwei verschiedene Klassifikatoren: einen Klassifikator für Aktivitätserkennung und einen Klassifikator für Anomalieerkennung. Eine schematische Beschreibung des Systems ist in Abbildung 7 dargestellt.

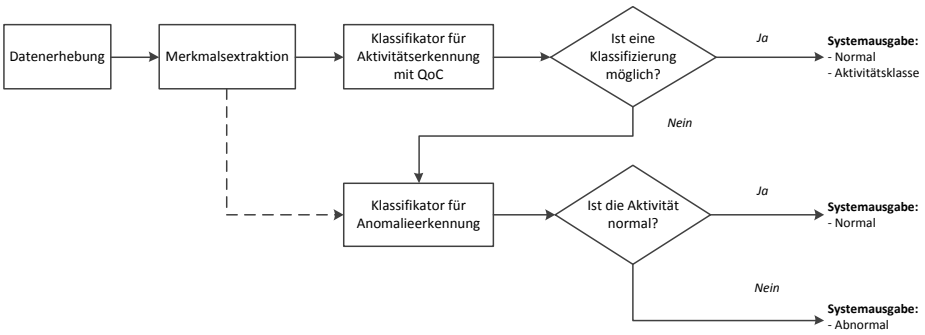


Abbildung 7. Kombiniertes System zur Erkennung von normalen und abnormalen Aktivitäten.

Ein Ensemble aus Klassifikatoren ist ein System, bei dem der Output der Klassifikatoren anhand verschiedener Techniken fusioniert wird. Das Ziel dabei ist es, die Leistung bezüglich der Erkennung insgesamt zu erhöhen. Die kombinierten Klassifikatoren müssen nicht vom gleichen Typ sein. Sie können sogar verschiedene Merkmale nutzen und mit unterschiedlichem Datensatz trainiert werden. Dies verstärkt die Divergenz zwischen den verwendeten Klassifikatoren, was auch der Grund für den Einsatz von Ensemblesystemen ist. Die unterschiedlichen Klassifikatoren haben in der Regel auch unterschiedliche Eigenschaften und somit auch eine unterschiedliche Erkennungsleistung. Daher machen sie verschiedene Fehler bei der Klassifizierung. Durch die Ensemblesysteme und eine geeignete Fusionierung lassen sich diese Unterschiede positiv ausnutzen, um die Nachteile der verwendeten Klassifikatoren zu kompensieren. (vgl. [Pol06])

Beim Entwurf des kombinierten Systems wird angenommen, dass keine Instanzen für Anomalien zur Verfügung stehen. Das System beruht auf dem Ansatz, dass die Aktivitäten, die nicht als eine normale Aktivität klassifiziert werden können, als eine Anomalie gekennzeichnet werden. Die beiden Klassifikatoren nutzen denselben Merkmalsvektor als Input bei der Klassifizierung. Der Klassifikator für Anomalieerkennung wird nur dann eingesetzt, wenn der Klassifikator für Aktivitätserkennung die aktuelle Aktivität zu keiner Aktivitätsklasse

mit einer ausreichend hohen Sicherheit zuordnen kann. Die Kombination der Klassifikatoren erlaubt die Vermutung, dass das System eine nicht so hohe False-Alarm-Rate aufweist wie die Systeme zur Anomalieerkennung, die nur auf einem One-Class-Klassifikator basieren. Im Folgenden werden die einzelnen Module des Systems aus Abbildung 7 beschrieben.

- **Datenerhebung**: Dieses Modul ist für die Erhebung der Daten aus den einzelnen Sensoren zuständig.
- **Merkmalsextraktion**: In diesem Modul werden die Merkmale aus den Sensordaten extrahiert und geeignet in Merkmalsvektoren zusammengefasst.
- **Klassifikator für Aktivitätserkennung**: Dieses Modul implementiert den Klassifikator, der die normalen Aktivitäten den entsprechenden Aktivitätsklassen zuordnen soll.
  Für die Klassifizierung wird ein stochastischer Klassifikator verwendet, z. B. Hidden Markov Model (HMM). So bekommt man als Ergebnis Wahrscheinlichkeiten, mit denen die aktuelle Aktivität den jeweiligen Aktivitätsklassen zugeordnet werden kann [Sch10, Kapitel 5]. Die Aktivität wird der Klasse mit der größten Wahrscheinlichkeit zugeordnet. Diese Wahrscheinlichkeit kann als einen QoC-Parameter interpretiert werden wie z. B. den Parameter *Wahrscheinlichkeit der Korrektheit*. Liegt die Wahrscheinlichkeit unter einem vordefinierten Schwellenwert, kann angenommen werden, dass eine ausreichend sichere Klassifizierung nicht möglich ist. In diesem Fall wird angenommen, dass es sich eventuell um eine unbekannte Aktivität handelt, also eine Anomalie. Diese Vermutung wird durch den Klassifikator für Anomalieerkennung überprüft.
- **Klassifikator für Anomalieerkennung**: Das Modul implementiert einen One-Class-Klassifikator, z. B. einen One-Class-SVM. Es liefert als Ergebnis ein Kennzeichen, ob die aktuelle Aktivität normal oder abnormal ist. Der Klassifikator ist so zu optimieren, dass die Aktivitäten, die als normal klassifiziert werden, mit hoher Wahrscheinlichkeit tatsächlich normal sind.

Vorteil des Systems ist die Möglichkeit, die Klassifikatoren mit verschiedenen Trainingsdatensätzen zu trainieren. In Bezug auf den Klassifikator für Aktivitätserkennung soll dies eine gute Erkennung der gewünschten Aktivitäten ermöglichen. Hier kann z. B. die NULL-Klasse nicht modelliert werden. Der Klassifikator für Anomalieerkennung kann mit allen verfügbaren normalen Dateninstanzen trainiert werden oder nur mit den Instanzen, die für das Training des ersten Klassifikators nicht benutzt wurden. Dies können z. B. nur die Instanzen der NULL-Klasse sein. Ein anderer Vorteil ist die Möglichkeit der Energieeinsparung, falls das System für eine eingebettete Plattform zu implementieren ist. Der Grund dafür ist die Tatsache, dass der Klassifikator für Anomalieerkennung nicht bei jeder Klassifizierung eingesetzt wird.
Ein Nachteil des Systems ist die erschwerte Einschätzung der Systemparameter, wenn keine Dateninstanzen für Anomalien zur Verfügung stehen. Ein solcher Parameter ist z. B. der Schwellenwert für die Abgrenzung der normalen und abnormalen Aktivitäten bei dem Klassifikator für Aktivitätserkennung.

7 Schlussfolgerung

Im Rahmen dieser Seminararbeit wurden die Ansätze von Quality of Context (QoC) und Anomalieerkennung in Bezug auf die Aktivitätserkennung erläutert. Es wurden Beispiele vorgestellt, wie diese Ansätze in der Praxis realisiert werden können. Auf Basis dieser Beispiele wurden die Hauptprobleme bei der Implementierung und dem Einsatz der Ansätze identifiziert und diskutiert.

QoC stellt ein Konzept dar, das verschiedene QoC-Parameter definiert, aber keine Aussage dazu macht, wie diese Parameter bezüglich eines Erkennungssystems berechnet werden können. In [VRL+09] ist ein Vorschlag gegeben, wie diese Parameter mit den Evaluationsmetriken der Aktivitätserkennung in Verbindung gesetzt werden können. Das von Berchtold et al. [BDR+07] entworfene System stellt auch ein Beispiel dafür dar, wie ein QoC-Parameter ermittelt werden kann. Diese Ansätze stellen grundsätzlich keinen Standard dar, der weit verbreitet eingesetzt wird.

Das Problem der Anomalieerkennung wird in der Regel domänenspezifisch definiert, so wie auch die entsprechende Lösung. Häufig stehen keine Dateninstanzen für Anomalien zur Verfügung. Die Erhebung solcher Instanzen ist grundsätzlich teuer und aufwendig. Nach der Verfügbarkeit von abnormalen Instanzen richtet sich die Auswahl der Vorgehensweise bei dem Entwurf von Systemen zur Anomalieerkennung. Am häufigsten werden die Ansätze für semiüberwachte und nicht überwachte Anomalieerkennung angewendet. Wegen der fehlenden Instanzen für Anomalien weisen in der Regel die Systeme, die diese Techniken verwenden, eine relativ hohe False-Alarm-Rate auf.

Es wurde ein kombiniertes System vorgeschlagen, das sowohl normale als auch abnormale Aktivitäten identifizieren soll. Für die Erkennung von Anomalien wird der folgende Ansatz angewendet: Alles, was nicht als eine normale Aktivität klassifiziert werden kann, wird als eine abnormale Aktivität gekennzeichnet. Durch die Kombination der beiden verschiedenen Klassifikatoren wird gleichzeitig eine hohe Erkennungsrate und eine niedrige False-Alarm-Rate bezüglich der Anomalien angestrebt.

Literatur

[BDR+07] M. Berchtold, C. Decker, T. Riedel, T. Zimmer, and M. Beigl. Using a context quality measure for improving smart appliances. In *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on*, pages 52–52. IEEE, 2007.

[BI04] L. Bao and S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive Computing*, pages 1–17, 2004.

[BKS03] T. Buchholz, A. Küpper, and M. Schiffers. Quality of context: What it is and why we need it. In *Proceedings of the workshop of the HP OpenView University Association*, pages 1–13, 2003.

[BKV+97] C. V.C Bouten, K. T.M Koekkoek, M. Verduin, R. Kodde, and J. D Janssen. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Transactions on Biomedical Engineering*, 44(3):136–147, March 1997.

[CBK09] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[CK11] L. Chen and I. Khalil. Activity recognition: Approaches, practices and trends. *Activity Recognition in Pervasive Intelligent Environment*, pages 1–31, 2011.

[DAS01] A. Dey, G. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of Context-Aware applications. *Human-Computer Interaction*, 16:97–166, December 2001.

[KAE11] T.L.M. Kasteren, H. Alemdar, and C. Ersoy. Effective performance metrics for evaluating activity recognition methods. *Proceedings-ARCS 2011*, 2011.

[Pol06] R. Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45, 2006.

[Sch10] J. Schenk. *Mensch-Maschine-Kommunikation : Grundlagen von sprach- und bildbasierten Benutzerschnittstellen*. SpringerLink : Bücher. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010.

[SLP11] J. H. Shin, B. Lee, and K. S. Park. Detection of abnormal living patterns for elderly living alone using support vector data description. *IEEE Transactions on Information Technology in Biomedicine*, 15(3):438–448, May 2011.

[VRL+09] C. Villalonga, D. Roggen, C. Lombriser, P. Zappi, and G. Tröster. Bringing quality of context into wearable human activity recognition systems. In K. Rothermel, D. Fritsch, W. Blochinger, and F. Dürr, editors, *Quality of Context*, volume 5786, pages 164–173. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[Wei91] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[YYP08] J. Yin, Q. Yang, and J. J Pan. Sensor-Based abnormal Human-Activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090, August 2008.