Hochschule Reutlingen
Reutlingen University

INF

Studiengang
Human-Centered
Computing

Uwe Kloos, Natividad Martínez, Gabriela Tullius (Hrsg.)

Wissenschaftliche Vertiefungskonferenz
HuC - Digital Future

Informatik-Konferenz an der Hochschule Reutlingen

16. November 2016

ISBN 978-3-00-054956-4

Impressum

Anschrift:
Hochschule Reutlingen
Reutlingen University
Fakultät Informatik
Human-Centered Computing
Alteburgstraße 150
D-72762 Reutlingen

Telefon: +49 7121 / 271-4002
Telefax: +49 7121 / 271-4042

E-Mail: wvk@reutlingen-university.de
Webseite: wvk.reutlingen-university.de

Organisationskomitee:
Prof. Dr. Uwe Kloos, Hochschule Reutlingen
Prof. Dr. Natividad Martínez, Hochschule Reutlingen
Prof. Dr. Gabriela Tullius, Hochschule Reutlingen

Oliver Bertram
Vadim Demkiv
Charalambos Evgenidis
Saskia Heine
Gregor Renka
Damir Stažić
Eva Witzel

Hochschule Reutlingen
Reutlingen University

Copyright: Hochschule Reutlingen, Reutlingen 2016
Herstellung und Verlag: Hochschule Reutlingen
ISBN 978-3-00-054956-4

Vorwort

Technik und Computer nehmen bereits heute eine dominante Rolle in unserem Leben ein und dieser Trend wird sich in den kommenden Jahren weiter verstärken. Dabei übernehmen Computer in Interaktion mit dem Menschen immer komplexere Aufgaben. Im Rahmen dieser Entwicklung hat sich in den letzten Jahren kontinuierlich die Rollenverteilung von Mensch und Maschine gewandelt. So ist die Maschine nicht länger nur ein Werkzeug, das lediglich vom Benutzer definierte Aufgaben abarbeitet. Vielmehr übernehmen Computer zunehmend Verantwortung für Teilbereiche unseres Lebens, z.B. in den Bereichen der Heimautomatisierung, der autonomen Fahrzeuge oder der computer-assistierten Medizin. Neben den vielfältigen technischen Herausforderungen und den angestrebten Verbesserungen unserer Lebensqualität gilt es immer auch mögliche Implikationen im Blick zu behalten und abzumildern, die etwa die körperliche Unversehrtheit, die Privatsphäre oder die Entscheidungsfreiheit des Menschen beeinträchtigen könnten. Dieses Thema des Zusammenspiels zwischen Mensch und Maschine ist das grundlegende Thema des Studiengangs Human-Centered Computing (huc) an der Fakultät Informatik der Hochschule Reutlingen.

Bereits zum 12. Mal findet in diesem Jahr am 16. November die Wissenschaftliche Vertiefungskonferenz statt. Im Rahmen der zugehörigen Lehrveranstaltung „Wissenschaftliche Vertiefung“ haben Studierende die Gelegenheit, sich in einem selbst gewählten Fachbereich zu vertiefen. Diese Vertiefung kann an der Hochschule, in einem Forschungsinstitut, in einem Unternehmen oder auch im Ausland erfolgen. Unter dem Thema „Digital Future“ stellen in diesem Tagungsband sieben Studierende die Ergebnisse ihrer Vertiefungsprojekte im Wintersemester 2016/17 vor.

Ich wünsche Ihnen anregende Einblicke bei der Lektüre der vorgestellten Projekte und freue mich auf eine erfolgreiche Konferenz mit interessanten Gesprächen und Diskussionen.

Warmbronn, 5. November 2016
Prof. Dr.-Ing. Michael Tangemann.

Inhaltsverzeichnis

Oliver Bertram
Reguläre Ausdrücke als Ansatz für die Extraktion domänenspezifischer Konzepte aus ophthalmologischen Befundtexten.....6

Vadim Demkiv
Labeling Tool zum teilautomatisierten überwachten Lernen.....16

Charalambos Evgenidis
Verringerung des Entwicklungs- und Wartungsaufwandes automatischer Web-Oberflächentests....26

Saskia Heine
Einsatzmöglichkeiten von Affective Computing zur Optimierung von Arbeitsumgebungen 36

Gregor Renka
Bildregistrierungsverfahren und ihre Genauigkeit bei Current Imaging-Daten 46

Damir Stažić
Sicherheitsanalyse für Cloud-basierte Produkte der Telemedizin 56

Eva Witzel
Analyse von Tools für einen automatisierten Continuous Delivery Prozess 66

Reguläre Ausdrücke als Ansatz für die Extraktion domänenspezifischer Konzepte aus ophthalmologischen Befundtexten *

Oliver Bertram
Hochschule Reutlingen
Oliver.Bertram@Student.Reutlingen-University.DE

Abstract

Im Rahmen eines Forschungsprojektes wird ein System entwickelt, bei welchem verschiedene Methoden des *Natural Language Processing* und *Information Retrieval* für die inhaltliche Erfassung und spätere Auswertung von ophthalmologischen (augenheilkundlichen) Befundtexten deutscher Sprache eingesetzt werden sollen. Die Problemstellung der Erkennung und Zuordnung von sprachlichen Beschreibungen medizinischer Sachverhalte innerhalb solcher Texte wird betrachtet. Ausgehend von aktuellen Arbeiten werden gängige Lösungsansätze erfasst. Eine auf Regulären Ausdrücken basierende, projektspezifische Lösung wird im Detail vorgestellt.

Schlüsselwörter

Natural Language Processing, Information Extraction, Named Entity Recognition, Disorder Mention Detection, Disorder Concept Identification

*

Betreuer Hochschule: Prof. Dr. Christian Thies
Hochschule Reutlingen
Christian.Thies@reutlingen-university.de

Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Oliver Bertram

CR-Kategorien

H.3.0 [**Information Search and Retrieval**]: General

1 Einleitung

Im klinischen Umfeld werden traditionell viele Patientendaten in unstrukturierten oder halbstrukturierten Texten festgehalten, wie z.B. Arztbriefen. Solche Texte enthalten unter Anderem Informationen über den aktuellen Zustand des Patienten, den Krankheitsverlauf und verordnete Therapien. Mithilfe von Methoden aus dem Gebiet der *Information Extraction* (IE) wird versucht, diese unstrukturierten Informationen in eine strukturierte Form zu übertragen. Ziel ist es, die Daten maschinell verarbeitbar, und damit für Anwendungen aus verschiedenen Bereichen nutzbar zu machen: z.B. in der elektronischen Datenverarbeitung, als Diagnoseunterstützung für behandelnde Ärzte oder für die Auswertung größerer Datenmengen in der medizinischen Forschung.

Die vorliegende Arbeit ist Teil eines kooperativen Projekts mit der Augenklinik Tübingen. Dabei soll ein System für die inhaltliche Erfassung und Verarbeitung von augenärztlichen Befundtexten in deutscher Sprache entwickelt werden. Ein langfristiges Ziel dieses Systems ist es, durch Methoden aus dem Bereich des *Information Retrieval*, automatische Diagnosevorschläge anhand ähnlicher Befundtexte in einer Datenbank liefern zu können. Damit können Nutzer des

Systems, z.B. Augenärzte, Unterstützung bei der Diagnosestellung erhalten. Aktuell wird das System vorbereitend zur Erstellung eines Korpus von annotierten Befunden und dem Aufbau eines Konzeptkatalogs verwendet.
Mit dem Begriff „Konzept“ sind hier Abstraktionen bestimmter medizinischer Entitäten oder Sachverhalte gemeint. Annotieren bezeichnet die von Experten durchgeführte, händische Markierung von Textbereichen und deren Verknüpfung mit bestimmten ophthalmologischen Konzepten, welche bei der Diagnosestellung von Relevanz sind. Die verwendeten Konzepte sind sehr spezifisch für die Domäne der Augenheilkunde, weshalb keine bestehenden Klassifikationssysteme oder Ontologien genutzt werden können. Beispielsweise stellen SNOMED CT[1] oder UMLS [2] nicht die hier benötigte Detailtiefe bereit. Teilziel des Gesamtprojekts ist der Aufbau eines solchen Konzeptkatalogs für die Domäne der Augenheilkunde, der alle für die Diagnose relevanten Merkmale eines Befundtextes umfasst, ähnlich wie in [3].
Um beim späteren Information Retrieval neue Befunde automatisch erfassen zu können, muss eine Merkmalsextraktion stattfinden. Analog zum Annotieren durch menschliche Experten, soll dies durch das automatisierte Finden und Zuordnen von Konzepten geschehen.
Die vorliegende Arbeit befasst sich mit Lösungsmöglichkeiten für dieses Problem der Automatisierung, ihrer Einordnung in die Literatur, sowie der Beschreibung eines konkret umgesetzten Ansatzes auf Basis der Verwendung von regulären Ausdrücken. Zunächst wird die Problemstellung formal dargestellt und analysiert. Darauf folgt die Einordnung des Themas in den Bereich des *Natural Language Processing* bzw. der *Information Extraction* und die Benennung relevanter aktueller Arbeiten. Nach der Erläuterung einiger projektspezifischer Besonderheiten, folgt die Vorstellung des praktisch umgesetzten Ansatzes. Die Arbeit schließt mit einer Diskussion des vorgestellten Ansatzes und einem Ausblick auf die nächsten Schritte.

[1] Systematized Nomenclature of Medicine Clinical Terms

[2] Unified Medical Language System

2 Problemformulierung

Das Finden und Zuordnen von Konzeptbeschreibungen in einem Befund, im Weiteren als „Dokument“ bezeichnet, lässt sich als zweistufiges Klassifikationsproblem formulieren. In einem ersten Schritt werden Textbereiche gesucht, welche sprachliche Beschreibungen von bestimmten ophthalmologischen Konzepten darstellen. Zweitens sind die gefundenen Textbereiche ihrem entsprechenden Konzept in einem Konzeptkatalog zuzuordnen, welcher eine Liste aller relevanten Konzepte enthält.
Ein Dokument D ist aus einer Sequenz von n Tokens $w_1, w_2, ..., w_n$ zusammengesetzt. In der ersten Stufe sind bestimmte inhaltlich zusammengehörende Textbereiche zu identifizieren, welche die Beschreibungen medizinischer Sachverhalte darstellen. Diese Textbereiche werden im Folgenden als „Konzeptbeschreibungen“ bezeichnet. Eine Konzeptbeschreibung kann aus einem oder mehreren Wörtern bestehen. Bei Letzterem können die einzelnen Wörter einer Beschreibung direkt aufeinanderfolgend sein, was jedoch nicht zwingend der Fall ist.
Für die Lösung des ersten Teilproblems wird ein zweiwertiger Klassifikator benötigt, der die Wörter eines Dokuments anhand der zwei Zielklassen „Teil einer Konzeptbeschreibung“ und „nicht Teil einer Konzeptbeschreibung“ einteilt. Tokens, die Teil einer Beschreibung sind, müssen darüber hinaus mit einer eindeutigen Beschreibungsnummer versehen werden, um sie im nächsten Schritt als zusammengehörig identifizieren zu können. Ein Dokument D kann danach auch durch die Menge seiner Konzeptbeschreibungen $B_D = \{b_1, b_2, ..., b_m\}$ beschrieben werden. Eine andere Sichtweise auf dieses erste Teilproblem ist es, eine *Segmentierung* der Tokens vorzunehmen, die für die weiteren Schritte von Relevanz sind.
Das zweite Teilproblem besteht darin, die Konzeptbeschreibungen eines Dokumentes

auf die Elemente eines Konzeptkatalogs $K = \{k_1, k_2, ..., k_n\}$ abzubilden, wobei die einzelnen k_i die Konzepte mit ihrer jeweiligen Kennzahl i darstellen. Gesucht wird also eine Abbildung $c : B_D \rightarrow K$.

3 Verwandte Arbeiten

Standard-Fragestellungen aus dem Bereich der maschinellen Sprachverarbeitung, die der hier betrachteten Problematik ähnlich sind, sind *Part-of-Speech-Tagging* (PoS-Tagging) und *Named Entity Recognition* (NER). Innerhalb eines typischen IE Systems werden diese beiden Schritte oft nacheinander durchgeführt [2]. Auch wenn die Zielsetzungen der beiden genannten Bereiche sich von der speziellen Problemstellung dieser Arbeit unterscheiden, sind die jeweiligen Lösungsansätze aufgrund der Ähnlichkeit dennoch von Relevanz. Sowohl PoS-Tagging, als auch NER werden in der Regel als Klassifikations-, bzw. Labelingaufgaben betrachtet. In beiden Fällen ist es das Ziel, einzelne Worte oder Wortgruppen der Texte bestimmten Kategorien bzw. Klassen zuzuordnen.

Besonders relevante Arbeiten sind unter den Begriffen *Concept Mention Detection* und *Disorder Concept Identification* zu finden, welche der hier untersuchte Fragestellung sehr nahe kommen. Beide wurden als Teil einer gemeinsamen Aufgabenstellung im Kontext der IE, der Extraktion und Zuordnung sprachlicher Beschreibungen von medizinischen Konzepten in Texten im Rahmen der *ShARe/CLEF eHealth 2013 Shared Task (1)* [10] formuliert.

3.1 PoS-Tagging

Bei PoS-Tagging wird versucht, die Wortarten der Wörter eines Textes innerhalb ihrer jeweiligen Satzstruktur festzustellen. Dabei sollen für einen gegebenen Satz alle Wörter entsprechend ihrer grammatikalischen Funktion markiert werden[5]. Die Anzahl an verwendeten Tags, nach denen die Wörter klassifiziert werden unterscheidet sich in der Literatur, wobei im NLP-Bereich generell große Anzahlen in den Tagsets der Trainingskorpora genutzt werden[5]. Für indogermanische Sprachen werden beim PoS-Tagging heutzutage Genauigkeiten um 96-97% erreicht[5].

Die Lösungsansätze für PoS-Tagging werden nach *regelbasierten* und *statistischen* Ansätzen unterschieden. Erstere versuchen über das Anwenden von Regelwerken, die typischerweise händisch von Experten zusammengestellt werden, erlaubte Tag-Folgen für untersuchte Sätze herzuleiten. Der Aufbau eines solchen Regelwerks ist enorm aufwändig und bedarf eines besonderen Fachwissens. Regelbasierte Ansätze wurden ab den 90er Jahren durch statistische Ansätze, vor Allem durch Markov-Modelle verdrängt.

3.2 Named-Entity Recognition

In der deutschsprachigen Literatur wird NER oft mit der Erkennung von Eigennamen übersetzt, was jedoch bei vielen Anwendungsbereichen zu kurz greift. Vielmehr werden bei NER in der Regel bestimmte zusammengehörige Informationseinheiten im Text gesucht. Dabei handelt es sich oft um textuelle Repräsentationen von Konzepten wie z.B. „Person“, „Organisation“, „Staat“, „Telefonnummer“ oder „Monat“. Typischerweise werden diese oft als einzelne oder mehrere zusammenhängende Tokens im Text ausgedrückt. So kann z.B. das Konzept <*Staat*> als das Token „Deutschland“ oder das Konzept <*Organisation*> als die Tokens „Hochschule Reutlingen“ ausgedrückt werden.

Welche Konzepte konkret gesucht werden, ergibt sich häufig aus der spezifischen Aufgabenstellung. So sind z.B. bei der Meinungsbildgenerierung für bestimmte Politiker anhand der Auswertung von Postings aus sozialen Medien vor Allem sprachliche Bezeichnungen dieser Politiker von Interesse.

In [6] wird eine Übersicht zur Entwicklung von NER-Ansätzen gegeben, die im Folgenden kurz zusammengefasst ist. Ähnlich dem PoS-Tagging, waren frühere Ansätze oft durch eine händische Zusammenstellung von bestimmten Regeln und deren Anwendung auf Testdaten geprägt. Aktuellere Ansätze versuchen den Einsatz maschi-

neller Lernverfahren, sei es für das Ableiten von Regelwerken oder das Trainieren von Sequenz-Labeling Algorithmen. Beides beruht dabei auf dem Lernen anhand von Trainingsbeispielen und fällt in den Bereich des *überwachten Lernens*, wobei verschiedene Lernverfahren und Algorithmen zum Einsatz kommen. Der Einsatz Überwachter Lernverfahren ist in der Literatur der letzten Jahre der häufigste Fall. Außerdem wurden Versuche unternommen, *halbüberwachtes Lernen* und *unüberwachtes Lernen* für NER umzusetzen. Für beide Ansätze werden nur wenige bzw. keine Trainingsbeispiele benötigt. Halbüberwachte Lernverfahren versuchen aus einer geringen Anzahl Beispiele (engl. *seeds*) selbstständig neue Beispiele in den Daten zu erkennen. Unüberwachte Lernverfahren setzen auf die statistische Analyse oft großer, nicht annotierter Korpora, z.B. durch Clustering-Verfahren. Die genannten Ansätze für Maschinelles Lernen können in der entsprechenden Literatur, z.B. [8] im Detail nachvollzogen werden.

3.3 Bewertung von PoS-Tagging / NER

Der Unterschied von PoS-Tagging zu der speziellen Problemstellung dieser Arbeit ist einerseits, dass dort versucht wird, jedes Wort eines gegebenen Satzes zu klassifizieren, während hier nur bestimmte Textteile von Interesse sind. Damit entfällt bei PoS-Tagging die erste Teilaufgabe der Segmentierung von relevanten Textbereichen aus Abschnitt 2, da eine Klassifizierung für alle Tokens durchgeführt wird. Außerdem werden beim PoS-Tagging die Wörter bezüglich ihrer grammatikalischen Funktion klassifiziert, im Gegensatz zu einer Einteilung nach ihrer Semantik.
NER kommt der hier untersuchten Fragestellung deutlich näher. Wenn die gesuchten Konzeptrepräsentationen nicht nur einzeln stehende Bezeichnungen, sondern auch sprachliche Formulierungen umfassen, ist die Ähnlichkeit zu der hier betrachteten Problematik sehr groß.

3.4 ShARe/CLEF eHealth 2013

Die ursprüngliche Aufgabenbeschreibung [9] der ShARe/CLEF eHealth 2013 Task 1 wurde „Named entity recognition and normalization of disorders“ genannt. Die Aufgabe für Teilnehmer bestand darin, in einem gegebenen Testdatensatz Erwähnungen medizinischer Störungen in klinischen Texten englischer Sprache zu erkennen und anhand UMLS *Concept Unique Identifier* (CUIs) zu klassifizieren. Während der Titel der Aufgabe nach einer typischen NER Aufgabe klingt, wird in der Präzisierung der Aufgabe deutlich, dass die Anforderungen darüber hinaus gehen. So werden sowohl einzeln stehende und zusammenhängende Bezeichnungen für medizinische Konzepte gesucht, als auch Beschreibungen in form „disjunkter“ Erwähnungen. Beispielsweise soll das Konzept mit dem CUI C0344720 sowohl in den Formulierungen „left atrial dilatation“ als auch „The left atrium is moderately dilated“ erkannt werden, wobei bei letzterem die Wörter „left atrium ... dilated“ relevant sind. Den Teilnehmern wurde neben einem Testdatensatz von 100 Dokumenten auch ein annotierter Trainingsdatensatz mit 200 Dokumenten zur Verfügung gestellt. Die von den teilnehmenden Teams entwickelten Methoden sollten anhand der üblichen Evaluationsmetriken *Precision*, *Recall* und *F-Score* (siehe z.B. [8]) ausgewertet werden, wobei die Teilaufgaben der Erkennung korrekter Textbereiche und die nachfolgende Zuordnung zu UMLS CUIs seperat bewertet wurden. Bei der Ergebnisanalyse ist zu beachten, dass die Ergebnisse der ersten Teilaufgabe die der zweiten direkt beeinflussen. Dies entspricht der formalisierten Problemformulierung aus Abschnitt 2. Die erste Teilaufgabe wurde von vielen Teilnehmern in ihren Veröffentlichungen als *Concept Mention Detection* und die zweite als *Disorder Concept Identification* bezeichnet.
In den folgenden Jahren wurden die Aufgaben weiterentwickelt und in weiteren Wettbewerben und dazugehörigen Konferenzen

bearbeitet. Dabei wurden die Anforderungen an die Art und Anzahl von Informationen, die aus den Texten zu extrahieren sind immer zahlreicher und spezifischer.

3.5 Methoden

Die bei der ShARe/CLEF eHealth 2013 eingereichten Arbeiten verwendeten viele unterschiedliche Ansätze, von welchen die meisten dem Ansatz des Überwachten Lernens folgten. Einige Teilnehmer verwendeten Frameworks für Information Retrieval, welche bereits für die englische Sprache vortrainiert sind: METAMAP und cTAKES.
Viele Arbeiten setzten die Kombination mehrerer Ansätze um: ein Framework, die Verwendung von UMLS Termlisten und Maschinelle Lernverfahren. Bei Letzteren wurden vor Allem *Conditional Random Fields* und *Support Vector Machines* verwendet. Auch die Arbeit mit den besten Ergebnissen bezüglich der Evaluationskriterien setzte eine solche Kombination um. Alle der eingereichten Arbeiten nutzten den Trainingskorpus, auch die regelbasierten Systeme. Eine ausführliche Darstellung der Ergebnisse der eingereichten Arbeiten ist in [7] wiedergegeben.

4 Spezielle Problemstellung

Das Projekt, in dessen Umfeld diese Arbeit angesiedelt ist, unterscheidet sich in einigen Aspekten von der grundsätzlich sehr ähnlichen Fragestellung der ShARe/CLEF eHealth 2013 Challenge.
Die Datensätze der Challenge waren klinische Texte in englischer Sprache, was es den Teilnehmern ermöglichte, vorhandene Frameworks wie cTAKES und METAMAP zu verwenden. Diese verwenden bereits vortrainierte Komponenten für (allgemein) medizinische Texte in Englisch. Die vorliegende Arbeit befasst sich mit deutschen Texten, so dass die besagten Frameworks nicht direkt genutzt werden können.
Der häufigste Ansatz bei dem Wettbewerb waren überwachte Lernverfahren, die sich den annotierten Trainingskorpus zu nutze machen. Zum aktuellen Stand liegt für diese Arbeit noch kein annotierter Korpus vor. Obwohl deutschsprachige, annotierte Korpora existieren, passen sie nicht zu der hier relevanten Domäne der Augenheilkunde. Vielmehr ist es ein Teilziel des Projektes, den Aufbau eines Korpus für diese spezifische Domäne in deutscher Sprache durch das Verfügbarmachen verschiedener Tools zu unterstützen.
Neben dem Korpus soll parallel der bereits erwähnte Katalog relevanter ophthalmologischer Konzepte aufgebaut werden, wozu ebenfalls projekteigene Tools genutzt werden sollen. Im Gegensatz dazu konnten die Teilnehmer der ShARe/CLEF eHealth 2013 Challenge auf einen vordefinierten und in diesem Sinne vollständigen Katalog zurückgreifen.
Da das händische Annotieren eines Korpus sehr arbeitsintensiv ist, wurde im Projekt das Ziel verfolgt, die menschlichen Annotatoren möglichst frühzeitig durch Automatisierung zu unterstützen. Dabei kam die Beobachtung zu tragen, dass viele der Annotationen auf sehr einfachen Textmustern basieren, z.B. einzelne Worte oder bestimmte Abkürzungen. Dies entspricht auch dem Sprachstil des Korpus, der aus sehr kurz gehaltenen, stark durch Fachbegriffe geprägter Textbefunde besteht. Solche Textmuster lassen sich leicht in Form Regulärer Ausdrücke modellieren, was die Grundidee für die Entwicklung eines einfachen automatischen Annotationssystem war.
Die Unterstützung durch das Tool soll derart funktionieren, dass mit jedem annotierten Befund, dem Tool bisher unbekannte Textmuster erfasst werden. Während des Annotationsvorgangs soll das Tool auf Basis der bereits annotierten Befunde gewisse Textanteile als mögliche Annotationen vorschlagen. Diese Vorschläge können während des Annotierens durch den Experten, entweder bestätigt oder verworfen werden. Auf diese Weise sollte die Erkennung einfacher und häufiger Annotationen schnell automatisiert werden und dafür sorgen, dass der Experte seine wertvolle Zeit effizienter nutzen kann. Z.B. um ungewöhnliche und seltener vor-

kommende Annotationen zu markieren, oder um einfach mehr Dokumente in der gleichen Zeit abzuarbeiten.

5 RegEx-Annotator

Die grundlegende Idee des Tools, das im Weiteren als „RegEx-Annotator“ bezeichnet wird, ist die Modellierung bestimmter Textmuster mittels regulärer Ausdrücke. Der Fokus liegt dabei auf Mustern bestehend aus einzelnen Wörtern oder einer Abfolge mehrerer Wörter. Es sollen auch solche Muster erfasst werden können, bei denen die eigentlich relevanten Wörter von unrelevanten Textteilen unterbrochen sind.

5.1 Beispiele von Textmustern

Nachfolgend drei Beispiele für die unterschiedlichen Fälle von Textmustern, die durch den Annotator abgedeckt werden:

1. „Drusen in der *Makula*.“
2. „*Makula trocken*.“
3. „*Papille* etwas *enger*.“

Jedes der Beispiele stellt den Inhalt einer korrekten Annotation dar, wie sie durch den Experten bereitgestellt werden könnten. Kursiv dargestellt sind jeweils die Wörter, die Teil der Annotation sind. Im ersten Fall tritt ein einzelnes Wort als Annotation auf, der einfachste Fall. Im zweiten Fall folgen zwei Worte einer Annotation direkt auf einander. Der dritte Fall ist ein Beispiel dafür, dass die Wörter einer Annotation durch unrelevanten Text unterbrochen sind. Die jeweiligen fachspezifischen Konzepte mit denen die Beispiele verknüpft sind könnten folgendermaßen formuliert sein:

1. Konzept 1: „Makula“
2. Konzept 1b: „trockene Makula“
3. Konzept 4c: „enge Papille“

Der exakte Aufbau des Katalogs, der alle benötigten Konzepte und deren Formulierungen umfasst, wird in dieser Arbeit nicht näher betrachtet. Anzumerken ist jedoch, dass in der aktuellen Version die medizinischen Konzepte in einer Baumstruktur hinterlegt sind. Semantisch zusammengehörige Konzepte sind dabei in den selben Zweigen des Baumes eingefügt, was etwa bei den Beispielkonzepten „Konzept 1: Makula“ und „Konzept 1b: trockene Makula“ deutlich wird.

5.2 Gleichartige Muster

Es wurde beobachtet, dass innerhalb der Annotationen gleichartige Muster auftraten. Gleichartig im dem Sinn, dass sie auf das selbe Konzept im Katalog verweisen und die gleichen Wörter markiert sind, die Abstände der Wörter sich jedoch unterscheiden. Um diese als einheitliche Muster zusammenzufassen, wurden die Abstände der Wörter untereinander als dynamische Bereiche mit minimaler und maximaler Länge definiert. Als Beispiel folgende zwei Annotationen:

1. „*Papille* etwas *enger*.“
2. „*Papille* außerdem etwas *enger*.“

In beiden Fällen ist ein gleichartiges Textmuster vorhanden, während die Annotationen auf das selbe Konzept „Papille eng“ verweisen.

5.3 Modellierung durch reguläre Ausdrücke

Aus den drei beschriebenen Fällen wurde ein gemeinsames Muster für reguläre Ausdrucke erarbeitet, das alle jeweiligen Besonderheiten abdeckt. Dazu werden die an einer Annotation beteiligten Wörter, ihre Anzahl, sowie die Abstände der Wörter in einen regulären Ausdruck *R* folgender Form gebracht:

```
R = ( (term1).{min, max}(term2) )
```

Dies stellt den Fall dar, dass zwei Wörter in einer Annotation markiert wurden. Dabei bezeichnet $term_1$ das erste Wort einer Annotation und $term_2$ das zweite. Für Annotationen mit mehr als zwei Wörtern, können diese nach dem gleichen Muster hinzugefügt werden. Die Variablen min, max stellen den Abstand zwischen den Wörtern $term_1$

und $term_2$ in der Annotation dar. Der Abstand wird innerhalb des regulären Ausdruck durch das Erlauben eines beliebigen Zeichens durch das Wildcard-Symbol „ . “ in maximal *max* und minimal *min* Wiederholungen abgebildet. Gleichartige Muster können so in einen einzelnen Ausdruck zusammengefasst werden.

5.4 Zahlen

Da innerhalb des verwendeten Korpus auch Zahlen, bzw. Zahlenbereiche angegeben werden, werden diese für den Annotator durch einen Vorverarbeitungschritt ersetzt. Dabei kommen ebenfalls reguläre Ausdrücke zum Einsatz, welche jedoch händisch zusammengestellt wurden. Eine Zahlenangabe wie z.B. „CDR 0.4 mm“ ist nach der Vorverarbeitung als „CDR <RN> mm“ dargestellt, wobei „<RN>“ ein regulärer Ausdruck für die jeweilige Art der numerischen Angabe ist. Bei der Generierung der Ausdrücke für die eigentlich gesuchten Textmuster können die regulären Ausdrücke für numerische Angaben schließlich direkt als Tokens behandelt und übernommen werden.

5.5 Groß-/Kleinschreibung

Innerhalb des Korpus sind Begriffe teilweise groß und teilweise klein geschrieben vorzufinden. Daher wurde im RegEx-Annotator eine Möglichkeit implementiert, die Erzeugung der regulären Ausdrücke *case-sensitive* oder *case-insensitive* durchzuführen. Durch das Ignorieren von Groß- und Kleinschreibung innerhalb der Annotationen, können wiederum gleichartige Muster zusammengefasst werden, die sich nur in der Verwendung von Großbuchstaben unterscheiden. Da dies jedoch der Annahme unterliegt, dass in der Verwendung von Großbuchstaben keine relevante Information enthalten ist, ist dieses Feature optional.

5.6 Term-Tree

Vor dem Zusammenstellen der einzelnen regulären Ausdrücke werden die Annotationen aus den Trainingsbeispielen in einer Baumstruktur zwischengespeichert. Diese dient vor Allem der Zusammenfassung gleichartiger Textmuster, wie in Abschnitt 5.2 beschrieben. Sie wird im Weiteren als „Term-Tree“ bezeichnet und nachfolgend erläutert.
Der Sinn des Term-Trees ist es, alle verfügbaren Trainingsbeispiele und ihre Annotationen in einer gemeinsamen Datenstruktur abzulegen. Dabei sollen die Daten so zusammengefasst werden, dass nachfolgend die gesuchten Textmuster in geeigneter Form für die Zusammenstellung der regulären Ausdrücke und deren Verknüpfung mit Konzept-IDs des Katalogs vorliegen.
Die Wurzel des Baumes trägt aus Implementierungsgründen keine Information und enthält lediglich eine Liste mit Referenzen zu ihren Kindknoten. Die Knoten des Baums repräsentieren die einzelnen Wörter, die an sämtlichen verfügbaren Annotationen aus den Trainingsbeispielen beteiligt sind. Neben der Zeichenkette des Worts verfügt ein Knotenobjekt über Attribute für die minimalen und maximalen Werte für den Abstand zum Vorgänger und eine optionale Konzept-ID.
Eine mögliche Instanz eines solchen Term-Trees ist in Abbildung 1 beispielhaft dargestellt. Die Information die in den Attributen der Knotenobjekte enthalten ist, für dieses Beispiel in Tabelle 1 angegeben. Der gezeigte Term-Tree enthält die Information für sechs gültige Textmuster, die in reguläre Ausdrücke überführt werden können. Das Einfügen neuer Beispiele geschieht schrittweise für jede Annotation. Bei einer Annotation mit nur einem Wort werden die Kinder des Wurzelknotens danach durchsucht, ob bereits ein Knoten vorhanden ist, welcher dieses Wort repräsentiert. Wenn dies nicht der Fall ist, wird ein neuer Knoten als Kind des Wurzelknotens hinzugefügt und die entsprechende Konzept-ID als Attribut gesetzt. Falls mehrere Wörter Teil der Annotation sind, wird das erste Wort an der Wurzel eingefügt und die nachfolgenden Wörter jeweils als Kind ihres Vorgängers mit den jeweiligen Abstandswerten. Auch dabei wird geprüft ob entsprechende Knoten bereits vorliegen, und lediglich die minimalen und maximalen Ab-

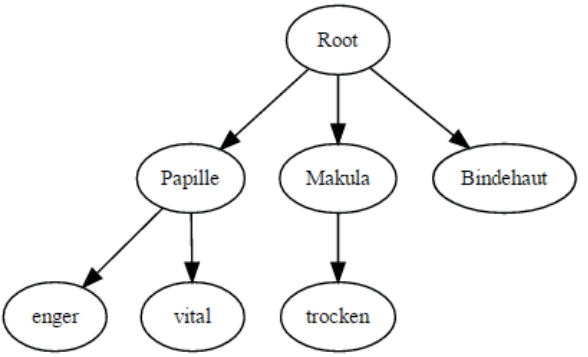

Abbildung 1: Beispielinstanz eines Term-Trees.

Knoten	Konzept-ID	min	max	Kindknoten
Root	-	-	-	Papille, Makula, Bindehaut
Papille	4	-	-	enger, vital
Makula	1	-	-	trocken
Bindehaut	16	-	-	-
enger	4c	1	6	-
vital	1d	1	1	-
trocken	1b	6	15	-

Tabelle 1: Die Information in den Knotenobjekten des Beispiels aus Abbildung 1.

standswerte aktualisiert werden müssen. Die Konzept-ID wird nur beim letzten Wort einer Annotation als Attribut gesetzt.
Um alle aus den Beispielen extrahierten und gültigen Textmuster zu erhalten müssen anschließend nur alle Pfade im Term-Tree gesucht werden, die zu einem Knoten mit einer gesetzten Konzept-ID führen. Wenn dieser Pfad gefunden ist, ist das Erstellen entsprechender regulärer Ausdrücke trivial und wird hier nicht weiter beschrieben. Die einzelnen regulären Ausdrücke werden dann in einer Map eingefügt, welche die Konzept-IDs als Schlüssel verwendet und jeweils auf eine Liste von regulären Ausdrücken verweist, die für die unterschiedlichen Schlüssel bekannt sind.
Das Training des Annotators erfolgt zum aktuellen Projektstand nach dem *Batch-Prinzip* oder *Offline-Lernen*, siehe z.B. [1]. Dabei werden für das Ableiten der Textmuster aus dem Trainingskorpus immer alle verfügbaren Trainingsbeispiele in einem Durchlauf abgearbeitet. Kommt ein neues Trainingsbeispiel hinzu, kann die Wissensbasis nicht inkrementell erweitert werden. Die Anzahl von Beispielen, die für das Training genutzt werden können ist vor Allem in der frühen Projektphase gering. Aus diesem Grund wird der momentane Trainingsansatz nicht als Problem bezüglich der Performance betrachtet, eine Weiterentwicklung in Richtung *Online-Lernen* ist dennoch vorgesehen. Die hier vorgestellte Struktur des Term-Trees ist gut dafür geeignet, in einem weiteren Schritt den Trainingsansatz des *Online-Lernens* elegant umzusetzen.

5.7 Automatische Annotation

Für das automatische Annotieren eines Befunds, werden die gesammelten regulären Ausdrücke durch Standard-Matching Funk-

tionen der RegEx-Implementierungen von Java an dem Befundtext geprüft. Dies stellt die Lösung für das erste Teilproblem aus Abschnitt 2 dar. Wenn ein bestimmtes Muster gefunden wird, wird eine ensprechende Annotation der beteiligten Wörter angelegt. Die Konzept-ID der Annotation wird über die im vorigen Abschnitt angesprochene Map gesucht, womit auch die zweite Klassifikationsaufgabe aus Abschnitt 2 gelöst wird.
Es kann vorkommen, dass einzelne Wörter mehrfach annotiert werden. Beispielsweise würden im Text „Papille eng“, sowohl das Konzept „Papille“, als auch das Konzept „Papille eng“ entsprechend markiert werden. Solche Verschachtelungen werden aktuell als gültig zugelassen. Das Auftreten von überlappenden Annotationen mit gleicher Konzept-ID wird in einem weiteren Schritt abgefangen. Dabei wird standardmäßig die Annotation mit dem größeren Textbereich übernommen - z.B. würden bei dem Text „Papille enger“ die Muster „*Papille eng*“ und „Papille enger“ gefunden werden.

5.8 Einordnung

Die hier entwickelte Methodik der Verwendung regulärer Ausdrücke ist in den Bereich der regelbasierten Ansätze einzuordnen. Die Regeln für die Erkennung von Beschreibungen augenheilkundlicher Konzepte sind dabei innerhalb einer Sammlung regulärer Ausdrücke und deren Verknüpfung mit einem Konzeptkatalog modelliert. Die Muster werden jedoch nicht von Hand durch Menschen erzeugt, sondern anhand von Trainingsbeispielen abgeleitet.

6 Diskussion

Der hier vorgestellte Ansatz für die Extrahierung von Textmustern ist aus der speziellen Problematik heraus entstanden, dass nur sehr wenige Trainingsdaten vorliegen. Ein größerer deutschsprachiger Korpus annotierter Befunde aus der Domäne der Augenheilkunde soll mit Unterstützung des RegEx-Annotators erst noch entstehen. Wenn eine größere Menge von Trainingsdaten erzeugt wurde, ist der Einsatz von gängigen Techniken aus der relevanten Literatur wie z.B. CRFs möglich. Dieses und andere Verfahren aus dem Bereich des maschinellen Lernens bieten die Möglichkeit, deutlich besser generalisierende Modelle für die gesuchten Textmuster zu liefern. Der RegEx-Annotator ist im aktuellen Zustand sehr stark an die Trainingsbeispiele und die markierten Textbereiche gebunden. Er kann zwar präzise Ergebnisse liefern, ist jedoch anfällig für dem menschlichen Betrachter gering erscheinende Unterschiede: z.B. Tippfehler oder eine umgekehrte Reihenfolge der Wörter eines Textmusters. Der Kontext markierter Textstellen wird ignoriert, obwohl dort ebenfalls relevante Informationen enthalten sein könnten. Als Vorteil des vorgestellten Ansatzes ist zu nennen, dass die abgeleiteten Muster und selbst die regulären Ausdrücke für den Menschen leicht lesbar und nachvollziehbar sind, im Gegensatz zu komplexeren Modellen wie etwa CRFs. Auch der aufwändige Prozess einer Merkmalsauswahl, der bei den meisten maschinellen Lernverfahren notwendig ist hier fast nicht vorhanden, da auf Token-Ebene gearbeitet wird.
Es sind verschiedene Möglichkeiten denkbar, den bestehenden Ansatz zu erweitern und ein gewisses Maß an Generalisierung zu erreichen. Den Umgang mit Tippfehlern könnte man etwa unter Berücksichtigung eines Maßes für Wortdistanzen, z.B. der *Levenshtein-Distanz* ermöglichen. Die aus den Beispielen abgeleiteten Abstände aufeinanderfolgender Wörter könnten durch eine probabilistische herangehensweise modelliert werden. Auch der Einsatz von weiteren Vorverarbeitungsschritten, z.B. *Stemming* oder *Lemmatizing* ist möglich. Angesichts des großen Angebots von Techniken aus dem maschinellen Lernen ist abzuwägen, ob der mit einer umfangreichen Erweiterung verbundene Mehraufwand für die Implementierung gerechtfertigt ist.
Ein Aspekt der noch zu untersuchen ist, ist das eventuelle Auftreten von Rückkopplungseffekten durch die Unterstützung von menschlichen Annotatoren durch den RegEx-Annotator. Vorgesehen ist es, dass je-

der vorannotierte Befund vom menschlichen Experten gegengeprüft und ergänzt wird - auch die Markierung falsch erkannter Textmuster. Ein gewisses Risiko besteht, dass fehlerhafte Textmuster vom Menschen übersehen werden und so den Weg in den finalen Korpus finden, welcher wiederum die Grundlage für das spätere Information Retrieval sein soll. Diese Art der Fehlerfortpflanzung muss möglichst vermieden werden um den Trainingskorpus sinnvoll einsetzen zu können.
Schließlich bietet der hier entwickelte Ansatz eine Möglichkeit, für die Evaluierung von anderen Lösungsansätzen in nachfolgenden Arbeiten eine „Baseline“ zu erhalten an der man die Güte der Ergebnisse messen kann.

Literatur

[1] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[3] G. Fette, P. Klügl, M. Ertl, S. Störk, and F. Puppe. Information extraction from echocardiography records. In *Report of the symposium "Lernen, Wissen, Adaptivität 2011 of the GI special interest groups KDML, IR and WM, LWA 2011, Magdeburg, 28.-30.September 2011*, pages 187–190, 2011.

[4] R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[5] N. Indurkhya and F. J. Damerau. *Handbook of Natural Language Processing*. Chapman & Hall/CRC, 2nd edition, 2010.

[6] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.

[7] S. Pradhan, N. Elhadad, B. R. South, D. Martínez, L. M. Christensen, A. Vogel, H. Suominen, W. W. Chapman, and G. Savova. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *JAMIA*, 22(1):143–154, 2015.

[8] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.

[9] H. Suominen and S. Abbott. *Task Descriptions - ShARe/CLEF eHealth 2013*, Oct. 2016. https://sites.google.com/site/shareclefehealth/task-description.

[10] H. Suominen, S. Salanterä, S. Velupillai, W. W. Chapman, G. Savova, N. Elhadad, S. Pradhan, B. R. South, D. L. Mowery, G. J. F. Jones, J. Leveling, L. Kelly, L. Goeuriot, D. Martinez, and G. Zuccon. *Overview of the ShARe/CLEF eHealth Evaluation Lab 2013*, pages 212–231. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Labeling Tool zum teilautomatisierten überwachten Lernen *

Vadim Demkiv
Reutlingen University
Vadim.Demkiv@gmail.de

Abstract

Diese Arbeit erörtert die Möglichkeiten zur Automatisierung der Datenakquise zum überwachten maschinellen Lernen. Das Anwendungsgebiet ist die Erkennung von Fußgängern in Bildaufnahmen aus der Sicht eines Fahrzeugs. Für die Datenakquise der Trainingsdaten werden in der Arbeit Optimierungsmöglichkeiten untersucht. Das umfasst einerseits Automatisierungsmöglichkeiten der Datenakquise selbst. Andererseits folgt auch eine Untersuchung vom Verbesserungspotential der Trainingsdaten, um die Trainingsdatenqualität und daraus resultierend die Erkennungsrate des generierten Klassifikators zu verbessern. Darüber hinaus werden auch die Möglichkeiten an Deskriptoren und Klassifikationsalgorithmen analysiert, um Aussagen darüber zu treffen, welche sich für die Erkennung von Fußgängern im Straßenverkehr eignen.

Schlüsselwörter

Machine learning; Supervised learning; Labeling tool; Object identification; Object recognition; Object detection.

CR-Kategorien

I.4.8 [**Scene Analysis**]: Object recognition;

*

Betreuer Hochschule: Prof. Dr.-Ing. Cristóbal Curio
Hochschule Reutlingen
cristobal.curio@reutlingen-university.de
Betreuer Hochschule: M.Sc. Dennis Ludl
Hochschule Reutlingen
Dennis.Ludl@reutlingen-university.de
Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Vadim Demkiv

1 Bedeutung von Lerndaten im maschinellen Lernen

Maschinelles Lernen wird in einer breiten Palette an Domänen angewandt. Ob es um die Optimierung von Suchalgorithmen bei Suchmaschinen geht, um die Zuordnung von Personen zu Bildern in sozialen Netzwerken, um Sprachassistenten oder um autonomes Fahren – sofern es sich um datengetriebene Systeme handelt, haben diese Dienstleistungen und Produkte des Alltags einen gemeinsamen Nenner: Ihnen liegt eine künstliche Intelligenz zugrunde, die ein umfangreiches datengetriebenes Training absolvieren muss. Im Rahmen dieser Arbeit geht es um Bildverarbeitung und das Training mit Bildausschnitten. Als Anwendungsgebiet wird die Erkennung von Fußgängern aus der Perspektive eines PKWs gewählt, da es sich dabei um eine zentrale Problemstellung bei der Entwicklung von autonomen Fahrzeugen handelt. Dementsprechend orientieren sich die folgenden Beispiele an diesem Anwendungsgebiet. Die wesentlichen Prinzipien können allerdings auch in anderen Anwendungsdomänen angewandt werden – im engeren Kontext auf andere Problemstellungen der Bildverarbeitung, weiter gefasst aber auch auf andere Bereiche der Daten- und Signalverarbeitung.

Um eine zuverlässige Erkennung zu erreichen müssen Klassifikatoren umfassend trainiert und weiterentwickelt werden. Die Trainingsdaten müssen für eine hohe Erkennungsrate derselben oder einer ähnlichen Datendomäne angehören. Unterscheidet sich diese, kann die Erkennungsrate deutlich sinken. (vgl. [17])

Es bleiben in solch einer Situation zwei Möglichkeiten: Einerseits kann eine Adaption an die neue Datendomäne erfolgen. Dabei wird ein bestehender Klassifikator, dessen Lerndaten aus einer anderen Datendomäne stammen, an eine neue angepasst. Mitunter Mozafari & Jamzad beschreiben, wie die Migration eines Klassifikators in eine fremde Datendomäne umgesetzt werden kann (vgl. [17]). Andererseits kann ein neuer Klassifikator auf Basis einer Masse an Positiven und Negativen aus der neuen Datendomäne generiert werden. Diese zweite Möglichkeit steht im Fokus dieser Arbeit, da die Erzeugung eines neuen Klassifikators – auch aus wirtschaftlicher Perspektive – einen besseren Weg darstellen kann. Die Weiterverwendung bestehender Klassifikatoren erhöht die Abhängigkeit nach außen und ist kann auch mit Lizenzrecht und damit einhergehenden Kosten verbunden sein. Das selbstständige Training eines neuen Klassifikators vermeidet diese Problematik.

Das Ziel dieser Arbeit ist daher die Analyse des Stands der Wissenschaft hinsichtlich der *Automatisierungsmöglichkeiten der Datenakquise* sowie der *Qualitätsverbesserung von Lerndaten*. Die Arbeit soll einen Überblick darüber schaffen, welche Teilprozesse bei der Datenakquise automatisiert werden können und worauf bei der Generierung von Lerndaten zu achten ist, um einen Klassifikator mit einer möglichst hohen Erkennungsrate zu schaffen. Parallel dazu findet die Entwicklung eines Labeling Tools statt, mit dem die Datenakquise nach den erarbeiteten Prinzipien sowie Verarbeitungselementen teilautomatisiert ablaufen kann.

Im nächsten Kapitel geht es um die Definition und Erläuterung diverser Ansätze und Methoden, die für das Verständnis der Generierung eines Klassifikators erforderlich sind. Im dritten Kapitel geht es um die Analyse des Automatisierungspotentials der Datenakquise sowie der zu berücksichtigenden Aspekte bei der Generierung von Lerndaten. Das vierte Kapitel beschreibt anschließend den praktischen Teil der Arbeit, die Entwicklung eines Labeling Tools, das die Datenakquise für das maschinelle Training teilautomatisiert unterstützt, gemäß den zuvor beschriebenen Verbesserungsmöglichkeiten. Das letzte Kapitel bietet schließlich einen Ausblick, in welche Richtung sich das maschinelle Lernen entwickeln könnte.

2 Vom Datensatz zum Klassifikator

Auf dem Weg vom Ordner mit Bildaufnahmen bis hin zum fertigen Klassifikator gilt es unterschiedliche Stationen zu durchlaufen. Um ein besseres Verständnis für die damit verbundenen Begrifflichkeiten zu schaffen, sollen in diesem Kapitel die wichtigsten Ausdrücke definiert und ein Verständnis für den Ablauf von der Datenakquise bis zum fertigen Klassifikator geschaffen werden.

2.1 Verarbeitungskette

Der *Lernvorgang* lässt sich in mehrere Elemente einer Verarbeitungskette einteilen. Die Grundlage jedes Trainings sind die *Trainingsdaten*. Zu diesen gehören sowohl Bildausschnitte vom Zielobjekt (*Positive*) als auch von anderen Objekten bzw. der Umgebung (*Negative*). Vereinfacht erklärt werden die Merkmale des Zielobjekts von denen der Umgebung abgegrenzt, um eine Entscheidungsgrundlage bei Erkennungen zu schaffen. Diese Merkmale enthalten meist diverse Attribute von Schlüsselpunkten - beispielsweise von Kanten von Objekten. Dabei definieren und ermitteln *Deskriptoren* diese Merkmale in Form von Vektoren, die anschließend von einem *Klassifikationsalgorithmus* weiterverarbeitet werden. Dieser wiederum verarbeitet mithilfe eines Algorithmus die Merkmale beider Klassen und definiert auf Grundlage dieser Daten die Grenze zwischen ihnen.

2.2 Training beim maschinellen Lernen

Um einen *Klassifikator* zu generieren, muss ein Klassifikationsalgorithmus einen Satz an Trainingsdaten verarbeiten. Ein Klassifikator entspricht dem daraus resultierenden Satz an Regeln, der zur Unterscheidung eines Ziel-

objekts von anderen Objekten dient. Je nach Einsatzgebiet können die Daten aus textuellen Informationen, Tonaufnahmen oder auch Bildern stammen. Bei diesen Daten ist im überwachten Lernen zusätzlich eine Kennzeichnung erforderlich, welche Merkmale zu welcher bekannten Klasse gehören. Bei unüberwachtem Lernen fällt die Kennzeichnung weg, da ein Clustering zusammenhängender Merkmale erfolgt. Letzteres stellt für das hiesigen Anwendungsgebiet allerdings keine Option dar, da die Klassen bekannt sind. Bei binären Klassifikatoren wird lediglich zwischen einer Klasse und allem anderen unterschieden. Bei Mehrklassenklassifikatoren wird zwischen mehreren Klassen sowie allem anderen unterschieden. Der Einfachheit halber ist in diesem Ansatz ein binärer Klassifikator das Ziel, also die Unterscheidung zwischen einem Objekt und allem anderen.

2.3 Vorklassifikator

Als *Vorklassifikator* wird hier ein Klassifikator bezeichnet, der die manuelle *Datenakquise* durch intelligente Vorschläge beschleunigen soll. Die Datenakquise umfasst die Sammlung von Trainingsdaten aus Rohdaten. Das Zielobjekt des Vorklassifikators sollte entsprechend mit dem Ziel des neuen Klassifikators übereinstimmen.

2.4 Bounding Box

Bei einer *Bounding Box* (BB) handelt es sich um einen rechteckigen Rahmen, der einen Interessensbereich (ROI, vom engl. *Region of Interest*) markiert. BB werden häufig in Zusammenhang mit Klassifikatoren genutzt, um erkannte Objekte zu markieren. Sie können aber auch beim Labeling genutzt werden, um manuell ROI in Bildern zu markieren.

Eine weitere Möglichkeit zur Markierung von ROI sind Polygone. Bei Polygonen wird ein Vieleck um die ROI gezeichnet, um diese besser von der Umgebung abzugrenzen. Als Segmentierung wird die präzise Zuordnung aller Bereiche eines Bildes bezeichnet. Im Kontext dieser Arbeit sind allerdings nur die BB interessant, da mit diesen die Datenakquise besonders zügig umgesetzt werden kann.

2.5 Metriken zur Quantifizierung der Erfolgsrate

Zur Evaluation von Klassifikatoren wird insbesondere das *Precision-Recall Modell* genutzt, das die zwei Metriken *Precision* und *Recall* nutzt. Für diese gilt:

$$Precision = \frac{Positive_{richtige}}{Positive_{richtige} + Positive_{falsche}}$$

$$Recall = \frac{Positive_{richtige}}{Positive_{richtige} + Negative_{falsche}}$$

(vgl. Abb. 1 und [16])

3 Auf dem Weg zur Klassifikation von Menschen in Bildaufnahmen

Für die Verbesserung der Erkennungsrate eines Klassifikators gibt es unterschiedliche Möglichkeiten. Der Fokus der Arbeit liegt in der Automatisierung der Datenakquise sowie Verbesserung der Qualität der Trainingsdaten. Aber auch die Elemente der Verarbeitungskette werden untersucht und geeignete Kandidaten vorgeschlagen – darunter die Art der Rohdaten sowie die Wahl des Deskriptors und Klassifikationsalgorithmus.

3.1 Unterstützung der Datenakquise

3.1.1 Ablauf bei vollständig manueller Umsetzung

Für das Training eines Klassifikators sind Positive und Negative notwendig. Um diese zu sammeln, müssten ohne ein Labeling Tool eine Reihe von Aktionen durchführt werden. Ausgehend von einem Ordner mit Bildern, die die Rohdaten bilden, müssen für jedes Bild ROI ermittelt und gespeichert werden. Das Ausschneiden und Speichern dieser ROI in gesonderten Dateien ist aufwändig. Auch das Datenmanagement inklusive der Trennung von Positiven und Negati-

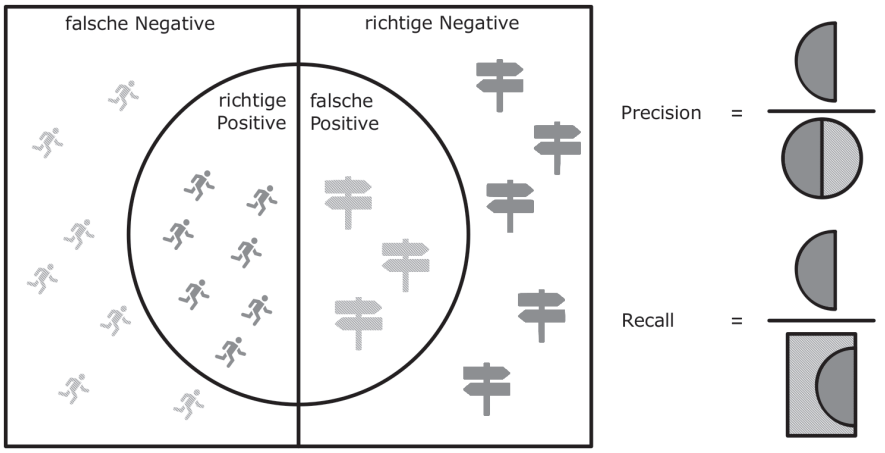

Abbildung 1: Schematische Abbildung von jeweils richtigen und falschen Positiven und Negativen. Zusätzlich sind die relevanten Anteile für die Berechnung der Metriken *Precision* und *Recall* dargestellt.

ven stellt einen deutlichen Aufwand dar. Da ein umfangreicher Datensatz für das Training benötigt wird, fällt die Datenakquise entsprechend aufwendig aus, da der genannte Ablauf für jedes einzelne Bild durchgeführt werden muss.

3.1.2 Automatisierungsmöglichkeiten

Es würde entsprechend zeitaufwendig sein, allein mit einem herkömmlichen Bildverarbeitungsprogramm diesen Vorgang durchzuführen. An dieser Stelle setzt das Labeling Tool mit der Frage an: Welche Teilprozesse können automatisiert werden?
Entsprechend des Ablaufs aus dem vorigen Abschnitt kann an folgenden Punkten eine Optimierung umgesetzt werden:

- Automatisiertes Navigieren durch Bilder innerhalb des Zielordners
- Vorschläge für Positive, die einen Großteil des Aufwands für die Markierung übernehmen
- Automatisiertes Speichern der Bildausschnitte
- Verwaltung bzw. Kennzeichnung von Positiven und Negativen
- Optional: Speichern der Position und Größe der Markierungen, um diese bei einer folgenden Sitzung einsehen und bearbeiten zu können

Über die Beschleunigung der Datenakquise hinaus liefert ein solches Tool beste Voraussetzungen dafür, eine Tabelle mit Informationen zu Positiven, Negativen und ihren jeweiligen Pfadangaben zu generieren. Ist bekannt, welcher Deskriptor verwendet werden soll, können auch direkt die zugehörigen Merkmalsvektoren erzeugt werden.
Zusammengefasst unterstützt ein Labeling Tool also die Navigation durch die Bilder, das Datenmanagement der Lerndaten und beschleunigt durch die Automatisierung diverser Teilprozesse die Datenakquise deutlich. Zudem verfügt es über die Kenntnis der relevanten Datenstrukturen, sodass die automatisierte Zusammenstellung aller erforderlichen Informationen zum maschinellen Lernen eine interessante Erweiterungsmöglichkeit darstellt.

3.1.3 Vorklassifizierung

In erster Linie hat die Vorklassifizierung den Zweck, intelligente Vorschläge für Positive zu liefern, um die manuelle Markierung zu beschleunigen.
Der zweite Nutzen liegt gerade in der Fehlbarkeit des Klassifikators. Denn ein mit einer abweichenden Datendomäne trainierter Klassifikator liefert vermehrt *falsche Positive* – verwechselt also andere Objekte mit dem Zielobjekt. Das kann wiederum von Vorteil sein, da so qualitativ hochwertige – da ähnliche – Negative geliefert werden. Diese können anschließend beim Training eingesetzt werden und so eine bessere Abgrenzung erreichen. Die falschen Positive allein werden als Negative nicht reichen. Eine manuelle Markierung weiterer Bereiche als Negative kann erforderlich sein, sofern der Umfang an falschen Positiven niedrig ist. Welche Bereiche des Bildes bzw. welche Objekte sich hierfür insbesondere eignen, wird im Unterkapitel 3.2 erläutert.

3.1.4 Adaptiver Vorklassifikator

Bei einem adaptiven Klassifikator handelt es sich um einen Klassifikator, der während der Laufzeit bereits das Lernmaterial verarbeitet und dadurch in den Folgeklassifikationen mehr richtige und weniger falsche Positive erkennt. Das macht im Produktivbetrieb Sinn, da so eine Anpassung an vorherrschende Umgebungsbedingungen erfolgen kann, wie beispielsweise an die Beleuchtung oder das Wetter.
Ein bekannter Algorithmus für die adaptive Klassifizierung ist *AdaBoost* (vgl. [9]). Beim *Boosting* selbst geht es um die Kombination mehrerer schwacher, grober Regeln für Vorschläge – hier von erkannten Objekten – zu einer präzisen Regel. Die Adaptivität entsteht durch die Anpassung an die Fehlerraten von schwachen Regeln während der Laufzeit durch die Verarbeitung erkannter Objekte.
Allerdings verhindert ein adaptiver Vorklassifikator auch gleichzeitig eine Verarbeitung im Hintergrund und damit eine flüssige Navigation zwischen den Bildern. Das liegt an der Verarbeitung der neuen Positive und Negative samt der Anpassung des Klassifikators zwischen den Bildern, was zu Wartezeiten und damit zur Beeinträchtigung des Arbeitsflusses führen kann. Eine Implementierung von Erkennungsvorschlägen für den eigentlichen Klassifikator könnte seine Effizienz deutlich steigern (vgl. [12]), bietet allerdings keine Garantie für einen flüssigen Ablauf, da die Berechnungen nach wie vor zwischen den Bildern stattfinden. Da eine flüssige, zügige Datenakquise in dieser Arbeit im Vordergrund steht, wird von der Nutzung eines adaptiven Vorklassifikators abgesehen (vgl. Kapitel 4.1). Ein weiteres Hindernis ist die rückwärtige Navigation des Labeling Tools.

3.2 Worauf bei der Datenakquise zu achten ist

Bei der Datenakquise ist nicht nur das Sammeln von Positiven, sondern auch von Negativen das Ziel. Negative sollten nicht rein zufällige Bildausschnitte sein, sondern gezielt ausgewählt werden. Das Sammeln von qualitativen Trainingsdaten ist das Ziel, um falsche Negative und falsche Positive zu vermeiden. In diesem Teilkapitel wird erläutert, worauf dabei zu achten ist.
Die Sammlung von Positiven dient der Vermeidung von falschen Negativen. Zu falschen Negativen können insbesondere ungewöhnliche Posen führen, aber auch ein schwieriger Hintergrund, geringe Kontraste, Verdeckung, getragene Objekte, Unter- und Überbelichtung sowie Erkennungsversuche in zu großen oder kleinen Skalierungen (vgl. [22]). Dementsprechend sollten die Rohdaten viele solcher Problemfälle bereitstellen, um qualitative Positive sammeln zu können. Dadurch steigt die Robustheit des Klassifikator gegenüber solchen falschen Negativen. Es könnte auch hilfreich sein, die Positive der Trainingsdaten zu duplizieren und die Duplikate zu manipulieren um so mehr Trainingsmaterial zu generieren. Zu den Manipulationsmöglichkeiten zählen unterschiedliche Skalierungen, JPEG-Artefakte, Rotation, Beleuchtung, Blur-Effekte und speziell

das *Salz und Pfeffer* Rauschen (vgl. [12]).
Die Sammlung von Negativen dient der Vermeidung von falschen Positiven. Zu falschen Positiven können insbesondere vertikale Strukturen und ein überladener Hintergrund führen, aber auch zu groß skalierte Bilder bzw. Bildausschnitte mit Menschen im unteren Teil des Bildes, oder auch zu gering skalierte Bilder bzw. Bildausschnitte, auf denen die Personen nur teilweise abgebildet sind (vgl. [22]). Um falsche Positive zu vermeiden, ist es daher hilfreich, gezielt auf diese Elemente zu achten und sie bei der Datenakquise als Negative zu markieren. Besonders wertvoll sind falsche Positive des Vorklassifikators, da diese sehr ähnliche Merkmale enthalten, die es besser abzugrenzen gilt. Durch eine Markierung dieser falschen Positive als Negative wird das erreicht.
Zur Vermeidung von falschen Positiven zu vertikalen Strukturen, wie Schildern oder Ampeln, könnte auch ein zusätzlicher Klassifikator eingesetzt werden, der diese Elemente erkennt und filtert. Bei der Datenakquise kann so ein Klassifikator auch dazu eingesetzt werden um gezielt Negative vorzuschlagen. *Histogram of oriented Gradients* (HoG, vgl. [6]) ist nicht nur besonders für die Erkennung von Menschen geeignet, wie im nächsten Teilkapitel gezeigt wird, sondern es ist auch sehr erfolgreich in der Erkennung von Schildern (vgl. [2]). Durch die Nutzung eines entsprechenden Vorklassifikators könnten gezielt solche Negative gesammelt werden, um eine deutliche Abgrenzung zwischen Schildern und Menschen zu erreichen und damit die häufigste Ursache für falsche Positive zu vermieden. Als Nebeneffekt gäbe es zusätzlich erkannte Schilder, die im autonomen Fahren ohnehin eine wesentliche Rolle spielen.

3.3 Elemente der Verarbeitungskette

Nachdem nun die Datenakquise, der diesbezügliche Nutzen eines Labeling Tools und die Eigenschaften qualitativer Trainingsdaten beleuchtet wurden, folgt nun ein Überblick zu den Elementen der Verarbeitungskette.

3.3.1 Art der Rohdaten

Die Art der Rohdaten stellt die Grundlage dafür dar, welche Informationen gewonnen bzw. generiert werden können. Aus Kostengründen sowie aufgrund der Allgegenwart der entsprechenden Kameras werden häufig RGB-Bilder genutzt. Denkbar sind aber auch zusätzliche Tiefeninformationen (RGB-D). Auch die Nutzung von Ferninfrarot (FIR) stellt eine vielversprechende Möglichkeit dar (vgl. [11, 14]). Sofern die zusätzlichen Kosten einer Infrarotkamera kein Hindernis darstellen, stellt dessen Nutzung eine interessante Möglichkeit dar, wertvolle Informationen zu erhalten – insbesondere auch, weil dadurch die zwei häufigsten Ursachen für falsche Positive – vertikale Strukturen und überladene Hintergründe – besser abgegrenzt werden können (vgl. [1, 11, 15]).

3.3.2 Deskriptoren

Als Deskriptor speziell für die Erkennung von Menschen wird häufig HoG genutzt (vgl. [1, 4, 6, 22, 24]).
Aufgrund der relativ simplen Implementierung und der Erweiterungsmöglichkeiten stellt HoG als Deskriptor für die Erkennung von Menschen einen guten Einstieg dar. Durch eine Kombination mit einer weiteren Merkmalsart (vgl. [22]) oder der Umsetzung eines Teil-basierten Ansatzes (vgl. [7]) lässt sich die Erkennungsrate ausbauen. Für bestmögliche Ergebnisse stellt es allerdings keine Option dar. Ein vielversprechender Ansatz nach Zhang et al., der in zwei bekannten Benchmarks an der Spitze steht, nutzt *Convolutional Feature Maps*, dazu mehr im nächsten Abschnitt (vgl. [23]).

3.3.3 Klassifikationsalgorithmen

Sehr verbreitet – insbesondere auch in der Kombination mit HoG – ist die Nutzung von *Support Vector Machines* (SVM, vgl. [4, 6, 20, 21, 22]). Die Kombination liefert akzeptable Ergebnisse bei einem relativ

geringen Implementationsaufwand und stellt daher eine gute Wahl für den Einstieg dar.
Um allerdings bestmögliche Ergebnisse zu erzielen, lohnt sich ein Blick auf den Caltech Benchmark (vgl. [3]). Die dort verglichenen Ansätze orientieren sich dabei an der von Dollár empfohlenen Metrik der durchschnittlichen Menge an falschen Negativen im Verhältnis zur Anzahl an falschen Positiven auf dem Bild (vgl. [8]). Eine grafische Darstellung der diesbezüglichen Ergebnisse jedes Ansatzes erleichtert den Vergleich von unterschiedlichen Ansätze. Auf der Webseite werden regelmäßig die Ergebnisse neuer Arbeiten hinzugefügt, sodass ein Überblick zum aktuellen Stand ersichtlich ist. In jedem Diagramm ist auch die Erkennungsrate der Kombination aus HoG und SVM abgebildet, da es trotz seines Alters aufgrund der Bekanntheit und des einstigen Erfolgs als Referenz dient.
Zhang et al. haben sich laut ihrer Arbeit mit ihrem Ansatz zum Benchmark Stand 2015 an die Spitze gearbeitet (vgl. [24]). Auch aktuell steht ein neuer Ansatz von Zhang et al. sowohl beim INRIA als auch beim ETH Datensatz – zwei bekannten Benchmarks, die auch im Caltech Benchmark vorkommen – an der Spitze (vgl. [23]). Dieser neue Ansatz nutzt eine Kombination aus *Region Proposal Networks* aus *Faster R-CNN* (vgl. [18]) und *Boosted Forests* (vgl. [10]), als Merkmale kommen wie zuvor erwähnt *Convolutional Feature Maps* mit *Deep Convolutional Features* (vgl. [13]) zum Einsatz .
Zusammengefasst lässt sich sagen, dass eine Kombination von SVM mit HoG einen guten Einstieg samt Erweiterungsmöglichkeiten bietet. Der Vorteil liegt hier in der vergleichsweise simplen und damit schnellen Implementierung. Wie auch die Ergebnisse im Caltech Benchmark schon suggerieren, bedarf es komplexerer Ansätze, um bestmögliche Ergebnisse zu erreichen. Aus aktueller Sicht empfiehlt sich dafür der neue Ansatz nach Zhang et al. (vgl. [23]). Ein Blick auf den Caltech Benchmark ist allerdings empfehlenswert, da die Ansätze kurzlebig sind und sich bereits innerhalb eines Jahres deutliche Sprünge abzeichnen können.

4 Zum Labeling Tool dieser Arbeit

Im Rahmen dieser Arbeit wurde ebenfalls ein praktischer Teil umgesetzt: die Entwicklung eines Labeling Tools nach den in Kapitel 2.3 genannten Kriterien. Auch ein Vorklassifikator wurde umgesetzt – allerdings kein adaptiver, da die Echtzeitfähigkeit dadurch nicht mehr gegeben wäre und der verzögerte Arbeitsfluss schwerwiegender ist als die gelegentlichen falschen Positive, die sogar einen Zusatznutzen als hochwertige Negative darstellen. In diesem Kapitel soll durch eine Zusammenfassung der Möglichkeiten und des Ablaufs ein Einblick in die praktische Umsetzung geschaffen werden.

4.1 Abgrenzung zu bisherigen Labeling Tools

Üblicherweise beschäftigen sich Labeling Tools, wie beispielsweise *LabelMe* (vgl. [19]), mit der manuellen Segmentierung von Bildern. Es geht also in erster Linie nicht um die Datenakquise von Bildausschnitten mit entsprechenden Objekten – auch wenn diese anschließend aus den Markierungen generiert werden können – sondern um eine relativ präzise Zuordnung von Bildbereichen zu Klassen mithilfe von Polygonen.
Das hier entwickelte Labeling Tool geht einen anderen Weg, denn die umfangreicheren Möglichkeiten der Markierung liefert keinen Mehrwert, wenn lediglich ein binärer Klassifikator das Ziel ist. In diesem Fall müssen lediglich Positive und Negative von der Zielklasse gesammelt, und damit im Labeling Tool zwei Arten von Markierungen umgesetzt werden. Die rechteckige Form einer BB ist dabei ausreichend. Dadurch, dass das Tool sich lediglich auf BB beschränkt, läuft die Datenakquise schneller ab, da nur Boxen gezogen werden müssen, statt mit einer Vielzahl von Punkten Polygone. Eine weitere Idee hinter dem Tool ist die direkte Generierung eines Klassifikators aus dem Labeling Tool. Dadurch entfällt praktisch der

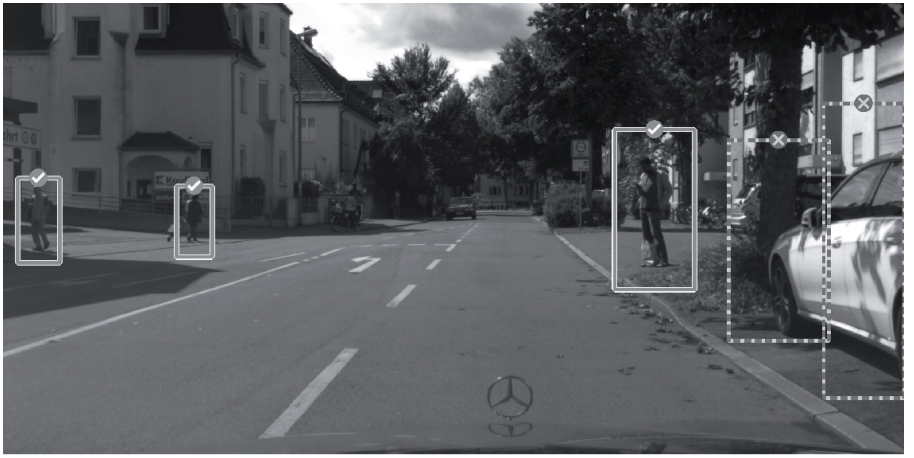

Abbildung 2: Beispiel eines Bildes aus dem Labeling Tool mit Vorschlägen eines Vorklassifikators für die Erkennung von Fußgängern. Die ursprüngliche Bildaufnahme stammt aus dem Cityscapes Datensatz (vgl. [5]) und wurde vom Labeling Tool mit BB versehen.

Export der gesammelten Trainingsdaten aus dem Tool und anschließende Import in ein weiteres Tool bzw. die manuelle Arbeit damit. Die Konfiguration der Lernanweisungen wird von Programmcode auf eine Benutzeroberfläche verlagert. Da so auch die Konfigurationsdatei gemäß des eingestellten Deskriptors und Klassifikators samt der zugehörigen Parameter vom Labeling Tool generiert werden kann, entfällt ein mühsamer Schritt.

Ein Labeling Tool, das sowohl die Datenakquise als auch die Generierung eines Klassifikators vereint, liefert nicht nur einen Mehrwert für Einsteiger, die sich so ohne Programmierkenntnisse mit maschinellem Lernen vertraut machen können, sondern auch für Forscher, die empirische Untersuchungen zu unterschiedlichen Kombinationen aus Rohdaten, Deskriptoren und Klassifikationsalgorithmen zügig durchführen wollen.

4.2 Funktionsumfang

Als Input erhält das Programm einen Ordner mit Bildern sowie optional einen Vorklassifikator für intelligente Vorschläge. Ein Starten der Vorklassifikation erzeugt einen Thread im Hintergrund, in dem alle Bilder verarbeitet und die Ergebnisse zu jedem Bild als JSON Datei hinterlegt werden. Der Benutzer kann Bild für Bild durch die Bildsammlung navigieren und entweder neue BB erzeugen, oder mit den Vorschlägen aus der Vorklassifikation interagieren. Die Interaktion mit den BB umfasst den Wechsel von Positiv zu Negativ und umgekehrt, sowie das Löschen bei fehlerhaften oder versehentlich erstellten Exemplaren. Ein Beispiel dafür, wie ein typisches Bild mit Vorschlägen in dem Labeling Tool aussieht, ist in Abb. 2 ersichtlich. Zwei Vorschläge wurden manuell als Negative markiert.

Der Benutzer kann jederzeit die bisherigen Ergebnisse allein speichern, oder zusätzlich auch alle bereits im Hintergrund berechneten Vorschläge des Vorklassifikators. Zweiteres bietet sich für einen ersten Eindruck über die Qualität vom Vorklassifikator an. Für ein Training eignen sich die Vorschläge des Vorklassifikators alleine allerdings nicht. Einerseits ist eine manuelle Kennzeichnung der falschen Positive als Negative erforderlich, andererseits müssen weitere Negative manuell gesammelt werden.

4.3 Erweiterungsmöglichkeiten durch modularen Ansatz

Der Weg für Erweiterungsmöglichkeiten hinsichtlich neuer Kombinationen aus Deskriptoren und Klassifikatoren ist insofern geebnet, als dass lediglich sieben Zeilen im Programm hinzugefügt werden müssen, um das Tool um ein neues Modul mit einem weiteren Klassifikationsansatz zu erweitern. Jedes Modul besteht aus drei Klassen. Eine Klasse dient der *Definition der Optionen* für den Vorklassifikator und das Training. Eine weitere Klasse setzt die *Vorklassifizierung* und Erarbeitung von Trainingsdaten samt deren Export als JSON Datei um, was als Ausgangsgrundlage für die manuelle Validierung dient. Die letzte Klasse definiert die *Umsetzung des Trainings*, also des maschinellen Lernens mit den ermittelten Positiven und Negativen. Für jede der drei Klassen ist ein Interface definiert, um modulübergreifend eine einheitliche Struktur zu schaffen.

5 Multitool – Labeling, Training und mehr

Zusammengefasst hat das im Rahmen dieser Arbeit entwickelte Labeling Tool die Einschränkung, dass dessen Fokus binäre Klassifikatoren sind. Zu den Vorteilen gehören, dass Labeling und Training bedienungsfreundlich vereint werden und zudem durch den modularen Aufbau Erweiterungsmöglichkeiten bestehen – beispielsweise durch die Einbindung weiterer Kombinationen aus Deskriptoren und Klassifikationsalgorithmen mit neuen Modulen.
Darüber hinaus wäre es – insbesondere im Hinblick auf die Erkennung von Fußgängern und den Vergleich der Erkennungsraten unterschiedlicher Ansätze – nützlich, bekannte Benchmark Datensätze einzubinden. Zu interessanten Kandidaten gehören beispielsweise die unterschiedlichen Datensätze aus dem erwähnten Caltech Benchmark (vgl. [3]). Um das Labeling Tool um eine Benchmark Funktion zu erweitern, müssen die automatisiert ermittelten Positive mit *Ground Truth* Daten abgeglichen werden. Dabei handelt es sich um Ergebnisse eines Labeling Durchgangs, der von einem Menschen evaluiert wurde und als Referenz dient. Durch den Vergleich der Ergebnisse des Klassifikators und der *Ground Truth* Daten können Metriken berechnet werden, darunter die hier vorgestellten Metriken *Precision* und *Recall*.

Literatur

[1] B. Besbes, A. Rogozan, A.-M. Rus, A. Bensrhair, and A. Broggi. Pedestrian detection in far-infrared daytime images using a hierarchical codebook of surf. *Sensors (Basel, Switzerland)*, 15(4):8570–8594, 2015.

[2] M. Beyeler. *OpenCV with Python blueprints: Design and develop advanced computer vision projects using OpenCV with Python*. Community experience distilled. Packt Publishing, Birmingham, UK, 2015.

[3] Caltech. Pedestrian detection benchmark, 04.09.2016.

[4] Y. Chen, L. Zhang, X. Liu, and C. Chen. Pedestrian detection by learning a mixture mask model and its implementation. *Information Sciences*, 2016.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. 2005.

[7] P. Dollár, B. Babenko, S. Belongie, P. Perona, and Z. Tu. Multiple component learning for object detection. In D. Forsyth, P. H. S. Torr, and A. Zisserman, editors, *Computer Vision - ECCV*

2008, volume 5303 of *Lecture Notes in Computer Science*, pages 211–224. Springer-Verlag, Berlin, 2008.

[8] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34, 2012.

[9] Y. Freund and R. E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

[10] J. Friedman, R. Tibshirani, and T. Hastie. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

[11] A. Gonzalez, Z. Fang, Y. Socarras, J. Serrat, D. Vazquez, J. Xu, and A. M. Lopez. Pedestrian detection at day/night time with visible and fir cameras: A comparison. *Sensors (Basel, Switzerland)*, 16(6), 2016.

[12] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2016.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 1097–1105, 2012.

[14] J. H. Lee, J.-S. Choi, E. S. Jeon, Y. G. Kim, T. T. Le, K. Y. Shin, H. C. Lee, and K. R. Park. Robust pedestrian detection by combining visible and thermal infrared cameras. *Sensors (Basel, Switzerland)*, 15(5):10580–10615, 2015.

[15] Y. Ma, X. Wu, G. Yu, Y. Xu, and Y. Wang. Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery. *Sensors (Basel, Switzerland)*, 16(4):446, 2016.

[16] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel. Performance measures for information extraction. In *In Proceedings of DARPA Broadcast News Workshop*, pages 249–252, 1999.

[17] A. S. Mozafari and M. Jamzad. A svm-based model-transferring method for heterogeneous domain adaptation. *Pattern Recognition*, 56:142–158, 2016.

[18] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[19] A. Torralba, B. C. Russell, and J. Yuen. Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484, 2010.

[20] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, Chichester, 1998.

[21] V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moskau, 1974.

[22] C. Wojek and B. Schiele. A performance evaluation of single and multi-feature people detection. In G. Rigoll, editor, *Pattern recognition*, volume 5096 of *LNCS sublibrary: SL 6 - Image processing, computer vision, pattern recognition, and graphics*, pages 82–91. Springer, Berlin and New York, 2008.

[23] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? *CoRR*, abs/1607.07032, 2016.

[24] S. Zhang, R. Benenson, M. Omran, J. H. Hosang, and B. Schiele. How far are we from solving pedestrian detection? *CoRR*, abs/1602.01237, 2016.

Verringerung des Entwicklungs- und Wartungsaufwandes automatischer Web-Oberflächentests

Charalambos Evgenidis
Hochschule Reutlingen
Alteburgstraße 110
Reutlingen, Deutschland

charalambos.evgenidis@student.reutlingen-university.de

Abstract

Diese Arbeit behandelt das Thema der Testentwicklung automatischer Web-Oberflächentests und der Reduzierung des Wartungsaufwandes jener Tests. Analysiert werden dabei Probleme und Schwierigkeiten die während der Testentwicklung bzw. Testwartung auftreten. Anschließend folgen die dazu passenden Lösungen, die stabile und wartungsarme Tests ermöglichen. Zu diesem Zweck werden verschiedene Werkzeuge vorgestellt, welche die Testentwicklung beschleunigen. Darüber hinaus wird kurz erläutert worin sich die diversen Werkzeuge voneinander unterscheiden. Im Besonderen wird dabei auf den fortschrittlichen und frei verfügbaren Robula+ -Algorithmus eingegangen, mit dessen Unterstützung die aufwändige Wartung der Tests weitestgehend minimiert werden soll. Im Zuge dessen, erfolgt eine Analyse des Algorithmus. Dabei werden Schwächen hervorgehoben und gelöst.
Zuvor wird auf die Einsatzgebiete automatischer Tests eingegangen und Probleme diskutiert, welche während der Testphasen und Testentwicklung auftreten, beginnend mit der Fragestellung, wie solche Tests im aktuellen Kontext technisch realisierbar sind.

Schlüsselwörter

Web testing, maintenance effort reduction, test cases fragility, CSS & XPath locator,

CR-Kategorien

D.2.5 [Software]: Testing and Debugging

E.1 [Data]: Data Structures

1 Einleitung

Vergleichsportale, Versicherungen, E-Banking und weitere anspruchsvolle dynamische Webseitentypen erfordern einen hohen Aufwand zur Sicherung deren Qualität. Um dabei möglichst effizient testen zu können, sind automatische Werkzeuge unbedingt notwendig [vgl. 4, S.93]. Die Webseite der Direktbank DKB (Deutsche Kreditbank) besteht bspw. aus über 900 verschiedenen, sich dynamisch ändernden Einzelseiten. Wird diese Menge an Seiten

---

Betreuer Hochschule: Prof. Dr. Martinez
Hochschule Reutlingen
Natividad.Martinez@reutlingen-university.de
Betreuer Firma: Dipl.-Inf. Dirk Feufel
Crealogix AG, Stuttgart
Dirk.Feufel@crealogix.com

Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Charalambos Evgenidis

regelmäßig von Menschen getestet, so kommt es neben hohen Personalkosten, die zwischen 40% und 80% des gesamten Entwicklungsbudgets betragen können [vgl. 10], auch zu Fehlern während des Testvorgangs. Neben der mit der Zeit nachlassenden Konzentration der Tester kommt der monotone Testablauf hinzu. Beides kann dazu führen, dass Fehler auf der Webseite, die sich wiederrum auf die Steuerung negativ auswirken oder ganze Prozessabläufe wie z.B. den einer Überweisung, unmöglich machen, übersehen werden. In jedem Fall senken derartige Fehler die Kundenzufriedenheit. Ferner ist es für die Webseitenbetreiber nur schwer möglich, Lasttests mit einer kleinen Gruppe von Testern durchzuführen. Des Weiteren müssen die Tester genaue Kenntnisse über die zu testende Webseite besitzen, um erwartete sowie unerwartete Fehler als solche zu erkennen. Die Tester müssen zur Fehlererkennung dabei unterschiedliche Browser verwenden. Dies ist notwendig um Browserspezifische Fehler, wie bspw. einer Schaltfläche die unter einem bestimmten Browser nicht korrekt funktioniert, zu ermitteln. Die Lösung zu den genannten Problemen bieten automatische Tests.

Automatische Tests ahmen das Verhalten echter Webseitennutzer nach und sind beliebig oft hintereinander und parallel ausführbar.

Regressionstests, welche nach einem vorgegebenen, immer wiederkehrenden Muster ablaufen, sichern die Qualität von Prozessabläufen [vgl. 7, S.62] wie bspw. dem Beantragen eines Kredites oder einer Versicherungsleistung.

Eine Webseite muss auch bei hoher Auslastung zuverlässig arbeiten. Aufgrund dessen wird ein hohes Besucheraufkommen auf einer Webseite durch Last- und Performancetests simuliert. Durch einen Lasttest lotet der Betreiber der Webseite aus, für wie viele gleichzeitige Nutzer der stabile Betrieb der Webseite garantiert werden kann. Dabei müssen "Testszenarien so definiert werden, dass sie Aussagen über das reale Verhalten zulassen" (siehe 8, S.305). Ist die Auslastung bei einer geringen Besucherzahl zu hoch, ist das Aufrüsten der Hardware oder eine softwaretechnische Lösung vom Webseitenbetreiber in Betracht zu ziehen. Darüber hinaus sind identische Tests mit unterschiedlichen Browsern von hoher Wichtigkeit, um browserspezifische Fehler zu erkennen und zu vermeiden. Ferner ist mithilfe automatischer Tests der gesamte Testvorgang detailgenau protokollierbar und dadurch ein schnelles und fokussiertes Eingreifen durch den Entwickler möglich. Nichtsdestotrotz ist der Entwicklungs- und Wartungsaufwand automatischer Tests hoch wie im nächsten Abschnitt beschrieben.

2 Wartungsaufwand automatischer Tests

Wie bei nahezu jeder Software üblich, müssen automatische Tests ebenfalls regelmäßig aktualisiert und gewartet werden. Der Wartungsaufwand ist unter Anderem abhängig vom Umfang der Tests, deren technischer Realisierung sowie dem zu testenden System. Dabei erhöht sich der Wartungsaufwand der Tests mit steigender Komplexität der Software nicht linear, sondern exponentiell [vgl. 1, S. ix]. Bei automatischen Webseiten-Oberflächentests ist die Wahrscheinlichkeit einer notwendigen Testüberarbeitung mit jeder neuen Veröffentlichung (dem sogenannten Release) der Webseite hoch. Gründe für diesen Umstand liegen beispielsweise in einer veränderten Seitenstruktur, neu generierten Identifikationsmerkmalen von Web-Elementen (siehe Abschnitt 4) oder neuem Inhalt. Derartige Änderungen finden während des Entwicklungsvorgangs einer Webseite teilweise täglich statt. Dies gilt selbst für Seiten, welche mithilfe eines Webseiten-Baukastens erzeugt worden sind. Je nach Software sind die daraus erzeugten Strukturen nach jedem Release, mit Ausnahme der neu eingefügten Inhalte durch den Benutzer, nahezu unverändert oder es wurden automatisch generierte Web-

Elemente mit neuen Identifikationsmerkmalen versehen. Im letzteren Fall ist eine starke Überarbeitung des korrespondierenden automatischen Tests notwendig, wohingegen im ersteren Fall nur ein kleiner Teil des Tests eine Überarbeitung erfordert.

3 Stand der Technik automatischer Weboberflächentests

Es existieren zwei Arten automatischer Tests für Webapplikationen. Die erste davon sieht das Testen einer Webapplikation über eine konventionelle Browseroberfläche vor, während die zweite das Testen ohne grafische Oberfläche ermöglicht. Letztere Methode ist deutlich performanter durch das Einsparen der grafischen Benutzeroberfläche und besonders für Lasttests geeignet, welche die Serverstabilität testen und messen sollen. Im Bereich der kostenlosen Werkzeuge ist Selenium der Marktführer. Mit Selenium sind Tests realisierbar, die mit als auch ohne grafische Benutzerschnittstelle auskommen. Weitere Werkzeuge in diesem Bereich sind HtmlUnit und PhantomJS.

Automatische Tests, die über die grafische Benutzerschnittstelle die Webapplikation steuern, nutzen hierfür entweder den DOM-Baum der HTML-Seite oder die Position der gerenderten grafischen Web-Elemente [vgl. 9]. Diese Arbeit behandelt erstere Methode.

Automatische Web-Oberflächentests protokollieren während des Testvorgangs parallel den sichtbaren Bereich der Applikation mithilfe von Bildschirmfotos oder -videos. Dies hilft dem Entwickler, etwaige Darstellungsfehler auf der Seite zu identifizieren und zu beheben. Neben Selenium sind hier das Robot-Framework zu nennen sowie die proprietären Programme JStudio SiteWalker und Ranorex.

Das Robot Framework nutzt Tests die in natürlicher Sprache geschrieben sind, während JStudio SiteWalker und Ranorex die Capture & Replay Methode zu Testaufzeichnung nutzen [vgl. 6, S. 213].

Bestimmte Testautomatisierungswerkzeuge wie z.B. *BadBoy* bringen ihren eigenen Browser mit. Der mitgelieferte Browser vereinfacht die Ansteuerung der Webapplikation, verhindert aber gleichzeitig das Testen auf unterschiedlichen Browsern, weswegen hier nicht weiter darauf eingegangen wird.

Grundsätzlich besitzen alle Testautomatisierungswerkzeuge die Schwäche, dass Tests, mit jeder neuen Veröffentlichung (dem sogenannten Release), mit hoher Wahrscheinlichkeit überarbeitet werden müssen. Thema dieser Arbeit ist es den dadurch steigenden Mehraufwand so weit wie möglich zu minimieren.

Worauf bei einem automatischen Test, welcher über eine grafische Benutzerschnittstelle abläuft, geachtet werden muss und wie die Ansteuerung eines Browsers gelingt, wird im nächsten Abschnitt diskutiert.

3.1 Ansteuerung

Automatische Web-Oberflächentests dürfen nicht von der Auflösung des Bildschirms und der Größe des Fensters der zu testenden Applikation abhängig sein [vgl. 1, S.53]. Ein solcher Test muss auch nach einer Änderung der grafischen Oberfläche funktionieren - sofern sich der Funktionsumfang und -ablauf der zu testenden Applikation im Laufe der Entwicklung nicht verändert hat.

Eine direkte Steuerung des Browsers ist mit *Javascript Injection* möglich. Das bedeutet, dass JavaScript-Code in die aktuelle Webseite eingeschleust und ausgeführt wird. Neben der Ausgabe von Pop-Ups sind auch Änderungen am Layout, Funktionsumfang und Farbgebung möglich. Ebenso sind Cookies manipulierbar und Inhalte auslesbar. Derartige Abfragen und Veränderungen sind ausschließlich Client-seitig möglich.

Der nächste Abschnitt beschreibt wie Web-Elemente – dazu gehören z.B. Textfelder und Schaltflächen – konkret angesteuert werden.

3.2 Browser mit XPath & CSS steuern

Selenium nutzt zur Browsersteuerung (wie jedes vergleichbare Werkzeug) Javascript Injection. Mithilfe von Javascript Injection kann Selenium zur direkten Ansteuerung von Web-Elementen, *XPath* (XML Path Language) und *CSS* (Cascading Style Sheets) nutzen.

XPath wird seit 1999 vom *World Wide Web Consortium* (W3C) zur Adressierung von Knoten in XML-Bäumen empfohlen (vgl. 3, Introduction). Mithilfe dieser Technik ist die Lokalisierung bzw. Adressierung von Web-Elementen in DOM-Bäumen (Document Object Model) möglich.

Das folgende Beispiel beschreibt einen einfachen XPath-Ausdruck, welcher das Schaltflächenelement *Auf gut Glück!* der Google Standardsuchseite anvisiert: *//input[@name='btnI']*. Die Zeichen *//* geben an, dass es sich hier um einen XPath-Ausdruck handelt. Der Bezeichner *input* deutet auf eine HTML-Schaltfläche hin. Die Bezeichnung *@name* bezieht sich auf das Attribut *name* der Schaltfläche und *btnI* ist der darin enthaltene Wert. Die Ausführung des XPath-Ausdrucks markiert damit das folgende HTML-Element des DOM-Baums der Google-Startseite:

<input type="submit" jsaction="sf.lck" name="btnI" aria-label="Auf gut Glück!" value="Auf gut Glück!"/>.

Denselben Effekt besitzt auch der folgende XPath-Ausdruck: *//input[@value='Auf gut Glück!']*. In diesem Fall wird auf das Attribut *value* mit dem Inhalt „*Auf gut Glück!*“ verwiesen, um das Element zu lokalisieren.

Neben XPath wird zur Lokalisierung von Web-Elementen CSS in der dritten Version genutzt. Konkret geschieht dies mit den sogenannten *CSS-Selectors*. Zwischen CSS und XPath gibt es neben einer unterschiedlichen Syntax geringfügige, jedoch vernachlässigbare Unterschiede in der Performance zugunsten von CSS. Dagegen ist mit XPath, im Gegensatz zu CSS, die Lokalisierung von Web-Elementen über deren Textinhalt möglich.

Der nächste Abschnitt geht näher auf die Identifikationsmerkmale von Web-Elementen und den dazugehörigen XPath- und CSS-Ausdrücken ein, mithilfe derer eine Lokalisierung und letztendlich eine automatische Ansteuerung möglich ist.

4 Die Grundproblematik bei der Web-Elementidentifikation

Selenium ist zur Entwickelung automatischer Web-Oberflächentests notwendig. Die für die Ansteuerung genutzten XPath- und CSS-Ausdrücke muss der Entwickler jedoch selbst oder mit Hilfe von Werkzeugen ermitteln. Die Firefox-Erweiterung Firepath aus Firebug z.B., generiert mit einem Klick auf das gewünschte Element derartige Ausdrücke automatisch und kann darüber hinaus auch manuell erzeugte Ausdrücke testen. Weitere Werkzeugen in dieser Kategorie sind *Selenium IDE* und Robula+.

Für die Schaltfläche aus Abschnitt 3.2 bildet Firepath folgenden XPath-Ausdruck:

html/body/div[1]/div[3]/form/div[2]/div[3]/center/input[2]

Selenium IDE, ebenfalls eine Firefox-Erweiterung, ist speziell für das schnelle Testen von Weboberflächen entwickelt worden. Mithilfe von *Capture & Replay* erstellt der Anwender Tests, indem er die Klickreihenfolge auf einer Webseite durch die Selenium IDE aufzeichnen lässt, um sie im Anschluss wieder automatisch ablaufen zu lassen. Dabei erzeugt die Selenium IDE folgenden XPath-Ausdruck für die bereits genannte Schaltfläche:

//input[@name='btnI']

Die offensichtlichen Unterschiede hinsichtlich Lesbarkeit, Länge und Struktur zwischen den beiden automatisch gebildeten XPath-Ausdrücken sind groß. Während der XPath-Ausdruck aus Firepath den vollständigen Pfad zum Web-Element wiedergibt, zielt der Ausdruck, den die Selenium IDE gebildet hat, auf ein Element-

Attribut. Für einen Web-Oberflächentest ist die Selenium-Variante wesentlich stabiler und damit geeigneter. Für automatische Tests kommen weitere relevante Punkte zu den drei bereits genannten hinzu.

Hinsichtlich der Wartbarkeit ist der XPath-Ausdruck aus der Selenium IDE wesentlich einfacher zu pflegen. Neben der besseren Lesbarkeit, ist die Wahrscheinlichkeit geringer, dass Änderungen am Ausdruck vorgenommen werden müssen sobald ein Update der Seite erscheint. Um das Gewicht dieses Problems hervorzuheben, seien im Anschluss ein paar Beispiele genannt.

Beispiel 1: Angenommen die Schaltfläche *Auf gut Glück!* wird in ein (Div-) Container-Element gesetzt, welches für den Benutzer nicht sichtbar ist. In diesem Fall ändert sich der DOM-Pfad des Elements und der XPath-Ausdruck aus Firepath funktioniert in dieser Variante, im Gegensatz zum Ausdruck aus der Selenium IDE, nicht mehr.

Beispiel 2: In einem weiteren Beispiel wird lediglich die Position der Schaltfläche mit der der Google-Suche vertausch. Auch in diesem Fall funktioniert der strukturorientierte XPath-Ausdruck nicht mehr, wohingegen der Attributs-orientierte Ausdruck aus der Selenium IDE weiterhin das korrekte Ergebnis liefert.

Beispiel 3: Im letzten Fall wird ein Div-Container am Anfang der HTML-Seite entfernt oder hinzugefügt. Auch in diesem Fall ändert sich der Pfad zur Lokalisierung der Schaltfläche und der XPath-Ausdruck aus Firepath liefert kein oder zumindest ein falsches Ergebnis.

Alle drei genannten Beispiele zielen auf die Problematik der Lokalisierung bzw. Identifikation von Web-Elementen mittels DOM-Pfad. Absolute XPath- und CSS-Ausdrücke sind für ihre Instabilität bekannt [vgl. 11]. Aufgrund dessen ist diese Art für automatische Web-Oberflächentest denkbar ungeeignet. Nichtdestotrotz ist die Lokalisierung automatisch generierter Web-Elemente nicht minder problematisch, wie im nächsten Abschnitt beschrieben wird.

4.1 Lokalisierungsproblem bei generischen Web-Elementen

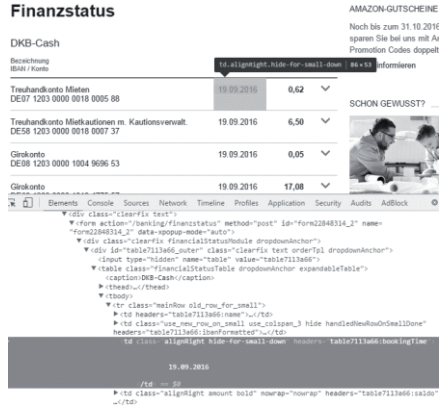

Abbildung 1: Mit XPath markiertes Web-Element des DKB-Online-Banking-Portals

Häufig stammen Webseiten (z.B. ein Blog oder ein Wiki) aus einem Webseiten-Baukasten oder sie wurden mithilfe eines Content-Management-Systems (CMS) bzw. einem Framework erzeugt. Derartige Werkzeuge bilden oft automatisch die Attribute der Web-Elemente einer Webseite wie beispielsweise einer *ID*, einem Type, einem *Header* oder einer sogenannten *Class*.

In Abb. 1, auf der Finanzstatusseite des DKB-Online-Banking-Portals ist das Datum der letzten Überweisung hervorgehoben. Weiter unten ist das Web-Element im DOM-Baum zu sehen. Es handelt sich um eine Zelle einer automatisch generierten Tabelle. Wird in diesem Fall der XPath-Ausdruck *//td[@class='alignRight hide-for-small-down']* verwendet, so sind neben dem gewünschten Element eine Reihe weiterer, verwandter Elemente markiert und damit eine eindeutige Lokalisierung, wie in Abb. 2 zu sehen ist, nicht möglich.

Dasselbe Problem ergibt sich ebenfalls bei der Verwendung des Header-Attributs (*td[@headers='table34ae52c:bookingTime']*).

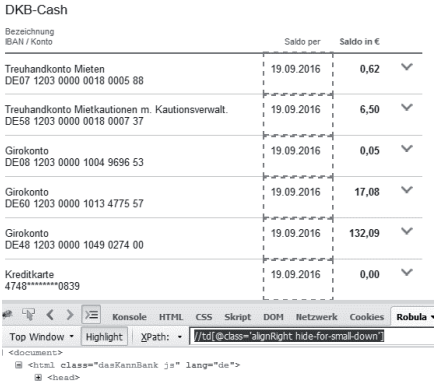

Abbildung 2: Nutzung eines nicht eindeutigen XPath-Ausdrucks

Dieses Attribut besitzt darüber hinaus eine Besonderheit: ein Teil des Attributwertes wird automatisch generiert. Erkennbar ist dieser Teil an seiner kryptischen Zahlen- und Buchstabenfolge *table34ae52c*. Mit einer erneuten Anmeldung auf der Webseite ändert sich dieser Teil automatisch und ist aufgrund dessen nicht für stabile automatische Web-Oberflächentests geeignet.

Alle Arten automatisch generierter Web-Elemente sind aufgrund genannter Probleme schwierig zu identifizieren bzw. zu lokalisieren und daher nur unter größerem Aufwand in stabile automatische Tests integrierbar.

Um das Problem der Lokalisierung der beschriebenen Zelle in der Tabelle zu lösen, ist in diesem Fall eine Kombination aus struktur- und attributsorientiertem XPath-Ausdruck notwendig. Der Ausdruck *//tr[1]/td[contains(@headers,'bookingTime')]* markiert in diesem Fall die gewünschte Zelle mit dem entsprechenden Datum. Der Teilausdruck *//tr[1]* zielt auf die zweite Zeile einer unbestimmten Tabelle der aktuellen Seite ab und */td* auf eine Spalte die ein Attribut *header* besitzt, welcher den Teil-String (*contains*) *bookingTime* enthält. Auf diese Weise wird der automatisch generierte Teil *table34ae52c:* nicht weiter beachtet. Auch in dieser Lösung fehlt ein eindeutiges Attribut, welches die entsprechende Tabelle repräsentiert. Da es möglich ist, identische Attribute mit gleichem Wertinhalt auf derselben HTML-Seite mehrmals zu nutzen, ist der Testentwickler gezwungen, die DOM-Struktur im XPath-Ausdruck, wie in diesem Fall, zu berücksichtigen. Die beste Lösung hierzu ist, den Pfad, welcher auf das entsprechende Element weist, möglichst kurz aber eindeutig zu halten. Für den obigen Fall ist der XPath-Ausdruck

//div[contains(@class,'financialStatus')]/div[1]/table//tr[1]/td[contains(@headers,'bookingTime')]*

dafür notwendig, da zwei weitere, nahezu identische, Tabellen auf der HTML-Seite existieren.

4.2 Verringerung des Wartungsaufwands

Mit Werkzeugen wie bspw. der Selenium IDE FirePath ist eine zügige Erzeugung des gewünschten XPath-Ausdrucks oder CSS-Selectors möglich, jedoch kaum für stabile Web-Oberflächentests geeignet. Manuell extrahierte Ausdrücke zur Lokalisierung von Elementen sind indes deutlich robuster, sie erfordern aber zur Ermittlung viel Zeit. Eine gute Alternative zu genannten Werkzeugen und der manuellen Ermittlung von Web-Elementen ist der Robula+-Algorithmus. Der intelligente Algorithmus soll wesentlich stabilere XPath-Ausdrücke erzeugen, um robuste Web-Oberflächentests zu ermöglichen [vgl. 5, S.2].

Im nächsten Abschnitt erfolgt die Analyse der Arbeitsweise des Werkzeugs.

5 Der Robula+- Algorithmus

Grund für die Entwicklung von Robula+ ist die Reduzierung des Wartungsaufwands bei Web-Oberflächentest. Dies wird durch möglichst beständige XPath-Ausdrücke erreicht, die zur Lokalisierung von Web-Elementen nötig sind [vgl. 5, S.180].

Der von Robula+ erzeugte XPath-Ausdruck ist im Vergleich zu den anderen Werkzeugen, wie in Abb. 3 zu sehen ist, wesentlich kürzer. Während die anderen Werkzeuge teilweise

oder ausschließlich DOM-Strukturinformationen in ihrem erzeugten XPath-Ausdruck verwenden, kommt der aus Robula+ stammende Ausdruck nahezu ohne derartige Informationen aus. Wie in Abschnitt 4 vermittelt, sollten robuste XPath-Ausdrücke bestenfalls keine oder so wenig Strukturinformationen wie möglich enthalten um die Wahrscheinlichkeit einer notwendigen Anpassung nach einem erneuten Release der Webseite zu minimieren. Ein von Robula+ erzeugter Ausdruck zielt bspw. auf den enthaltenen Text eines Attributes um das Element zu lokalisieren, während andere Werkzeuge den kompletten DOM-Pad dafür nutzen. Inwiefern Robula+ dies bewerkstelligt, wird im nächsten Abschnitt beschrieben.

5.1 Funktionsweise

Der Robula+-Algorithmus nutzt heuristische Methoden zur Bildung von XPath-Ausdrücken. Beginnend mit dem letzten Knoten *xp* (d.h. dem Zielelement) sucht Robula+ nach dem passenden Ausdruck hin zum DOM-Wurzelelement [vgl. 5, S.202].

Der Algorithmus durchläuft die angegebenen Funktionen in folgender Reihenfolge: transfConvertStar, transfAddID, transfAddText, transfAddAttribute, transfAddAttributeSet, transfAddPosition und transfAddLevel (vgl. 5, S.184). Jede Funktion besitzt eine Vorbedingung. Die erste Funktion bspw. wird solange ausgeführt, bis ein sogenanntes tag-Element aus einer Menge *L* am Zielknotet ermittelt worden ist. Das kann eine Zeile *tr* oder eine Spalte *td* einer Tabelle sein, ein Eingabeelement *input* oder weitere tag-Elemente aus der Menge L. Im Idealfall besitzt der Zielknoten bereits ein nutzbares tag-Element, andernfalls wird das nächstgelegene tag-Element am Zielknoten, wie das folgende Beispiel veranschaulicht, genutzt:

Idealfall (N=1): //input

Weitere Möglichkeiten (N=3): //table/tr/td

N stellt dabei die Länge von xp dar.

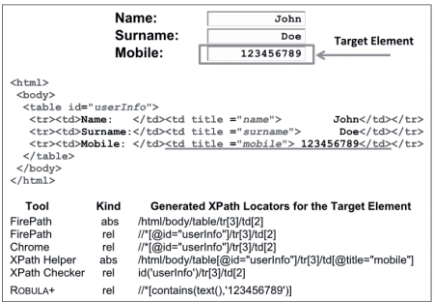

Abbildung 3: Knotenvergleich Robula+ und vergleichbaren Werkzeugen [5, S. 182]

Die Suche der Funktion transfConvertStart beginnt immer mit dem Anfangsknoten *//**. Der Asteriskus wird während des Verarbeitungsvorgangs mit einem gültigen tag-Element aus der Menge L ersetzt, bis der gesuchte Teilausdruck, wie im unteren Beispiel zu sehen, ermittelt wurde (vgl. 5, S.185).

Ablauf: *//* →//*/td→//*/tr/td →//table/tr/td*

Nachdem die Funktion transfConvertStar abgearbeitet ist, erfolgt die Bestimmung einer eindeutigen ID durch die Funktion transfAddID. Die Vorbedingung für transfAddID ist erfüllt, sofern der aktuelle Ausdruck xp an der Stelle N keine eindeutige Aussage enthält, die für die Identifikation des Zielelements ausreicht. Diese Vorbedingung gilt für alle weiteren Funktionen bis auf die beiden letzten Funktionen transfAddPosition und transfAddLevel. Falls eine eindeutige ID existiert, wird diese als Aussage dem Ausdruck xp hinzugefügt und das Ergebnis (z.B. *//input[@id='btn1']*) ausgegeben. Die darauf folgende Funktion transfAddText fügt das Attribut *[contains(text(), ...)]* hinzu, falls keine ID existiert. Nach diesem Schema arbeitet auch die Funktionen transfAddAttribute. Ist bis zu diesem Zeitpunkt kein eindeutiger Ausdruck zustande gekommen, folgt die Funktion transfAddAttributeSet. Im Gegensatz zu transfAddAttribute werden hier mehrere Attribute dem Ausdruck xp hinzugefügt, wie im Beispiel *//input [@name='data' and @class='value']*.

Anzumerken ist, dass nicht nur zu N_{Max} Attribute hinzugefügt werden können, sondern auch in niedrigeren Level. Aus dem undefinierten XPath-Ausdruck xp = *//tr/td* mit N=2 wird bspw. der eindeutige Ausdruck *//tr[@class='example']/td* auf N_{Max-1} erzeugt. Für diesen Umstand sorgt die letzte Funktion transfAddLevel am Ende dieses Abschnitts.

Die Funktion transfAddPosition fügt, sofern notwendig, die entsprechende Position des Zielelements dem bestehenden Ausdruck xp hinzu. Ist das Zielelement in einer Zelle enthalten, welche nicht eindeutig über ein Attribut identifiziert werden kann, so wird die Position (hier:*[2]*) der Zelle dem Ausdruck hinzugefügt wie im Beispiel *//tr[@class='example']/td[2]*.

Ist bis zur letzten Funktion transfAddLevel kein eindeutiger XPath-Ausdruck gefunden, so fügt transfAddLevel dem Ausdruck xp einen weiteren Level hinzu. Das unten aufgeführte Beispiel verdeutlicht den gesamten Vorgang.

1. Start des Algorithmus (xp=//* ; N=1)
2. Aufruf transfConvertStar (xp=//td ; N=1)
3. Durchlaufen aller Funktionen bis auf transfAddLevel und transfAddPosition (xp=//td ; N=1)
4. Aufruf transfAddPosition (xp=//td[2] ; N=1)
5. Aufruf transfAddLevel (xp=//*/td[2] ; N=2)
6. Aufruf transfConvertStart (xp=//tr/td[2] ; N=2)
7. Aufruf transfAddID (xp=//tr[@id='name']/td[2] ; N=2)
8. Ausgabe xp=**//tr[@id='name']/td[2]**

Robula+ erlaubt darüber hinaus eine Anpassung seiner verwendeten Attribute. Wie eine Anpassung möglich ist, wird im nächsten Abschnitt beschrieben.

5.2 Blacklist und Prioritätsliste

Robula+ nutzt zur Ausschließung bzw. Priorisierung von HTML-Attributen, welche innerhalb der Funktionen transfAddAttribute und transfAddAttributeSet genutzt werden, eine Blacklist bzw. eine Prioritätsliste. Attribute, welche in der Blacklist auftreten, sind nie Teil des von Robula+ erzeugten XPath-Ausdrucks. Die in der Prioritätsliste enthaltenen Attribute dagegen sind, je nach Position in der Liste, stärker oder schwächer priorisiert.

Durch die Blacklist und die Prioritätsliste ist der Robula+ Algorithmus skalierbar. Mithilfe der Blacklist kann Robula+ Attribute ausschließen, welche nicht bzw. weniger robuste XPath-Ausdrücke hervorbringen. Im Gegensatz zur Blacklist ist die Prioritätsliste für die Bevorzugung von Attributen notwendig. Standardmäßig sind die Attribute *name, class, title, alt* und *value* in der Prioritätsliste enthalten. Diese Reihenfolge ergibt sich aus den Evaluierungsergebnissen der Robula+-Entwickler, wobei das ID-Attribut, unabhängig von der Liste, den höchsten Stellenwert einnimmt [vgl. 5, S.188].

Mit Hilfe beider Listen ist Robula+ auf automatisch generierte DOM-Strukturen abstimmbar. Eine Webseiten-Baukastensoftware nutzt bspw. bevorzugt bestimmte Attribute oder Attributs-Konstellationen. Je nachdem ob diese Attribute stabile oder ständig wechselnde Identifikationsmerkmale besitzen, sind sie entweder der Prioritätsliste oder der Blacklist hinzuzufügen.

Trotz der Listen ist es für Robula+ nicht immer möglich einen zuverlässigen XPath-Ausdruck zu erzeugen. Wann Probleme eintreten beschreibt der nächste Abschnitt.

5.3 Analyse der Schwächen des Robula+-Algorithmus

Aufgrund der mannigfaltigen Anzahl unterschiedlicher DOM-Strukturen und den automatisch generierten Attributen von Web-Elementen ist es nahezu unmöglich, für jedes zu lokalisierende Web-Element, einen stabilen und zugleich einfachen XPath-Ausdruck zu erzeugen. Dem Robula+-Algorithmus ist es bspw. nicht möglich, automatisch generierte IDs zu ignorieren

oder nur stabile Bestandteile der ID für die Erzeugung des XPath-Ausdrucks zu nutzen.

Beispiel 1- Prioritätsproblem:

HTML: *<input class="r12465f3" headers="saldo"></td>*

Robula+: *//input[@class='r12465f3']*

Stabiler Ausdruck:
//input[@headers='saldo']

Im ersten Beispiel besitzt das Web-Element ein generiertes, eindeutiges class-Attribut und ein eindeutiges header-Attribut. Aufgrund der höheren Priorität des class-Attributes wird das header-Attribut ignoriert. Nur im Fall eines nicht-eindeutigen class-Attributes wird das stabile header-Attribut von Robula+ verwendet.

Beispiel 2: Problem der automatisch generierten (kryptischen) ID:

HTML: *<td id="table2631ffa5:saldo">... </td>*

Robula+: *//td[@id='table2631ffa5:saldo']*

Stabiler Ausdruck:
//tr[3]/td[contains(@id,'saldo')]

Im zweiten Beispiel besitzt das Web-Element ein automatisch generiertes, eindeutiges ID-Attribut. Gewünscht ist die Lokalisierung der *saldo*-Spalte der dritten Zeile einer Tabelle. Der Robula+-Ausdruck würde bereits nach einer erneuten automatischen Generierung der ID nicht mehr funktionieren, da der generierte Anteil nicht aus dem XPath-Ausdruck entfernt bzw. ignoriert wird. Ferner ist aufgrund der Eindeutigkeit der ID keine spezifische Zeile im Robula+-Ausdruck enthalten.

Beispiel 3 – Blacklist-Problem:

HTML:

```
<div class="clearfix successBox">
  <button class="span1"
  href="/booking">
</div>
```

Robula+: *//div[@class=successBox']*
/button[@class='span1']

Stabiler Ausdruck:
//button[contains(@href,' ,'booking')]

Durch den Eintrag des href-Attributes in die Blacklist, beachtet Robula+ dieses Attribut im dritten Beispiel nicht weiter. Trotz der Eindeutigkeit des href-Attributs und des daraus wesentlich kürzeren XPath-Ausdrucks erzeugt Robula+ einen deutlich längeren und instabileren Ausdruck.

6 Verbesserung von Robula+

Dieser Abschnitt beschäftigt sich mit der Verbesserung von Robula+. Ziel ist die Generierung stabilerer XPath-Ausdrücke und die damit einhergehende Wartungsreduktion der Web-Oberflächentests.

6.1 Automatische Bewertung der Prioritäten

Der gegenwärtige Robula+-Algorithmus nutzt eine starre Liste für die Ermittlung der bevorzugten Attribute zur Generierung der XPath-Ausdrücke. Um das geschilderte Problem aus Abschnitt 5.3 (Beispiel 1) zu lösen, muss Robula+ in der Lage sein die vorhandenen Attribute auf ihre Beständigkeit überprüfen zu können. Besitzt ein Attribut mit hoher Priorität, wie beispielsweise *class*, einen automatisch generierten, kryptischen Anteil im Inhalt, so wird (sofern vorhanden) das Attribut mit der nächsthöheren Priorität gewählt, welches keinen derartigen Inhalt besitzt. Ob eine kryptische Zeichenfolge vorliegt, ist durch den Abgleich mit einem Wörterbuch möglich. Enthält jedes priorisierte Attribut kryptische Anteile, so wird dieser im XPath-Ausdruck, wie im nächsten Abschnitt beschrieben, ignoriert.

6.2 Kryptische Zeichenfolgen in Attributen ignorieren

Falls alle Attribute kryptische Anteile enthalten, erfolgt ein weiterer Wörterbuchabgleich mit dem am höchsten priorisierten Attribut. Auf diese Weise ist die Vermeidung des kryptischen Teils im XPath-Ausdruck möglich. Der erneute Abgleich mit dem Wörterbuch wird durch eine Zwischenspeicherung des Ergebnisses während des

ersten Abgleichvorgangs aus Performancegründen vermieden. Geht durch das Entfernen der kryptischen Zeichenfolge die Eindeutigkeit verloren, so werden durch die Methode *transfAddAttributeSet* oder *transfAddLevel* weitere Attribute bzw. weitere Level dem XPath-Ausdruck hinzugefügt.

6.3 Lösen des Blacklist-Problems

Das im dritten Beispiel des Abschnitts 5.3 beschriebene Blacklist-Problem lässt sich, ohne den Sinn der Blacklist ad absurdum zu führen, nicht ohne weiteres umgehen. In diesem Fall sind die Gründe der Eintragung eines Attributes in die Blacklist essentiell. Handelt es sich bspw. um ein Attribut, welches unerwünschte Pfadangaben enthält, ist es sinnvoller eine Methode in Betracht zu ziehen, die den irrelevanten Teil des Pfades entfernt, statt das Attribut in die Blacklist einzutragen. Schlussfolgernd sind Attribute in die Blacklist einzutragen, welche durch weitere Textfiltermethoden nicht nutzbar gemacht werden können oder sich mit jedem Release vollständig ändern.

7 Fazit

Die Motivation dieser Arbeit liegt in der Minimierung des Aufwands bei der Erstellung und Wartung automatischer Web-Oberflächentests. Beantwortet wurden die Fragen wofür diese Art von Test benötigt wird und wie sie technisch realisiert werden. Nachfolgend ist auf die Probleme der Erstellung und Wartung automatischer Tests eingegangen worden und im Zuge dessen, erfolgte die Vorstellung diverser Werkzeuge wie bspw. der Selenium IDE. Abschließend wurde Robula+ vorgestellt, der aktuell fortschrittlichste Algorithmus zur automatischen Erzeugung von XPath-Ausdrücken [vgl. 5, S.180]. Dabei ist auf die Schwächen des Algorithmus eingegangen worden sowie auf die Lösung der vorgestellten Probleme.

8 Literatur

[1] T. Bucsics, M. Baumgartner, R. Seidl, S. Gwihs: *Basiswissen Testautomatisierung – Konzepte, Methoden und Techni*ken; 2. Auflage; dpunkt.verlag GmbH, Heidelberg, ISBN 978-3-86490-024-2, 2015

[2] U. Gundecha: *Selenium Testing Tools Cookbook*; 2. Auflage; Packt Publishing Ltd., Birmingham, Vereinigtes Königreich, ISBN 978-1-78439-251-2, 2015

[3] J. Robie, M. Dyck, J. Spiegel: *XML Path Language (XPath) 3.1*; World Wide Web Consortium (MIT, ERCIM, Keio, Beihang), 2015;https://www.w3.org/TR/xpath-31 (Abruf 13.09.16)

[4] P. Tonella, F. Ricca, A. Marchetto: *Recent advances in web testing – Advancen in Computers*; 93:1–51. DOI:10.1016/ b978-0-12-800162-2.00001-4, 2014

[5] M. Leotta, A. Stocco, F. Ricca, P. Tonella: *ROBULA+: an algorithm for generating robust XPath locators for web testing*; Journal of Software: Evolution and Process, Universität Genova, Italien, DOI:10.1002/smr.1771, 2016

[6] K. Franz: *Handbuch zum Testen von Web-Applikationen*; Springer-Verlag Berlin Heidelberg, Deutschland, ISBN 978-3-540-24539-1, 2007

[7] R. Nörenberg: *Effizienter Regressionstest von E/E-Systemem nach ISO 26262*; KIT Scientific Publishing, Karlsruhe, Deutschland, ISBN 978-3-86644-842-1, 2012

[8] S. Kleuker: *Qualitätssicherung durch Softwaretest*; Springer Fachmedien, Wiesbaden, Deutschland, ISBN 978-3-8348-0929-2, 2013

[9] M. Grechanik, Q. Xie, C. Fu: *Maintaining and evolving GUI-directed test scripts*; Proceedings of the 31st International Conference on Software Engineering, ICSE 2009, IEEE, 408–418, DOI:10.1109/ icse.2009.5070540, 2009

[10] B. Beizer: Software Testing Techniques; 2. Auflage.VanNostrand Reinhold Co., New York, USA, ISBN 0-442-20672-0, 1990

[11] M. Leotta, D. Clerissi, F. Ricca, C. Spadaro: Comparing the maintainability of Selenium WebDriver test suites employing different locators - A case study; 1st International Workshop on Joining AcadeMiA and Industry Contributions to testing Automation, Jamaica, Seiten 53–58, ISBN: 978-1-4503-2161-7, 2013

Einsatzmöglichkeiten von Affective Computing zur Optimierung von Arbeitsumgebungen *

Saskia Heine
Reutlingen University
Saskia.Heine@Student.
Reutlingen-University.DE

Abstract

Die Arbeit stellt die Einsatzmöglichkeit von Systemen zur Erkennung und Reaktion auf Gemütslagen von Benutzern im Kontext der Optimierung von Arbeitsumgebungen dar. Zu diesem Zweck wird zunächst auf Messmöglichkeiten von Stress- und Belastungszuständen eingegangen, da deren Ermittlung die Grundvoraussetzung solcher Systeme bildet. Daraufhin werden ergonomische Methoden und Konzepte mit der Zielsetzung, Arbeitsbedingungen für den Menschen zu verbessern, gezeigt. Im nächsten Schritt wird der aktuelle Stand der Technik beschrieben, auf dessen Basis dann ein weiterer Ansatz erarbeitet wird, der im darauffolgenden Schritt in den Kontext der Ergonomie eingeordnet wird. Abschließend erfolgen eine Zusammenfassung der Arbeit sowie ein Ausblick auf zukünftige Entwicklungen in diesem Bereich.

Schlüsselwörter

Affective Computing, Ergonomie, Emotionsmessung, Arbeitsplatzgestaltung

CR-Kategorien

J.3 [Computer Applications]: Life and medical sciences, Health

1 Einleitung

Emotionen gelten als essentielle Bestandteile der menschlichen Kommunikation, da deren Vermittlung zu größeren Teilen der Informationsübertragung beiträgt als der gesprochene Inhalt. Nachdem dieser Aspekt in der Informatik lange Zeit vernachlässigt wurde, gab es in den letzten Jahren im Bereich Affective Computing diverse Entwicklungen, die darauf abzielen, eine Emotionserkennung vornehmen zu können. Ziel hierbei ist es, dass der Computer ein Verständnis für die Stimmungslage seines Benutzers erlangt und somit in der Lage ist, ihn in höherem Maß in der Bedienung zu unterstützen [6, S.424]. Es ist also nicht länger erstrebenswert, dass ein Computer schlicht praktische Aufgaben erledigen soll. Vielmehr soll er sich in die emotionale Natur und Gewohnheit des Menschen integrieren, wodurch die Akzeptanz beim Nutzer gesteigert werden soll [19, S.15].

Affective Computing ist ein Bestandteil des Gebiets KI (Künstliche Intelligenz). Es gibt verschiedene Ansprüche, die an ein System mit künstlicher Intelligenz gestellt werden. Einen davon formulierte Winston wie folgt:

---

*
Betreuer Hochschule: Prof. Dr.-Ing. Michael Tangemann
Hochschule Reutlingen
Michael.Tangemann@Reutlingen-University.de

Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Saskia Heine

"The study of the computations that make it possible to perceive, reason, and act" [8, S.68]

Dies bedeutet, dass ein solches System Personen bei der Entscheidungsfindung unterstützen oder sogar selbständig Probleme lösen soll. Grundvoraussetzung hierfür ist die Fähigkeit solcher Programme, gewisse Situationen zu erkennen und auf Basis dieser Erkenntnis Empfehlungen für weiteres Handeln oder selbständige Aktionen zuordnen zu können [7].

2 Erkennungsmöglichkeiten von Befindlichkeitszuständen

Innerhalb dieses Kapitels werden Möglichkeiten zur für Affective Computing essentiellen Erkennung von psychischen und physischen Befindlichkeiten von Benutzern in ihrer Vorgehensweise vorgestellt sowie anhand ihrer spezifischen Vorzüge und Nachteile eingeordnet. Die Erkennung dieser Befindlichkeiten lässt sich dabei auf den in Abbildung 1 gezeigten typischen Ablauf darstellen.

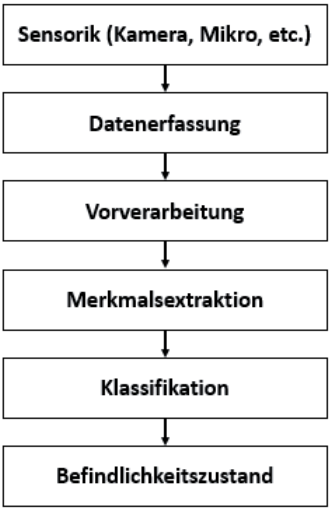

Abbildung 1: Ablauf der automatischen Emotionserkennung [6, S.425]

Die verschiedenen Möglichkeiten, die im Nachfolgenden dargestellt werden, stellen in Abbildung 1 die Variabilität der verwendeten Sensorik dieser Ablaufkette dar.

2.1 Akustische Erkennung

Durch die gegebene Komplexität ist es zur Emotionserkennung aus Sprache wichtig, viele Merkmale zu berücksichtigen, die eine Repräsentation einer Emotion darstellen können. Bei der akustischen Erkennung ist das am häufigsten berücksichtigte Charakteristikum der Merkmalsextraktion (in Abbildung 1 als vierter Schritt aufgeführt) die sogenannte Prosodie einer Sprache. Diese setzt sich aus der Intensität, Intonation sowie der Dauer einer Sprachsequenz zusammen [18, S.45]. Die Intensität entspricht der physikalischen Analyse des Schalldrucks der Sprachsequenz und ist sowohl ein komplexer als auch umstrittener psychoakustischer Zusammenhang zwischen der Empfindung der Lautstärke eines Tons sowie der Frequenz und Dauer. Die Problematik hierbei ist der Mangel an Informationen bezüglich der Richtung und Distanz der Aufnahme. Die Intonation wird als Sprachgrundfrequenz betrachtet, welche durch die akustisch wahrgenommene Tonhöhe einer Sprachsequenz beschrieben wird. Durch die Intonation wird somit eine Aussage über die Tonhöhe innerhalb eines Sprachakts getroffen, die sich jedoch nicht kulturell übertragen lässt, da Aussprache und Betonungen von Worten und Sätzen kulturell bestimmt sind und somit die Intonation beeinflussen. Zuletzt wird bei der Prosodie die Dauer der Sprachsequenz anhand einer Näherung aus der Länge stimmhafter Abschnitte geschätzt. Im nächsten dargestellten Schritt aus Abbildung 1 erfolgt eine Klassifikation, die durch Analyse der Merkmale der Prosodie auf eine Emotion schließt. Klassifikationen können hierbei unter anderem anhand von Entscheidungsbäumen, künstlichen neuronalen Netzen oder Support-Vektor-Maschinen erfolgen. [6, S.427]

Die akustische Erkennung bietet den Vorzug eines berührungs- und sensorapplikationsfreien Messvorgangs.

Darüber hinaus wird das Gefühl einer zu starken Überwachung durch die Verwendung einer Mikrofonaufnahme im Vergleich zur visuellen Informationsextraktion vermieden. Außerdem wird davon ausgegangen, dass durch zukünftig noch vermehrt eingesetzte sprachgesteuerte Mensch-Maschine-Schnittstellen die Verfügbarkeit verwertbarer Sprachäußerungen ansteigt [18, S.2]. Dem gegenüber stehen jedoch neben den bereits erwähnten Schwierigkeiten der Prosodie stark vom Anwendungsgebiet abhängige mögliche Einschränkungen. So kann es je nach Einsatzgebiet etwa unüblich beziehungsweise unnatürlich sein, während der Bedienung einer Maschine zu sprechen. Überdies können störende Hintergrundgeräusche die Sprachaufnahme sowie die Präzision der Erkennung stark beeinträchtigen.

2.2 Optische Erkennung

Bei der optischen Erkennung wird die Mimik eines Benutzers analysiert. Die Mimikerkennung setzt sich aus den beiden Bereichen des herkömmlichen 2D-Verfahrens und des immer stärker eingesetzten 3D-Verfahrens zusammen. 2D-Verfahren bedienen sich einer zweidimensionalen geometrischen Vermessung der charakteristischen Gesichtsmerkmale wie Augen, Augenbrauen, Mund oder Nase. Die jeweiligen Positionen und Abstände dieser Merkmale dienen als Grundlage zur Bestimmung einer Emotion. Hierfür wird im ersten Schritt das zu analysierende Bild nach einer hautfarbenen Region abgesucht. Innerhalb dieser detektierten Region werden dann die Gesichtsmerkmale wie die Augen, die die dunkelsten Punkte in dieser Region sind, oder die Nase, die den hellsten Punkt darstellt, bestimmt. Stehen diese Merkmale fest, ist es ein Leichtes, weitere Merkmale wie Augenbrauen oder Mundwinkel zu finden. Die Winkel- und Distanzverhältnisse der bestimmten Merkmale bilden daraufhin die Grundlage zur Klassifizierung einer Emotion. Ein Beispiel für dieses Verfahren liefert die am Fraunhofer-Institut für Integrierte Schaltungen IIS in Erlangen entwickelte Bibliothek SHORE. SHORE ermöglicht neben der Emotionserkennung ebenfalls die Bestimmung des Geschlechts sowie das ungefähre Alter einer Person [6, S.426], [10].

Das 2D-Verfahren ist mit Hilfe handelsüblicher Kameras möglich. Dem gegenüber steht das neuere 3D-Verfahren, das auf die dreidimensionale Erfassung, beispielsweise mittels Streifenprojektion, setzt und anhand dessen durch zusätzliche Informationen eine höhere Genauigkeit der Erkennung sowie bessere Posenunabhängigkeit erreicht werden soll [6, S.426].

Wie bei der akustischen Erkennung besteht auch bei der optischen Erkennung der Vorteil eines Messvorgangs, der keine Anbringung von Sensoren an den Personen erfordert, da lediglich eine Kamera vonnöten ist. Allerdings besteht auch bei diesem Verfahren eine starke Abhängigkeit der Umgebung. Unzureichende oder stark wechselnde Lichtverhältnisse erschweren die Erkennung erheblich. Außerdem ist es notwendig, dass der Benutzer sich permanent im Fokus der Kamera befindet, was einerseits je nach Anwendungsszenario die Bewegungsmöglichkeit einschränken und andererseits ein unangenehmes Gefühl von Überwachung beim Benutzer hervorrufen kann.

2.3 Physiologische Signale

Neben der akustischen und optischen Erkennung von Befindlichkeitszuständen besteht die Möglichkeit der Erkennung anhand von Vitalparametern. Hierbei werden Sensoren am Benutzer angebracht, die beispielsweise die Hautleitfähigkeit, die Atemfrequenz, die Muskelanspannung, die Gehirnaktivität oder auch die Herzratenvariabilität messen. Ein im Jahr 2008 entwickelter Prototyp von Jonghwa Kim und Elisabeth André führt eine automatische Emotionserkennung anhand von verschiedenen Vitaldaten aus. Im ersten Schritt bestand das Projekt von Kim und

André aus einem Testlauf, bei dem drei verschiedene männliche Testpersonen im Alter zwischen 25 und 38 Jahren zunächst vier Musiktitel auswählten, die ihrer Meinung nach jeweils zu einer der Emotionen Freude, Ärger, Traurigkeit und Zufriedenheit führten. Der Testlauf umfasste eine Dauer von drei Monaten, währenddessen jedes Mal, wenn die Probanden einen der von ihnen gewählten Songtitel hören wollten, physiologische Signale, die Aufschluss über die Muskelanspannung (EMG), die Herzratenvariabilität, die Hautleitfähigkeit und die Atemfrequenz geben, erfasst wurden. Aus den Ergebnissen des Testlaufs wurde daraufhin ein Algorithmus entwickelt, der die Erkennung der Emotion auf Basis der gemessenen Vitaldaten automatisch vornimmt. Das Experiment von Kim und André hat gezeigt, dass ein direkter Zusammenhang zwischen einer erlebten Emotion und den physiologischen Eigenschaften des Körpers besteht [16, S.115]. Die Genauigkeit der Klassifizierung wird dabei durch die Auswahl der Sensoren beeinflusst. So ist beispielsweise die Temperatur ein schlechter Indikator für die Erkennung des Befindlichkeitszustands, da sich diese nur sehr träge verändert. [6, S.429] Gegenüber dem akustischen und dem optischen Verfahren zur Bestimmung von Gemütszuständen hat die Erkennung anhand von Vitaldaten den Nachteil, dass die Sensoren am Körper des Benutzers angebracht sein müssen oder je nach Art der Sensorik zumindest in ständigem Kontakt mit dieser sein müssen. Ein Beispiel für Letzteres ist das Anbringen von Sensoren im Lenkrad oder auch Sitz eines Fahrzeugs. Ein Konzept hierfür hat etwa das FZI Forschungszentrum Informatik in Kooperation mit der Daimler AG erarbeitet. Die in Sitz und Lenkrad integrierten Sensoren sind in der Lage, EKG und Puls, Hautwiderstand, Blutsauerstoffgehalt und Fingertemperatur zu messen. [11] Voraussetzung für eine permanente Messung ist hierbei jedoch die Einhaltung einer festgelegten Position der Hände am Lenkrad, sodass die verbauten Sensoren Vitaldaten erfassen können. Am Körper des Benutzers angebrachte Sensoren etwa in Form eines Handschuhs oder auch Armbands, auch „Wearables“ genannt, haben dagegen den Vorteil, dass die Bewegungsfreiheit des Benutzers nicht eingeschränkt wird und dass im Gegensatz zur optischen sowie akustischen Erkennung Umgebungsbedingungen wie Lichtverhältnisse oder Störgeräusche die Messung nicht beeinflussen.

2.4 Multimodale Erkennung

Die bisherige Darstellung der verschiedenen Möglichkeiten zur Erkennung des Befindlichkeitszustands eines Benutzers zeigt, dass jedes Verfahren seine individuellen Vorzüge und Schwachstellen aufweist. Insbesondere kann festgehalten werden, dass unter bestimmten Umgebungsbedingungen die Genauigkeit der Erkennung beeinträchtigt werden kann. Daher entstand die Methode der multimodalen Erkennung, deren Ziel eine höhere Erkennungsgenauigkeit anhand der Verwendung mehrerer Eigenschaften ist. Genau betrachtet wird diese Methode auch vom Menschen selbst bei der Interaktion mit anderen verwendet, da ein Mensch zur Bestimmung des Befindlichkeitszustands ebenfalls mehrere Merkmale wie Stimmlage, Mimik, Körperhaltung, Gestik und in manchen Situationen ebenfalls das Erröten oder Schwitzen seines Gegenübers berücksichtigt. Durch die Verwendung mehrerer Eigenschaften kann jedoch nicht nur die Erkennungsgenauigkeit verbessert werden, sondern darüber hinaus kann auch zwischen einer größeren Auswahl von Zuständen unterschieden werden. Außerdem ist ein multimodales Erkennungssystem robuster gegenüber Ausfällen. Kann beispielsweise das optische Verfahren durch sehr ungünstige Lichtverhältnisse kein Ergebnis mehr erzielen, so kann die Erkennung aufgrund der anderen Eigenschaften weiterhin durchgeführt werden. [6, S.430]

3 Ergonomie

Dieses Kapitel stellt einige Grundlagen der Ergonomie dar. Es werden verschiedene Komponenten und Methoden vorgestellt, um eine Verständnisgrundlage für mögliche Erweiterungen durch den Einsatz von Affective Computing zu schaffen.

3.1 Aufgaben und Ziele

Die Aufgabe der Ergonomie ist einerseits die Anpassung der Arbeit an die Eigenschaften und Fähigkeiten des Menschen sowie andererseits ebenfalls die Anpassung des Menschen an die Arbeit. Ersteres soll durch die Gestaltung von Arbeitsplatz und –raum, Arbeitsmittel wie Maschinen und Werkzeuge, Arbeitsumgebung, worunter unter anderem Klima, Beleuchtung, Lärm oder Gefahrstoffe zählen, Arbeitsorganisation, -aufgabe und –ablauf sowie Arbeitsergebnis erfolgen. Die Anpassung des Menschen an die Arbeit dagegen ergibt sich durch die Auswahl von Personen mit der Anforderung entsprechenden Eigenschaften und Fähigkeiten, durch Ausbildung sowie durch Gewöhnung wie beispielsweise an bestimmte Klimaverhältnisse. Anhand dieser Anpassungen soll erreicht werden, dass der Mensch bei der Ausführung seiner Arbeit weder über- noch unterfordert ist. Dies kann erfolgen, wenn das Verhältnis von Belastung und Leistungsfähigkeit ausgeglichen ist. Unter Belastung versteht man hierbei sowohl körperliche Belastungen wie beispielsweise von Muskeln, Skelett oder Kreislauf, psychischen Belastungen sowie Umgebungs-Belastungen wie etwa Strahlung, Lärm, Klima oder Gefahrstoffe. Die Folge von Belastungen wird als Beanspruchung bezeichnet und beschreibt zum Beispiel Schmerz, Änderung der Sinnesorgane oder auch Erhöhung der Pulsfrequenz. Dabei hängt die Beanspruchung von Faktoren wie Höhe und Dauer der Belastung sowie den Eigenschaften und Fähigkeiten des Menschen ab. [5, S.6ff]

Eine optimale Beanspruchung liegt dann vor, wenn die Belastung der Arbeit genau der Leistungsfähigkeit des Menschen entspricht. Gesundheit, Motivation, Zufriedenheit und Wohlbefinden sind die aus diesem Zustand der optimalen Belastung resultierenden Ergebnisse. Ist dagegen das Verhältnis nicht ausgeglichen, so sind Unzufriedenheit, Unfälle, Beschwerden, Demotivation oder Krankheit die möglichen Folgen. Diese können sowohl bei Unterbeanspruchung, wenn die Leistungsfähigkeit die Belastung übersteigt, als auch bei Überbeanspruchung, zu der es bei zu hoher Belastung im Verhältnis zur Leistungsfähigkeit kommt, vorkommen. Man unterscheidet grundsätzlich in diesem Kontext zwischen zwei verschiedenen Arten von Belastungen, der physischen und der psychischen Belastung. Diese werden in den folgenden beiden Abschnitten näher betrachtet.

3.2 Physische Belastungen

Zu physischen Belastungen kann es insbesondere durch die Handhabung von Lasten, durch erzwungene Körperhaltungen, durch Bewegung unter ungünstigen räumlichen Bedingungen, durch häufig wiederkehrenden Einsatz kleiner Muskelgruppen oder durch überhöhten Kraftaufwand, vor allem bei handwerklichen Tätigkeiten, kommen. Es ist zu beachten, dass größere körperliche Belastungen auch dann bedeutsam sind, wenn sie selten oder nur in Ausnahmefällen auftreten, da für gewöhnlich bei seltenen Anforderungen technische Hilfsmittel mangelhaft oder gar nicht vorhanden sowie Übung und Erfahrung gering sind. Körperliche Beanspruchung betreffen hauptsächlich die drei Fälle Skelett, Kreislauf und Muskeln. Die genannten Beanspruchungen treten in der Regel gleichzeitig auf, wobei Gesundheitsschäden eher selten durch Kreislauf- und Muskelbeanspruchung verursacht werden. Dagegen sind Gesundheitsschäden durch Verschleißerscheinungen am Skelett relativ häufig. [15, S.81]

3.3 Psychische Belastungen

Durch intelligenten Technikeinsatz hat sich die früher eher durch schwere körperliche

Arbeit sowie durch schädigende Einwirkung der Umweltfaktoren gekennzeichnete Belastung in der Arbeitswelt in den letzten Jahren vermehrt zu psychomentalen und psychosozialen Belastungen hin verschoben. Vermehrt wird diese Art der Belastung oftmals auch gerade durch den Wegfall der klassischen physischen Belastung wahrgenommen. Demnach sind nicht nur ungünstige Arbeitsplatzgestaltungen oder unergonomische Arbeitsabläufe Ursachen für arbeitsbedingte Gesundheitsgefahren, sondern auch Faktoren wie Zeitdruck, Konflikte oder soziale Isolation. Darüber hinaus können ebenfalls Belastungen aus dem persönlichen Umfeld wie finanzielle Sorgen, Zukunftsängste oder Probleme mit Kindererziehung und Partnerschaft den Arbeitsprozess beeinflussen. Im Arbeitsumfeld werden psychische Belastungsfaktoren wie monotone Arbeitsbedingungen, schwierige Kunden, steigende Arbeitsmenge, ständige Überwachung, Isolation, Gruppendruck oder auch schlechte Aufstiegschancen häufig genannt. [15, S.90] Langzeitfolgen von psychischer Fehlbelastung können sich äußerst komplex äußern. So kann etwa das körperliche Befinden durch Schlafstörungen, Kopfschmerzen, Verschlechterung des allgemeinen Gesundheitszustands oder auch durch erhöhte Häufigkeit von Herz-Kreislauf-Erkrankungen beeinträchtigt werden. Außerdem kann das psychische Erleben durch Unzufriedenheit, depressive Verstimmung, Neigung zu Überreaktion oder Erleben von Kontrollverlust belastet werden. Auch durch das Verhalten des Menschen können Fehlbelastungen erkannt werden. Dies kann sowohl das Arbeitsverhalten, etwa durch Leistungsschwankungen oder –rückgang, Verlust an Können und Kompetenz, Fehlzeiten oder Krankheitstage, als auch das Sozialverhalten, beispielsweise durch Aggressionen gegenüber anderen, Rückzug, Isolation oder vermehrter Konsum von Nikotin, Alkohol oder Medikamenten, betreffen. [15, S.93]

Die besondere Herausforderung bei der Bewertung von psychischer Belastung ist die im Vergleich zur physischen Belastung noch deutlich stärker ausgeprägte Individualität der Leistungsfähigkeit, was bedeutet, dass die Fähigkeit, psychische Belastungen zu ertragen, noch deutlicher von Person zu Person abweichend ist als bei physischen Belastungen.

4 State Of The Art

Innerhalb dieses Kapitels soll der aktuelle Stand der Entwicklung im Bereich Affective Computing für die ergonomische Gestaltung von Arbeitsumgebungen anhand eines derzeit in Umsetzung befindlichen Projekts des Fraunhofer Instituts dargestellt werden.

Gegenwärtig beschäftigt sich das NeuroLab des Fraunhofer Institut für Arbeitswirtschaft und Organisation IAO im Rahmen des Projekts EMOIO, welches in Kooperation mit Partnern aus Industrie und Wirtschaft im Oktober 2015 eröffnet wurde, mit der Fragestellung, inwiefern Assistenzsysteme bei der Unterstützung der Mitarbeiter durch die Erkennung emotionaler Zustände und die entsprechende Reaktion darauf verbessert werden können. Es gilt dabei herauszufinden, wie Mitarbeiter ihren PC-Arbeitsplatz beziehungsweise die Interaktion mit unterschiedlichen Maschinen erleben, ob der Umgang mit denselben eher belastend oder motivierend wirkt und ob sie bei der Nutzung in der Lage sind, sich gut zu konzentrieren. Zu diesem Zweck wurde in der Grundlagenphase eine Studie durchgeführt, bei der circa 60 Probanden mit einem Assistenzsystem arbeiten. Während der Mensch-Technik-Interaktion werden neurophysiologische Daten in Form von Elektroenzephalographie (EEG) sowie von funktionaler Nahinfrarotspektroskopie (fNIRS) erhoben, welche darüber Aufschluss geben, wie sich die Interaktion mit dem Assistenzsystem auf das Gehirn auswirkt. Die Probanden sollten während des Tests unter Anderem verschiedene Ziele in einem Menü erreichen, wobei zufällig unterschiedliche Adaptionsverhalten der

Software eingebaut wurden. Die Ergebnisse dieses Tests zeigen, dass die Ausprägungen der unterschiedlichen Adaptionen deutlich unterschieden werden können. Nach der Grundlagenphase, in der die Daten offline analysiert werden, streben die Forscher in einem nächsten Schritt die Entwicklung eines onlinefähigen Algorithmus an, der die Reaktionen in Echtzeit zu erkennen vermag. In einer dritten Phase soll dann erforscht werden, welches sinnvolle Adaptionsverfahren das Assistenzsystem in jeweiligen Anwendungsszenarien wie etwa einer adaptiven Webschnittstelle, ein System zur Mensch-Roboter-Kooperation oder ein Fahrerassistenzsystem umsetzen sollte. [14]

5 Konzept: Unterstützung der Aufmerksamkeitsfähigkeit

In diesem Kapitel wird nun ein Konzept zur Unterstützung der Aufmerksamkeitsfähigkeit erarbeitet und vorgestellt. Zum besseren Verständnis wird zunächst der Hintergrund näher beleuchtet, woraufhin dann auf die Zielsetzung des Konzepts eingegangen wird. Daraufhin erfolgt die Darstellung einer möglichen Umsetzung.

5.1 Hintergrund

Das Auftreten von Unterbrechungen und Multitasking während der Arbeit hat sich im Rahmen zahlreicher Veränderungen in der Arbeitswelt drastisch erhöht. Sowohl die Arbeitsunterbrechung als auch das Multitasking wirken sich negativ auf die psychische Befindlichkeit eines Arbeitenden aus und werden als belastend erlebt, da durch diese hohe Anforderungen an die Aufmerksamkeit und an das Arbeitsgedächtnis gestellt werden. [2, S.5] Insbesondere Unterbrechungen und ihre Auswirkungen wurden unter dem Begriff „Interruption Sciences“ bisher in einigen Studien untersucht. Als Grundlage dieser Forschung dient das Ziel-Aktivierungsmodell von Altmann und Trafton [1], nach dem während einer Unterbrechung zwei unterschiedliche Aufgaben, die Unterbrechungsaufgabe sowie die zuvor ausgeführte, unterbrochene Aufgabe, im Arbeitsgedächtnis in Konflikt geraten. Aus diesem Konflikt können dann Leistungseinbußen resultieren. Bailey und Konstan [3] konnten in ihrer Studie etwa nachweisen, dass Unterbrechungen zu einer längeren Bearbeitungszeit der primären und der Unterbrechungsaufgabe führen und dass als Konsequenz die Wahrnehmung von Stress und Ärger ansteigt. Die „Interruption Studies“ zeigen, dass Unterbrechungen durch die Belastung des Arbeitsgedächtnisses eine höhere Arbeitsbelastung, schlechtere Leistung und negative Emotionen nach sich ziehen können. [2. S.13]

Unterbrechungen während der Arbeit können in verschiedenen Formen auftreten. Bei Bildschirmarbeitsplätzen ist eine besonders häufige Form der Unterbrechung das Erscheinen einer Benachrichtigung über eine eingehende E-Mail oder auch das Klingeln des Telefons, das einen Mitarbeiter von seiner zu erledigenden Aufgabe ablenkt.

5.2 Zielsetzung

Das Ziel dieses Konzepts ist die Unterstützung der Aufmerksamkeitsfähigkeit eines Mitarbeiters an einem Bildschirmarbeitsplatz durch ein intelligentes Assistenzsystem, das in der Lage ist, zu erkennen, wie konzentriert dieser seiner aktuellen Aufgabe nachgeht. Ermittelt das System, dass der Mitarbeiter gerade sehr konzentriert arbeitet, werden Schritte eingeleitet, sodass er in dieser Aufgabe möglichst nicht unterbrochen wird. So werden beispielsweise Pop-Ups von eingehenden E-Mails unterbunden oder eingehende Telefonanrufe nicht durchgestellt. Stellt das System fest, dass der Mitarbeiter schon sehr lange konzentriert und eventuell gestresst ist, so kann es auch eine Pause vorschlagen.

5.3 Beschreibung Konzept

Abbildung 2 stellt eine denkbare Architektur des Assistenzsystems dar.

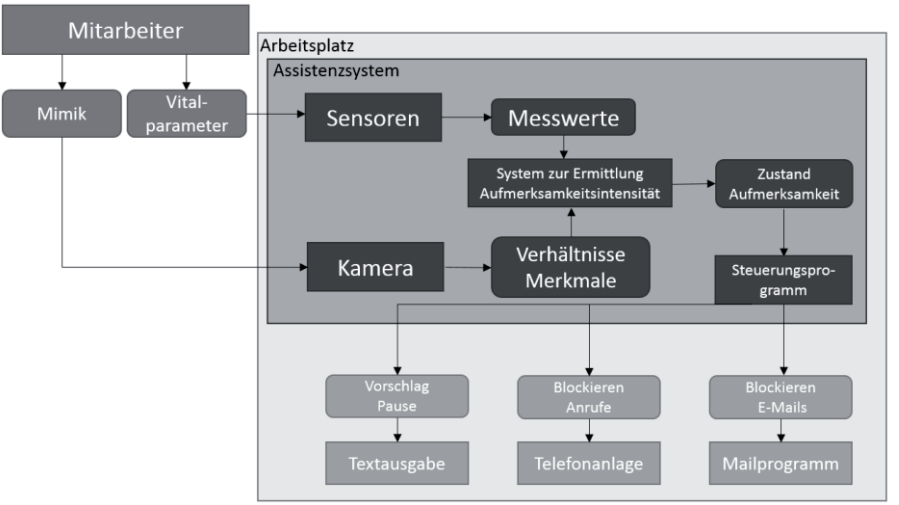

Abbildung 2: Komponenten des Gesamtsystems

Das Assistenzsystem ist Bestandteil des Systems Arbeitsplatz. Der Mitarbeiter interagiert mit dem Arbeitsplatz durch Steuerungen der Tastatur, Maus oder des Telefons. Während der Interaktion werden die Vitaldaten des Mitarbeiters ausgelesen sowie die Mimik analysiert. Es handelt sich hierbei also um eine multimodale Erkennung, wie in 2.4 erklärt. Für die Ermittlung des Aufmerksamkeitszustands eignet sich hierbei für die Messung der Vitalparameter besonders der Einsatz einer Elektroenzephalographie (EEG), wie auch im in Abschnitt 4.1 vorgestellten Projekt verwendet. Zusätzlich bietet sich bei der Art des Arbeitsplatzes ebenfalls die Verwendung einer optischen Erkennung an. Die akustische Erkennung hingegen ist in dieser Situation eher ungeeignet, da der Mitarbeiter während der Bearbeitung einer Aufgabe am Computer normalerweise eher nicht spricht. Es gibt daher wie in Abbildung 2 ersichtlich Sensoren, die die elektrische Aktivität des Gehirns messen, sowie eine Kamera, die eine Analyse der Mimik vornimmt. Die Messwerte der Elektroenzephalographie und die Winkel- und Distanzverhältnisse der Gesichtsmerkmale werden an das System zur Ermittlung der Aufmerksamkeitsintensität weitergegeben. Dieses wertet daraufhin alle erhaltenen Daten aus, speichert die Daten und bestimmt den aktuellen Zustand des Mitarbeiters. Dieser wird dann an das Steuerungsprogramm weitergegeben. Ist der Mitarbeiter gerade sehr konzentriert, so werden über das Mailprogramm Benachrichtigungen blockiert sowie über die Telefonanlage Unterbrechungen durch eingehende Anrufe unterbunden. Ist der Mitarbeiter schon sehr lange konzentriert und möglicherweise gestresst, wird ihm über eine Textausgabe am Bildschirm eine Pause vorgeschlagen.

6 Einordnung des Konzepts in den Kontext Ergonomie

In diesem Kapitel erfolgt eine Einordnung des erarbeiteten Konzepts in den Kontext Ergonomie. Das Konzept zielt darauf ab, die Arbeitsbedingungen für den Menschen angenehmer zu gestalten. Es verfolgt in diesem Kontext den Ansatz, dieses Ziel anhand der Vermeidung psychischer Belastungen durch Verhindern von Arbeitsunterbrechungen zu erreichen. Es

zielt also darauf ab, das bekannte Problem von Arbeitsunterbrechungen zu verhindern oder zumindest zu mildern. Es stellt somit einen Lösungsansatz in einem Bereich dar, für den es bisher keine Lösung gibt.

Bei dem Konzept kommt die Auswertung von Vitalparametern zum Einsatz, weshalb ein Anbringen von körpernaher Sensorik notwendig ist. Da Arbeitsunterbrechungen jederzeit auftreten und jederzeit zu negativen Emotionen führen können, ist auch eine dauerhafte Anbringung der Sensorik am menschlichen Körper nötig. Ein solches Konzept ist also nur sinnvoll, wenn die Sensorik so unauffällig wie möglich ist, da sonst die Gefahr besteht, dass das Tragen dieser als sehr unangenehm empfunden wird und sich dadurch selbst negativ auf den emotionalen Zustand des Arbeitenden auswirkt.

7 Zusammenfassung und Ausblick

Innerhalb dieser Ausarbeitung wurde im Zusammenhang von Affective Computing zunächst auf gängige Methoden zur Ermittlung von Befindlichkeitszuständen eingegangen. Im darauffolgenden Schritt wurde eine Einführung in die Ergonomie mit den Zielen und Methoden gegeben. Es wurde herausgearbeitet, dass es eine Vielzahl von Faktoren bei der Verbesserung von Arbeitsbedingungen zu beachten gilt. Außerdem wurde verdeutlicht, dass durch die individuellen Eigenschaften eines Arbeitenden auch die Arbeitsbedingungen individuell angepasst werden müssen. Dass der unterstützende Einsatz von Affective Computing an dieser Stelle sinnvoll sein kann, zeigt das in Kapitel 5 erarbeitete Konzept sowie die Tatsache, dass in diesem Bereich, wie in Kapitel 4 vorgestellt, aktuell geforscht wird. Ob und wie weit sich der Einsatz von Affective Computing in diesem Bereich durchsetzen wird, hängt wohl maßgeblich einerseits von der Erkennungsgenauigkeit des Befindlichkeitszustands und andererseits von der für den Nutzer möglichst störfreien Ermittlung desselben ab. Gelingt es, beide Faktoren zu erfüllen, eröffnen sich durch Affective Computing eine Vielzahl an Möglichkeiten, Arbeitsbedingungen für den Menschen zu verbessern.

8 Literaturverzeichnis

[1] Erik M. Altmann, J.Gregory Trafton. Memory for goals: an activation based model. Cognitive Science 26 (2002)

[2] Anja Baethge, Thomas Rigotti. Auswirkung von Arbeitsunterbrechungen und Multitasking auf Leistungsfähigkeit und Gesundheit. Bundesanstalt für Arbeitsschutz und Arbeitsmedizin 2013; ISBN: 978-3-88261-714-6

[3] Brian P. Bailey, Joseph A. Konstan. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. Computers in Human Behavior (2006). Online verfügbar unter: http://www.interruptions.net/literature/Bailey-CHB06_1.pdf Besucht am 20.08.2016

[4] Eva Bamberg, Antje Ducki, Anna-Marie Metz. Gesundheitsförderung und Gesundheitsmanagement in der Arbeitswelt. Hogrefe Verlag GmbH & Co.KG 2011; ISBN: 978-3-8409-2371-5

[5] Marcel Brand, Florian Klompmaker, Peter Schleining, Fabian Weiß. Automatische Emotionserkennung – Technologien, Deutung und Anwendungen veröffentlicht in Informatik_Spektrum_35_6_2012, Springer-Verlag 2012; DOI: 10.1007/s00287-012-0618-3

[6] BG Norddeutsche Metall-Berufsgenossenschaft. Mensch und Arbeitsplatz http://www.gesundheitimbetrieb.de/fileadmin/upload/downloads/BGI_523_Me

nsch_und_Arbeitsplatz.pdf
Besucht am 09.08.2016

[7] Prof. Dr. Gerhard Brewka. Materialien zur Vorlesung Wissensbasierte Systeme - Einführung Online verfügbar unter http://www.informatik.uni-leipzig.de/~brewka/lehre1.html
Besucht am 12.09.2016

[8] Ulrike Brüggemann. Modellierung der Alltagsorganisation. University of Bamberg Press 2009; ISBN: 978-3-923507-42-9

[9] Christoph Busch, Alexander Nouak. Das EU-Projekt „3D Face“. 3-D-Gesichsterkennung für die unbeaufsichtigte Grenzkontrolle http://3dface.org/files/papers/busch-BSI2007-3dface-project-overview.pdf
Besucht am 12.09.2016

[10] Fraunhofer Institute IIS, SHORETM – Pioneering Facial Analysis. Website. Online verfügbar unter http://www.iis.fraunhofer.de/content/dam/iis/de/doc/il/bsy/bs/shore_pioneering_facial_analysis.pdf
Besucht am 03.08.2016

[11] FZI Forschungszentrum Informatik, Lenkrad mit integrierter Vitalsensorik. Website 2009 Online verfügbar unter http://www.insitex.de/indexe50f.html?option=com_content&view=article&id=96&Itemid=149
Besucht am 04.08.2016

[12] Dr. Ulrich Glitsch. Grundlagen der ergonomischen Arbeitsgestaltung http://www.lak-nds.net/lak20130905/vortraege/12_glitsch/glitsch.pdf
Besucht am 11.08.2016

[13] Britta Hofmann. Einführung in die ISO 9241-10. Website, 2005. Online verfügbar unter http://www.fit-fuer-usability.de/archiv/iso9241-10/;
Besucht am 12.9.2016

[14] Innovisions. Damit Technik uns individuell assistiert – Emotionserkennung für eine menschengerechte Arbeitsgestaltung. Online verfügbar unter http://innovisions.de/beitraege/volltext/1/damit-technik-uns-individuell-assistiert/
Besucht am 18.08.2016

[15] Hans-Jürgen Kiepsch, Christian Decker, Gudrun Harlfinger-Woitzik. BGI 523: Mensch und Arbeitsplatz http://elbdeich.net/pdf/bgi523%20Menschen%20am%20Arbeitsplatz.pdf
Besucht am 15.08.2016

[16] Jonghwa Kim, Elisabeth André. Emotion Recognition Based on Physiological Changes in Music Listening. In: IEEE TPAMI 30(12) 2008; DOI: 10.1109/TPAMI.2008.26

[17] Reinhard Koether, Bernhard Kurz, Uwe A. Seidel, Franz Weber. Betriebstättenplanung und Ergonomie – Planung von Arbeitssystemen. Carl Hanser Verlag München 2010; ISBN: 978-3-446-42598-9

[18] Jarek Krajewski. Akustische Stimmmessung – Automatische Detektion von Befindlichkeitszuständen. W. Sarges & D. Scheffer 2008. Online verfügbar unter http://www.wirtschaftspsychologie-wuppertal.de/fileadmin/wieland/lehrstuhl/Krajewski/Krajewski__J.__2008_._Akustische_Stimmmessung___Automatische_Detektion_von_Befindlichkeitszustaenden._In_W._Sarges___D.pdf
Besucht am 12.9.2016

[19] Paolo Petta, Catherine Pelachaud, Roddy Cowie. Emotion-Oriented Systems – The Humaine Handbook. Springer-Verlag Berlin Heidelberg 2011; ISBN: 978-3-642-15183-5

Bildregistrierungsverfahren und ihre Genauigkeit bei Current Imaging-Daten

*

Dipl.Ing.(FH) Gregor Renka
Hochschule Reutlingen
Gregor.Renka@Reutlingen-University.de

Abstract

Für die Fehleranalyse in der Halbleiterfertigung gewinnt das Current Imaging immer mehr an Bedeutung. Für ein neues Current Imaging System existiert bislang kein Verfahren um Bilder aus einem Rasterelektronenmikroskop und denen des Current Imaging zu überlagern. Um damit korrelative Mikroskopie betreiben zu können, wird ein Bildregistrierungsverfahren benötigt, welches bei dieser Art von Daten gute Ergebnisse in einer akzeptablen Zeit erzielt. Als einen Schritt in diese Richtung, wurde in dieser Arbeit die Funktionsweise rigider und nicht-rigider Registrierungsverfahren betrachtet und die Genauigkeit ihrer Arbeitsweise bei Current Imaging Bildern untersucht. Als Referenz dienten manuell registrierte Bilder. Die Ergebnisse des Vergleichs sollen bei der Wahl eines geeigneten Registrierungsverfahrens helfen. Das ausgewählte Verfahren soll Teil eines automatisierten Prozesses werden, um die Fehleranalyse zu vereinfachen.

Schlüsselwörter

Current Imaging, Bildregistrierung, Template Matching, Korrelative Mikroskopie

1 Einleitung

In der Halbleiterfertigung stellt jede Verkleinerung der Strukturen die Fehleranalyse vor neue Herausforderungen. Ihre Durchführung wird immer aufwendiger und zeitraubender. Die Anwendung korrelativer Mikroskopie mit einem Rasterelektronenmikroskop (REM) und einem in situ Current Imaging System, erlaubt es mehr Informationen in kürzerer Zeit zu gewinnen.

Beim Current Imaging (CI) wird ein kleiner Bereich der Halbleiterprobe mit einer an einem sogenannten Mikromanipulator[1] montierten Messspitze gescannt. Dabei kann eine Spannung zwischen Probe und Spitze oder zwischen zwei Stellen auf der Probe angelegt werden (Abbildung 1)[11]. Welche Variante benutzt wird, hängt im wesentlichen von der Probe ab und der Art des Fehlers, den man sucht. Beim scannen, fährt die Messspitze über Bereiche die leitend sind oder nicht. Durch die angelegte Spannung resultiert daraus ein Stromfluss, der als bildgebendes Signal benutzt wird. Je nach Polarität der angelegten Spannung, lassen sich so fehlerhafte P-N-Übergänge[2] finden.

Zu den enormen zeitlichen Vorteilen des in

*

Betreuer Hochschule: Prof. Dr.-Ing. Cristobal Curio
Hochschule Reutlingen
cristobal.curio@reutlingen-university.de

Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Dipl.Ing.(FH) Gregor Renka

[1] Roboter mit drei Achsen, der Schritte im Nanometer-Bereich ausführen kann

[2] P-N-Übergänge entstehen wenn positiv und negativ leitendes Material (dotiertes Silizium) zusammengebracht wird. Dazwischen entsteht durch Rekombination eine elektrisch neutrale Schicht. Diese wird als P-N-Übergang bezeichnet.

situ Current Imaging, kommt der Informationsgewinn durch die räumliche Zuordnung des gescannten Bereichs zum REM-Bild hinzu, wenn beide Bilder registriert werden. Für diese Art von Bildern existiert bislang keine Lösung für die Registrierung.
Als einen weiteren Schritt in Richtung einer Lösung des Registrierungsproblems der CI-Bilder, wird in dieser Arbeit die Genauigkeit verschiedner Registrierungsverfahren miteinander verglichen. Es handelte sich dabei um echte CI-Bilder aktueller und zukünftiger Halbleitertechnologie. Der Vergleich soll dazu dienen das am besten geeignete Registrierungsverfahren für diese Art von Bilddaten zu finden.

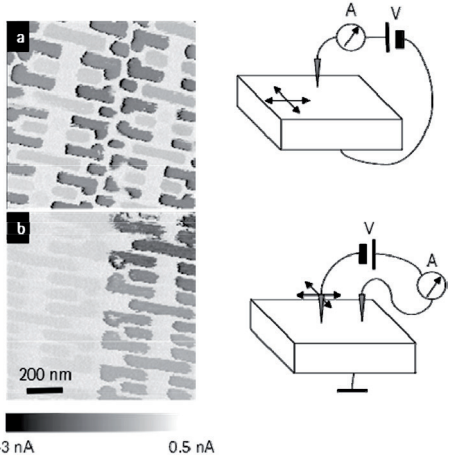

Abbildung 1: Stromfluss bei Verwendung unterschiedlicher Pfade: (a) Stromquelle am Bulk-Signal, Lineartisch scannt, während die stationäre Messspitze den Stromfluß misst. (b) Stromquelle an der scannenden Messspitze, Bulk und zweite stationäre Messspitze dienen als Stromsenke. Der Stromfluß wird über die zweite Messspitze gemessen [11]

2 Einsatzbereiche für Bildregistrierung

Das Haupteinsatzgebiet der Bildregistrierung findet sich in der medizinischen Bildverarbeitung. Dort müssen häufig Bilder aneinander angepasst werden, die mit verschiedenen bildgebenden Verfahren (Modalitäten) aufgenommenen wurden. Wie bereits im Kapitel 1 beschrieben, erhöht sich durch die Überlagerung der Bilder der Informationsgewinn. So werden in der Medizin z.B. häufig MRT-Bilder mit PET-Bildern überlagert [7]. Auf MRT-Bildern werden Weichteilgewebe gut abgebildet, während auf PET-Bildern bestimmte Stoffwechselprozesse gut sichtbar gemacht werden können. Durch ihre Überlagerung lässt sich genauer bestimmen, in welchem Bereich, z.B. im Gehirn, ein bestimmter Stoffwechselprozess stattfindet.
Im Life-Science-Bereich gewinnt die Bildregistrierung, mit der immer häufiger eingesetzten korrelativen Mikroskopie, zunehmend an Bedeutung. Immer bessere Lichtmikroskope, erlaubten es Forschern lange Zeit das Innere von Zellen zu erforschen. Mittlerweile ist die Zellforschung an einem Punkt angelangt, wo Lichtmikroskope nicht mehr die benötigte Vergrösserung erreichen können. Rasterelektronenmikroskope (REM, engl. Scanning electron microscopes, SEM) haben die nötige Vergrößerung und so entstand die korrelative Mikroskopie [19]. Lichtmikroskope können große Bereiche abbilden und sind bei lebenden Zellen einsetzbar. Damit eignen sie sich gut um innerhalb einer Zelle Bereiche von Interesse zu finden. Rasterelektronenmikroskope werden dann eingesetzt um Teile der interessanten Bereiche hochaufgelöst darzustellen.
Ein weiteres Anwendungsgebiet ist die Montage einzelner Satellitenbilder zu einem Gesamtbild [7] [22]. Aufgrund der Erdkrümmung und der von Bild zu Bild wechselnden Position des Satelliten, entstehen Verzerrungen, die dann mit Registrierungsverfahren aneinander angeglichen werden.
Grundsätzlich kann man sagen, dass sich die Bildregistrierung auf folgende drei Arten von Bilddaten anwenden lässt:

- Bilder, die das gleiche Objekt zeigen, aber aus verschiedenen Positionen aufgenommen wurden (multiview).

- Bilder, die das gleiche Objekt zeigen, aber zu verschiedenen Zeitpunkten aufgenommen wurden (multitemporal).

- Bilder, die das gleiche Objekt zeigen, aber mit verschiedenen Sensoren (Bildgebungsverfahren) aufgenommen wurden (multimodal).

In den meisten Fällen handelt es sich um multimodale Daten. So ist es auch beim Current Imaging. Für die CI-Bilder existiert kein gemeinsames Koordinatensystem oder andere Hilfen, die es erlauben würden, die Bilder aus beiden Geräten schnell und einfach zu registrieren.
Die CI-Bilder stellen die Registrierung noch vor zwei weitere Herausforderungen. Aufgrund der Funktionsweise des Current Imaging, ist in einem CI-Bild nie das gleiche zu sehen, wie im REM-Bild. Das REM bildet alle sichtbaren Strukturen auf der Oberfläche einer Probe ab, während das CI nur die Bereiche sieht, in denen ein Stromfluß gemessen werden kann (Abbildung 1)[11].
Bedingt durch den Piezoantrieb des Mikromanipulators, der für die Aufnahme der CI-Bilder benutzt wird, treten an den Rändern der CI-Bilder Verzerrungen auf. Piezos sind Kristalle, die sich ausdehnen oder zusammenziehen, wenn eine Spannung an sie angelegt wird. Man spricht vom inversen Piezoeffekt [13]. Bei jeder Richtungsumkehr während des Scan-Vorgangs muss die im Piezo aufgebaute Spannung im Kristallgitter erst abgebaut werden. Das wird als Hysterese bezeichnet. Der Spannungsabbau während der Richtungsumkehr sorgt für eine unregelmäßige Bewegung der Messspitze, die dann im Bild als Verzerrung am Rand zu sehen ist.

3 Verwendete Registrierungsverfahren

3.1 Rigide Verfahren - merkmalsbasiert

Merkmalsbasierte Registrierungsverfahren gelten als schnelle Verfahren. Die Registrierung basiert auf der Zuordnung lokaler Merkmale. Ihre Anzahl ist im Verhältnis zur Anzahl der Pixel im Bild gering und damit auch der Rechenaufwand [20][5].
Als lokales Merkmal kann alles dienen, was sich von seiner direkten Umgebung unterscheidet. Als Unterschiede gelten Helligkeit, Farbe und Struktur. Ecken, Kanten und runde Strukturen sind somit typische Merkmale, die gut zu detektieren sind [1][5].
Die wichtigste Eigenschaft eines guten Merkmals ist seine eindeutige Position im Bild. Diese Position darf sich bei Betrachtung aus unterschiedlichen Perspektiven bei Rotation und Skalierung nicht ändern. Eine weitere wichtige Eigenschaft ist die Ausgeprägtheit des Merkmals. Bei einem Gebäude z.B. wird ein rundes Fenster zwischen rechteckigen Fenstern immer zuverlässig erkannt. Lokale Merkmale erlauben es damit Korrespondenzen in Bildern zu finden, selbst wenn die darin enthaltenen Objekte teilweise verdeckt sind, aus verschiedenen Perspektiven aufgenommen wurden oder Störungen vorhanden sind.
Detektoren für lokale Merkmale nutzen Änderungen der Bildhelligkeit oder der lokalen Steigung für die Detektion. Unter lokaler Steigung ist eine gerichtete Änderung der Helligkeit oder Farbe zu verstehen.
Zu jedem gefundenen Merkmal gehört ein sogenannter Deskriptor [5]. Dieser wird aus den Bereichen um das Merkmal berechnet. Deskriptoren sind im Grunde Vektoren, die die Bereiche um die Merkmale herum repräsentieren. Sie erlauben den Vergleich der Merkmale und der sie umgebenden Bereiche, unabhängig von Skalierung und Rotation.
Bevor die eigentliche Registrierung beginnen kann, müssen die in beiden Bildern korrespondierenden Merkmale gefunden wer-

den, die höhst wahrscheinlich zueinander gehören. Das Finden der Merkmalspaare ist damit ein Optimierungsproblem, denn es gilt die beste Lösung aus mehreren möglichen Lösungen zu finden. Wie in [14] beschrieben, wurden viele Algorithmen zur Lösung des Problems veröffentlicht, aber nicht wie man für das eigene Problem den geeigneten Algorithmus auswählt und seine Parameter einstellt. Der am häufigsten verwendete Ansatz ist die Suche nach dem nächstliegenden Nachbarn (Nearest Neighbor Search oder Proximity Search). Hierfür wurden bereits verschiedene Lösungen vorgeschlagen. Die einfachste ist eine lineare Suche. Vom Merkmal in einem Bild wird die Distanz zu allen Merkmalen im anderen Bild ermittelt. Die kürzeste Distanz zwischen zwei sich ähnelnden Merkmalen ist die Lösung [2].
Wenn ausreichend Merkmalspaare gefunden wurden, wird die Registrierung mit Hilfe einer geometrischen Transformation abgeschlossen [10]. Dabei wird aus den zueinander gehörenden Merkmalen in beiden Bildern eine Transferfunktion berechnet. Seien die Koordinaten der Merkmale in beiden Bildern,

$$(x_i, y_i) \text{ und } (X_i, Y_i) \text{ mit } i = 1, ..., n \qquad (1)$$

dann bestimmt die geometrische Transformation eine Transferfunktion, die aus den Teilen f_x und f_y besteht (2).

$$\begin{aligned} X_i &\approx f_x(x_i, y_i) \\ Y_i &\approx f_y(x_i, y_i) \end{aligned} \quad \text{für} \quad i = 1, ..., n \qquad (2)$$

Transferfunktionen werden in folgende Kategorien eingeteilt:

- Similarity - unterstützt Translation, Rotation und lineare Skalierung
- Affine - unterstützt Translation, Rotation und nicht-lineare Skalierung
- Projective - unterstützt Translation, Rotation, nicht-lineare Skalierung und Verkippung

Für die Bestimmung der Transferfunktionen wurde die Kategorie Similarity verwendet. Sie ist auch bekannt als Transformation kartesischer Koordinaten (3). Dabei stellen s, γ und (h, k) die Unterschiede in Skalierung, Rotation und Translation der Bilder dar.

$$\begin{aligned} X &= xs\cos\gamma - ys\sin\gamma + h \\ Y &= xs\sin\gamma + ys\cos\gamma + k \end{aligned} \qquad (3)$$

Für die Bestimmung der Parameter h und k wird die Prozedur "Least Median of Squares"genutzt. Sie beginnt mit zufällig gewählten Werten für h und k.

1. Finde für jeden Punkt im REM-Bild den Abstand zum nächstgelegenen Punkt im CI-Bild, nachdem es mit den Parametern h und k transformiert wurde.
2. Ordne die (quadrierten) Abstände für alle Punkte im REM-Bild aufsteigend.
3. Ist der Median-Wert der Abstände innerhalb der geforderten Toleranz, stoppe an dieser Stelle. Das bedeutet, dass mit den aktuellen Werten für h und k mindestens 50% der Punkte nach der Transformation innerhalb der geforderten Toleranz liegen. Andernfalls wähle einen neuen Punkt aus beiden Bildern aus, wähle neue Werte für h und k und gehe zu Punkt 1.

3.2 Rigide Verfahren - flächenbasiert

Die flächenbasierten Registrierungsverfahren sind auch als Korrelationsverfahren, intensitätsbasierte Verfahren oder Template Matching bekannt. Dabei werden für die Registrierung keine Merkmale gesucht sondern Bildbereiche oder die kompletten Bilder verwendet [22].
Das bekannteste flächenbasierte Verfahren ist die Kreuzkorrelation. Sie vergleicht die Intensitäten in beiden Bildern, ohne den Inhalt zu analysieren. Dadurch ist sie aber anfällig auf Änderungen der Intensität (z.B. verursacht durch Rauschen), schwankende Helligkeit und die Nutzung verschiedener Sensoren zur Bildaufnahme.

Die normierte Kreuzkorrelation wird daher häufiger verwendet, da sie gegenüber der Kreuzkorrelation ein paar Vorteile aufweist [12]. Formel 4 zeigt wie die Werte der Korrelationsmatrix c bei der Kreuzkorrelation berechnet werden. I ist das Bild in dem nach Template T gesucht wird. Die Summe ist über x, y im Bereich mit dem Template an der Position u, v. Es ist leicht erkennbar, dass die Werte in c abhängig von der Helligkeit der Bilder sind.

$$c(u,v) = \sum_{x,y} I(x,y)T(x-u,y-v) \quad (4)$$

Verändert sich die Energie (5) im Bild I mit der Position, kann das Matching fehlschlagen. Das exakte Matching zwischen dem Template und einem Bildbereich zum Beispiel, kann einen niedrigeren Wert ergeben als das Matching zwischen dem Template und einem hellen Bildbereich [12].

$$\sum I^2(x,y) \quad (5)$$

Dem kann man mit der normierten Kreuzkorrelation entgegenwirken. Durch die Normierung werden die Werte in der Korrelationsmatrix c auf einen festen Wertebereich von -1 ... +1 abgebildet (6). Dabei ist $\overline{I}_{u,v}$ der Mittelwert der Region unter dem Template und $\overline{T}$ der Mittelwert des Templates. Die Position des größten Wertes in der Korrelationsmatrix c entspricht der Position mit dem besten Match im Bild. Das überlagern der beiden Bilder muss in einem separaten Schritt erfolgen.
Bedingung für die Verwendung der normierten Kreuzkorrelation ist, dass das Bild in dem das Template gesucht wird, größer ist als das Template. Das Template darf auch nicht aus nur einem Wert bestehen, d.h. keine einheitliche Helligkeit besitzen. Damit ist die normierte Kreuzkorrelation nicht auf beliebige Bilder anwendbar und bedarf einer Vorverarbeitung der Bilder. Das Matching selbst folgt einer festen Prozedur:

1. Abhängig von der Bildgröße wird die Kreuzkorrelation in der Raum- oder Frequenz-Domäne berechnet
2. Lokale Summen werden durch Berechnung des Integralbildes berechnet [12]
3. Die lokalen Summen werden für die Normierung der Kreuzkorrelation verwendet (6)

Ein automatisches Bildregistrierungsverfahren arbeitet direkt mit den kompletten Bildern. Es handelt sich um einen iterativen Prozess (Abbildung 2), der die Einstellung folgender Parameter erfordert, bevor die Registrierung beginnen kann:

- Transformationstyp - Definiert die Art der Transformation
  - Monomodal: Beide Bilder mit dem selben Sensor aufgenommen.
  - Multimodal: Bilder mit unterschiedlichen Sensoren aufgenommen.
- Metrik - Ähnlichkeitsmaß
  - Mutual Information (MI): Maß für die statistische Abhängigkeit zweier Variablen, bzw. die Menge an Information, die ein Bild über das andere enthält [18].
  - Mean Squares (MS): die Differenz korrespondierender Pixel in jedem Bild wird quadriert und daraus der Mittelwert berechnet.

$$c(u,v) = \frac{\sum_{x,y}\left[I(x,y)-\overline{I}_{u,v}\right]\left[T(x-u,y-v)-\overline{T}\right]}{\left\{\sum_{x,y}\left[I(x,y)-\overline{I}_{u,v}\right]^2 \sum_{x,y}\left[T(x-u,y-v)-\overline{T}\right]^2\right\}^{0.5}} \quad (6)$$

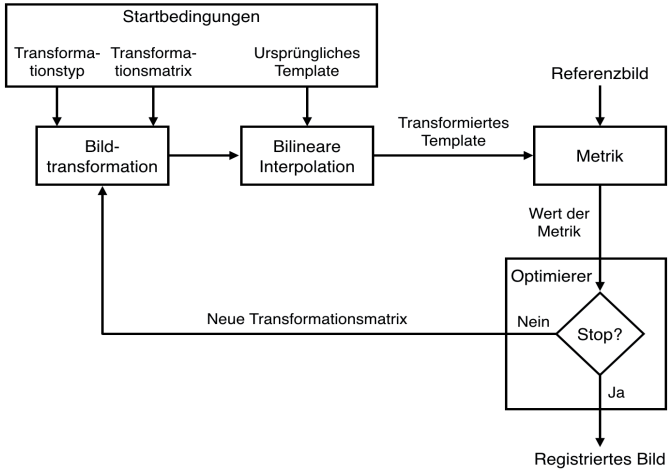

Abbildung 2: Iterationsschleife der automatischen flächenbasierten Registrierung

- Optimierer
  Enthält mehrere Parameter zur Optimierung der Registrierung, sowie die Abbruchkriterien. Die Registrierung wird abgebrochen, wenn das Ähnlichkeitsmaß (Metrik) den eingestellten Schwellwert überschritten hat oder die maximale Anzahl an Iterationen erreicht wurde.

Die erste Iteration der Registrierung beginnt mit der Transformation des Templates durch bilineare Interpolation [21], basierend auf dem zuvor eingestellten Transformationstyp und einer automatisch ermittelten Transformationsmatrix (Abbildung 2). Durch Vergleich des Referenzbildes und des transformierten Templates ermittelt die gewählte Metrik den Wert für die Ähnlichkeit. Der Optimierer überprüft dann das Abbruchkriterium. Die Registrierung wird beendet, wenn die maximale Anzahl der Iterationen erreicht wurde oder der Wert für die Ähnlichkeit sein Maximum erreicht hat. Trifft nichts davon zu, erstellt der Optimierer eine neue Transformationsmatrix und die nächste Iteration beginnt.

3.3 Nicht-rigide Verfahren

Eine der ersten Anwendungen der nicht-rigiden Registrierung war das Erkennen sich verändernder Objekte, wie das menschliche Gesicht [9] oder handgeschriebene Buchstaben [8]. Darauf aufbauend, entwickelten Bajcsy und Broit [3] ein effizienteres Verfahren für 2D und 3D Bilder.
Nicht-rigide Registrierungsverfahren sind iterative Verfahren, die mit den kompletten Bildern arbeiten. Sie eignen sich besonders gut, wenn Bilder registriert werden müssen, bei denen eines Verzerrungen aufweist. Sie eignen sich nicht für Bilder, die sich z.B. nur in Skalierung und/oder Rotation unterscheiden. Abhängig von den verwendeten Daten kann also eine Vorregistrierung oder Vorverarbeitung nötig sein, bevor ein nicht-rigides Verfahren verwendet werden kann.
Bei der nicht-rigiden Registrierung spricht man von einem elastischen Modell. Wie Broit in seiner Doktorarbeit [6] schreibt, ist es von der Theorie der Elastizität abgeleitet. Man muss sich das Bild als ein elastisches Gewebe vorstellen, auf das externe Kräfte gegen interne Kräfte einwirken. Die externen Kräfte sind die, die bei der Registrierung

verändert werden um das Bild lokal zu verändern und so bestmöglich an das Referenzbild anzugleichen. Die internen Kräfte beruhen auf einem Elastizitätsmodell. Es ist ein mathematisches Modell, das beschreibt wie sich feste Objekte durch äußere Krafteinwirkung verformen. Die bekannteste Kenngröße ist das Elastizitätsmodul E (7), mit σ als Kraft/Fläche und ε als Dehnung.

$$E = \frac{\sigma}{\varepsilon} = konstant \tag{7}$$

Das Ergebnis der Registrierung muss man sich wie die Verformung eines elastischen Materials vorstellen. Als Verformung gilt biegen, strecken und stauchen.
Bei der Registrierung wird ein regelmäßiges Gitter über das Referenzbild gelegt. Das entspricht dem erwähnten elastischen Gewebe. Dieses Gitter wird durch die externen Kräfte, die aus den Unterschieden der beiden Bilder berechnet werden und die internen Kräfte, entsprechend verformt (Abbildung 3). Die Kreuzungspunkte sind dabei die Angriffspunkte für die Kräfte. Das verformte Gitter stellt am Ende die Zuordnung des einen Bildes zum anderen Bild dar [3]. Da dieses Verfahren bei größeren Unterschieden deutlich sichtbare Verzerrungen im Bild verursacht, wurde in [4] ein verbessertes Verfahren vorgestellt, welches eine hierarchische Verarbeitungsstruktur aufweist. Dabei wird statt einem Gitter mit mehreren Gittern gearbeitet. Das Verfahren beginnt mit einem groben Gitter. Jeder Kreuzungspunkt wird so lange verschoben, bis das lokale Maximum des gewählten Ähnlichkeitsmaßes an diesem Punkt erreicht ist. Das so ermittelte Zwischenergebnis bildet den Startpunkt für die nächste Verarbeitungsstufe, bei der ein feineres Gitter benutzt wird. Dieser Ablauf wiederholt sich so lange, bis alle Stufen (Hierarchien) durchgelaufen sind.

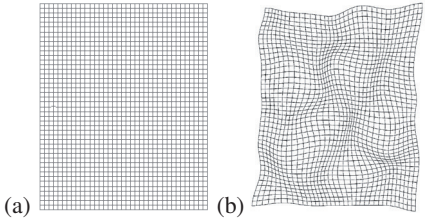

Abbildung 3: Nicht-rigide Registrierung (a) Gitter vor der Registrierung (b) Gitter nach der Registrierung [15]

4 Daten und Methoden

Als Daten für den Vergleich dienten drei Bildpaare (A,B,C) unterschiedlicher Proben. Sie wurden mit dem gleichen REM und CI-System aufgenommen und so gewählt, dass möglichst viele der Abbildungsvarianten der CI-Bilder abgedeckt wurden. Bei den Proben handelte es sich um Halbleiterproben aus aktueller Produktion und Prototypen neuer Technologien. Informationen über Hersteller, Strukturgröße und Produktart fallen unter die Geheimhaltung und dürfen nicht genannt werden.
Als Testumgebung wurde Matlab R2016a mit dem Computer Vision Packet verwendet. Das Computer Vision Packet stellt gängige merkmals- und flächenbasierte Funktionen für die rigide Registrierung zur Verfügung. Für die nicht-rigide Registrierung bietet Matlab als einzige Funktion 'Imregdemons' an, daher wurde die Medical Image Registration Toolbox (MIRT) von Andriy Myronenko installiert [17][16][15]. Sie erlaubt die nicht-rigide Registrierung auf Basis der folgenden Ähnlichkeitsmaße durchzuführen:

- Correlation Coefficient (CC)
  Dimensionsloses Maß für den Grad des linearen Zusammenhangs, also der Stärke des Zusammenhangs zweier Bilder.

- Mutual Information (MI)
  Dimensionsloses Maß für die statistische Abhängigkeit zweier Variablen, bzw. die Menge an Information, die ein Bild über das andere enthält [18].

- Residual Complexity (RC)
  Maß für die Komplexität des verblei-

benden Störbildes zweier registrierter Bilder [16].

- Sum of Absolute Differences (SAD)
  Einfachstes Ähnlichkeitsmaß. Bildet die Differenzen aus Pixeln zweier Bilder und summiert ihre absoluten Werte auf. Je ähnlicher die Bilder sind, umso kleiner der Wert.

- Sum of Squared Differences (SSD)
  Bildet die Differenzen der Pixel zweier Bilder, wie bei SAD, quadriert sie und summiert sie dann auf.

Alle Funktionen wurden mit den gleichen Daten und Einstellungen getestet. Manche Funktionen erforderten ein anderes Datenformat als Input. Die Daten wurden dafür entsprechend konvertiert.
Um die Ergebnisse der einzelnen Registrierungsverfahren vergleichen zu können, wurden sie mit einer Referenz-Registrierung verglichen, die manuell erstellt wurde. Der Vergleich wurde mit der normierten Kreuzkorrelation durchgeführt. Als Maß für die Ähnlichkeit diente der maximale Wert der Korrelationskoeffizienten aus der Korrelationsmatrix. Eine 1 entspricht dabei voller Korrelation, eine 0 bedeutet keine Korrelation.

5 Ergebnisse

Die Tabellen 1 und 2 zeigen die ermittelten normierten Korrelationskoeffizienten für die verwendeten Registrierungsverfahren und Datensätze. Auffällig ist, dass für die merkmalsbasierte rigide Registrierung (Tabelle 1) keine Ergebnisse vorliegen. Der Grund ist eine, für die Registrierung, zu geringe Anzahl korrespondierender Merkmale. Die Ursache dafür liegt im Datensatz. Wie im Kapitel 3.1 beschrieben, werden Merkmale gesucht, die ausgeprägt sind und eine feste Position haben. Das trifft auf die verwendeten Daten nicht zu, da sie Strukturen beinhalten, die sich in regelmäßigen Abständen wiederholen. Damit ist diese Bedingung nicht erfüllt. Dazu kommt, dass nicht alle im REM-Bild gefundenen Merkmale auch im CI-Bild gefunden werden. Wie im Kapitel 2 beschrieben, bildet das CI-System nur die Strukturen ab, wo ein Stromfluss gemessen werden kann (Abbildung 1). Das sorgt für den Wegfall weiterer Merkmale. Bei den anderen rigiden Verfahren liefert die normierte Kreuzkorrelation für jeden Datensatz das beste Ergebnis. In Abbildung 4 ist das Ergebnis der normierten Kreuzkorrelation für einen Datensatz als Beispiel dargestellt.
Bei der nicht-rigiden Registrierung (Tabelle 2) sind die Ergebnisse nicht mehr so eindeutig. Gut erkennbar ist, dass die Matlab-Funktion `Imregdemons` die schlechtesten Ergebnisse erzielt. Die MIRT-Funktion liefert gute Ergebnisse bei den Ähnlichkeitsmaßen Correlation Coefficient (CC) und Residual Complexity (RC).

	A	B	C
Merkmale	—	—	—
Normxcorr2	0.977	0.705	0.898
Imregister (MI)	0.942	0.692	0.879
Imregister (MS)	0.949	0.568	0.876

Tabelle 1: Korrelationskoeffizienten aus dem Vergleich der rigiden Registrierung

	A	B	C
Imregdemons	0.758	0.586	0.376
MIRT2D (CC)	0.904	0.812	0.743
MIRT2D (MI)	0.899	0.816	0.724
MIRT2D (RC)	0.868	0.830	0.749
MIRT2D (SAD)	0.896	0.820	0.735
MIRT2D (SSD)	0.898	0.817	0.747

Tabelle 2: Korrelationskoeffizienten aus dem Vergleich der nicht-rigiden Registrierung

6 Diskussion

Die Ergebnisse dieses Vergleichs zeigen, dass nicht jedes Registrierungsverfahren für diese Art von Daten geeignet ist. Es existiert auch kein ideales Registrierungsverfahren, welches zur Lösung des Problems führt. Bei der nicht-rigiden Registrierung hängt es

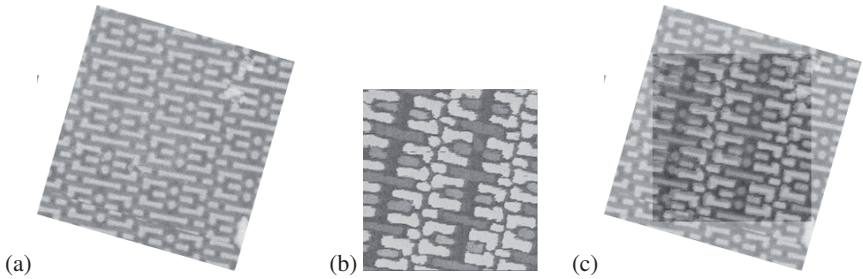

Abbildung 4: Beispiel einer Registrierung (a) Referenzbild (b) Template (c) Mit normierter Kreuzkorrelation registriertes Bild

vermutlich vom Datensatz ab, welches Ähnlichkeitsmaß die besseren Ergebnisse erzielt. Um das zu prüfen, sind weitere Untersuchungen mit mehr Datensätzen notwendig.
Die Untersuchung der verschiedenen Registrierungsverfahren hat gezeigt, dass alle ihre Vor- und Nachteile haben. Rigide Verfahren passen Skalierung, Rotation und Translation gut an, während die nicht-rigiden Verfahren sehr gut Verzerrungen korrigieren können. Da die Current Imaging Daten Verzerrungen enthalten und bei der Registrierung mindestens skaliert und rotiert werden müssen, macht es Sinn eine Kombination aus einem rigiden und nicht-rigiden Verfahren zur Lösung des Problems in Betracht zu ziehen.
Die hier gewonnen Erkenntnisse werden in die begonnene Arbeit in [19] einfließen, um so die bestmögliche Lösung für das Registrierungsproblem der REM- und CI-Bilder zu finden.

7 Zusammenfassung

In dieser Arbeit wurde ein neues Current Imaging System vorgestellt, welches die Fehleranalyse in der Halbleiterentwicklung beschleunigen und vereinfachen kann. Die gängigen Bildregistrierungsverfahren wurden vorgestellt. Es wurde gezeigt vor welche Herausforderungen die Current Imaging Daten die Registrierungsverfahren stellen. Anhand von drei Datensätzen aktueller Halbleiterproben, wurde die Genauigkeit der Registrierung, durch den Vergleich mit manuell registrierten Daten, für jedes Verfahren bestimmt.
Die normierte Kreuzkorrelation lieferte bei der rigiden Registrierung die besten Ergebnisse. Bei der nicht-rigiden Registrierung waren es die Ähnlichkeitsmaße Correlation Coefficient (CC) und Residual Complexity (RC). Hier wird eine Abhängigkeit von den Current Imaging Daten vermutet, daher sind weitere Untersuchungen mit mehr Datensätzen erforderlich.

8 Danksagung

Ich bedanke mich bei der Firma Kleindiek Nanotechnik GmbH. Sie stellte die Bilddaten des REM- und CI-Systems für diese Arbeit zur Verfügung.

Literatur

[1] *A Combined Corner and Edge Detector*. University of Manchester, August 1988.

[2] *A quantitative analysis and performance study for similarity search methods in high dimensional spaces*, number 24, New York, USA, 1998. VLDB Conference.

[3] R. Bajcsy and C. Broit. Matching of deformed images. *Proc. Int. Conf. Pattern Recognition*, pages 351–353, 1982.

[4] R. Bajcsy and S. Kovačič. Multiresolution elastic matching. *Computer Vi-*

sion, Graphics, and Image Processing, 46(1):1–21, April 1989.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[6] C. Broit. *Optimal Registration Of Deformed Images*. PhD thesis, University of Pennsylvania, January 1981.

[7] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(ACM 0360-0300/92/1200-0325):326–376, December 1992.

[8] D.J.Burr. A dynamic model for image registration. *Comput. Graph. Image Process.*, 15:102–112, 1981.

[9] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22:67–92, 1973.

[10] A. Goshtasby. Image Registration by Local Approximation Methods. *Image Vision Comput.*, 6(4):255–261, November 1988.

[11] M. Kemmler, A. Rummel, K. Schock, and S. Kleindiek. Correlating SEM and SPM for Nanoprobing in Failure Analysis. Kleindiek Nanotechnik GmbH, Aspenhaustr. 25, 72770 Reutlingen, Germany, 2015. IPFA.

[12] J. Lewis. Fast Normalized Cross-Correlation. *Vision Interface*, 1995.

[13] T. Melz. *Technische Piezokeramiken: Grundlagen für Auslegung und orbitalen Betrieb von Aktoren*. TU Darmstadt, 2002.

[14] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. International Conference on Computer Vision Theory ans Applications, February 2009.

[15] A. Myronenko and X. Song. Image registration by minimization of mapping complexity. *Computer Vision and Pattern Recognition*, pages 17–24, 2009.

[16] A. Myronenko and X. Song. Image registration by minimization of residual complexity. *Computer Vision and Pattern Recognition*, pages 49–56, 2009.

[17] A. Myronenko and X. Song. Intensity-based image registration by minimizing residual complexity. *IEEE Trans. on Medical Imaging*, 29(11):1882–1891, 2010.

[18] S. Rahunathan, D. Stredney, P. Schmalbrock, and B. Clymer. Image registration using rigid registration and maximization of mutual information. Poster presented at the 13th Annual Medicine Meets Virtual Reality Conference, January 2005.

[19] G. Renka. Template Matching für korrelative Mikroskopie mit Current Imaging. Arbeit aus Bildverarbeitung, July 2016.

[20] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *9th European Conference on Computer Vision*, volume 1, pages 430–443, 2006.

[21] S. Wang and K. Yang. An image scaling algorithm based on bilinear interpolation with VC++. *Techniques of Automation and Applications*, July 2008.

[22] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(ISSN: 0262-8856):977–1000, 2003.

Sicherheitsanalyse für Cloud-basierte Produkte der Telemedizin *

Damir Stažić
Hochschule Reutlingen
damir.stazic@Student.
Reutlingen-University.DE

Abstract

Die aufkommende Vernetzung in der Umwelt spiegelt sich im Bereich des *Internet of Things* wider, so sind z.B. Fitness Tracker oder vernetzte Gewichtswaagen Teil des alltäglichen Lebens. Diese Entwicklung wird sich auch im medizinischen Bereich ausbreiten, die erste Grundlage bildet das Gesetz. Diese Arbeit betrachtet ein fiktives Beispiel eines E-Health Services, welcher nach den aktuellen Gesetzen und Normen analysiert wird. Es werden die gesetzlichen und normativen Rahmenbedingungen besprochen und wie diese Einfluss auf die Entwicklung nehmen.

Schlüsselwörter

Datenschutz, E-Health, Gematik, Continuous Deployment

CR-Kategorien

D.2.1 [**Requirements/Specifications**]: Elicitation methods (e.g., rapid prototyping, interviews, JAD), Languages, Methodologies (e.g., object-oriented, structured), Tools; D.2.2 [**Design Tools and Techniques**]: Computer-aided software engineering; D.2.5 [**Testing and Debugging**]: Code inspections and walk-throughs, Diagnostics, Monitors, Testing tools (e.g., data generators, coverage testing); D.2.7 [**Distribution, Maintenance, and Enhancement**]: Documentation, Enhancement; D.2.9 [**Management**]: Life cycle, Programming teams, Software configuration management,Software process models (e.g., CMM, ISO, PSP), Software quality assurance (SQA)

1 Einleitung

Die mobile Vernetzung von Gegenständen nimmt kontinuierlich zu. Nach Gartner[16] werden im Jahr 2020 über 25 Mrd. vernetzter Geräte mit dem Internet verbunden sein. Darunter werden auch medizinische Anwendungen zum Alltag gehören[8]. In den aktuellen Entwicklungen werden zur Überwachung der Vitalparameter, Smartwatches (z.B. Apple Watch) bzw. Fitness-Tracker (z.B. Fitbit) eingesetzt. Die aufgenommenen Daten werden auf mobilen Geräten wie z.B. dem Smartphone oder über die Cloud verarbeitet und visuell für den Benutzer aufbereitet. Gerade für die Umsetzung der mobilen Telemedizin, im Bereich des Internet of Things, müssen Datenschutz und Datensicherheit nach medizinischen und technischen Standards gewährleistet werden. Als Beispiel dient ein fiktiver E-Health-Service der aus zwei Komponenten besteht, einer Dienstleistung (siehe Kap. 3) und einem Blutzuckermessgerät. Durch beide Komponenten generiert der Patient Echtzeitdaten über seinen Gesundheitszustand. Diese Daten werden bezugnehmend für die individuelle Lebensweise ausgewertet. Das ermöglicht dem Patienten eine personalisierte Medikation auf die individuelle Lebensweise sowie die Digitalisierung und Dokumentation seiner Werte. Im nächsten Schritt werden

*

Betreuer Hochschule: Prof. Dr. Schöller
Hochschule Reutlingen
Marcus.Schoeller@reutlingen-university.de
Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Damir Stažić

die Daten des Patienten über die Gematik-Schnittstelle (siehe Kap. 2.1) in die elektronische Gesundheitsakte übertragen und gespeichert. Somit stehen die aktuellen Daten den behandelnden Ärzten und Kliniken zur Verfügung.

2 Gesetzgebung

Die Digitalisierung des Gesundheitssektors in Deutschland wurde zum 1. Januar 2006 mit der Einführung der elektronischen Gesundheitskarte (eGK) durch die *Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH* (Gematik) eingeführt[6]. Als weitere Entwicklung im Gesundheitsbereich wurde das E-Health-Gesetz [5] 2015 verabschiedet. Die Telematikinfrastruktur vernetzt den Gesundheitssektor und wird in der Bundesrepublik Deutschland eingeführt, welche bis Mitte 2018 abgeschlossen sein soll. Für die Konzeption, Entwicklung und Überwachung der Umsetzung ist die Gematik[11] verantwortlich.

2.1 Gematik

Die an die Gematik übertragene Aufgaben der Konzeption, Entwicklung und Überwachung beruhen auf dem § 291b, Abs. 1, Sozialgesetzbuch (SGB) V:
"**Satz 1:** Im Rahmen der Aufgaben nach § 291a Abs. 7 Satz 2 hat die Gesellschaft für Telematik. 1. die funktionalen und technischen Vorgaben einschließlich eines Sicherheitskonzepts zu erstellen, 2. Inhalt und Struktur der Datensätze für deren Bereitstellung und Nutzung festzulegen, 3. Vorgaben für den sicheren Betrieb der Telematikinfrastruktur zu erstellen und ihre Umsetzung zu überwachen, 4. die notwendigen Test- und Zertifizierungsmaßnahmen sicherzustellen. ...; hierbei sind durch die Gesellschaft für Telematik Interoperabilität, Kompatibilität und das notwendige Sicherheitsniveau der Telematikinfrastruktur zu gewährleisten."

3 Der Anwendungsfall

Der Anwendungsfall betrachtet ein E-Health-Service (nach dem Medizinproduktgesetz), welcher von einem Patienten mobil bzw. bei sich zu Hause verwendet werden kann. Die Verwendung ist wie bei einem Blutzuckermessgerät, bei dem ein Tropfen Blut auf einem Teststreifen in das Messgerät eingeführt wird. Die gemessenen Werte werden an den Betreiberserver übertragen, die Datenübertragung erfolgt über das Internet. Dabei werden die Daten vom Betreiber in einer Private Cloud gespeichert und wieder aufbereitet. Der Benutzer kann seine Daten mit verschiedenen Geräten über das Internet abrufen und z.B. für Ärzte / Krankenhäuser freigeben. Der Anwendungsfall ist in Abb. 1 dargestellt.

3.1 Anforderung

Eine Anwendung die medizinische sowie personalisierte Daten über das Internet versendet, muss dementsprechend abgesichert sein. Mit der Absicherung sind die relevanten Gesetze sowie Normen gemeint, die eingehalten werden müssen. Die Rahmenbedingungen werden hier durch das E-Health-Gesetz [5] sowie das Bundesdatenschutzgesetz (BDSG) [3] vorgegeben. Die Gematik stellt dabei die Normative für die Umsetzung der Telematikinfrastruktur dar, diese richtet sich nach den aktuellen Gesetzen sowie Normen aus. Im Gematik-Dokument *Gesamtarchitektur* [13, Kap. 3.3.3.1.3] werden die Anforderungen für die *Anwendungen zur Wahrnehmung der Versichertenrechte* wie folgt beschrieben: „ ... Durch die Einführung der Telematikinfrastruktur darf sich die datenschutzrechtliche Position der Versicherten nicht verschlechtern. Der Versicherte hat die Datenhoheit für alle seine Anwendungen und Gesundheitsdaten in der Telematikinfrastruktur. Der Grundsatz der Freiwilligkeit der Speicherung von Gesundheitsdaten muss bewahrt werden, d.h.

- Die Versicherten müssen darüber entscheiden können, welche ihrer Gesundheitsdaten aufgenommen und welche gelöscht werden und ob und welche Daten sie einem Leistungserbringer zugänglich machen.

- Die Patienten haben das Informations-

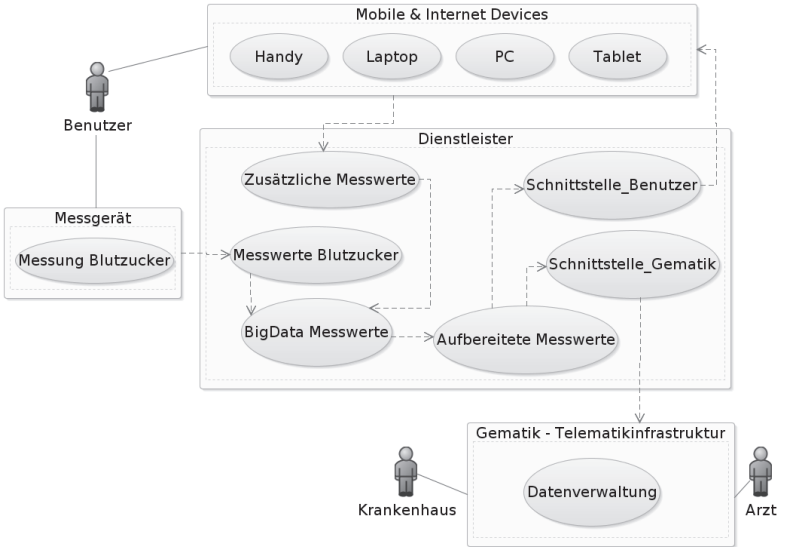

Abbildung 1: Anwendungsfall - Blutzuckermessgerät

und Leserecht über ihre gespeicherten Daten und alle diese Daten betreffenden Vorgänge.

Daher müssen die Versicherten Zugang zu einer Anwendung haben, die ihnen die Wahrnehmung dieser Rechte erlaubt. ... “ Dies spiegelt sich auch in der Anforderung für alle Dienstleistungen und Geräte, die in der Telematikinfrastruktur betrieben werden sollen, wieder. Dabei ist der Begriff „Informationssicherheit“ in der ISO 27000 [7] durch folgende Punkte definiert: **Vertraulichkeit**, **Integrität**, **Verfügbarkeit**, **Authentizität**, **Zurechenbarkeit**, **Nicht-Abstreitbarkeit**, **Verlässlichkeit**. Betrachtet man die Geräte sowie Software die einen medizinischen Hintergrund haben, so fallen diese alle unter Medizinprodukte-Richtlinie 93/42/EWG der EU (MPR) [10]. Dabei beschreibt das MPR die Anforderungen für medizinische Software wie folgt: „ ... bei Produkten, die Software enthalten oder bei denen es sich um medizinische Software an sich handelt, die Software entsprechend dem Stand der Technik validiert werden muss, wobei die Grundsätze des Software-Lebenszyklus, des Risikomanagements, der Validierung und Verifizierung zu berücksichtigen sind ...“ Dies bezieht sich auf zwei wichtige Normen, die in der Produktentwicklung eingesetzt werden müssen:

- DIN EN ISO 14971:2013-04 Medizinprodukte - Anwendung des Risikomanagements auf Medizinprodukte
- DIN EN 62304:2013-10 Medizingeräte-Software - Software-Lebenszyklus-Prozesse

Wie man erkennen kann, summieren sich die Anforderungen von der sicheren Telematikinfrastruktur bis hinunter zur medizinischen Anwendung bzw. dem Gerät, dass sich mit der TI verbindet.

3.2 Architektur

Auf die Architektur und die technische Umsetzung wird in dieser Arbeit nur oberflächlich eingegangen. Dies soll nur dem Verständnis und der Vollständigkeit dienen. Ziel

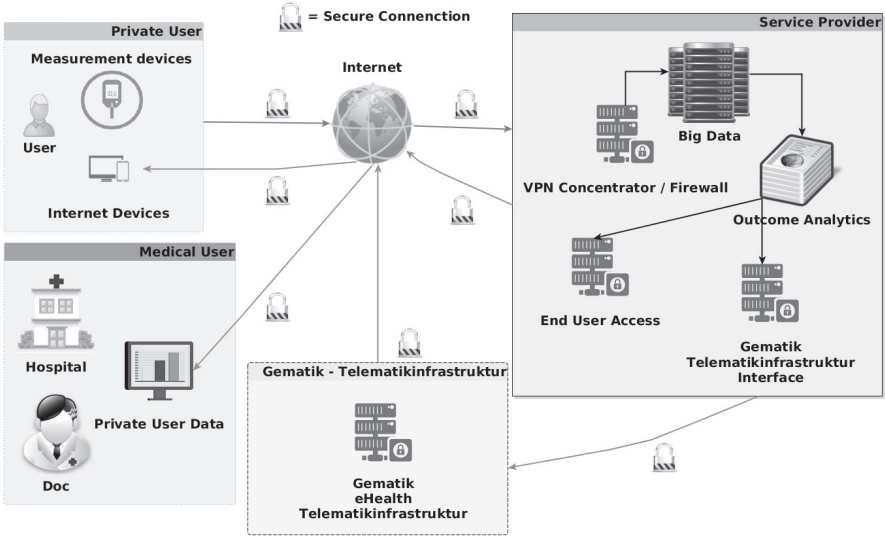

Abbildung 2: Architektur - Blutzuckermessgerät

der Architektur ist die technische Umsetzung und Einhaltung der Anforderung sowie den daraus folgenden gesetzlichen sowie normativen Vorgaben. Ein wichtiger und grundlegender Punkt für diese Architektur ist die Informationssicherheit.

3.2.1 Beschreibung

Wie in der Abb. 2 zu sehen ist, kann man die Architektur in vier Bereiche aufteilen. Der erste Bereich ist dem *Private User* zugeteilt, dieser hat Geräte die sich mit dem Internet verbinden können. Nach dem Anwendungsfall aus Kapitel 3 überträgt das Messgerät die Messwerte an den Dienstleistungsbetreiber über das Internet. Der Benutzer kann sich über seine Geräte wie PC oder Handy mit dem Dienstleister verbinden und seine Werte online betrachten bzw. für andere Personen freigeben. Im zweiten Bereich ist der Service Provider, dieser muss beim Verbindungsaufbau die „Informationssicherheit“ überprüfen. In erster Folge wird die **Authentizität** des Gerätes überprüft, darunter fällt z.B. das Indentitätszertifikat X.509 welches für die verschlüsselte Verbindung eingesetzt wird. Dieses wird auf Gültigkeit überprüft, ist diese gegeben, kann eine sichere Verbindung aufgebaut werden. Im nächsten Schritt werden die Zertifizierung und die verwendeten Protokolle vom Messgerät überprüft. Sind diese zu alt oder die Zertifizierung ungültig, kann die **Integrität** und **Verlässlichkeit** nicht sichergestellt werden. Wurde die „Informationssicherheit“ festgestellt, können die Daten vom *Private User* zum Service Provider übertragen werden. Die Daten werden vom Service Provider für den *Private User* und die Gematik aufbereitet und zur Verfügung gestellt. Der dritte Bereich ist die Gematik-Infrastruktur, diese beinhaltet unter anderem die elektronische Patientenakte (ePA). Die Gematik bietet die gesetzliche Schnittstelle für medizinische Anwendungen im Gesundheitsbereich an. Hier werden die vom *Private User* medizinischen Daten abgelegt, welche dann für medizinische Anwendungen (Hausarzt, Krankenhaus usw.) freigegeben werden können. Im vierten Bereich arbeitet der *Medical*

User (hier Hausarzt bzw. Krankenhaus) mit den vom *Private User* freigegebenen Daten. Diese Daten können zur Diagnose hinzugezogen werden.

3.3 Sicherheits-Strategie

Für die Übertragung von personalisierten Gesundheitsdaten ist eine Sicherheits-Strategie unerlässlich. Dadurch sollen die Risiken und Schadenhöhe minimiert und ein vertretbares Restrisiko ermittelt werden. Dieses Kapitel orientiert sich am Gematik-Dokument *Übergreifendes Sicherheitskonzept* [12], da dies durch Experten zusammengestellt und überprüft worden ist, stellt es die *State of the Art* Umsetzung dar. In der Abbildung 3 stellen die Pfeile die vier Hauptpunkte dar, welche zur Minimierung der Risiken eingesetzt werden. **Starke Kryptoalgorithmen** werden für die Übertragung und Speicherung von sensiblen Daten beim Dienstleister benutzt. Durch die Anwendung von Kryptoalgorithmen wird sichergestellt, dass die Daten beim Dienstleister nicht durch Dritte bzw. Einbruch gelesen werden können. Dazu werden zusätzliche technische sowie physikalische Maßnahmen ergriffen, um die Krypto-Schlüssel zu schützen. Die **geräteindividuelle Verschlüsselung** dient zur Sicherstellung, dass die Daten vom Gerät verschlüsselt und signiert werden. Der Dienstleister überprüft die Signatur und kann als einziger die Daten entschlüsseln. Sollte eine geräteindividuelle Verschlüsselung kompromittiert sein, kann diese ohne Probleme aus dem Netzwerk ausgeschlossen werden. **Proaktive Prozesse / schnelle Reaktion** dienen zur kontinuierlichen Überwachung und Verbesserung des Systems. Darunter fallen die Empfehlungen und Best practices vom Bundesamt für Sicherheit in der Informationstechnik (BSI), neue Gesetze bzw. Normen die in das System eingepflegt werden müssen. Mit der Umsetzung von **Datenvermeidung und Datensparsamkeit** werden nur die nötigsten Daten gespeichert. Dabei werden die Daten nicht zusammenhängend gespeichert, damit bei einem unberechtigten Zugriff keine

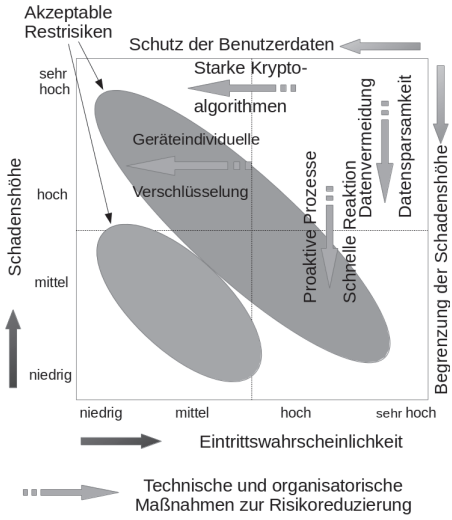

Abbildung 3: Sicherheits-Strategie, entnommen und modifiziert aus [12]

Rückschlüsse gezogen werden können. Genauer wird das durch die Gesetze *§3a BDSG* [3] und *§78b SGB X* [4] gefasst.

3.4 Einführung Sicherheitsmanagement

Unter Sicherheitsmanagement versteht man die Einführung eines Informationssicherheitsmanagementsystems (ISMS), dass nach der ISO 2700x-Normenfamilie sowie die Anwendung der IT Infrastructure Library (ITIL) umgesetzt wird. Die Begriffsbestimmung wird nach dem Gematik-Dokument *Übergreifendes Sicherheitskonzept der Telematikinfrastruktur* [12, Kap. 8] wie folgt definiert: „Sicherheitsmanagement ist die Gesamtheit von Verfahren und Prozessen zur Aufrechterhaltung der primären Schutzziele Vertraulichkeit, Integrität und Verfügbarkeit von Informationsobjekten sowie der nachgeordneten Schutzziele Authentizität, Nachvollziehbarkeit und Nichtabstreitbarkeit von Zugriffen auf Basis eines Risikoansatzes umfasst Sicherheitsmanagement die Planung, Implementierung,

Überwachung und Verbesserung von organisatorischen und technischen Maßnahmen zur Informationssicherheit.“

Wie man erkennen kann, hat die *Einführung des Sicherheitsmanagements* ihren Ursprung in der Anforderung aus Kapitel 3.1. Im **ersten Schritt** wird die Security policy definiert, hier werden die zu schützenden Informationen analysiert und bestimmt darauf wird der Umfang des Schutzes festgelegt. Zusätzlich müssen gesetzliche und unternehmerische Anforderungen berücksichtigt werden. Der **zweite Schritt** legt den Geltungsbereich fest auf dem die ISMS gelten soll. Mit den Informationen aus den ersten zwei Schritten kann das Risk Management auf die Anforderungen und Situation gestaltet werden. Die Umsetzung erfolgt in den **Schritten drei** und **vier** welche durch die ISO/IEC 27005 Informationssicherheits-Risikomanagement angeleitet wird. Die letzten zwei **Schritte fünf** und **sechs** dienen der kontinuierlichen Instandhaltung, Kontrolle und Verbesserung des ISMS. Diese werden zusätzlich in der ISO 2700x-Normenfamilie beschrieben.

4 Analyse

Die Analyse des Anwendungsfalls konzentriert sich auf den softwarebasierten Teil, die Analyse der Hardware ist nicht Teil dieser Arbeit.

4.1 Normativ

Um den Anwendungsfall wie in Abb. 1 umsetzen zu können, müssen die Regularien für medizinische Anwendungen herangezogen werden. Wie in dem Anwendungsfall beschrieben, werden Daten generiert, verarbeitet, gespeichert und können weitergeleitet werden. Für die Verarbeitung personalisierter Daten steht an erster Stelle das BDSG. Da es sich um medizinische Daten handelt gilt besonders §3 Abs. 9 [3] des BDSG welches ein hohes Schutzniveau vorschreibt.

Im Weiteren wird der Anwendungsfall (Kap. 3) als E-Health-Service beschrieben und fällt somit unter die MPR [10]. Die MPR fordert für die Entwicklung medizinischer Software-Produkte den Einsatz aktueller Technik sowie Methoden. Darunter fallen der Software-Lebenszyklus, Risikomanagement, Validierung und Verifizierung. Für diese Umsetzung werden folgende internationale bzw. harmonisierte Normen, in aktueller Fassung, herangezogen.

- DIN EN ISO 9000 Qualitätsmanagementsysteme
- DIN EN ISO 9001 Qualitätsmanagementsysteme - Anforderungen
- DIN EN ISO 13485 Medizinprodukte - Qualitätsmanagementsysteme - Anforderungen für regulatorische Zwecke
- DIN EN ISO 14971 Medizinprodukte - Anwendung des Risikomanagements auf Medizinprodukte
- ISO 31000 Risikomanagement
- DIN EN IEC ISO 31010 Risikomanagement
- DIN EN IEC 60601-1 Medizinische elektrische Geräte
- DIN EN IEC 61907 Zuverlässigkeit von Kommunikationsnetzen
- DIN EN IEC 62304 Medizingeräte-Software - Software-Lebenszyklus-Prozesse
- DIN EN IEC 62366 Medizinprodukte
- DIN EN 80001-1 Anwendung des Risikomanagements für IT-Netzwerke, die Medizinprodukte beinhalten
- BSI IT-Grundschutz [1]

Die Querverweise in den Normen wurde aufgrund des Umfangs nicht berücksichtigt.

4.2 Technisch

Aus technischer Sicht arbeiten die Anwendung (mobil/web), der Service und die Hardware (Blutzuckermessgerät) als Komponenten zusammen. Daher müssen diese dementsprechend abgestimmt sein. Für die Entwicklung von Software wird eine entsprechende Entwicklungsumgebung benötigt die bestimmte Anforderungen umsetzen kann. Durch die im fachlichen Bereich (z.B. Web, Service) erstellten Anforderungen (Requirements-Engineering) werden diese wie folgt technisch umgesetzt. Hierfür wird die Continuous Delivery (CD) Methode gewählt (Abb. 4), diese deckt einen großen Teil der fachlichen Anforderungen ab und besitzt einen hohen Automatisierungsgrad. Dadurch steigert die Zahl der Deployments auf mehrere Male pro Tag an. Das Resultat sind häufigere und schnellere Feedbacks auf neue Features und den Code. Daraus folgt, dass auch die Tests automatisiert werden müssen und dies führt zur Reproduzierbarkeit von den Tests. Wie man in der Abb. 4 sieht, können verschiedene **Teams** an einem oder mehreren Quelltexten arbeiten. Durch eine **Source Repository** können verschiedene Software-Entwicklungsstränge angelegt und versioniert werden. Dabei werden die Konfigurationen der jeweiligen Software (Applikation, Service, Embedded) je nach Einsatz und Anforderung erstellt. Der **Continuous Integration Server** (CI) führt erste automatische Buildvorgänge aus, diese beinhalten verschiedene Tests auf Funktion, Integration und Qualität. Daraus werden Reports generiert welche die Entwicklung zusätzlich dokumentieren. In einem weiteren Schritt werden nach dem CI weitere automatisierte **Tests und qualitätssichernde Maßnahmen (QA)** anhand der Anforderungen durchgeführt, hier wird zusätzlich geprüft ob diese auch umgesetzt werden. Im letzten Schritt erfolgt die Auslieferung der Software. Alle weiteren Anforderungen die vom CD nicht abgedeckt werden, müssen durch zusätzliche Tools vervollständigt und normgerecht umgesetzt werden. Dies ermöglicht eine komplette und transparente Dokumentation des Entwicklungsprozesses.

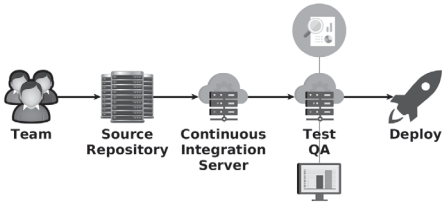

Abbildung 4: Continuous Delivery Umgebung

4.3 Qualitativ

Qualitativ steht und fällt das Software-Projekt mit dem Requirements Engineering (RE) [2, S. 439]. Das RE ist eine grundlegende Planung und Entwicklung für ein Software Projekt. Durch die Anforderung, Konzeption, Planung, Entwicklung sowie Involvierung aller beteiligten Personen (Stakeholder), wird eine Spezifikation erstellt nach welcher entwickelt wird. Die folgenden Beispiele [9, 14] sind aus [2] entnommen. Nach einer Umfrage [9] zu gescheiterten Software Projekten in den Jahren 2005 und 2007 sind folgende Aufschlüsse entstanden:

- von den Projekten sind 15,5% in 2005 und 11,5% in 2007 vorzeitig abgebrochen worden. Dabei waren die fünf wichtigsten Abbruchgründe:
- 33% keine Einbindung der Management Ebene
- 33% zu viele Änderungen der Anforderungen und des Umfangs
- 28% das Fehlen benötigter Management-Fähigkeiten
- 28% überziehen des Budgets
- 22% das Fehlen benötigter Programmier-Fähigkeiten

Für Teams die RE-Methoden einsetzen wollen, ist die Einführung von RE Best Practices sehr wichtig, da dies unnötige Fehler und Kosten verhindert [2, S. 439]. Ein

Beispiel für RE Best Practices [14] und der gewonnene Nutzen sind in der Tabelle 1 aufgeführt. Durch die konsequente und gewissenhafte Anwendung von RE Methoden wird die richtige Entwicklungsrichtung vorgegeben. Nachträgliche Anforderungsänderungen oder Implementierungen gefährden das Projekt und treiben die Kosten in die Höhe [15, S. 7]. Als Nachschlagewerke für das RE wird auf Balzert [2] und Partsch [15] verwiesen, für spezifische Informationen sollten die Normen *ISO/IEC/IEEE 29148 - System-und Software-Engineering - Lebenszyklus-Prozesse* sowie die *DIN EN IEC 62304 Medizingeräte-Software - Software-Lebenszyklus-Prozesse* herangezogen werden.

5 Fazit

Für die Software Entwicklung von medizinischen Anwendungen, seien es Applikationen oder Gerätesoftware, sind die gesetzlich vorgegebenen Rahmenbedingungen sowie Normen zu beachten. Das Verstehen der Rahmenbedingungen und Normen, hilft bei der Bestimmung von richtigen Anforderungen für das RE. Hierbei ist der zeitliche Faktor wichtig und sollte nicht unterschätzt bzw. mit ausreichend Zeit bemessen werden [2, 15]. Da medizinische Informationen über das Internet übertragen werden, fordert die Gesetzeslage eine Integration von Informationssicherheit und Datenschutz. Diese fließen in einer hohen Gewichtung in das RE ein. Die aus dem RE resultierenden Spezifikationen werden dann für die Software Entwicklung eingesetzt. Das BDSG [3] stellt klare Vorgaben zur Verarbeitung, Speicherung und Verwaltung von privaten medizinischen Daten. Dieses Gesetzt sollte nicht unterschätzt werden. Eine Entwicklungsumgebung wird nach den Spezifikationen aufgebaut und deckt einen bestimmten Bereich für Qualitätsanforderungen, Testings und Reports ab. Wichtig ist dabei die Überprüfung und Dokumentation zur Einhaltung der gesetzlichen Anforderungen in der Softwareentwicklung. Eine saubere und transparente Dokumentation der Softwareentwicklung ist für eine Zertifizierung der Software als Medizinprodukt unabdingbar.

6 Ausblick

Um eine Software Entwicklungsumgebung für medizinische Anwendungen unter Einhaltung aktueller Gesetzgebung sowie Normen auf dem neusten Stand zu halten, ist ein umfangreiches Know-How in diesen Bereichen gefragt. Ein Weg dies zu erreichen ist der Ansatz für eine zentralisierte Modellierung der Konfiguration einer Entwicklungsumgebung. Dabei kann die Modellierung über eine Domain Specific Language (DSL) erfolgen. In der DSL sollten die Anforderungen und Abhängigkeiten der Software Normen modelliert werden. Dadurch kann die gesamte Software Entwicklungsumgebung automatisiert konfiguriert und erstellt werden. Anforderungen, Testings, QA und Deploys können als Module hinzu modelliert werden, bis der gesamte Prozess die Anforderungen erfüllt. Die Zentralisierung der Konfiguration, ermöglicht die Software Entwicklungsumgebungen schneller, transparenter und konsistenter mit Modellen und Tests aufzubauen.

Literatur

[1] It-grundschutz. Web. Abgerufen am 30.05.2016 `https://www.bsi.bund.de/DE/Themen/ITGrundschutz/itgrundschutz_node.html`.

[2] H. Balzert. *Lehrbuch der Softwaretechnik Basiskonzepte und Requirements Engineering*. Springer-Verlag, Berlin Heidelberg New York, 3. aufl. edition, 2009.

[3] Bundestag. Bundesdatenschutzgesetz in der fassung der bekanntmachung vom 14. januar 2003 (bgbl. i s. 66), das zuletzt durch artikel 1 des gesetzes vom 25. februar 2015 (bgbl. i s. 162) geändert worden ist. Web. Abgerufen am 30.05.2016 `https://www.gesetze-im-internet.de/bdsg_1990/index.html`.

Tabelle 1: Best Practices für RE, Quelle: [14]

Fokus	Best Practice	Einführungs-kosten	Anwendungs-kosten	Nutzen
Wissen	Einbinden der Benutzer und Kunden	Gering	Mittel	Besseres Verständnis der „echten Bedürfnisse“
Wissen	Identifizierung und Konsultation mit allen RE Quellen	Gering - Mittel	Mittel	verbesserte RE Abdeckung
Wissen	Zuweisung von erfahrenen Mitarbeitern	Mittel - Hoch	Mittel	Bessere Leistungseinschätzung
Ressourcen	15 - 30 % der Ressourcen für das RE einplanen	Gering	Mittel - Hoch	Umsetzung einer hoch qualitativen Spezifikation im Projekt
Ressourcen	Anforderungsschablonen und Beispiele zur Verfügung stellen	Gering - Mittel	Gering	Verbesserung der Spezifikationsqualität
Ressourcen	Gute Beziehungen zu Stakeholdern halten	Gering	Gering	Besseres eingehen auf Kundenwünsche
Prozess	Priorisieren der Anforderungen	Gering	Gering - Mittel	Fokus auf die wichtigsten Kundenwünsche
Prozess	Ergänzende Modelle parallel mit Prototypen entwickeln	Gering - Mittel	Gering	Eliminierung von Widersprüchen und Inkonsistenzen in der Spezifikation
Prozess	Nachverfolgungsmatrix aktuell halten	Mittel	Mittel	Explizite Verweise zwischen Anforderungen und Ergebnissen
Prozess	Anwendung von Peer Reviews, Szenarien und Walk-Throughs zum Validieren und Verifizieren der Anforderungen	Gering	Mittel	Genauere Spezifikation und höhere Kundenzufriedenheit

[4] Bundestag. Das zehnte buch sozialgesetzbuch - sozialverwaltungsverfahren und sozialdatenschutz - in der fassung der bekanntmachung vom 18. januar 2001 (bgbl. i s. 130), das durch artikel 7 des gesetzes vom 21. juli 2016 (bgbl. i s. 1768) geändert worden ist, 21.07.2016. Abgerufen am 30.05.2016 https://www.gesetze-im-internet.de/sgb_10/index.html.

[5] Bundestag. ehealth-gesetz, 28.12.2015. Abgerufen am 30.05.2016 http://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&jumpTo=bgbl115s2408.pdf.

[6] Bundestag. Gesetz zur modernisierung der gesetzlichen krankenversicherung, 28.12.2015. Abgerufen am 30.05.2016 http://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&jumpTo=bgbl103s2190.pdf.

[7] DIN. Deutsches institut für normung e. v. - informationstechnik - it-sicherheitsverfahren - informationssicherheitsmanagementsysteme - Überblick und terminologie (iso/iec 27000:2009), 2009. Abgerufen am 30.05.2016 http://www.beuth.de/de/norm/din-iso-iec-27000/140845763.

[8] B. G. I. e. V. (bvitg e. V.). Telemedizinlösungen für eine bessere versorgung. Web. Abgerufen am 30.05.2016 http://www.conhit.de/media/cit/cit_dl_all/presse_2/pressemappen_themenfelder/vernetzte_medizin/deutsch_7/2016_Themenfeld_Vernetzte_Medizin_Telemedizinloesungen_fuer_eine_bessere_Versorgung-2.pdf.

[9] K. E. Emam and A. G. Koru. A replicated survey of it software project failures. *IEEE Software*, 25(5):84–90, Sept 2008.

[10] EU. Richtlinie 93/42/ewg des rates vom 14. juni 1993 über medizinprodukte, der rat der europÄischen gemeinschaften. Web. Abgerufen am 30.05.2016 http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1993L0042:20071011:de:PDF.

[11] Gematik. Gematik. Web. Abgerufen am 30.05.2016 https://www.gematik.de.

[12] Gematik. Übergreifendes sicherheitskonzept, 10.03.2008. Abgerufen am 30.05.2016 https://www.gematik.de/cms/media/dokumente/release_0_5_3/release_0_5_3_datenschutz/gematik_DS_Sicherheitskonzept_V2_2_0.pdf.

[13] Gematik. Gesamtarchitektur, 18.03.2008. Abgerufen am 30.05.2016 https://www.gematik.de/cms/de/spezifikation/release_0_5_3/release_0_5_3_architektur/gesamtarchitektur_r053.jsp.

[14] H. F. Hofmann and F. Lehner. Requirements engineering as a success factor in software projects. *IEEE Software*, 18(4):58–66, Jul 2001.

[15] H. Partsch. *Requirements-Engineering systematisch - Modellbildung für softwaregestützte Systeme*. Springer-Verlag, Berlin Heidelberg New York, 2. aufl. edition, 2010.

[16] J. Rivera and R. van der Meulen. Gartner says 4.9 billion connected "things"will be in use in 2015. Web. Abgerufen am 30.05.2016 https://www.gartner.com/newsroom/id/2905717.

Analyse von Tools für einen automatisierten Continuous Delivery Prozess *

Eva Witzel
Reutlingen University
Witzel.Eva@Student.Reutlingen-University.DE

Abstract

Um Prozesse innerhalb der Softwareentwicklung zu beschleunigen und zu automatisieren, gibt es eine Reihe von Tools. Zu einer Toolchain gehören u.a. die Kontrolle einer Versionsverwaltung für Source Code, automatisierte Build- und Unit-Tests sowie das abschließende Deployment der Software. Innerhalb dieser Arbeit werden spezifische Anforderungen an eine kontinuierliche Softwareentwicklung definiert. Dazu gehören u.a. ein konfigurierbares Feedback System, die Möglichkeit einen Build reproduzieren zu können, sowie die Verwendung von Hardware innerhalb eines Entwicklungsprozesses. Das kann beispielsweise dann der Fall sein, wenn die Software für weitere Tests auf einem Embedded-System installiert wird. Ziel dieser Arbeit ist die Analyse aktueller Lösungen, um auszuwählen, welches Tool die Anforderungen am besten erfüllt.

Schlüsselwörter

Continuous Delivery, Continuous Software Engineering, Testing, DevOps

CR-Kategorien

D.2.9 [**Software Engineering**]: Management

*

Betreuer Hochschule: Prof. Dr.-Ing. Schöller
Hochschule Reutlingen
Marcus.Schoeller@reutlingen-university.de
Wissenschaftliche Vertiefungskonferenz
16. November 2016, Hochschule Reutlingen
Copyright 2016 Eva Witzel

1 Einführung

In Zeiten, in denen viele Bereiche des täglichen Lebens mehr und mehr automatisiert werden, stellt sich die Frage, inwieweit Softwareentwicklungsprozesse automatisiert und optimiert werden können. Das Verfahren des Continuous Delivery (CD) soll es ermöglichen, Software schneller an den Kunden auszuliefern. Die Basis dafür bietet ein weitestgehend automatisierter Prozess innerhalb der Entwicklung. [14, S. 13] Für den Begriff Continuous Delivery gibt es keine allgemeingültige Definition. Martin Fowler definiert diesen Begriff folgendermaßen:

> "Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time". [13]

Software soll duch automatisierte Build- und Testvorgänge jederzeit in die Produktion ausgeliefert werden können. Um die Einführung eines CD Prozesses zu vereinfachen, gibt es eine Reihe von Tools. Diese Tools bieten die Möglichkeit, Source Code automatisch zu bauen, zu testen und zu deployen. Somit stellt sich die Frage, wie die Funktionalitäten der Tools im Detail aussehen und in welchen Bereichen diese sich unterscheiden. Die Auswahl der richtigen Toolchain ist noch entscheidender, wenn das Softwareprojekt viele spezifische Anforderungen und Abhängigkeiten hat.
Aus diesem Grund werden innerhalb die-

ser Arbeit aktuelle Lösungen vorgestellt und evaluiert. Dazu werden spezifische Kriterien definiert, mit denen vor allem die Adaptivität der Tools untersucht werden soll. Adaptivität bedeutet in diesem Fall die Fähigkeit, die Toolchain an spezielle Bedürfnisse eines Entwicklungsprozesses anpassen zu können. Das kann z.B. die Verwendung von Embedded-Systems sein oder die Möglichkeit, Konfigurationeinstellungen zu versionieren. Die Evaluation soll zeigen, welche Unterschiede es gibt und welche Toolchain für welchen Einsatzzweck und v.a. für die definierten Anforderungen am besten geeignet ist.
Im folgenden Abschnitt werden zunächst die Grundlagen eines CD Prozesses erläutert. Darauffolgend werden die Bewertungskriterien definiert, um in Abschnitt 4 die Lösungsansätze vorzustellen und in Abschnitt 5 zu evaluieren. Abgeschlossen wird diese Arbeit mit einer Diskussion und einem Fazit.

2 Grundlagen des Continuous Delivery

Damit es möglich ist, Software jederzeit ausliefern zu können, müssen nahezu alle Schritte des Entwicklungsprozesses automatisiert werden. Die einzelnen Komponenten des Prozesses, sowie die Struktur der eigentlichen Delivery Pipeline werden im Folgenden genauer erklärt. In Abbildung 1 sind die einzelnen Schritte einer Delivery Pipeline abgebildet.

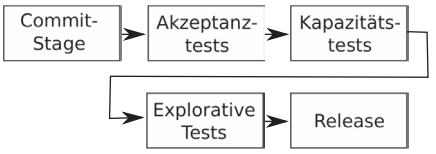

Abbildung 1: Aufbau einer Continuous Delivery Pipeline, vgl. [12, Figure 1.1]

Der Begriff *Pipeline* wird im Bereich CD sehr häufig verwendet. Es ist damit, wie in Abbildung 1 dargestellt, eine lineare Abfolge einzelner Komponenten gemeint, die nacheinander ausgeführt werden.

2.1 Commit-Phase

Eine Pipeline wird mit jedem Commit, also mit einer Änderung am Source Code gestartet. Daraufhin wird der Source Code kompiliert. Schlägt bereits dieser Schritt fehl, wird der Entwickler informiert und die weitere Durchführung abgebrochen. Nach einem erfolgreichen Build werden Unit-Tests und ggf. eine statische Quellcode-Analyse durchgeführt. Ziel der Commit-Phase ist es, Fehler so schnell wie möglich zu erkennen, um dem Entwickler nach kurzer Zeit Feedback geben zu können. [12, S. 120ff]

2.2 Durchführung von Tests

Nach der Commit-Phase sind weitere Tests notwendig, auf die im Folgenden genauer eingegangen wird.

Akzeptanztest

Akzeptanztests beschreiben die Funktionen, die das System erfüllen muss. Die Tests müssen, wenn möglich, in der Umgebung laufen, die auch beim Kunden vorhanden ist. [12, S. 122f] Um Akzeptanztests überhaupt ausführen zu können, muss die Testumgebung entsprechend konfiguriert werden. Daraufhin wird die Software auf dem Zielsystem installiert und getestet. [12, S. 187ff] Auch nicht funktionale Anforderungen wie *Security* oder *Availability* können mit Akzeptanztests abgedeckt werden. [12, S. 188]

Kapazitätstest

Nach erfolgreicher Durchführung aller Akzeptanztests werden Kapazitätstests durchgeführt, um nicht funktionale Anforderungen zu testen. Hierbei wird u.a. getestet, ob die Software die gewünschte Performance liefert und ausreichend leistungsfähig ist. [14, S. 157ff]

Explorative Tests

Explorative Tests werden in Form von manuellen Tests und ohne definierten Testplan durchgeführt. [14, S. 173ff] Trotz der vielen automatisierten Tests kann in bestimmten Fällen ein manueller Test notwendig sein. [14, S. 173ff] In bestimmten Fällen können

auch nicht funktionale Anforderungen wie die *Usability* oder das *Look and Feel* manuell getestet werden. [14, S. 173ff]

2.3 Release

Laut [12, S. 129ff] ist es Ziel, einen komplett automatisierten Release Prozess zu erreichen und theoretisch jeden Tag einen Release durchführen zu können. Mit dem CD Prozess ist sichergestellt, dass jede neue Version ausreichend getestet ist. Die Auslieferung in die Produktion ist im Prinzip kein neuer Schritt, da das Deployment bereits vor den Akzeptanztests durchgeführt wird. [14, S. 181ff]

3 Definition von Bewertungskriterien

Damit die Toolchains in den Funktionalitäten miteinander verglichen werden können, werden in diesem Abschnitt zunächst einige Anforderungskriterien definiert. Für diese Kriterien wurde größtenteils die Definition einer CD Pipeline genauer untersucht, um festzustellen, welche Aufgaben eine Toolchain in diesem Bereich erfüllen muss.

3.1 Automatisierter Build-Prozess

Die Durchführung eines automatisierten Build-Prozesses beinhaltet unterschiedliche Komponenten. Grundlegend dabei ist, dass auf Änderungen im Source Code mit einem automatisierten Build reagiert wird. Außerdem soll es möglich sein, Änderungen nur in bestimmten Branches zu betrachten.

Da viele Projekte sehr spezifische Projektstrukturen haben, muss sich der Prozess auf verschiedene Anforderungen anpassen können. Es gibt unterschiedliche Build-Systeme, Programmiersprachen und IDEs, die je nach Projekt in unterschiedlichen Konstellation vorkommen können.

Noch komplexer ist der Build-Prozess bei der Entwicklung von Embedded-Software. Dabei muss beachtet werden, dass oftmals auch die Erstellung des Board Support Packages (BSP), also des Betriebssystems, mit zur Entwicklung gehört. Somit gibt es auch hierbei sehr spezifische Anforderungen an den Build, die vom Betriebssystem, dem verwendeten Embedded-System, sowie dem Framework für die Erstellung des BSP abhängen.

Für das Auslösen eines Builds gibt es unterschiedliche Möglichkeiten. Die Toolchain soll die Auslösemöglichkeiten **Manuell**, **Zeitlich**, **Poll** und **Event** bieten. [10, S. 80f] Mit einem Poll wird ein Repository auf Änderungen überprüft. Ein Event kann ein Commit in einem Repository oder der erfolgreiche Durchlauf eines anderen Builds sein, der nach Beendigung einen neuen Build startet.

3.2 Statische Quellcode-Analyse

Um die Qualität der Software zu gewährleisten, ist es sinnvoll eine automatische Analyse des Quellcodes, am Ende von jedem Build, durchzuführen. Für die Toolchain bedeutet das, dass diese entweder eigene Werkzeuge für die Analyse mitbringen, oder die Integration von externen Analysewerkzeugen erlauben muss. [10, S. 17]

3.3 Automatisierte Tests

Automatisierte Tests decken einen Großteil der CD Pipeline ab und müssen daher auch mit Hilfe der Toolchain aufgesetzt und durchgeführt werden können. Auch hierbei gilt, dass entsprechende Test-Frameworks ggf. in die Pipeline integriert werden müssen. [12, S. 122ff] Beispiele für solche Frameworks sind *JUnit* und *Googletest* für die Ausführung von Unit-Tests. Bei der Entwicklung von Embedded Software ist es wichtig, auch die Ergebnisse von Tests die auf dem Embedded System laufen zu generieren und dem Entwickler in Form von Feedback zur Verfügung zu stellen.

3.4 Konfigurationsmanagement

Ein Softwareprodukt besteht meistens aus vielen Einzelkomponenten, die wiederum Abhängigkeiten untereinander aufweisen. Somit ist es u.a. Aufgabe des Konfigurati-

onsmanagement, Abhängigkeiten zwischen Softwarekomponenten zu organisieren, sowie Code-Änderungen zu kontrollieren und versionieren. [12, S. 31ff]
Laut [12, S. 24ff] ist es für einen funktionierenden Delivery Prozess unbedingt notwendig, alles unter Versionskontrolle zu haben. Das System für die Versionsverwaltung ist nicht zwingend Bestandteil des CD Prozesses, muss jedoch kompatibel mit der Toolchain sein.
Im Gegensatz zur Versionsverwaltung geht es innerhalb des Build-Managements nicht um die Verwaltung des Quellcodes, sondern um die Versionierung und Verwaltung der Build-Artefakte. [12, S. 38f] Für einen Release muss klar definiert sein, aus welchen Unterkomponenten sich eine Hauptkomponente zusammensetzt, sodass dieser Stand auch nach Release wiederhergestellt werden kann.

3.5 Feedback System

Nach der Durchführung von Builds oder Tests soll der zuständige Entwickler so schnell wie möglich ein Feedback von dem Prozess bekommen. [12, S. 11ff] Ein Feedback-Mechanismus kann beispielsweise das Verschicken einer E-Mail sein. Diese Benachrichtigungen sollen Team spezifisch eingerichtet werden können. Je nach Projekt oder Modul sollen E-Mails nur an bestimmte Personen oder auch an das komplette Team verschickt werden. [10, S. 207ff]
Damit die Entwickler auf das Feedback reagieren können, soll die Toolchain eine übersichtliche Visualisierung der Build- und Test-Ergebnisse bereitstellen. Häufig wird das in Form einer Web-Oberfläche und einem Dashboard realisiert. [12, S. 321f]

3.6 Deployment

Damit die Tests aussagekräftige Ergebnisse liefern können, müssen diese zum Teil bereits auf dem Zielsystem ausgeführt werden. Der Vorgang der Installation der Software auf dem System wird auch als Deployment bezeichnet. [12, S. 160ff] Der Vorgang des Deployment wird etwas komplexer, sobald es sich beispielsweise um die Entwicklung von Embedded-Software handelt. Im Gegensatz zu Webanwendungen wird externe Hardware benötigt, auf der die Software installiert wird. Für die Durchführung von Tests nach dem Deployment, muss die Umgebung soweit flexibel sein, dass beispielsweise Testdaten aus einer Datenbank an das System übergeben und die darauffolgenden Testergebnisse ausgelesen werden können.

3.7 Reproduzierbarkeit

Reproduzierbarkeit bedeutet in diesem Fall, dass ein bestimmter Softwarestand eines Release auch nach Jahren wiederhergestellt werden kann. Außerdem soll es möglich sein, die Toolchain zu wechseln, ohne die komplette Konfiguration aller Projekte erneut vornehmen zu müssen. Das führt dazu, dass die Konfiguration eines Build-Prozesses in irgendeiner Art versionierbar sein muss, also nicht ausschließlich über eine Web-Oberfläche durchgeführt wird.

3.8 Usability

Der letzte Punkt umfasst die Benutzerfreundlichkeit sowie die Bedienbarkeit des Tools. Da die Entwickler täglich mit dem Tool arbeiten, um neue Projekte zu erstellen oder Testergebnisse zu betrachten, sollte das Tool eine übersichtliche Web-Oberfläche sowie eine gute Bedienbarkeit aufweisen.

3.9 Verfügbarkeit

Softwareprojekte werden teilweise über Jahre weiterentwickelt und gepflegt. Daher ist es wichtig, dass auch hinter der Toolchain ein Team oder eine Firma für den Support und die Weiterentwicklung steht.

4 Aktuelle Lösungsansätze

In folgendem Abschnitt werden einige Lösungsansätze vorgestellt, die das Ziel verfolgen, Softwareentwicklungsprozesse zu automatisieren. Dieser Abschnitt dient dazu, einen Überblick über unterschiedliche Systeme zu geben, um diese innerhalb der Evaluation miteinander zu vergleichen.

4.1 Concourse

Die Prinzipien von *Concourse* sollen dem Benutzer einen einfachen Wechsel zwischen unterschiedlichen CD Tools ermöglichen, da alle Konfigurationseinstellungen der Projekte in einzelnen Dateien organisiert werden. [8] Mit *Concourse* können einfache bis sehr komplexe Pipeline Systeme aufgebaut werden. Eine Übersicht über den Aufbau einer Pipeline ist in Abbildung 2 dargestellt.

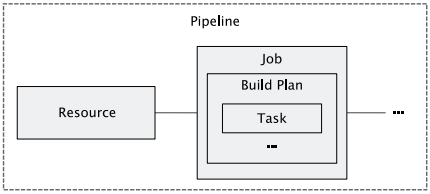

Abbildung 2: Übersichtsdarstellung der Komponenten in Concourse

Die Pipeline setzt sich aus Ressourcen und Jobs zusammen. Eine Ressource stellt eine bestimmte Form einer Entität dar, die einen Job auslösen kann, z.B. ein Git Repository oder einen zeitlichen Trigger. Der Job setzt sich aus verschiedenen Aktionen zusammen, die mit Hilfe einer Domänenspezifischen Sprache (DSL) in einem Build Plan definiert werden. Ein Build Plan kann verschiedene Aktionen wie z.B einzelne Tasks durchführen, in denen Skripte in isolierten Docker Containern ausgeführt werden. [8] Das Ergebnis eines Jobs kann wiederum an die nächste Ressource für die darauffolgenden Jobs weitergegeben werden.

4.2 GoCD

Der Aufbau einer Pipeline ist ähnlich wie bei *Concourse* in verschiedene Hauptkomponenten gegliedert, siehe Abbildung 3.
Ein Task definiert eine einzelne Aktion, die ausgeführt wird. Jobs bestehen aus einer Reihe von Tasks. Sobald ein Task fehlschlägt, schlägt auch der Job fehl. Eine Stage enthält wiederum mehrere Jobs. Die eigentliche Pipeline wird aus mehreren Stages aufgebaut, die in einer bestimmten Reihenfolge ausgeführt werden. Der Trigger definiert, wann eine Pipeline ausgeführt werden soll, z.B. über eine Änderung im Repository oder zeitlich. [9]

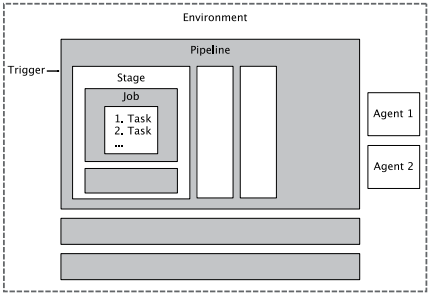

Abbildung 3: Übersichtsdarstellung der Komponenten in GoCD

Alle zuvor vorgestellten Komponenten, wie Tasks oder Jobs, werden auf Agenten ausgeführt. Ein Server überwacht die Events des Triggers und gibt die entsprechenden Jobs an Agenten weiter. Pipelines und Agenten können in beliebiger Anzahl und Konstellation gruppiert und in einer eigenen Umgebung von anderen Komponenten isoliert werden. [9]

4.3 Jenkins

Jenkins ist ein sehr bekanntes Tool, das über eine Web-Oberfläche konfiguriert werden kann. Das Tool bietet eine große Auswahl an Plugins. Auch *Jenkins* bietet eine eigene DSL an, mit der die Pipeline definiert wird. Mit dieser DSL kann ein *Jenkinsfile* erstellt werden. [6] Die Definition einer Pipeline besteht auch hier aus mehreren Komponenten, siehe Abbildung 4.
Ein Step definiert, wie auch bei den bereits vorgestellten Tools, eine einzelne Aktion. Eine Stage besteht aus einem oder mehreren Steps . Auch hier wird nach dem Prinzip mit Master und Slave, bzw. Agent gearbeitet (siehe 4.2 GoCD). [6]

4.4 Buildbot

Das Tool *Buildbot* vereint Continuous Integration, Continuous Deployment und Relea-

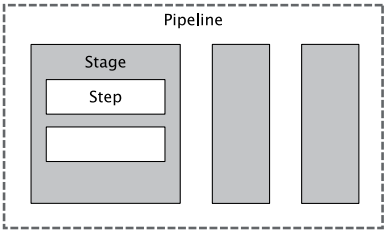

Abbildung 4: Übersichtsdarstellung der Komponenten in Jenkins

se Management. Ähnlich wie bei *Jenkins* und *GoCD* gibt es auch hier ein Master-Slave-Prinzip. Der Master entscheidet dabei, was wie und wo gebaut werden soll und sendet das entsprechende Kommando an einen der Slaves. Die einzelnen Slaves können auf verschiedenen Maschinen und Plattformen laufen. [7] Die Hauptkomponenten von *Buildbot* werden im Folgenden genauer vorgestellt, siehe Abbildung 5.

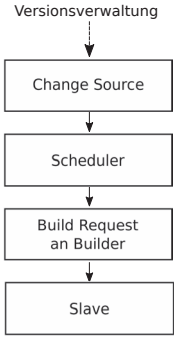

Abbildung 5: Ablaufdiagramm für die Ausführung eines Build in Buildbot

Die Change Source reagiert, sobald sich in einem Repository einer Versionsverwaltung etwas ändert. Die Change Source verhält sich also ähnlich zu den bereits vorgestellten Komponenten Ressource und Material. Auch hier können zeitliche Trigger verwendet werden. Die Scheduler entscheiden, wann Builds ausgeführt werden sollen. Dabei gibt es verschiedene Arten, die beliebig konfiguriert werden können. Um einen Build zu starten, sendet der Scheduler eine Anfrage an den Builder. Dieser wird ausgeführt, sobald ein Slave verfügbar ist. Ein Builder definiert, wie ein entsprechender Build durchgeführt werden soll und wird wiederum durch einzelne Build Steps definiert.

4.5 Team Foundation Server

Der *Team Foundation Server (TFS)* ist eine proprietäre Lösung von Microsoft, wobei die Express Version auch kostenlos und mit zu fünf Teammitgliedern verwendet werden kann. Ein großer Unterschied zu den bisher vorgestellten Tools ist, dass *TFS* eine eigene Versionsverwaltung beinhaltet. Des Weiteren unterstützt *TFS* agile Softwareentwicklungsprozesse wie *Kanban* oder *Scrum*. So können Backlogs erstellt, sowie Arbeitsaufgaben mit Code oder Builds verknüpft werden. [5]
Für die Erstellung eines neues Prozesse können in TFS eine Reihe von Build Steps mit unterschiedlichen Funktionen definiert werden. Es können Build Systeme wie z.B. *Gradle* oder Utilities in Form von Shell-Skripten verwendet werden, um spezifische Builddefinitionen zu erstellen. Für die anschließende Durchführung von Tests und dem Deployment bietet TFS unterschiedliche Plugins an. Außerdem können Umgebungsvariablen und verschiedene Trigger für das Auslösen eines Builds verwendet werden, so wie bei den anderen Tools auch. [5]

5 Evaluation

In diesem Abschnitt werden die Lösungsansätze bezüglich der zuvor definierten Bewertungskriterien evaluiert. Um die Anzahl der Tools zu reduzieren, wird diese mit Hilfe einer Vorauswahl eingeschränkt. Bei dem Tool, das nicht im Detail untersucht wird, handelt es sich um den *Team Foundation Server* von Microsoft. *TFS* kann am wenigsten alle der hier definierten Anforderungen erfüllen.
Das Problem hierbei ist, dass die Builddefinition ausschließlich innerhalb des Web-Tools sowie auf dem *TFS*-Server in Form

von Konfigurationsdateien vorhanden ist. Für den Entwickler gibt es keine Möglichkeit, die Builddefinition beispielsweise in Form einer DSL zu erstellen und zu versionieren. Ein weiterer Nachteil ist, dass ein erfolgreicher Durchlauf einer Builddefinition keine andere Builddefinition triggern kann.

5.1 Vergleich

Für die Evaluation wurden alle Tools installiert und praktisch untersucht. Zum Teil kam eine einfache Beispielanwendung zum Einsatz, mit der exemplarisch die Konfiguration einer Builddefinition gezeigt werden kann. Die Informationen zu den Funktionalitäten werden somit aus praktischen Tests, der Web-Oberfläche des jeweiligen Tools sowie vereinzelt aus den Dokumentationen [7](Buildbot), [8](Concourse), [9](GoCD) und [6](Jenkins) entnommen.
GoCD und *Jenkins* sind webbasierte Tool, dessen Konfigurationen größtenteils in der Web-Oberfläche stattfinden. Im Gegensatz dazu, werden bei *Concourse* und *Buildbot* Konfigurationsdateien verwendet.

Automatisierter Build-Prozess

Alle Tools bieten die Möglichkeit, Source Code innerhalb einer Versionsverwaltung zu überwachen und auf Änderungen zu reagieren. Mit allen Tools können die in den Anforderungen definierten Trigger verwendet werden.
Für die Definition von Buildschritten bieten die Tools teilweise sehr verschiedene Lösungen an. Grundsätzlich können jedoch bei allen Tools einfache Shellbefehle eingesetzt werden. *Jenkins* und *Concourse* arbeiten mit einer eigenen DSL, mit der einzelne Buildschritte definiert werden können. Bei *Concourse* muss sich der Entwickler zunächst in die Syntax der Sprache einarbeiten. Bei Concourse gibt es jedoch ein Problem mit der Standardeinstellung des Docker Containers, dieser muss für die Ausführung eines make Befehls erst angepasst werden. *Jenkins* bietet für die Verwendung der DSL einen Übersetzer an, mit dem u.a. der Inhalt eines Shell-, oder Batch-Skriptes in ein *Jenkinsfile* konvertiert werden kann.
Buildbot verwendet zwar keine eigene DSL, bietet jedoch den Vorteil, dass das Tool auf Python basiert und die Prozessschritte somit ebenfalls in Python definiert werden können. Ein Nachteil ist jedoch, dass in *Buildbot* kein Pipeline-Konzept wie bei den zuvor vorgestellten Tools verwendet wird. Im Gegensatz zu *TFS* kann jedoch ein Build durch einen anderen Build ausgelöst werden, sodass es trotzdem möglich ist, eine CD Pipeline aufzubauen.

Statische Quellcode-Analyse & Automatisierte Tests

Da alle Tools mit Shellbefehlen und Skripten arbeiten können, ist grundsätzlich auch die Verwendung einer statischen Quellcode-Analyse sowie automatisierten Tests kein Problem. Jenkins bietet mit Hilfe von Plugins zusätzlich die Möglichkeit, die Ergebnisse entsprechend auszuwerten und zu visualisieren. Für die statische Quellcode-Analyse bietet Jenkins u.a. die Integration von *Coverity* und *Protecode* an. Im Bereich der Unit-Tests werden u.a. *MS-Test*, *Googletest*, *JUnit* und *CppUnit* unterstützt. Grundsätzlich können die Testergebnisse bei allen Tools in der Ausgabe des Builds betrachtet werden.

Konfigurationsmanagement

Je nach Tool, kann der Entwickler Versionsverwaltungen wie *Git*, *Mercurial* oder *SVN* nutzen. Die Integration von *Git* bieten alle Tools an. *GoCD* unterstützt zudem die Versionsverwaltung des Microsoft *Team Foundation Servers* und *Perforce*.
Bei allen Tools wird ein Build nummeriert und die aktuelle Commit ID des Repositorys angegeben. Für ein bestimmtes Release- oder Build-Management kann es jedoch sein, dass sich die Buildnummer aus mehr Informationen als nur einer Zahl zusammensetzen muss. Für diese Modifizierung bieten *Jenkins* und *GoCD* gute Lösungen an. Bei *Jenkins* kann ein Plugin verwendet werden, während *GoCD* dieses Feature automatisch enthält. In *Concourse* gibt es keine Möglichkeit, die Buildnummer anzupassen.

Feedback System

Die Funktion einer E-Mail Benachrichtigung ist in *Jenkins*, *GoCD* und *Buildbot* enthalten. Bei *Concourse* kann diese Funktion nur über ein Third-Party Projekt hinzugefügt werden. *Jenkins* und *Buildbot* bieten mehrere Möglichkeiten bei der Konfiguration von E-Mail Benachrichtigungen an.
Jedes Tool bietet dem Entwickler die Möglichkeit, beispielsweise nach Erhalt eines Feedbacks, das Buildergebnis in der Web-Ansicht zu betrachten. *GoCD* und *Concourse* verwenden für die Darstellung einen typischen Pipeline Aufbau. Detaillierte Informationen zu den Buildergebnissen erhält der Entwickler, wenn eine der Komponenten ausgewählt wird. Innerhalb der Detailsansicht kann dann das Ergebnis der einzelnen Buildschritte betrachtet werden. Abbildung 6 zeigt, wie der Benutzer in *Concourse* mit einem Mausklick in der Gesamtansicht in die Detailansicht gelangt.

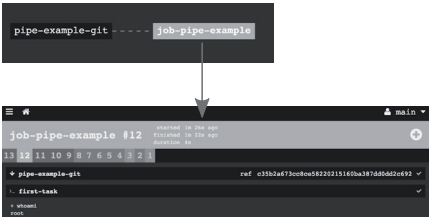

Abbildung 6: Gesamtansicht und Detailansicht einer Pipeline in Concourse

Deployment

Das Deployment beschränkt sich in dieser Analyse auf die Installation auf einem Embedded System. *Jenkins*, *GoCD* und *Buildbot* sind offen gestaltet, sodass mit Hilfe der Shellbefehle problemlos eine Verbindung mit dem Embedded-System aufgebaut werden kann (Infrastruktur vorausgesetzt). Bei der Verwendung von *Concourse* muss der Docker Container entsprechend konfiguriert sein.
Bei *Jenkins* und *GoCD* gibt es zusätzlich Plugins, die *XebiaLabs* Deployment auf unterschiedliche Zielsysteme ermöglichen. Bei *GoCD* können zusätzlich *Ansible*, *Puppet* und *Chef* mit eingebunden werden.

Reproduzierbarkeit

Grundsätzlich können die Builddefinitionen aller Tools versioniert werden. Bei *Buildbot* gestaltet sich die Versionierung am Schwierigsten. Die Konfiguration basiert auf einer Konfigurationsdatei, in der alle wichtigen Server- und Build-Einstellungen vorgenommen werden. Wenn allein die Builddefinition versioniert werden soll, sind zusätzliche Python-Skripte notwendig.
GoCD bietet eine bessere Lösung an. Hierbei existiert eine globale Konfigurationsdatei im XML-Format, die über die Web-Oberfläche oder direkt in der Datei selbst geändert werden kann. Bei *Jenkins* und *Concourse* ist der Prozess durch die Verwendung von eigenen DSLs sehr einfach. Die Dateien können ineinander verschachtelt und direkt aus verschiedenen Repositories eingebunden werden. *Jenkins*, *Buildbot* und *GoCD* bieten außerdem die Möglichkeit, einen bestimmten Build in der Web-Oberfläche nochmals zu starten. In *Jenkins* kann für den Rebuild auch das *Jenkinsfile* verändert werden, um beispielsweise etwas zu testen.

Usability

Bei *Jenkins* und *Buildbot* ist die Web-Oberfläche und das Design veraltet. *Jenkins* bietet dafür einen guten Konfigurationsassistenten für die Definition der einzelnen Buildschritte an. Die Web-Oberfläche kann bei *Buildbot* sehr einfach und dadurch auch sehr übersichtlich gehalten werden, da alle Einstellungenin den Konfigurationsdateien stattfinden. Dieser Vorteil wird auch von *Concourse* genutzt. Die Web-Oberfläche von *GoCD* ist zu Beginn etwas undurchsichtig. Es ist nicht ganz klar definiert, wo die Pipeline betrachtet und wo erstellt werden kann.

Verfügbarkeit

Jenkins und *GoCD* sind die Tools, die bereits am längsten bestehen. *Jenkins* wurde 2006 unter dem ursprünglichen Namen *Hud-*

son gegründet. Hinter diesem Projekt steht ein großes Entwicklerteam und eine sehr gute Dokumentation. [4]
GoCD wird seit 2008 von *ToughtWorks Inc.* entwickelt. [11] Obwohl es sich hierbei um ein Open Source Tool handelt, gibt es die Möglichkeit, zusätzlichen Support einzukaufen.
Buildbot wird seit 2011 entwickelt. *Buildbot* ist ein Teil der *Software Freedom Conservancy Inc.* und hat eine große Community, die stetig für die Weiterentwicklung des Tools sorgt. [1]
Concourse ist das Neueste aller Tools. Unterstützt wird dieses durch *Pivotal Software Inc*. [3] Bis auf die Webseite und einige Blog-Beiträge gibt es noch nicht viele Informationen zu diesem Tool. Die ersten Release Notes sind von Februar 2016. [2]

5.2 Ergebnisse

Für eine bessere Übersicht sind die Ergebnisse der Evaluation in Tabelle 1 nochmals zusammengefasst.

GoCD kann alle Anforderungen erfüllen. Die Vorteile bei *GoCD* liegen in der übersichtlichen Darstellung des Konfigurationsassistenten. Außerdem hat *GoCD* eine besonders gute Integration verschiedener Versionsverwaltungen. Dafür wird bei *GoCD* die Versionierung der Builddefinition nicht ganz so gut gelöst, da keine DSL verwendet werden kann.

Jenkins Die Bewertung für *Jenkins* fällt von allen Tools am besten aus. Die Verwendung eines *Jenkinsfiles* bietet eine einfache Versionierung und Erstellung. Außerdem gibt es eine große Auswahl an Plugins. So hat der Entwickler die Möglichkeit, viele Funktionen in Form eines Plugins hinzuzufügen.

Buildbot kann grundsätzlich auch alle Anforderungen erfüllen. Mit Hilfe von Python können die Funktionen von *Buildbot* voll ausgenutzt und sogar erweitert werden. Auch hier wird keine DSL verwendet, mit der die Definition eines Builds erstellt werden kann.

Concourse ist von allen Tools das Neueste und schneidet in bestimmten Bereichen eher schlechter ab. Das liegt hauptsächlich an fehlenden Funktionen wie den E-Mail Benachrichtigungen. Vorteile von *Concourse* sind das schlichte und moderne Design der Web-Oberfläche, sowie die Verwendung einer eigenen DSL.

6 Diskussion und Fazit

Innerhalb dieser Arbeit wurden verschiedene Tools analysiert und evaluiert. Zuvor wurden spezifische Anforderungen definiert, um zu untersuchen, welches der Tools für diese am besten geeignet ist. Es hat sich gezeigt, dass *Jenkins* die Anforderungen insgesamt am besten erfüllen kann. Auch *GoCD* ist *Jenkins* sehr ähnlich, arbeitet jedoch ohne DSL und ohne zusätzliche Plugins.
Bei kleineren, nicht ganz so komplexen Projekten, kann jedoch auch ein leichtgewichtigeres Tool ausreichend sein. Bei der Verwendung von sehr neuen Tools, wie z.B. *Concourse*, stellt sich die Frage, ob dieses auch die nächsten fünf bis zehn Jahre auf dem Markt bleibt und entsprechend unterstützt und weiterentwickelt wird. Außerdem kommt es auch auf das Projektteam und den Wissensstand der Entwickler an, welches Tool am besten geeignet ist. Vor allem bei Tools wie *Concourse* oder *Buildbot*, die ausschließlich über Konfigurationsdateien gesteuert werden, sind gewisse Vorkenntnisse notwendig.
Abschließend kann gesagt werden, dass *Jenkins* für eine große Bandbreite an Projekten am besten geeignet ist. Letztendlich liegt die Entscheidung für die richtige Toolchain jedoch individuell bei jedem Projektteam und sollte auch individuell für ein Projekt getroffen werden. Die Analyse dieser Arbeit hat gezeigt, welche Vorteile das jeweilige Tool in welchen Bereichen bietet und dient somit als gute Entscheidungsgrundlage für die Wahl einer Toolchain.

Literatur

[1] Buildbot about us. `http://buildbot.net/about.html`. Besucht am 11.10.2016.

Tabelle 1: Darstellung der Ergebnisse der Evaluation

	GoCD	Jenkins	Buildbot	Concourse
Automatisierter Build-Prozess	++	++	+	+
Statische Quellcode-Analyse & automatisierte Tests	+	++	+	+
Verwendung von Versionskontrolle	++	+	+	+
Build-Management	++	+	o	o
Feedback E-Mail Notification	+	++	++	-
Visualisierung von Buildergebnisse	+	+	+	++
Deployment	++	++	+	+
Builddefinition versionierbar	+	++	o	++
Build reproduzierbar	+	++	+	-
Usability	+	+	++	++
Verfügbarkeit	+	+	+	o

++ Die Anforderung wird besonders gut erfüllt, die Funktion bietet weitere Features oder wurde besonders gut umgesetzt

+ Die Anforderung wird erfüllt

o Die Anforderung wird eingeschränkt erfüllt, sodass die Umsetzung der Funktion mit mehr Aufwand verbunden ist

- Die Anforderung wird nicht erfüllt

[2] Cocnourse release notes. `https://concourse.ci/downloads.html`. Besucht am 13.10.2016.

[3] Concourse community. `https://concourse.ci/community.html`. Besucht am 13.10.2016.

[4] Jenkins press information. `https://jenkins.io/press/`. Besucht am 11.10.2016.

[5] Team foundation server. `https://www.visualstudio.com/en-us/docs/overview`. Besucht am 02.10.2016.

[6] Jenkins. `https://jenkins.io/doc/`, 05.10.2016. Besucht am 02.10.2016.

[7] Buildbot. `https://docs.buildbot.net/0.8.12/`, 16.09.2015. Besucht am 02.10.2016.

[8] Concourse: Ci that scales with your project. `https://concourse.ci/single-page.html`, 23.09.2016. Besucht am 02.10.2016.

[9] Gocd - open source continuous delivery server. `https://docs.go.cd/current/`, 29.09.2016. Besucht am 02.10.2016.

[10] P. M. Duvall, S. Matyas, and A. Glover. *Continuous integration: Improving software quality and reducing risk*. Addison-Wesley, 2007. ISBN: 0321336380.

[11] M. Fowler. Open-sourcing thoughtworks go. `http://martinfowler.com/articles/go-interview.html`, 2014. Besucht am 11.10.2016.

[12] J. Humble and D. Farley. *Continuous delivery: reliable software releases through build, test, and deployment automation*. Pearson Education, 2010. ISBN: 0321670221.

[13] Martin Fowler. Continuous delivery. `http://martinfowler.com/bliki/ContinuousDelivery.html`. Besucht am 22.09.2016.

[14] E. Wolff. *Continuous Delivery: Der pragmatische Einstieg*. dpunkt.verlag, 2016. ISBN: 9783864919312.

Wir danken unserem Sponsor der WVK 2016

Ihr Partner für Software Engineering

www.systecs.com