Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Computergestützte Modellierung und Simulation
Prof. Dr.-Ing. André Borrmann

Webbasierte Integration von Sensordaten in digitale Brückenbauwerke

Nepomuk Wolf

Bachelorthesis
für den Bachelor of Science Studiengang Umweltingenieurwesen

Autor: Nepomuk Wolf
Matrikelnummer: [redacted]
Betreuer: Sebastian Esser, M.Sc.

Ausgabedatum: 01. Januar 2020
Abgabedatum: 26. Juli 2020

Abstract

Brücken sind wichtige Komponenten unserer Infrastruktur. Sie ermöglichen Mobilität, Verkehr und Transport, ein Ausfall hat negative Konsequenzen für viele Bereiche. Ein komplettes Versagen dieser Bauwerke kann sogar katastrophale Folgen haben und viele Menschenleben kosten. Es ist daher von großer Wichtigkeit zuverlässige Monitoring-Lösungen zu entwickeln, um einen sicheren, dauerhaften Betrieb dieser Bauwerke zu gewährleisten.
Die Überwachung geschieht normalerweise überwiegend in festen Zeitabständen oder beim Feststellen eines Schadens durch Fachpersonal vor Ort und ist mit Ungenauigkeit und Aufwand verbunden. Daher wird viel Forschung betrieben, um die Überwachung einfacher, genauer und günstiger zu gestalten. Ein Ansatz ist dabei das sogenannte Structural Health Monitoring (SHM), bei dem der Zustand kontinuierlich und automatisiert mithilfe von Sensoren erfasst wird. Davon erhofft man sich weniger Wartungsarbeit und vor allem eine deutlich frühere Erkennung von Schäden. Gerade mit der schnellen Entwicklung in den Bereichen Sensortechnik und Internet of Things (IoT) gewinnt dieses Vorgehen immer mehr an Relevanz. Auch von staatlicher Seite werden deshalb viele Projekte dieser Art gefördert, in Auftrag gegeben oder mitgetragen. Langfristiges Ziel ist es, alle Bereiche des Verkehrs- und Infrastruktursektors der fortschreitenden Digitalisierung anzupassen. Systeme dieser Art sind bereits erfolgreich in Verwendung, werden allerdings noch nicht in der Breite eingesetzt. In der Baubranche liegt die Antwort auf die Digitalisierung und die steigenden Anforderungen in der immer häufigeren Verwendung von digitalen Bauwerksmodellen sowohl in der Planungs- und Bauphase als auch während des Betriebs. Die Entwicklung und Einführung dieser als Building Information Modeling (BIM) bekannten Technologie wird ebenfalls von staatlicher Seite gefördert.
Im Rahmen dieser Arbeit wurde eine webbasierte Anwendung entwickelt, die die beiden Bereiche (IoT und BIM) verknüpft. Zu diesem Zweck wurde eine Website eingerichtet, die sowohl das digitale Bauwerksmodell als auch Sensordaten visualisiert. Die räumliche Lage der Sensoren wird im Modell dargestellt und die Messwerte dementsprechend mit dem Bauwerk verbunden. Zusätzlich zur eigentlichen Monitoring-Funktion ist eine benutzerfreundliche Handhabung essenziell, damit Überwachungssysteme auch in der Praxis den größtmöglichen Nutzen erzielen. Deshalb wurde auf eine verständliche Darstellung des Modells, der Sensoren und deren Messergebnisse Wert gelegt. Diese Webanwendung demonstriert prototypisch die Integration von Sensordaten in Forge-Applikationen.
Im ersten Teil dieser Abschlussarbeit wird ein Überblick über das sensorgestützte Brückenmonitoring sowie ein Einblick in die aktuelle Forschungslage gegeben. Wichtige Themenbereiche bilden dabei, neben der Erfassung von Messwerten, die Übertragung, Verarbeitung, Speicherung und Darstellung von Sensordaten im Brückenmonitoring. Im zweiten Teil wird anschließend die Entwicklung der Anwendung vorgestellt und deren Funktionsweise dargelegt. Insbesondere werden das Sensornetzwerk, die Architektur der Server, die Datenbank und der Aufbau des Webinterfaces beleuchtet.

Inhaltsverzeichnis

Abkürzungsverzeichnis

2D	Zweidimensional
3D	Dreidimensional
ADC	Analog-to-Digital Converter
API	Application Programming Interface
BIM	Building Information Modeling
CAD	Computer Aided Design
CM	Condition Monitoring
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
GPIO	General Purpose Input Output
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IFC	Industry Foundation Classes
IoT	Internet of Things
JS	Java Script
JSON	Java Script Object Notation
NDT	Non Destructive Testing
NoSQL	Not only SQL
NPM	Node Package Manager
OSS	Object Storage Service
RDBMS	Relational Database Management System
REST	Representational State Transfer
RPI	Raspberry Pi
SHM	Structural Health Monitoring
SQL	Structured Query Language
TUM	Technische Universität München
UI	User Interface
URL	Uniform Resource Locator
WebGL	Web Graphics Library
XML	Extensible Markup Language

Teil I

Sensorgestütztes Brückenmonitoring

Kapitel 1

Einführung

Die sensorische Überwachung und Steuerung von Gebäuden ist eine gängiges Verfahren in der Industrie. Sowohl in der Bauphase als auch während des Betriebs werden in vielen Gebäuden, gleich ob es sich um Wohn-, Arbeits- oder öffentliche Gebäude handelt, unterschiedlichste Parameter wie z.B. Luftqualität, Temperatur, Feuchtigkeit oder Lichtverhältnisse erfasst. Diese Werte werden zur Steuerung von Gebäudefunktionen verwendet oder sie dienen dazu frühzeitig ein drohendes Versagen einzelner Systeme zu erkennen.
Aktuell wird dieses Vorgehen zunehmend auch auf andere Bauwerkstypen, insbesondere Infrastrukturbauwerke, ausgeweitet. Das Monitoring von z.B. Brückenbauwerken ist im Moment noch recht zeit- und arbeitsaufwändig und verlangt zusätzlich den Einsatz von Inspektoren vor Ort sowie unter Umständen aufwändige Prüfverfahren. Die Bundesanstalt für Straßenwesen leitet daher zur Zeit ein Forschungsprojekt mit dem Namen „Intelligente Brücke“, das sich der Frage widmet, in wie weit Brücken mit Hilfe von Sensoren automatisch überwacht werden können. Ziel ist es Mängel frühzeitig zu erkennen und Unglücke zu vermeiden während gleichzeitig der Arbeitsaufwand minimiert wird. Schadensindikatoren lassen sich so frühzeitig erkennen auch ohne Inspektion vor Ort. Spätestens mit dem Einsturz des Polcevera Viadukts in Genua im August 2018 ist der Nutzen und die Notwendigkeit für eine frühzeitige Schadenserkennung durch automatische Überwachung von Brücken deutlich im Bewusstsein der Gesellschaft angelangt. Besonders für das Katastrophenmanagement sind Echtzeitinformationen über den Zustand von Infrastrukturbauwerken von großer Wichtigkeit und ermöglichen dadurch erst konkrete und aktuelle Planung.
Diese wissenschaftliche Arbeit soll untersuchen, in wie weit eine Echtzeitüberwachung von Infrastrukturbauwerken, vor allem in der Betriebsphase, mit den momentan verfügbaren Mitteln umsetzbar ist. Diese Überwachung sollte sowohl benutzerfreundlich als auch kostengünstig sein. Zu diesem Zweck wird eine Anwendung mithilfe der Autodesk Platform „Forge“ erstellt, die dabei helfen soll Brücken oder andere Infrastrukturbauwerke einfacher zu überwachen.
Die Anwendung soll dazu in der Lage sein 3D Brückenmodelle zu importieren und in einem Viewer darzustellen. Zusätzlich soll es dem Nutzer möglich sein Sensoren verschiedener

Art in das digitale Modell einzufügen. Die tatsächliche Brücke (im Rahmen dieser Arbeit das prototypische Modell einer Brücke) wird mit den korrespondierenden Sensoren ausgestattet welche auf http-Anfrage die Sensordaten an die Forge-Anwendung weitergeben. Diese Daten werden dann in das Modell integriert und liefern so in Echtzeit Informationen über den Zustand der Brücke. Von kritischen Temperatur- und Feuchtigkeitswerten bis hin zu der Information, ob die Brücke überhaupt noch befahrbar ist, lassen sich je nach verwendeten Sensoren unterschiedliche Parameter überwachen und in der Anwendung graphisch darstellen. Dabei steht hier weniger ein konkreter Anwendungsfall im Vordergrund, sondern vielmehr die Überprüfung der Praktikabilität und Machbarkeit eines solchen Vorgehens anhand eines prototypischen Modells. Für das Auslesen, Verarbeiten und Weiterleiten der Sensordaten werden Raspberry Pis als Server verwendet. Diese sind sowohl relativ leicht zu handhaben, als auch kostengünstig.
Das Vorgehen könnte in seiner Methodik auf reale Brücken übertragen werden oder in größerem Maßstab auf ganze Infrastrukturnetze ausgeweitet werden und so eine nutzerfreundliche Echtzeitüberwachung ermöglichen.

Kapitel 2

Konventionelle Methoden

Es soll zunächst das konventionelle Vorgehen zur Überwachung von Brückenbauwerken dargestellt werden, um die Unterschiede zu einem sensorgestützten Monitoring zu verdeutlichen. Damit sollen die Vor- und Nachteile einer automatischen Überwachung veranschaulicht werden.

2.1 Visuelle Inspektion

Es gibt klassischerweise zwei Vorgehensweisen den Zustand einer Brücke zu überprüfen und einzuordnen. Zum einen kann die Brücke optisch inspiziert werden, zum anderen können spezielle Prüfverfahren verwendet werden (Sargsyan *et al.*, 2019). Diese lassen sich in zerstörende und zerstörungsfreie Verfahren kategorisieren, wobei in den meisten Fällen zerstörungsfreie Verfahren zum Einsatz kommen. Im Verlauf eines zerstörenden Prüfverfahrens wird das geprüfte Material, wie es der Name nahelegt, zerstört.
Die Inspektion durch Fachpersonal ist i.d.R. der erste Schritt einer Zustandsbestimmung und liefert eine grobe Einschätzung des (augenscheinlichen) Zustands der Brücke. Hierbei lässt sich ein Überblick über den Zustand gewinnen und optisch erkennbare Schäden können mit ihrer genauen Lage entdeckt und klassifiziert werden. In einzelnen Fällen reicht dieses Verfahren aus, um den Zustand der Brücke hinreichend einzuordnen und geeignete Maßnahmen für die Instandhaltung festzulegen und zu veranlassen (Kuchekar & Deshpande, 2017).
Kuchekar & Deshpande (2017) beschreiben das generelle Vorgehen und die verwendeten Werkzeuge für eine Inspektion folgendermaßen: Bevor mit der Inspektion begonnen werden kann, sollten eine Reihe von Informationen eingeholt werden, um die Ergebnisse der Inspektion zu ergänzen sowie festzustellen auf welche Bauteile und Parameter besonders geachtet werden sollte. Insbesondere sollten neben Bauplänen auch Protokolle vergangener Inspektionen und Reparaturen herangezogen werden. Ebenso geben Baujahr und Konstruktionsmethode Aufschluss über potenzielle Mängel und Schwachpunkte. Mit diesen Informationen kann die

Inspektion genauer auf den Zustand und die Art der Brücke abgestimmt werden und potenziell gefährdete Elemente können im Vorhinein identifiziert werden. Bei der Begehung werden optisch erkennbaren Mängel inspiziert und klassifiziert. Besonders wichtig ist dabei, ob es sich um strukturell relevante Defekte handelt, welche die Standsicherheit des Bauwerks gefährden. Dabei werden in erster Linie Schadensmerkmale wie Risse, Abrasionen, Korrosionen und Konstruktionsfehler erkannt. Alle Mängel werden in einem Bericht protokolliert und mittels fotographischer Dokumentation ergänzt. Zusätzlich müssen die Fundamente überprüft und der Zustand des Bodens festgehalten werden. Es ist notwendig, die Brücke auf Abweichungen von den Bauplänen hin zu überprüfen.
Eine Anwendung dieser Vorgehensweise ist z.B. bei Agarwal *et al.* (2019) an der Kund-Mala Bridge in Indien zu finden. Die Genauigkeit und Zuverlässigkeit von visuellen Brückeninspektionen im Allgemeinen wurde von Phares *et al.* (2004) untersucht. Zu diesem Zweck wurden die Inspektionsprotokolle von 49 Ingenieuren/innen miteinander verglichen. Die Inspekteure untersuchten dabei zehn unterschiedliche Brücken mit den gängigen zerstörungsfreien Prüfmethoden. Die Protokolle bestanden aus handschriftlichen Notizen, Photos und Zustandsbewertungen sowohl übergeordneter Strukturen, als auch einzelner Bauteile. Phares *et al.* (2004) kamen zu dem Ergebnis, dass sowohl die Bewertung des Gesamtzustandes als auch die Dokumentation der vorhandenen Schäden und deren Umfang zwischen den Inspekteuren signifikant variierten. Die Bewertung ist stark subjektiv und in mindestens 56% der Fälle inkorrekt. Außerdem variierte die Anzahl der aufgenommenen Photos stark zwischen einer bis neunzehn Aufnahmen. Daraus wird deutlich, dass eine Inspektion keine eindeutigen Ergebnisse liefert. Schäden werden teilweise nicht erkannt oder von unterschiedlichen Inspektoren verschieden bewertet. Die Inspektion als „primitive Methode“ (Kuchekar & Deshpande, 2017) liefert einen ersten Eindruck des Zustands eines Bauwerkes, der in den wenigsten Fällen vollkommen zutreffend oder vollständig ist. Sie hilft aber festzulegen, welche Bauteile welcher Art von Tests unterzogen werden müssen, um eine genaue Aussage über den Zustand machen zu können (Kuchekar & Deshpande, 2017).

2.2 Spezielle Prüfverfahren

Aus den oben genannten Gründen stellt die visuelle Inspektion meist nur den ersten Schritt dar, gefolgt von einer Reihe spezieller Prüfverfahren, welche den genauen Zustand der gesamten Struktur und einzelner Bauteile genauer bestimmen. Die in der visuellen Inspektion festgestellten Auffälligkeiten werden dabei mit den geeigneten Methoden näher untersucht, um Schäden bestätigen, lokalisieren und quantifizieren zu können. Es kommen dabei eine Vielzahl verschiedener Methoden zum Einsatz, je nachdem um welches Material es sich handelt und welche Art von Schaden erkannt werden soll. Eine Zusammenstellung der wichtigsten zerstörungsfreien Methoden (engl: Non Destructive Testing (NDT)) ist in Abbildung 2.1 zu

finden. Die Auswahl ist hier auf Beton und Stahl beschränkt, da es sich dabei um die Hauptbaustoffe von modernen Brückenbauwerken handelt.

Concrete NDT Methods		
Acoustic Emission	Impact Echo Testing	Nuclear Method
Electrical (half-cell) Method	Infrared Thermography	Pachometer
Delamination Detection Machinery	Laser Ultrasonic Testing	Rebound and Penetration
Ground Penetrating Radar	Magnetic Method	Smart Concrete
Electromagnetic Methods	Neutron Probe	Ultrasonic Testing
Pulse Velocity		

Steel NDT Methods	
Acoustic Emission	Radiographic Testing
Corrosion Sensors	Ultrasonic Testing
Smart Paint	Eddy Current
Penetrant Testing	Robotic Inspection
Magnetic Particle	

Abbildung 2.1: Zerstörungsfreie Prüfmethoden für Beton und Stahl nach Lee & Kalos (2015)

Auf die einzelnen Methoden soll im Rahmen dieser Arbeit nicht näher eingegangen werden. Es soll lediglich ein grober Überblick über die verwendeten und zur Verfügung stehenden Methoden zur Zustandsbestimmung von Beton und Stahlkonstruktionen gegeben werden. Einige dieser Methoden können ebenfalls im Kontext einer sensorgestützten Überwachung automatisiert angewendet werden, in der Regel kommen aber andere Testverfahren zum Einsatz (vgl. 3.3). Wichtig ist zu bemerken, dass diese Prüfverfahren vor Ort von einem Fachkundigem, beispielsweise Ingenieur oder Techniker, ausgeführt werden müssen.
Die gewonnenen Daten der visuellen Inspektion, und auch der speziellen Prüfverfahren, die daraufhin durchgeführt werden, liegen in Form von Protokollen, meist handschriftlich, vor und werden selektiv manuell digitalisiert oder analog abgelegt. Einige Messgeräte speichern Ergebnisse intern ab und müssen dann einzeln ausgelesen werden. Auch hier geschieht die Digitalisierung und Speicherung der Daten nicht automatisch, sondern erfordert Arbeit der Inspekteure oder Ingenieure. Es gibt damit zunächst keinen zentralen Ort, an dem die Daten zu einem Objekt zur weiteren Verarbeitung oder Speicherung zusammengeführt werden. Weiterhin werden Schäden, die zwischen einzelnen Inspektionsterminen auftreten, erst bei der nächsten Inspektion und nicht bereits beim Entstehen bemerkt. Genau an diesen Punkten bringt ein sensorgestütztes Monitoring Vorteile und gewinnt deshalb immer mehr an Bedeutung. Ob und inwieweit eine automatische Überwachung eine Inspektion vor Ort komplett ersetzen oder nur als Ergänzung dieser eingesetzt werden kann, ist eine andere Frage und hängt auch mit den jeweiligen rechtlichen Rahmenbedingungen zusammen.

Kapitel 3

Sensorgestütztes Monitoring

Das Konzept der permanenten automatischen Überwachung eines Objektes, um dessen Zustand zu überprüfen, ist in anderen Branchen bereits geläufig und wird serienmäßig durchgeführt. Beispielsweise im Bereich Maschinenbau und insbesondere bei der Luftfahrzeuginstandhaltung wird diese Methode standardmäßig — unter dem Namen Condition Monitoring (CM) — eingesetzt. Die Intention dahinter ist die Verringerung von Wartungsarbeiten und eine frühere und umfassendere Erkennung von Schäden. Auch im Bereich von Infrastrukturbauwerken wie z.B. Brücken steigt das Interesse nach einer automatisierten, kontinuierlichen Zustandsbestimmung um Schäden und Gefahren frühzeitig erkennen oder gar prognostizieren zu können. Die automatisierte, kontinuierliche Überwachung von Bauwerken mit Hilfe von Sensoren und der dazugehörigen Datensammlung und -verarbeitung wird als Structural Health Monitoring (SHM) bezeichnet.

> *„Structural health monitoring is the integration of a sensory system, a data acquisition system, a data processing and archiving system, a communication system, a damage detection system and a modeling system to acquire knowledge about the integrity and load-worthiness of in-service structures on either a temporary or continuous basis. The objective of an SHM system is to monitor the behaviour of a structure accurately and efficiently, so as to assess its performance under various service loads, to detect damage or deterioration, and to determine the health or condition of a structure.“* (Bakht & Mufti, 2015)

Brückenbauwerke sind dabei ein häufiges Zielobjekt solcher Systeme. Ein Versagen kann hier katastrophale Folgen nach sich ziehen. Die Zustandsüberwachung ist daher grundsätzlich vorgeschrieben und wird in der Regel durch periodisch durchgeführte Inspektionen umgesetzt. SHM bietet dagegen den Vorteil, dass auch die Schäden und Probleme, die zwischen den einzelnen Inspektionsterminen auftreten, erkannt werden. Zum Thema SHM wurde und wird viel Forschungsarbeit geleistet und es existieren einige Forschungsprojekte speziell zu SHM bei Brückenbauwerken. Beispielsweise hat die Bundesregierung einen Projektcluster mit dem

Namen „Intelligente Brücke“ in Auftrag gegeben, an dem unter anderem auch die Technische Universität München (TUM) mitgewirkt hat. Trotz der vielfältigen Forschung und Pilotprojekte in diesem Gebiet (vgl. hierzu Kapitel 5) finden SHM-Systeme bei Brückenbauwerken in der Praxis aber noch keine allzu breite Verwendung (Webb *et al.*, 2015).

3.1 Gründe und Zielsetzungen

Wie im Kapitel 2 festgestellt, bringt die visuelle Inspektion und die Schadensüberwachung mit periodisch durchgeführten Tests gewisse Probleme mit sich. Zum einen findet die Datenerhebung nicht kontinuierlich, sondern in festgelegten Zeitabständen statt, zum anderen bietet die subjektive Bewertung durch verschiedene Inspekteure gewisse Ungenauigkeiten und birgt damit die Gefahr Defekte zu übersehen. Aus diesen Gründen bietet sich eine sensorgestützte Überwachung besonders wichtiger Infrastrukturbauwerke an, bei denen ein Versagen unbedingt vermieden werden muss.
Es gibt inzwischen eine ganze Reihe an Forschungsprojekten und Artikeln zum Themenbereich sensorgestütztes Brückenmonitoring die daran arbeiten Methoden zu entwickeln und zu erforschen um Brücken langfristig durchgängig automatisch zu überwachen. Trotz der großen Präsenz dieses Themas in der Forschung ist der flächendeckende Übergang in die Praxis bisher noch nicht erfolgt. Dass die meisten Brücken nach wie vor manuell mittels Inspektionen vor Ort überwacht werden hat mehrere Gründe. Nicht zuletzt ist die praktische Umsetzung für den Betreiber finanziell oft nicht lukrativ und mit technischen Schwierigkeiten verbunden. Es müssen teilweise kostenintensive Sensoren und Hardware verbaut werden. Die Verarbeitung der Daten, die Bereitstellung und Handhabung der Software sowie die konkrete Berechnung des Zustandes der Brücke aus den gemessenen Daten ist unter Umständen sehr aufwändig. Der Mehraufwand durch Schulung des Personals (in neue Technologien und Software) und Investitionskosten kann ein Hindernis darstellen. Daher wird oft auf vertraute Methoden zurückgegriffen statt auf neue Lösungen zu setzen.
Dennoch gibt es gute Gründe ein SHM-System zu installieren, wie es schon der standardmäßige Einsatz in anderen Branchen beweist. Zum einen vermeiden automatische Monitoring Systeme die erwähnten Unsicherheiten und Ungenauigkeiten einer Visuellen Inspektion. Bei korrekter Funktion der Sensoren sind die gelieferten Werte und damit auch die abgeleitete Zustandsbewertung im Rahmen der Sensorgenauigkeit eindeutig. Bei gleichen Bedingungen und gleichem System ergibt sich damit auch die gleiche Bewertung, was sowohl eine bessere Vergleichbarkeit als auch eine höhere Konsistenz der Ergebnisse mit sich bringt. Zum anderen werden Schäden, die zwischen den Inspektionen auftreten bei einer Visuellen Inspektionsstrategie erst bei der nächsten geplanten Inspektion erkannt. Ein dauerhaftes Monitoring erkennt diese Schäden direkt bei der Entstehung. Auch können durchgängigen Messreihen die durch SHM produziert werden für bestimmte Schadensarten aussagekräftiger sein als Momentaufnahmen des Zustands.

Die Gründe ein SHM-System einzuführen hängen eng mit der Zielsetzung und dem erwarteten Nutzen zusammen. Webb *et al.* (2015) definieren fünf Nutzungskategorien in die sich laut den Autoren alle vorliegenden SHM-Systeme (bei Brückenbauwerken) einordnen lassen. Demnach ist für den Erfolg eines solchen Projektes entscheidend die Zielsetzung vor Beginn des Projektes klar zu definieren um das System genau auf den gewünschten Nutzen auszurichten. Diese sind im Folgende kurz vorgestellt.

1. **Anomaly Detection:** Das Auffinden von Anomalien oder Unregelmäßigkeiten in den Sensorwerten stellt eine verhältnismäßig primitive Methode des Monitoring dar. Erfasst werden hier nur Änderungen der überwachten Parameter. Eine Schwierigkeit besteht darin, die Änderungen der Werte, welche durch Umwelteinflüsse und normale Belastung hervorgerufen werden von denen zu trennen, die durch Schäden verursacht werden.

2. **Sensor Deployment Studies:** Diese Kategorie befasst sich primär mit der Erforschung oder Demonstration neuer Sensortechnologie und weniger mit dem Zustand des Bauwerks. Es können so Ansatzpunkte zur Weiterentwicklung und Verbesserung der Sensortechnik geliefert werden.

3. **Model Validation:** Bei der Entwicklung und dem Entwurf eines Bauwerks werden zwangsläufig Annahmen gemacht und Modellparameter festgelegt. Diese Annahmen auf ihre Richtigkeit zu überprüfen bzw. zu falsifizieren ist für die Forschung und die Entwicklung neuer Bauwerke sehr wertvoll. Eine Reihe von SHM-Systemen dienen daher dem Zweck die gemessenen Sensorwerte mit dem Modell des Bauwerks zu verknüpfen, um Aussagen über die getroffenen Designannahmen zu machen.

4. **Threshold Check:** Eine sehr einfache Methode die Sensorwerte zu interpretieren, ist die Überprüfung von Grenzwerten. Das System warnt, sobald ein bestimmter Wert überschritten wird oder kumulativ eine kritische Belastungs- oder Schadenmenge erreicht ist.

5. **Damage Detection:** Die letzte und — aus der Perspektive der Datenverarbeitung — komplexeste Kategorie ist die direkte Erkennung und Identifizierung von Schäden. Dabei macht es einen großen Unterschied ob jede Art von möglicherweise auftretendem Schaden erkannt werden soll, oder vordefinierte Schadenszenarien existieren über die sich die Überprüfung einschränken lässt. In jedem Fall sind wesentlich aufwändigere Algorithmen notwendig um Art, Ort und Ausmaß auftretender Schäden verlässlich zu erkennen.

Zweifelsohne ist auch eine Kombination mehrerer Kategorien innerhalb eines SHM-Projektes möglich. Wichtig ist lediglich, dass die Zielsetzung im Vorfeld klar definiert wird. Es besteht sonst die Gefahr, Daten zu sammeln, die auf den ersten Blick sinnvoll erscheinen bei späterer Datenauswertung aber keine brauchbaren Informationen liefern (Webb *et al.*, 2015).

3.2 Aufbau eines SHM Systems

Zwischen den verschiedenen SHM-Systemen bestehen große Unterschiede hinsichtlich der verwendeten Hardware (z.B. Sensortypen, Mikrokontroler, Speichermedien usw.) und der implementierten Software (Algorithmen, Datenbank, Programmierungssprache). Dennoch sind bei fast allen Systemen die grundlegenden Komponenten und Prozesse identisch(Dhakal *et al.*, 2013). Die folgenden Elemente sind stets vorzufinden:

- **Datensammlung:** Verwendet werden Sensoren abhängig von der Zielsetzung des Systems.
- **Datenübertragung:** beispielsweise die Verkabelung der Elemente bzw. Verknüpfung über kabellose Netzwerke sowie die Entscheidung in welchem Format die Daten übertragen werden.
- **Datenspeicherung:** Typ und Architektur der Datenbank sowie Speichermedien fallen unter diese Kategorie.
- **Datenverarbeitung:** Das Filtern und Verstärken von Signalen oder das Bereinigen von Datenreihen ist hier einzuordnen.
- **Datenauswertung:** Hierzu zählen unter anderem die Algorithmen zur Schadenserkennung.

Zu ergänzen ist hier noch, dass die Daten bzw. die Auswertung zugänglich gemacht werden muss, um genutzt werden zu können. Dies kann in Form von manuellem Mining aus der Datenbank geschehen, oder aber durch Aufarbeitung und Visualisierung der Daten über eine Plattform. Je nach Art der Daten und Intention der Überwachung lassen sich benutzerspezifische Datenbankabfragen erstellen und die Daten als Einzelwerte, tabellarisch oder in Form von Diagrammen darstellen (vergleiche Hierzu Kapitel Datendarstellung). Bei Glisic *et al.* (2010) findet sich eine Zusammenstellung aller zentralen Aufgaben innerhalb eines SHM Systems bezogen auf den gesamten Lebenszyklus. Diese Liste geht über die oben beschriebenen grundlegenden Komponenten hinaus und beinhaltet auch Prozesse in der Planungsphase, dem Betrieb und dem Rückbau. Sie ist deshalb wesentlich umfangreicher. Der grundsätzliche Aufbau mit den zentralen Komponenten wurde im zweiten, dem praktischen Teil, dieser Arbeit umgesetzt. Hierbei fanden Datenverarbeitung und Datenauswertung nur minimale Beachtung.

Core Monitoring Activities				
Monitoring strategy	**Installation**	**Maintenance**	**Data management**	**Closing activities**
Monitoring aim	Installation of sensors	Providing for electrical supply	Measurements	Interruption of monitoring
Selection of parameters	Installation of accessories (cables…)	Providing for communication lines	Storage of data	Dismantling of monitoring system
Selection of monitoring systems	Installation of reading units	Implementation of maintenance plans for hardware	Providing for access to data	Storage of monitoring components
Design of sensor network	Installation of software	repairs and replacements	Viszalization	
Schedule of monitoring	Interfacing with users		Export of Data	
Data exploitation			Data Interpretation and analysis	
Costing			The use of data	

Abbildung 3.1: Haupt- und Unterkategorien der Komponenten und Aufgaben innerhalb eines SHM Systems nach Glisic *et al.* (2010)

3.3 Überwachte Merkmale und verwendete Sensor Typen

Die verwendeten Sensortypen sind nicht nur aus Perspektive des verantwortlichen Ingenieurs wichtig, sie spielen auch eine große Rolle für den Entwickler des SHM-Systems auf Ebene der Datenübertragung und -speicherung sowie für die Darstellung der Messwerte und Ergebnisse. Die genutzten Sensoren sind direkt für die Art und Menge der zu verarbeitenden Daten verantwortlich und damit für das Design und die Dimensionierung von Speicherlösungen und Übertragungskanälen äußerst relevant. Aus diesem Grund lohnt sich ein Blick auf die häufig verwendeten Sensortypen, auch wenn der Fokus dieser Forschungsarbeit nicht auf Tragwerkslehre oder Sensortechnik liegt. Auch die Darstellung hängt maßgeblich von der Art der Daten ab. Temperaturdaten können z.B. anders visualisiert werden als die Daten von Beschleunigungssensoren für die Vibrationsmessung. Die Architektur und das Design einer Anwendung für SHM-Systeme müssen daher berücksichtigen, welche Sensoren Verwendung finden sollen und welche Daten zu erwarten sind.

Die verwendeten Sensortypen und die zu überwachenden Merkmale stehen daher in direkter Abhängigkeit zueinander. Die Festlegung welche Werte erfasst werden sollen diktiert welche Sensortypen verwendet werden müssen. Erfasst werden häufig geometrische Parameter wie z.B. Verschiebung oder Rotation, allerdings spielen auch Umwelteinflüsse bei der

Überwachung eine große Rolle. Die verwendeten Sensoren und die gewünschten Messwerte sind sehr vielfältig und abhängig vom Ziel der Überwachung. Es lassen sich aber einige Sensorkategorien hervorheben, da sie in SHM-Projekten bei Brückenbauwerken häufig Verwendung finden. Nach Rio *et al.* (2013) können folgende Kategorien als besonders relevant identifiziert werden:

- **Verschiebungsensoren:** Die Verschiebung zweier Punkte zueinander ist eine häufig verwendete Messgröße. Es lassen sich so z.B. Entfernungsänderungen oder das Verrücken von Bauteilen bestimmen. Angewendet werden hierzu z.B. Differentialtransformatoren oder die laserbasierte Distanzmessung.
- **Rotationsensoren:** Mit Rotationssensoren wird die relative Lage zweier Elemente zueinander, oder die absolute Neigung eines Bauteils in Abhängigkeit einer fixierten Referenzebene gemessen.
- **Dehnungsensoren:** Die Dehnung wird mittels Dehnungsmessstreifen, Piezoresistoren oder Schwingsaitenaufnehmern gemessen. Durch die Längenveränderung im Verhältnis zur ursprünglichen Gesamtlänge eines Bauteils, kann auf die vorliegende Spannung geschlossen werden.
- **Temperatursensoren:** Die Temperatur ist in vielen Projekten eine wichtige Messgröße. Sie wird häufig benötigt um Werte anderer Sensoren zu validieren oder zu normalisieren. Einige Parameter wie z.B. das Schwingungsverhalten oder Längenänderungen können ebenso stark durch Temperaturvariationen wie durch Schäden beeinflusst werden (Farrar & Jauregui, 1998). Es ist daher für einige Messverfahren notwendig, die Temperatur mit zu erfassen um die Daten von dadurch hervorgerufenen Effekten zu bereinigen.

Diese Auflistung kann noch um die folgenden zwei Punkte ergänzt werden, da die nachfolgenden Sensortypen in der mehrzahl der Projekten, die im Rahmen dieser Arbeit betracht wurden, Verwendung fanden.

- **Beschleunigung:** Eine zentrale Rolle nehmen Beschleunigungssensoren bzw. Schwingungsaufnehmer ein. Bauteile oder sogar größere Strukturen wie komplette Bauwerke besitzen ein spezifisches Schwingungsverhalten. Das Schwingungsverhalten(Resonanzfrequenz, Modaldämpfung oder Schwingungsform) eines Elements ist abhängig von den physischen Eigenschaften wie Masse oder Steifigkeit. Form und Frequenz der Schwingung sind dabei ebenfalls abhängig vom Zustand des Materials bzw. des Bauteils. Durch Schäden im Bauwerk oder defekte Verbindungen verändert sich das Schwingungsverhalten. Der Vergleich der aktuellen Schwingungsform mit der Schwingungsform des schadenfreien Bauteils gibt so Aufschluss über dessen Zustand (Farrar & Jauregui, 1998).

- **Piezoelektrische Sensoren** Piezoelektrische Sensoren sind keinem speziellen Parameter zuzuordnen. Sie können verschiedene Wertveränderungen wie Druck, Temperatur, Beschleunigung oder Spannung feststellen. Da sie bei vielen Forschungsprojekten zum Einsatz kommen, sollen sie auch hier Erwähnung finden. So werden piezoelektrische Sensoren für das Brückenmonitoring z.B. bei Abdelgawad & Yelamarthi (2017),Alavi *et al.* (2019), Aono *et al.* (2019), Aono *et al.* (2016) oder Hasni *et al.* (2018) verwendet.

Ferner existiert neben diesen häufig verwendeten Sensortypen, eine Vielzahl von alternativer Sensoren, die in anderen Projekten Verwendung finden. Zusätzlich zur Temperaturmessung werden weitere Klimadaten erhoben. Niederschlag und Feuchtigkeit beispielsweise stellen häufige Ursachen für die Korrosion von Bauteilen dar. Darüberhinaus fungiert Wind mitunter als große Lastquelle. Diese Umweltparameter helfen Störgrößen aus anderen Messergebnissen zu beseitigen. Ebenso finden GPS und Kameras Verwendung beim Monitoring. Insbesondere, wenn Faktoren wie der betriebliche Auslastungszustand eine Rolle spielen und daher erfasst werden sollen, können Kameras (in Kombination mit automatischer Bildverarbeitung- und Auswertung) verwendet werden. Eine Zusammenstellung mehrerer SHM-Projekten bei Brückenbauwerken zusammen mit den verwendeten Sensortypen findet sich bei Webb *et al.* (2015)

3.3.1 Sensorgenauigkeit und Wartung

Messwerte sind immer mit gewissen Unsicherheiten und Messfehlern behaftet und auch Vorhersagemodelle bilden die Realität nicht perfekt ab, sondern stellen nur eine Annäherung dar. Diese Unsicherheiten können nicht vermieden werden, gefährden aber die Aussagekraft der Ergebnisse des Systems. Es ist daher wichtig, die Fehlerquellen so klein wie möglich zu halten und vor allem die Fehler zu quantifizieren. So kann gewährleistet werden, dass die Sicherheit oder Wahrscheinlichkeit der Richtigkeit von Ergebnissen bekannt ist. Nur so ist eine Entscheidungsfindung und das Festlegen geeigneter Maßnamen auf Grundlage der vom SHM-System gelieferten Ergebnisse sinnvoll möglich (Webb *et al.*, 2015).
Die Genauigkeit von Sensoren wird in der Regel vom Hersteller angegeben und wird in der Produktion regelmäßig überprüft. Dennoch kann während des Betriebs die Genauigkeit der Sensoren mit der Zeit nachlassen, so dass verbaute Sensoren regelmäßig überprüft, geeicht und gegebenenfalls gewartet werden müssen. Diese Überwachung kann manuell erfolgen, d.h. durch periodisch durchgeführte Inspektionen. Dies birgt den Nachteil, dass der manuelle Inspektionsaufwand vom Bauwerk auf das Monitoring-Netz verlagert wird. Auch wenn der Aufwand so potenziell geringer ist, kann eine Inspektion vor Ort nicht vermieden werden.
Eine andere Methode besteht darin das Monitoring der Sensoren in das Netzwerk an sich zu integrieren. Das kann auf zwei verschiedene Arten durchgeführt werden. Entweder es werden Sensoren verwendet, die selbst ihren Zustand validieren können oder das Sensornetz wird so

konzipiert, dass die Auswahl und Lage der Sensoren eine gegenseitige Überprüfung der korrekten Funktion ermöglichen (Farrar & Worden, 2007). Diese Fähigkeit zur Selbstüberprüfung der Sensoren oder ganzer Systemteile garantiert die Funktionsfähigkeit durchgängig und ist daher und auch wegen des geringeren Aufwandes einem manuellen Monitoring überlegen. Erfasst werden sollte dabei sowohl der Zustand des kompletten Systems, als auch der Zustand aller wichtigen Teilkomponenten wie Sensoren, Verkabelung oder Energieversorgung (Glisic *et al.*, 2014).

3.4 Datenspeicherung

Ein zentraler Baustein eines jeden IoT-Systems ist die Lösung zur Speicherung und Bereitstellung der erfassten Daten. Eine geeignete Datenbank muss den Anforderungen des Projektes gerecht werden. Dabei spielen mehrere Fragen eine Rolle: Wie viele Daten fallen an? Welche Struktur haben die Daten? Wie oft soll geschrieben und ausgelesen werden? Wie stark sind die Daten verknüpft? Abhängig davon, kann eine passende Speicherlösung gewählt werden.

3.4.1 Datenmenge

> *„The amount of data generated is a function of the number of sensors and of their acquisition and sampling rates. It can vary by large amounts depending if the sensors are mainly static with lower sampling rates, or dynamic, with higher sampling rates. The management of all the original raw data, as well as of all the post-processed data, during the entire life-cycle of the infrastructure, that can add up to hundreds of Giga, of even Terabits, can constitute a problem, in particular if all the original data is kept for future processing.“* (Rio *et al.*, 2013)

Bei der dauerhaften Überwachung von Bauwerken können große Mengen an Daten generiert werden. Die konkrete Datenmenge hängt von folgenden Faktoren ab und kann über diese beeinflusst werden.

- **Sensoranzahl:** Je größer das Bauwerk und je umfassender das Monitoring, desto größer die Datenmenge. Die Anzahl der Sensoren ist auch davon abhängig, welche Schadensszenarien erkannt werden sollen, bzw. welche Parameter überwacht werden und wie dicht das Sensornetzwerk aufgebaut ist.
- **Auslesefrequenz:** Je öfter die Sensoren ausgelesen werden, desto größer ist die Datenmenge. Datentyp und Genauigkeit der Daten: Wie viel Speicherplatz einzelne Variablen einnehmen, variiert stark zwischen den Datentypen. Unter MySQL z.B. liegen TINYINT bei 1 Byte Speicherverbrauch pro Wert und können durchaus geeignet sein relativ grobe Temperaturwerte zu speichern. Sollen Werte mit höherer Genauigkeit oder aus einem größeren Wertebereich abgespeichert werden, müssen andere Datentypen mit höherem Speicherverbrauch gewählt werden wie FLOAT (4 Byte) oder INT (4 Byte).
- **Kompression:** Einige Daten lassen sich zur Speicherung komprimieren, so verwenden Tong *et al.* (2019) z.B. eine Fourier Transformation für die Kompression von Schwingungsdaten. Hier muss eine Abwägung stattfinden zwischen Geschwindigkeit der Kompression und gewünschter Kompressionrate.

- **Vorfilterung:** Werden die Daten vor der Speicherung gefiltert, können unter Umständen nur Daten gespeichert werden, die für die Anwendungsziele relevant sind. So ist es z.B. denkbar Werte, die sich in einem definierten normalen Bereich bewegen zu verwerfen und nur solche abzuspeichern, die gewisse Grenzwerte überschreiten oder andere Bedingungen erfüllen.

Wie umfangreich die Datenmenge tatsächlich ausfällt, lässt sich also beim Anlegen des Systems durch Designentscheidungen beeinflussen.

3.4.2 Datenbanktypen

In der Praxis lassen sich die verwendeten Datenbanken größtenteils in zwei Kategorien einordnen: Relationale (Structured Query Language (SQL)) und nicht-relationale Datenbanken (Not only SQL (NoSQL)). In diesem Abschnitt soll auf die Unterschiede sowie die Vor- und Nachteile beider Datenbanktypen eingegangen werden.

Relationale Datenbanken

In relationalen Datenbanken werden die Daten in Form von sogenannten Relationen abgespeichert. Relationen können in Form von Tabellen dargestellt werden. Der Name der Tabelle wird Entität genannt. Jede Spalte dieser Tabelle beschreibt ein Attribut. Die Datensätze werden als Tupel bezeichnet und bestehen aus allen Attributswerten eines Elementes. In der Tabelle stellen sie die Zeilen dar. Dabei muss i. d. R. für jedes Attribut der Relation auch ein passender Wert vorhanden sein. Die Werte, die das zugehörige Attribut in einem Tupel beschreiben, können auf einen bestimmten Wertebereich beschränkt werden. Dieser Bereich wird als Domäne bezeichnet. Es können hier aber auch Nullwerte (nicht die Zahl Null) zugelassen sein, sodass einzelne Attribute in einem Tupel nicht mit Werten belegt sein müssen (Steiner, 2009). Ein Tupel muss eindeutig identifizierbar sein, d.h. auch er darf nicht mehrmals in einer Relation vorkommen. Aus diesem Grund wird jeder Relation ein Schlüssel zugeschrieben. Dieser Schlüssel ist ein Attribut oder eine Menge aus Attributen, welche für jeden Tupel einen eindeutigen Wert annehmen. Dieser Schlüssel kann neben einem natürlichen Attribut, das sich ebenfalls als Schlüssel eignet, auch ein künstliches Merkmal sein, welches speziell erzeugt wird, um einen Datensatz eindeutig zu identifizieren (Meier, 2007). Die Matrikelnummer ist ein adäquates Beispiel für einen künstlichen Schlüssel. In einer Datenbank gibt es beliebig viele dieser Relationen, die untereinander in Beziehung stehen können. Über den Assoziationstyp ist dabei festgelegt, wie viele Tupel aus einer Relation einem Tupel einer anderen Relation zugewiesen werden können.
Relationale Datenbanken haben einige Mängel; insbesondere, wenn es um Flexibilität geht. Relationen müssen mit allen Attributen im Vorhinein definiert werden und ein späteres Einfügen neuer Attribute ist nur schwer möglich. Auch der Datentyp der für ein Attribut eingesetzt werden kann, muss von vornherein festgelegt werden und kann im Nachhinein

nicht mehr an geänderte Nutzungsanforderungen angepasst werden. Treten Daten mit einem anderen Typ auf, kann nicht ohne weiteres darauf reagiert werden. Zusätzlich kann es bei komplexen Relationsmodellen zu Performanceproblemen kommen. Das liegt daran, dass bei Abfragen, Tabellen miteinander verknüpft werden müssen und aus dieser Gesamtmenge erst die Ergebnisse gefiltert werden. Dadurch kann es unter Umständen zu sehr großen temporären Speicheraktionen kommen, die neben Speicherplatz auch Rechenzeit in Anspruch nehmen (Lapp, 2017).

Nicht-Relationale Datenbanken

Mit der Anforderung neben klassischen Datensätzen auch Textdokumente, Graphiken und Bilder in derselben Datenbank abzuspeichern, geht der Bedarf nach einer erweiterten Datenbanktechnologie einher und damit auch die Entwicklung sogenannter NoSQL Datenbanken. Der Begriff NoSQL ist nicht eindeutig definiert und abgegrenzt wie es bei SQL der Fall ist. Daher werden mehrere verschiedene Datenbanksysteme darunter zusammengefasst. Die Daten selbst werden je nach Typ als Schlüssel-Wertepaare (in Dokumentenspeichern), Spalten/-Spaltenfamilien oder als Graphen abgespeichert. Das zugrundeliegende Modell ist dabei nicht (oder nicht nur) relational. Daten können als zusammenhangslose Elemente abgelegt werden. Das Lesen, Schreiben oder Ändern eines Elementes ist damit potenziell deutlich schneller. Es ist somit möglich Datensätze und Datentypen ganz unterschiedlicher Art zusammen abzuspeichern. Einsatz finden Datenbanken dieser Art insbesondere im Bereich Big Data, wenn die Daten nicht zentral sondern auf unterschiedlichen Systemen zugleich abgespeichert werden (Meier, 2018). Dies lässt sich darauf zurückführen, dass NoSQL-Systeme sich deutlich leichter skalieren und auf unterschiedliche Systeme aufteilen lassen (Lapp, 2017).

Relationale Datenbanken sind bislang wesentlich verbreiteter als NoSQL Lösungen (Meier, 2018) vor allem, bezüglich des Speicherns verhältnismäßig kleiner Datenmengen. Insbesondere bei Sensordaten fallen die Nachteile von relationalen Datenbanken nicht stark ins Gewicht. Messwerte lassen sich sehr gut in einem relationalen Datenschema darstellen und bleiben selbst, unabhängig vom eigentlichen Wert, Struktur und Datentyp, konstant. Zusätzlich existiert für relationale Datenbanken ein breitgefächertes Angebot an Schnittstellen und Zwischen-Layern, die eine flexible Einbindung ermöglichen. Bei NoSQL Datenbanken ist die Implementierung hingegen in einigen Fällen komplizierter (Lapp, 2017). Es ist daher nicht außergewöhnlich, dass auch für die meisten SHM-Projekten bei Brückenbauwerken relationale Datenbanken Verwendung finden.

3.5 Datenverarbeitung

Die Verarbeitung der Daten stellt einen weiteren wichtigen Schritt bei der Einrichtung eines SHM-Systems dar. Die rohen Messdaten müssen oftmals für die weitere Verwendung vorbereitend bearbeitet werden. Welche Verarbeitungsschritte genau notwendig sind, hängt von den Daten selbst und der Zielsetzung des Monitorings ab, d.h. welche Informationen letzlich gewonnen werden sollen. Außerdem ist die Qualität, in der die Daten vorliegen ausschlaggebend. Im Rahmen der praktischen Arbeit (die den zweiten Teil dieser Abschlussarbeit darstellt) wird keine Datenverarbeitung implementiert, die Sensordaten werden direkt in die Datenbank eingepflegt. Da die Datenverarbeitung bei den meisten Monitoring-Projekten jedoch eine relevante Rolle spielt, wird an dieser Stelle ein grober Überblick vermittelt.

- **Digitalisierung analoger Messwerte:** Einige Sensortypen wandeln analoge Messwerte bereits automatisch in ein digitales Format um. Andere liefern indessen analoge Messwerte. Zur weiteren Verwendung in einem Computersystem müssen diese Werte mithilfe eines Analog-to-Digital Converter (ADC) umgewandelt werden. Farrar *et al.* (2006) listen daher auch ADCs als elementare Komponenten des SHM-Systems.

- **Herausrechnen von Störgeräuschen aus den Signalen:** Messwerte von Sensoren sind mit Störungen behaftet, die durch unterschiedliche Einflüsse hervorgerufen werden können. Einerseits verfügen die Sensoren selbst nur eine gewisse Genauigkeit, andererseits werden häufig auch Effekte von den Sensoren erfasst, die nicht zum eigentlichen Ziel der Überwachung gehören. Auch Umwelteinflüsse können Messergebnisse verfälschen oder diese so überlagern, dass ein Erkennen der notwendigen Merkmale nicht ohne Weiteres möglich ist. Eine durch Temperaturänderung hervorgerufene Variation der Schwingungseigenschaften kann ebenso signifikant sein, wie eine durch Schäden verursachte (Farrar & Jauregui, 1998). Für die Eliminierung dieser Effekte aus den Datensätzen werden unterschiedliche Methoden verwendet. Für einige Messmethoden ist dazu die Erfassung der störenden Umwelteinflüsse (wie z.B. die Temperatur) notwendig, um die daraus resultierenden Effekte bestimmen und herausrechnen zu können. Andere Methoden dagegen lassen sich anwenden ohne den Wert der Umweltparameter selbst zu kennen (Zhang *et al.*, 2013).

- **Komprimierung der Daten:** Für die Speicherung und die Übertragung ist es von Vorteil die Größe der Daten vorab zu reduzieren. Je nach Datentyp existiert eine Vielzahl anwendbarer Methoden zur Datenkomprimierung. Bei (Fraser *et al.*, 2010) werden beispielsweise Bilder für die Fahrzeugerkennung zur Speicherung in das JPEG Datenformat konvertiert und dabei eine Kompressionsrate von 70% erreicht. Je nach benötigter Auflösung der Bilder sind auch höhere Kompressionsraten denkbar. Dabei wird deutlich, dass u.U. enormes Potenzial zur Einsparung von Speicherplatz und Kosten besteht.

- **Vorbereiten der Daten für die Analyse:** Eine Hauptaufgabe der Datenverarbeitung ist es, neben dem Entfernen von Störgeräuschen, die Sensordaten für die Schadenserkennung vorzubereiten. Signale müssen so bearbeitet werden, dass gewünschte Merkmale erkennbar und die in den Daten enthaltenen Informationen zugänglich gemacht werden. Zu den verwendeten Methoden zählen: Fast-Fourier-Transformation, Kreuz-Korrelation, Wavelet-Transformation oder Principal-Component-Analysis (Mahmud *et al.*, 2018).

3.6 Datenauswertung

Die Auswertung der bereits verarbeiteten und gespeicherten Daten stellt die letzte Komponente eines klassischen SHM-Systems dar (siehe Abschnitt 3.2). Ziel ist es, aus den gesammelten und (vor-)verarbeiteten Daten Schadensmerkmale zu extrahieren, um letztendlich eine Aussage über den aktuellen Zustand des Bauwerks zu treffen. Wie auch schon bei der Datenverarbeitung wird im praktischen Teil dieses Projektes die Datenauswertung nicht (bzw. nur minimal) implementiert. Dennoch soll knapp erwähnt werden, auf welche Art und Weise die Daten in einem realen Monitoring-Projekt analysiert und ausgewertet werden. In den meisten Fällen ist die Intention der Datenauswertung das Erkennen, Lokalisieren und Quantifizieren von Schäden.
Die Schadenserkennung kann dabei nach An *et al.* (2019) in zwei Oberkategorien unterteilt werden: globale und lokale Methoden. Globale Methoden nutzen Eigenschaften der gesamten Struktur, wie z.B. Eigenfrequenzen, Schwingungsformen oder Elastizitätsmatrix, um Schäden festzustellen. Ein deutlicher Nachteil ist die relative unempfindlich gegenüber lokalen Defekten. Lokale Methoden dagegen sind häufig deutlich genauer beim Erkennen von Schäden. Sie geben allerdings nur indirekt Auskunft über den Zustand der gesamten Struktur und benötigen zusätzlich i. d. R. a priori Wissen über den Ort des Schadens. Sie werden deshalb häufig eingesetzt, wenn die Entwicklung von bereits bekannten Schadensparametern, wie z.B. Ermüdungsrissen exakt überwacht werden soll (An *et al.*, 2019).
In den meisten Fällen werden Methoden zur Schadenserkennung eingesetzt, die auf dem Schwingungsverhalten der Strukturen oder einzelner Elemente beruhen. Zur Erfassung dieser Daten werden Beschleunigungssensoren verwendet (vgl. 3.3). Die Schwingungseigenschaften eines Bauteils hängen von seinen physikalischen Eigenschaften ab und werden dadurch von Schäden beeinflusst. Das aktuelle Schwingungsverhalten, wie z.B. Resonanzfrequenzen oder Mode-Shape-Curvatures werden mit denen des unbeschädigten Bauwerkes verglichen. Aus den Abweichungen lässt sich folglich auf Lage und (begrenzt auch) auf das Ausmaß des Schadens schließen (Farrar & Worden, 2007). Daneben werden auch Methoden, die auf den Einsatz von Ultraschall, Radar, akustischen Emissionen sowie der Verschiebung oder Dehnung von Bauteilen basieren, verwendet. Auch Verfahrensweisen wie Deep Learning und neuronale Netze werden immer häufiger verwendet (An *et al.*, 2019).

3.7 Darstellung der Ergenisse

Die Darstellung oder Visualisierung der Daten stellt eine wichtige Komponente eines SHM-Systems dar, auch wenn sie in vielen Projekten wenig Beachtung erfährt und unter den Komponenten von SHM-Systemen häufig nicht zu finden ist. Sensordaten müssen nicht nur gesammelt und gespeichert, sondern auch in einer Art und Weise zugänglich und anschaulich gemacht werden, die den Ansprüchen des Endnutzers gerecht werden. Die konkreten Ansprüche können dabei recht unterschiedlich ausfallen, je nachdem welche Funktion der Nutzer im Bezug zum Bauwerk einnimmt. Es können drei besonders wichtige Kategorien bzw. Grundsätze für die Visualisierung bei SHM Projekten identifiziert werden (Glisic *et al.*, 2014).

1. **SHM-Daten:** Visualisierung der SHM-Daten sowohl in Echtzeit als auch den zeitlichen Verlauf der Messwerte.
2. **Metadaten:** Hierzu gehören Informationen zum Status der Sensoren sowie deren Lage im Objekt. Auch Pläne des Bauwerks (oder digitale Modelle) lassen sich unter dieser Kategorie finden.
3. **Warnhinweise:** Schadensindikatoren oder Grenzwertüberschreitungen sollten deutlich erkennbar und auffällig sein.

Die graphische Darstellung der Daten hängt in Umfang und Form von den Ansprüchen des jeweiligen Nutzers ab. Eine aus Befragungen zusammengestellte Liste der möglichen Interessensgruppen und deren Anforderungen an die Visualisierung und Datenverfügbarkeit findet sich ebenfalls bei Glisic *et al.* (2014):

1. **Brückenmanagement:** Überblick über den generellen Zustand des Bauwerks sowie ein effektives Warnsystem. Dieses sollte bei Schäden, Anomalien oder dem Überschreiten von Grenzwerten automatisch eine Warnung verschicken.
2. **Verantwortliche Ingenieure:** Darstellung des Verlaufs der Werte sowie Zugang zu Echtzeitinformationen.
3. **Brückenbetreiber:** Verkehrsdaten, Wetterdaten und Zustand der Verkehrswege.
4. **Forschung und Wissenschaft:** Je nach Schwerpunkt oder Forschungsinteresse können hier Kombinationen aller oben genannten Anforderungen auftreten.

Ein SHM System sollte auf die Bedürfnisse der Zielgruppe abgestimmt werden d.h. auch die Auswahl der zu visualisierenden Daten sowie die Art und Weise der Visualisierung anpassen. Glisic *et al.* (2014) ziehen daraus den Schluss, dass eine einzige Plattform für die Darstellung

der Daten nicht allen Ansprüchen gleichzeitig gerecht werden kann. Die Visualisierung wie auch schon das gesamte SHM-System sollte daher einen klar definierten Nutzen sowie eine definierte Zielgruppe besitzen und dementsprechend darauf ausgerichtet werden. Abgeleitet daraus ergibt sich welche Daten in welcher Form dargestellt werden sollen. In jedem Fall muss die Darstellung verständlich und prägnant sein. Die folgenden fünf Datenkategorien wurden dabei als besonders relevant eingeordnet und sollten in jedem SHM-System zur Verfügung stehen.

- Metadaten, wie die Position der Sensoren oder Pläne des Bauwerks.
- Zustandsinformationen des Systems wie den Status der Sensoren oder anderer Systemteile.
- Konfigurationsdaten wie die Einheiten der Messwerte oder die festgelegte Ausleseintervalle.
- Warnhinweise z.B. bei Überschreiten von Grenzwerten oder dem Ausfall von Sensoren.
- Schließlich die eigentlichen SHM Daten z.B. in Form der Messwerte.

Diese Datenkategorien wurden bei der Entwicklung der Anwendung im zweiten Teil dieser Abschlussarbeit bestmöglich berücksichtigt und integriert (vgl. Abschnitt 10.4). In aktuellen Forschungsprojekten findet sich sehr häufig die Visualisierung der Daten in Tabellenform und als Diagramme. I.d.R. wird aber auf den Aspekt der Verständlich- und Nachvollziehbarkeit weniger Wert gelegt als auf andere Komponenten eines SHM-System.

Kapitel 4

Building Information Modeling

BIM gewann in den letzten Jahren für die gesamte Baubranche zunehmend an Bedeutung. Die Bundesregierung hat es sich zum Ziel gesetzt, ab 2020 alle Infrastrukturbauprojekte mithilfe von BIM durchzuführen (Bundesregierung, 2019). Insbesondere bei öffentlichen Projekten im Infrastruktursektor, zu denen Brückenbauwerke zählen, kommt BIM eine große Relevanz zu. Eine allgemeine Definition findet sich bei Borrmann *et al.* (2015)

> *„Unter einem Building Information Model (BIM) versteht man ein umfassendes digitales Abbild eines Bauwerks mit großer Informationstiefe. Dazu gehören, neben der dreidimensionalen Geometrie der Bauteile, vor allem nicht-geometrische Zusatzinformationen, wie Typinformationen, technische Eigenschaften und Kosten. Der Begriff Building Information Modeling beschreibt entsprechend den Vorgang zur Erschaffung, Änderung und Verwaltung eines solchen digitalen Bauwerkmodells mithilfe entsprechender Softwarewerkzeuge.“* (Borrmann *et al.*, 2015)

Das wirft die Fragen auf, ob und inwieweit das sensorgestützte Monitoring als Teil von BIM aufgefasst werden kann. Beziehungsweise inwieweit das Monitoring mit den Werkzeugen von BIM verbessert und erleichtert werden kann. Die Verbindung wird besonders deutlich, wenn das digitale Modell des Bauwerks in die Überwachung mit einfließt, wie es im Teil 2 dieser Arbeit der Fall ist.
In der aktuellen Forschung sind Versuche zu finden SHM und die damit verbundene Sensorik in BIM zu integrieren. Die Sensoren sollen damit selbst Teil des digitalen Modells werden. Damit wären Lage und Eigenschaften der Sensoren bereits durch das Modell bekannt und es besteht die Möglichkeit z.B. Grenzwerte für die einzelnen Sensoren aus dem Modell abzuleiten. Weiterhin sind alle Informationen in einer zentralen Datei zusammengeführt und zugänglich. Dieses Vorgehen findet aber noch keine nennenswerte Verwendung in der Praxis. Obwohl es gerade für Infrastrukturbauwerke, wie beispielsweise Brücken von besonderem Interesse wäre. Zum einen ist hier eine regelmäßige Inspektion vorgeschrieben, um die Sicherheit des Bauwer-

kes zu gewährleisten, zum anderen liegen in aller Regel (zumindest für aktuelle Bauprojekte) digitale Modelle der Bauwerke vor, die für das Monitoring genutzt werden könnten. In Bereichen wie z.B. dem Facility Management ist die Integration von Sensoren in BIM und die Nutzung des Modells für das Monitoring eines Bauwerks längst etabliert (Borrmann *et al.*, 2015).

Es gibt eine Vielzahl an Datenformaten und Software zur Erstellung, Verwaltung und Verwendung solcher Modelle; die sich je nach Nutzen und Absicht unterscheiden. Zumeist sind diese Formate proprietärer Natur und nur bedingt herstellerübergreifend nutzbar. Fundamental mit dem Begriff BIM verknüpft, ist daher das Datenformat Industry Foundation Classes (IFC). Dabei handelt es sich um ein standardisiertes, herstellerneutrales Datenformat für den Austausch von digitalen Gebäudemodellen (Borrmann *et al.*, 2015). Nahezu alle Software im BIM Bereich besitzt die Möglichkeit IFC Dateien zu importieren und zu exportieren. Mit dem Format und seiner Datenmodellierungssprache EXPRESS können Bauwerke sehr detailliert modelliert werden und insbesondere die semantischen Beziehungen zwischen den Bauteilen präzise abgebildet werden. Aufgrund der Neutralität und der Offenheit des Datenformats, hat es sich in den letzten Jahren als das zentrale Format für BIM-Projekte durchgesetzt und bildet bereits die Grundlage für die meisten BIM Projekte aus staatlicher Hand (Borrmann *et al.*, 2015).

Dementsprechend findet sich bei Rio *et al.* (2013) der Vorschlag, das Datenaustauschformat IFC zu erweitern, um auch bauwerkstypische Sensoren und deren gelieferte Werte abbilden zu können. IFC besitzt bereits zwei Klassen mit deren Hilfe Sensoren als Entität dargestellt werden können: IfcSensor und IfcSensorType. Die in der SHM überwiegend verwendeten Sensortypen (vgl. hierzu Abschnitt 3.3) fehlen aber unter den vordefinierten IfcSensorTypes. Hier sind bis jetzt hauptsächlich Sensortypen zur Gebäudeüberwachung und Facility Management vertreten.

Tabelle 4.1 zeigt eine Auflistung der momentan unter IfcSensorTypeEnum definierten Sensoren[1]. Wie bereits erwähnt, lässt sich erkennen, dass wichtige Sensoren für das SHM in dieser Liste fehlen. Insbesondere Sensoren zur Distanz-, Neigungs-, Vibrations- und Dehnungsmessung sind entscheidend, um den Zustand von Brückenbauwerken festzustellen und sind daher in den meisten SHM Systemen zu finden. Aus diesem Grund schlägt Rio *et al.* (2013) eine Erweiterung dieser Liste um die in Tabelle 4.2 aufgelisteten Definitionen vor.

Es ist gegenwärtig (noch) nicht möglich die Messwerte der Sensoren ebenfalls unter IfcSensor oder IfcSensorType mit abzuspeichern und diese damit zu einem Teil des Modells zu machen. Hier kommt außerdem die Schwierigkeit hinzukommen potenziell sehr große Datenmengen innerhalb des Modells abspeichern zu müssen. Dies lässt die Dateigröße schnell unhandlich werden.

[1] https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD1/HTML/schema/ifcbuildingcontrolsdomain/lexical/ifcsensortypeenum.html Stand: 15.07.2020

CO2SENSOR	senses or detects carbon dioxide
CONDUCTANCESENSOR	senses or detects electrical conductance
CONTACTSENSOR	senses or detects contact
FIRESENSOR	senses or detects fire
FLOWSENSOR	senses or detects flow in a fluid
FROSTSENSOR	senses or detects frost on a window
GASSENSOR	senses or detects gas concentration
HEATSENSOR	senses or detects heat
HUMIDITYSENSOR	senses or detects humidity
IDENTIFIERSENSOR	reads a tag, such as for gaining access to a door
IONCONCENTRATIONSENSOR	senses or detects ion concentration
LEVELSENSOR	senses or detects fill level
LIGHTSENSOR	senses or detects light
MOISTURESENSOR	senses or detects moisture
MOVEMENTSENSOR	senses or detects movement
PHSENSOR	senses or detects acidity
PRESSURESENSOR	senses or detects pressure
RADIATIONSENSOR	senses or detects radiation
RADIOACTIVITYSENSOR	senses or detects atomic decay
SMOKESENSOR	senses or detects smoke
SOUNDSENSOR	senses or detects sound
TEMPERATURESENSOR	senses or detects temperature
WINDSENSOR	senses or detects airflow speed and direction
USERDEFINED	User-defined type
NOTDEFINED	Undefined type

Abbildung 4.1: Unter IfcSensorTypeEnum definierte Sensortypen

VIBRATINGWIRE	senses or detects vibration
LVDT	senses or detects linear displacement
INCLINOMETER	senses or detects angles of slope
FOILSTRAINGAUGE	senses or detects the strain of an object

Abbildung 4.2: Vorgeschlagene Erweiterung zu IfcSensorTypeEnum von Rio *et al.* (2013)

> *„However, the process of obtaining a dynamic monitoring of the structural behavior, where the obtained data can be quickly and accurately transferred from sensors to BIM and consequently to a specific structural analysis software, potentially allowing for effective decision making and in-time interventions, is not part of current BIM functionality.“* (Rio *et al.*, 2013)

Eine umfangreiche Datenspeicherung wie es für die geplante Anwendung notwendig ist, muss folglich in eine Datenbank ausgelagert werden. Wobei auch die Verknüpfung externer Daten mit dem Sensorelement des Modells nicht leicht zu bewerkstelligen ist.
Grundsätzlich kann man SHM als Teil von BIM verstehen. Es handelt sich um einen Anwendungsfall zur Zustandsüberwachung eines Bauwerks während der Betriebsphase und weist damit Ähnlichkeiten zu anderen BIM Anwendungen, wie z.B. Facility Management auf. Obgleich die verwendeten Sensortypen, die überwachten Parameter sowie die Zielsetzung des Monitorings verschieden sind, lässt sich das digitale Modell auch im SHM gewinnbringend

weiter nutzen. Die Integration von Sensoren ist im Datenformat IFC bereits angelegt und die Speicherung der Messwerte — zumindest theoretisch — im Modell möglich. Bis zum gegenwärtigen Zeitpunkt fehlen jedoch für das SHM entscheidende Sensortypen.

Kapitel 5

Beispiele aus der Forschung

Zu den Themen Brückenmonitoring und ferner SHM lassen sich eine Vielzahl an Forschungsarbeiten finden. Mit den stetig wachsenden Möglichkeiten in den Bereichen IoT und Sensortechnik nimmt auch die Forschungstätigkeit bzgl. automatisierter Lösungen für das Bauwerksmonitoring zu.
Im Kontext dieser Arbeit ist es interessant, vor allem die Forschungsprojekte näher zu betrachten, bei denen die Forschungsfrage und Zielsetzung ähnlich der dieser Abschussarbeit ist. Viele der verfügbaren wissenschaftlichen Arbeiten zum SHM bei Brückenbauwerken beschäftigen sich mit dem Thema entweder aus Bauingenieurs- oder aus sensortechnischer Sichtweise. Im Verhältnis dazu haben wenige Arbeiten den Fokus auf Informatik, Datenverarbeitung und -speicherung gelegt. Das hängt möglicherweise auch damit zusammen, dass Informatik als Methode selten fachspezifische Forschung und Lösungen bietet, sondern sich auf ein breites Gebiet anwenden lässt. Und dennoch gibt es eine Reihe an Forschungsprojekten, die sich spezifisch mit dem Brückenmonitoring beschäftigen und auch eine deutliche Darstellung der verwendeten Softwarelösungen bieten. Im Folgenden werden einige relevante Arbeiten vorgestellt, um einen Eindruck und Überblick zur aktuellen Forschungslage und den Stand der Technik zu bieten. Diese Auswahl ist dabei weder vollständig, noch repräsentativ für die Gesamtheit der Brückenmonitoring-Projekte. Vielmehr werden beispielhaft einige Projekte vorgestellt, die Ansätze beinhalten, die für diese Arbeit relevant sind. Besonderes Augenmerk liegt zugleich auf Übertragung, Speicherung und Visualisierung der Daten. Interessant zu beobachten ist dabei, dass in keiner der im Rahmen dieser Arbeit untersuchten Forschungsarbeiten eine Verknüpfung der Sensordaten mit einem digitalen Modell des Bauwerks zu finden war.

5.1 IoT-System für das Brückenmonitoring

Tong *et al.* (2019) haben ein IoT-basiertes System zur Zustandsüberwachung von Brückenbauwerken entwickelt. Der Fokus liegt auf automatisiertem Langzeitmonitoring und der Auswertung und Darstellung der Daten in Echtzeit. Zu diesem Zweck wurde ein kabelloses Sensornetzwerk, bestehend aus Beschleunigungssensoren mit sehr niedrigem Energiebedarf, genutzt. Zusammen mit den verwendeten Solarpanels ist das System damit energetisch autark. Die von den Sensoren erfassten Daten werden an ein Gateway gesendet und von dort in Echtzeit auf eine Cloud Plattform geladen — ebenfalls kabellos. Eine MySql-Datenbank dient der Speicherung, Verwaltung und gezielten Bereitstellung der Daten. Die Cloud-Plattform verwendet Alibaba Cloud Service als Background-Server und einen Apache-HTTP-Server als Webserver. Die Cloud-Plattform dient dem Empfangen, Speichern, Visualisieren und Analysieren der Sensorwerte und fungiert somit als Terminal. Nutzer haben die Möglichkeit, über die Web-Plattform neue Projekte anzulegen, Gateways und Sensoren für die Projekte hinzuzufügen sowie den Echtzeit-Status des Bauwerkes einzusehen. Dazu werden die Sensordaten graphisch in Form von Diagrammen dargestellt und die Daten auch zum Download bereitgestellt. Die Plattform bietet eine Registrierungs- und Loginfunktion, um verschiedene Nutzerkonten mit unterschiedlichen Berechtigungen anzulegen. Zusätzlich wurde ein Warnmechanismus bei Grenzwertüberschreitungen implementiert. Der schematische Aufbau des entwickelten Systems ist in Abbildung 5.1 dargestellt.

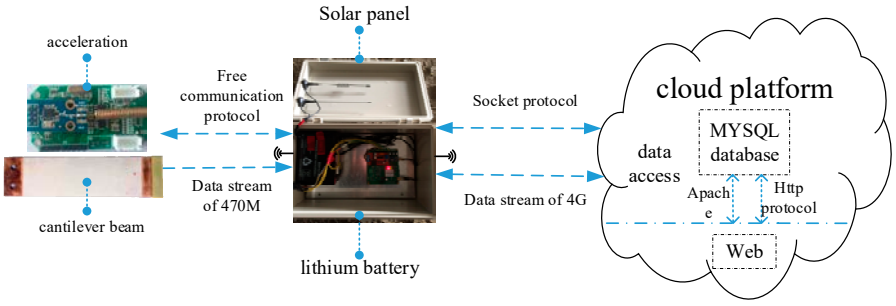


Abbildung 5.1: Schema des entwickelten IoT Systems von Tong *et al.* (2019)

5.2 SHM-Datenverwaltung mit MySQL und MATLAB

Bei Koo *et al.* (2011) von der University of Sheffield findet sich die Entwicklung eines Datenmanagement-Systems mit dem Ziel, große Datenmengen, die bei SHM-Projekten anfallen, effizient zu speichern, abzurufen und darzustellen. Das System basiert dabei auf einer MySQL-Datenbank und wurde für die Überwachung von drei Brückenbauwerken in Großbritannien bereits eingesetzt: der Tamar Bridge in Plymouth, der Humber Bridge in Hull und dem Arts Tower in Sheffield. Die Messwerte werden jeweils dezentral in Datenbanken vor

Ort eingespeist. Die Daten werden anschließend in regelmäßigen Abständen an den zentralen Server weitergeleitet, wo sie in eine zentrale Datenbank eingepflegt werden. Die Datenverarbeitung sowie die Datenanalyse werden mithilfe von Matlab durchgeführt. Dazu werden die Daten in regelmäßigen Intervallen abgefragt und die Ergebnisse der Datenverarbeitung wieder in die Datenbank eingepflegt. Auf diese Weise werden neben den rohen Messreihen auch Tabellen für 30 Minuten Durchschnittswerte abgelegt. Nutzer können auf die Daten entweder über das Matlab-Interface oder über ein Web-Interface zugreifen. Die Visualisierung der Zeitreihen wurde in Java Skript unter Einbindung der Google Chart Tool API geschrieben und erlaubt es dem Nutzer, den gewünschten Betrachtungszeitraum selbst festzulegen. Hier wird die Absicht verfolgt, die große Menge anfallender Daten für Nutzer verfügbar zu machen und anschaulich zu visualisieren. Die Daten werden in Form von Linien- und Scatterplots dargestellt. Die Datenbank enthält für jeden Sensortyp (wie z.B. Beschleunigungssensor, Webcam, Temperatursensor etc.) eine eigene Tabelle, in der die Messwerte zusammen mit einem Zeitstempel abgelegt werden. Ein Schema der entwickelten Anwendung ist in 5.2 zu sehen.

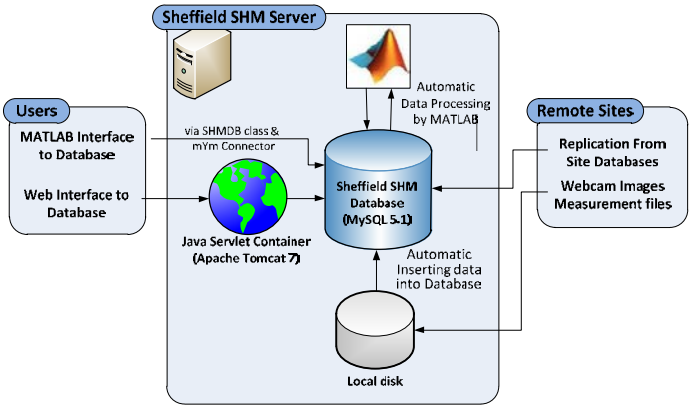


Abbildung 5.2: Sheffield SHM Data Management System von Koo *et al.* (2011)

5.3 Sensornetzwerk für das Structural Health Monitoring

Gleichermaßen findet sich bei Fraser *et al.* (2010) die Entwicklung und Installation eines Sensornetzwerks für das SHM bei Brückenbauwerken. Ziel war, Beschleunigungs- und Videodaten durchgängig und zeitsynchronisiert zu erfassen, zu speichern und die Daten online über ein Webportal verfügbar zu machen. Als Sensoren wurden 20 Beschleunigungssensoren und eine Sony XCD-X710CR Digitalkamera verwendet. Via Bildverarbeitung und automatischer Fahrzeugerkennung wurden die Verkehrsbelastung des Bauwerkes sowie die Position und die Geschwindigkeit jedes Fahrzeuges bestimmt. Die daraus resultierenden Vibrationen wurden mit dem Videomaterial korreliert, um den Zusammenhang zwischen Fahrzeug und dynamischem Verhalten der Brücke zu untersuchen. Da der Zusammenhang zwischen unterschied-

lichen Messwerten im Vordergrund steht, werden die Messwerte mit exakten Zeitstempeln versehen. Die ausgelesenen und verarbeiteten Sensordaten werden von einem lokalen Rechner an einen Hauptserver weitergeleitet. Alle Komponenten des Systems sind über eine high-speed-wireless Internetverbindung verknüpft, die eine Datenübertragung in Echtzeit erlaubt. Zur Speicherung der Daten wird eine DB2-Datenbank von IBM verwendet. Dabei handelt es sich um eine relationale Datenbank (wie sie in den meisten SHM-Projekten Verwendung findet), die zusätzlich in der Lage ist, Bilder und Videos zu speichern. Um die Daten zugänglich zu machen, wurde ein Web Portal entwickelt. Dies ermöglicht, alle Daten von der Datenbank zu durchsuchen, nach Zeit oder maximaler Beschleunigung sortieren zu lassen und Sensor- und Videodaten herunterzuladen oder auf der Website selbst zu betrachten. Für die graphische Darstellung der Daten wurden Graphen mit Zoomfunktion mithilfe von Matlab erstellt. Die durchgeführte Datenanalyse und die Beschreibung der Bildverarbeitung nehmen einen großen Teil der Arbeit von Fraser *et al.* (2010) ein, spielen im Rahmen dieser Abschlussarbeit aber nur eine untergeordnete Rolle und werden deshalb nicht weiter erläutert.

5.4 Visualisierung und Bereitstellung heterogener Monitoringdaten

Bei Glisic *et al.* (2014) findet sich die Darlegung eines umfassenden SHM-Systems. Besonders interessant ist, dass der Fokus der Arbeit auf die Datenvisualisierung gelegt wurde, welche bei den meisten anderen Projekten dagegen nur peripher behandelt wurde. Ziel war es, die große Datenmenge, die bei SHM-Systemen anfällt, zugänglich zu machen und verständlich zu visualisieren. Das SHM-System ermöglicht eine umfangreichere Überwachung des Bauwerkes. Z.B. liefert es neben Informationen zur strukturellen Sicherheit auch Daten über die betriebliche Auslastung sowie die Unterhaltung und Wartung. Ziel ist es, die Brücke auf unbegrenzte Zeit weiter zu betreiben. Die anfallenden Daten sind aus diesem Grund sehr heterogen und von umfangreichem Volumen. Das stellt einige Herausforderungen an die Visualisierung und Speicherung der Daten. Verwendet werden dynamische sowie statische Dehnungssensoren, Rotationssensoren, Temperatursensoren, Wetterstationen (mit Erfassung von Feuchtigkeit, Windstärke/-richtung und Niederschlag) und Videoüberwachung. Die Erfassung der Daten erfolgt über getrennte Subsysteme (ähnlich dem in dieser Arbeit verwendeten Raspberry Pi), welche die Daten über ein lokales Netzwerk an einen zentralen Server weiterleiten. Über den zentralen Server erhalten Nutzer Zugriff auf die Daten sowie das System. Neben der Visualisierung von vergangenen und aktuellen Sensordaten lassen sich über den zentralen Server auch die Konfiguration des SHM-Systems ändern sowie Metadaten einsehen. Für diese unterschiedlichen Aufgaben werden verschiedene Plattformen verwendet, die unterschiedliche Berechtigungen erfordern. Die Plattform, welche die Konfiguration der Parameter erlaubt (Festlegung von Grenzwerten,Kompressionsraten, etc..), ist dementsprechend pass-

wortgeschützt. Die Daten der Sensoren werden vorverarbeitet und komprimiert. In Form von Tabellen, Diagrammen, Übersichtsplänen und Videostreams werden die Daten visualisiert mit dem Ziel, eine leicht verständliche Darstellung des Systems und der gelieferten Messwerte bereit zu stellen. Die Frage, ob das System sich intuitiv und somit benutzerfreundlich bedienen lässt, steht bei den meisten Forschungsprojekten nicht im Vordergrund, spielt aber für den praktischen Nutzen eines solchen Systems eine wesentliche Rolle. Die Nutzer eines SHM-Systems sind i.d.R. nicht die Entwickler und besitzen — unter Umständen — nicht die gleichen (Fach-)Kenntnisse im Umgang mit der Software. Daher ist eine unkomplizierte und benutzerfreundliche Handhabung essenziell für den Erfolg und Nutzen der Überwachung.

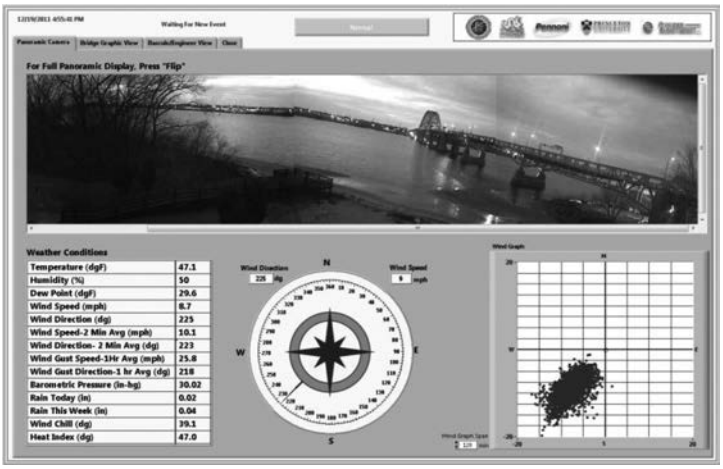


Abbildung 5.3: Plattform für die Darstellung von Echtzeitdaten bei Glisic *et al.* (2014)

Kapitel 6

Mögliche Vorbehalte

Die kontinuierliche Überwachung via Sensoren eröffnet neue Möglichkeiten Defekte theoretisch bereits deutlich früher zu erkennen. Graduelle Veränderungen werden deutlicher wahrgenommen und der Zustand kann insgesamt genauer bestimmt werden. Die theoretische Möglichkeit alle Fehler in frühen Stadien zu erkennen (und womöglich zu verhindern) bringt eine Verantwortung mit sich. Ein Versagen, das erkannt werden könnte, muss auch erkannt und behoben werden. Je mehr Daten vorliegen, desto mehr lässt sich daraus ableiten. Und genau hier liegt die Schwierigkeit, denn der Umgang mit großen Datenmengen ist unter Umständen nicht trivial und der Prozess, aus dem Rauschen die wichtigen Informationen zu extrahieren, kann aufwändig sein. Ist das Überwachungssystems in Betrieb, liefern die Sensoren fortlaufend neue Werte. Eine Flut an Daten wird generiert und es besteht die reelle Gefahr, trotz umfassender Information sicherheitsrelevante Daten zu übersehen.
Dieser Vorbehalt bedeutet nicht, dass das sensorgestützte Monitoring einer Inspektion vor Ort unterlegen ist; es lässt überdies zu, deutlich frühere und genauere Aussagen über potenzielles Versagen zu treffen. Es hat lediglich zur Folge, dass beim Entwurf eines SHM-Systems folgende Fragen gestellt werden müssen:

- Welche Daten werden erhoben?
- Welche Schlüsse können aus den Daten gezogen werden, d.h. welcher Schaden kann erkannt werden?
- Auf welche Weise werden diese Merkmale zuverlässig detektiert?
- Welche Konsequenzen werden aus den Daten gezogen, d.h. wie sieht die Handlungsstrategie bei erkanntem Schaden aus?

Die Beantwortung dieser und ähnlicher Fragen verdeutlicht, welche Szenarien verhindert werden können und welche nicht. Weiterhin lässt sich damit auch die Frage nach Verantwortung

und Schuld bei Versagen des Systems einfacher beantworten. Hier findet sich die grundlegende ethische Fragestellung zwischen dem Verhältnis von Wissen und Verantwortung zueinander, wieder.

Im ersten Teil dieser Arbeit wurde das sensorgestützte Brückenmonitoring vorgestellt. Die Vorteile gegenüber dem klassischen Vorgehen einer Inspektion vor Ort wurden dargelegt und die einzelnen Komponenten eines solchen Systems expliziert. Zusätzlich wurde ein Blick auf eine Auswahl an Forschungsprojekten geworfen die sich mit dem sensorgestützten Brückenmonitoring und seiner Implementierung beschäftigen. Im folgenden zweiten Teil dieser Arbeit wird die Entwicklung einer webbasierten Anwendung beschrieben, die diese Kategorien und Prinzipien umsetzt. Eine zentrale Rolle nimmt dabei die Forge-Plattform der Firma Autodesk ein.

Teil II

Entwicklung einer Forge basierten Anwendung zum sensorgestützen Brückenmonitoring

Kapitel 7

Zielsetzung

Wie in Teil 1 dieser Abschlussarbeit bereits dargestellt, ist die visuelle Inspektion von Brückenbauwerken aufwändig und relativ ungenau. Die Zustandserfassung findet in großen Intervallen statt und erfordert die physische Anwesenheit eines Inspekteurs. Für die Betreiber oder andere Nutzer von Brückenbauwerken kann es von großem zeitlichen und finanziellen Nutzen sein, wenn diese Überwachung durchgehend und automatisch möglich ist. Darüber hinaus bietet eine solche sensorgestützte Überwachungsmethode den Vorteil, dass der Zustand des Bauwerks auch aus der Ferne und nicht nur vor Ort beurteilt werden kann. Im Rahmen dieser Arbeit soll untersucht werden, in weit die Überwachung des Zustandes eines Brückenbauwerks über eine Webanwendung in Verbindung mit einem Sensornetz nutzerfreundlich umzusetzen ist. Dazu wurde eine prototypische Webanwendung entwickelt, die es ermöglichen soll, die Überwachung relevanter Parameter bei Brückenbauwerken unkompliziert und vor allem anschaulich zu gestalten sowie online zugänglich zu machen. Folgende Anforderungen wurden dabei als Ziele für die Anwendung definiert:

- **Webbasiert:** Die Anwendung wurde als Webanwendung konzipiert. Der Endnutzer soll die Anwendung von überall aus dem Netz erreichen können ohne im Vorfeld weitere unterstützende Software installieren zu müssen. Die einzige technische Voraussetzung ist ein aktueller Browser mit der Unterstützung von Web Graphics Library (WebGL).
- **Darstellung des Bauwerks als 3D-Modell:** Zur besseren Visualisierung und Benutzerfreundlichkeit soll —neben den grundlegenden Sensordaten — vor allem auch das Modell des Bauwerks im Browser veranschaulicht werden.
- **Erfassung und Darstellung von Sensordaten:** Zur Erfassung soll ein Sensornetzwerk angelegt werden, welches Messwerte für das Bauwerk liefert. Diese Daten sollen graphisch und semantisch mit dem Modell verknüpft werden. Dabei sollen die Lage der Sensoren, aktuelle Messwerte inklusive Messwertreihen angezeigt werden.

- **Anpassbarkeit durch den Nutzer:** Der Nutzer soll in der Lage sein mit dem Modell zu interagieren, die Anzeige der Sensordaten zu steuern und das Sensornetz selbst anzupassen. Zusätzlich soll es möglich sein über das Webinterface neue Sensoren in das Netz zu integrieren, ohne auf den Quellcode zugreifen zu müssen.

Zur Umsetzung dieser Anforderungen ist ein zentrales Element die Einbindung von Autodesk Forge, einer cloudbasierten Entwickler-Platform von der Firma Autodesk. Diese stellt eine Reihe Application Programming Interfaces (APIs) zur Verfügung, die es unter anderem ermöglichen soll, Computer Aided Design (CAD)-Modelle und andere ingenieurspezifische Daten über die Cloud abzurufen und zu nutzen. Damit ist es möglich, Webanwendungen zu entwickeln, die die Funktionalitäten von CAD-Programmen besitzen, ohne dass der Endnutzer dafür ergänzende Software installieren muss[1]. Die Anwendung wurde dabei als Proof-of-Concept geplant und nicht für die tatsächliche Überwachung eines konkreten realen Objekts. Sie soll vielmehr auf unterschiedliche Brückenbauwerke und Monitoringziele anpassbar sein und die generelle Funktionsweise demonstrieren. Das entwickelte Sensornetz ist aus diesem Grund nur ein Platzhalter und in Bezug auf Netzgröße sowie verwendete Sensortypen, variabel. Auf diese Weise lassen sich theoretisch mit der Anwendung die unterschiedlichsten Parameter (z.B. aktuelle Verkehrsbelastung, Umweltdaten oder Schadensmerkmale) überwachen und, abhängig von den verwendeten Sensorentypen, darstellen. In den folgenden Abschnitten werden Aufbau und Funktionsweise der entwickelten Anwendung komponentenweise und detailliert dargestellt.

[1] https://forge.autodesk.com

Kapitel 8

Konzept der Anwendung

Die Anwendung besteht im Kern aus zwei Hauptkomponenten: dem zentralen Server und dem Sensornetz. Das Sensornetz ist für die Sammlung und Übertragung der Sensormesswerte zuständig. Es stellt das Datenerfassungssystem dar und besteht aus Sensoren, Verkabelung und einem Raspberry Pi (RPI). Über diesen werden die Sensoren ausgelesen und die Messwerte an den Hauptserver weitergeleitet. Der Hauptserver dagegen stellt auf der einen Seite die Website zur Verfügung und dient auf der anderen Seite als zentraler Knoten für die Überprüfung und Verwaltung der Daten. Als Speicherlösung wird eine relationale Da-

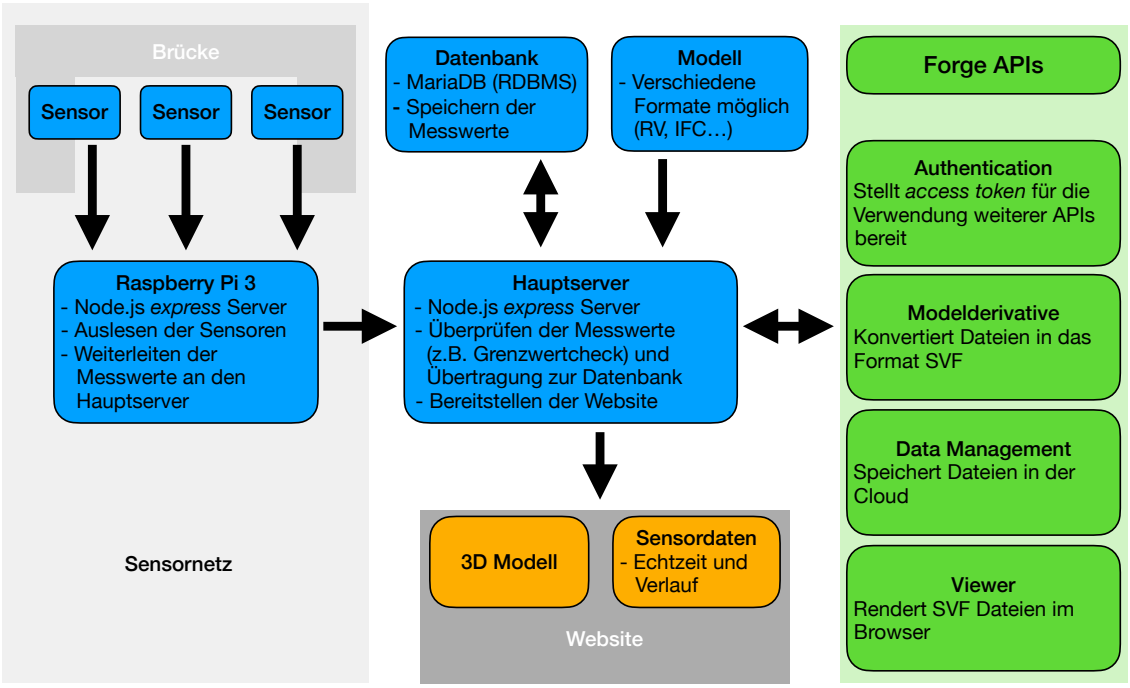


Abbildung 8.1: Anwendungsschema

tenbank verwendet. Gespeichert werden zum einen die Messwerte der Sensoren und zum anderen Informationen über die Sensoren selbst, wie die Lage im Bauwerk, Erreichbarkeit im Netzwerk oder die Auslesefrequenz. Die Verbindung zwischen RPI und Hauptserver wird kabellos über ein lokales Netzwerk realisiert, die Kommunikation läuft dabei über Hypertext Transfer Protocol (HTTP). Die Webplattform ermöglicht Endnutzern den Zugang zu den Messwerten und stellt daneben das Bauwerk als Modell dar. Auch die räumliche Lage der Sensoren wird in die Anzeige des Modells integriert. Die Messwerte werden dabei in Echtzeit auf die Website übertragen. Zusätzlich wird der Verlauf in Form eines Graphen dargestellt. Die Modelle können vom Nutzer auf den Hauptserver geladen und konvertiert werden. Sie bleiben dort für die weitere Nutzung gespeichert. Ebenfalls besteht die Möglichkeit, neue Sensoren direkt über die Website in das Netzwerk einzufügen und in das Modell zu integrieren. Für alle Funktionen bezüglich der Bauwerksmodelle wie Upload, Verwaltung, Konvertierung und Rendern werden APIs von Autodesk-Forge eingebunden. Hierzu wurde die Anwendung auf der Forge-Platform von Autodesk registriert. Ein Schema mit allen Komponenten des entwickelten Systems ist in Abbildung 8.1 zu sehen.

Kapitel 9

Sensornetzwerk

Die erste Komponente der Anwendung ist das Sensornetzwerk. Im Kontext eines SHM-Systems werden hier die Aufgaben der Datensammlung und Datenübertragung verwirklicht. Es besteht aus den Sensoren, der Verkabelung und einem RPI zum Auslesen der Sensoren sowie zum Bereitstellen der Sensormesswerte im lokalen Netzwerk. Die konkret für dieses Projekt verwendete Hardware ist problemlos durch Bauteile mit ähnlicher Funktion austauschbar. Insbesondere die verwendeten Sensoren stellen gewissermaßen Platzhalter dar und könnten durch beliebige Sensortypen ersetzt oder um diese erweitert werden. Der verwendete RPI ist durch andere Einplatinencomputer austauschbar und auch deren Anzahl ist variabel und kann von der Anzahl der benötigten Sensoren sowie deren räumliche Lage und Entfernung zueinander abhängig gemacht werden. Im Rahmen dieses Projektes war es notwendig, die generelle Funktionsweise und Machbarkeit der Integration mit einer Forge-Anwendung zu demonstrieren. Es wurde daher ein Sensornetz bestehend aus einem RPI sowie drei Sensoren zur Temperatur- und Feuchtigkeitsmessung entwickelt und verwendet.

9.1 Raspberry Pi 3

Zum Auslesen der Sensoren und als Verbindung zum Hauptserver wird ein Raspberry Pi Modell B verwendet. Der RPI ist ein unkomplizierter, kostengünstiger Einplatinencomputer mit einem Quad Core 1.2 GHz Broadcom BCM2837 64 bit Prozessor und 1 GB RAM Arbeitsspeicher. Zum Laden des Betriebssystems und als Speicher wird eine externe Micro SD Karte verwendet. Damit lässt sich auch die Größe des Speichers den individuellen Anforderungen anpassen. Diese Hardware besitzt mehrere USB-Ports sowie je einen Ethernet- und HDMI-Anschluss. Kabellos kann er über Wireless LAN oder eine Bluetooth-Verbindung angesteuert werden. Dies erspart lange Verkabelungen zwischen RPI und dem Hauptserver. Weiterhin können Sensoren auf diese Weise kabellos mit dem RPI verbunden werden. Die Verbindung zwischen RPI und Hauptserver wird über ein lokales kabelloses Netzwerk her-

gestellt. Die Kommunikation wird durch die Verwendung von HTTP ermöglicht. Zusätzlich besitzt der RPI 40 General Purpose Input Output (GPIO)-Pins, die ansteuerbar und programmierbar sind. Über diese Pins werden die Sensoren mit dem RPI verbunden und können ausgelesen werden. An jeweils zwei der ingesamt vier Pins liegt eine Spannung von 3,3 bzw. 5 Volt an, weitere acht dienen der Erdung. Diese zwölf Pins sichern die Stromversorgung für Sensoren oder andere Bauteile, falls diese keine eigene oder externe Energiequelle besitzen. Die restlichen 28 GPIO-Pins können dazu verwendet werden, die Sensoren auszulesen. Die genaue Pinbelegung kann dem Anhang entnommen werden. Die Anzahl der verknüpfbaren Sensoren ist limitiert durch die Anzahl der Pins sowie die benötigten Pins pro Sensor. Die verwendeten Sensoren des Typs DHT11 benötigen (neben Stromversorgung und Masseverbindung) einen weiteren Pin zum Auslesen. Bei anderen Sensortypen kann die Anzahl der benötigten Pins variieren und demnach auch höher ausfallen. Der RPI selbst benötigt eine Stromversorgung mit einer Spannung von 5 Volt und einer Stromstärke von 2,1 Ampere über einen Micro-USB-Port. Es ist ebenfalls möglich, eine unabhängige Stromversorgung z.B. über Solarpanels und Akkus einzurichten.
Als Betriebssystem stehen mehrere Optionen zur Auswahl. Neben dem offiziell unterstützen Betriebsystem Raspberry Pi OS, werden Betriebsysteme von anderen Anbietern bereitgestellt, darunter: Ubuntu, OSMC oder die speziell für den RPI entwickelte Windows-Distribution. Eine Vielzahl davon ist frei verfügbar und quelloffen. Für dieses Projekt wurde Raspberry Pi OS (32-bit) Lite als Betriebssystem gewählt. Dabei handelt es sich um eine minimalistischer Version des Raspberry Pi OS, ohne graphischer Benutzeroberfläche und mit möglichst schlanker Softwareausstattung [1]. Die Grundlage bildet das freie Betriebssystem Debian Buster, welches auf dem Linux-Kernel basiert und eines der am meist verbreiteten Linux Distributionen darstellt [2]. Der Vorteil eines so leichten Betriebssystems ist der verringerte Ressourcenbedarf in Bezug auf Rechenleistung und Speicher sowie damit einhergehend, die höhere Geschwindigkeit. Zur Verwendung als herkömmlicher Webserver und zum Auslesen und Verarbeiten von Sensordaten ist es insofern bestens geeignet. Die Bedienung des RPI erfolgt ausschließlich über Konsolenbefehle. Aufgrund seiner Leistung, universelle Anwendbarkeit und der Preisklasse wird er in einer Vielzahl von IoT-Projekten eingesetzt. Kumar & Rajasekaran (2016) verwenden ihn beispielsweise speziell für das Brückenmonitoring.

9.1.1 Node.js Express Server

Um die Messwerte der Sensoren über das Netzwerk verfügbar zu machen, wurde der RPI als simpler Webserver programmiert. Wie auch beim Hauptserver wurde hier die Sprache Node.js verwendet. Um die notwendigen Serverfunktionalitäten bereit zu stellen, wurde *express.js* verwendet, ein minimalistisches Web-Framework für Node.js [3]. Bei Serverstart wird

[1] https://www.raspberrypi.org/downloads/raspberry-pi-os/, Stand: 15.07.2020
[2] https://www.debian.org/releases/stable/index.de.html, Stand 22.07.2020
[3] https://expressjs.com, Stand: 05.07.2020

eine Liste aller, für diesen RPI registrierten Sensoren, vom Hauptserver angefragt. Diese beinhaltet die Information, welcher Sensor an welchem GPIO-Pin installiert ist, um das korrekte Auslesen der Sensoren zu ermöglichen.
Die Information, welche Sensor ID mit welchem GPIO-Pin korrespondiert, wird in einer Java Script Object Notation (JSON) Datei lokal abgespeichert, damit nicht für jeden Auslesevorgang eine Anfrage an den Hauptserver geschickt werden muss. Bei Neustart des Servers, bzw. bei Änderungen im Sensornetz kann diese Datei mit den aktuellen Daten überschrieben werden. Auf dem Server wurde lediglich eine einzige Route, nämlich </read> eingerichtet. Eine HTTP GET Request an diesen Punkt triggert das Auslesen eines bestimmten Sensors. Welcher Sensor ausgelesen werden soll, wird über die Sensor ID festgelegt. Diese wird als Query Parameter in der Uniform Resource Locator (URL) mit übergeben. Existiert der Sensor mit der angefragten ID auf dem RPI, wird dieser ausgelesen und der Messwert zusammen mit einem Zeitstempel als Antwort zurückgegeben. Dabei ist die Funktion zum Auslesen des Sensors abhängig vom Sensortyp. Auf diese Weise ist es möglich, mit verhältnismäßig wenig Aufwand neue Sensortypen in das Netz einzuführen. Dazu ist es lediglich notwendig, eine dem neuen Typ korrespondierende Methode zum Auslesen der Sensoren zu implementieren. Beim Auslesen wird dann die JSON Datei geladen, die ID, die als Request Parameter in der HTTP-Anfrage übergeben wurde, gesucht und der dazugehörige Pin mit der entsprechenden Funktion angesteuert. Folgende Funktion wird beispielsweise ausgeführt, um Sensoren des Typs DHT11 auszulesen:

Funktion für das Auslesen der Sensoren des Typs DHT11

```
1 function (sensor) {
2         var timestamp = Math.round(Date.now()/1000);
3         var readout = sensorLib.read(11,sensor.pin);
4         var data = [timestamp, sensor.sensor_id, readout.temperature,
              readout.humidity];
5         return data
6     }
```

Der RPI wird eindeutig über seine Netzwerkadresse identifiziert und kann über diese angesprochen werden. Für jeden Sensor wird die Netzwerkadresse des korrespondierenden RPI deshalb in der Datenbank festgehalten.

9.2 Sensoren

Die Art der verwendeten Sensoren ist sehr variabel und vom jeweiligen Monitoringsystem und -ziel abhängig. Für eine prototypische Anwendung kommt es daher darauf an, die Einbindung verschiedener Sensortypen zu ermöglichen und in erster Linie die generelle Funktionsweise mithilfe beliebiger Sensoren zu zeigen. Es wurden deshalb für dieses Projekt drei Senso-

ren des Typs DHT11 verwendet. Dabei handelt es sich um kombinierte Feuchtigkeits- und Temperatursensoren. Das technische Datenblatt diese Sensors liegt dieser Arbeit im Anhang bei. Es ist durchaus denkbar, dass diese oder ähnliche Sensoren für das Brückenmonitoring tatsächlich Verwendung finden. Umweltmesswerte, wie Feuchtigkeit und Temperatur spielen in realen Projekten häufig eine Rolle und werden auch in einigen der im 5 vorgestellten Forschungsarbeiten verwendet. Die Genauigkeit des DHT11 ist allerdings mit ±5 % RF und ± 1 °C für die meisten Anwendungen möglicherweise nicht ausreichend. In der vorgestellten Anwendung ist die Genauigkeit nicht relevant, da keine reale Brücke überwacht wird und daher keine realistische Schadensberechnung durchgeführt wird. Lediglich die bloße Erzeugung von Daten ist notwendig, um die Funktionsweise der Anwendung vorzuführen und zu testen. Die Anwendung ist dabei vollkommen flexibel, wie viele verschiedene Parameter ein Sensor erfasst oder welchen Datentyp diese haben. Zum Auslesen der Sensoren wird das Modul *node-dht-sensor* verwendet, welches über den Node Package Manager (NPM) verfügbar ist. Für jeden Sensortyp, der in das System aufgenommen werden soll, muss, wie im vorigen Abschnitt erwähnt, eine eigene Funktion definiert werden, die wahrscheinlich auch die Einbindung eigener Module benötigt.

9.3 Verkabelung

Die Sensoren sind auf einem Breadboard verkabelt und über Jumper-Kabel mit dem Raspberry Pi verbunden. Die konkrete Verkabelung ist dabei abhängig vom Sensortyp und kann in der Regel dem Datenblatt des Sensors entnommen werden. Für die Kabellänge zwischen Sensor und RPI gibt es einen im Datenblatt spezifizierten Maximalwert, bis zu dem das Auslesen des Sensors einwandfrei funktioniert. Dieser sollte beim Anlegen des Sensornetzes beachtet und nicht überschritten werden. Für den DHT11 z.B. beträgt die maximale Kabellänge 20 Meter. Bei größeren Abständen muss entweder ein weiterer RPI verwendet, das Signal verstärkt oder auf eine kabellose Lösung ausgewichen werden. Um den Auslesevorgang stabil zu halten, ist es beim DHT11 notwendig, einen Pull-up-Resistor zu verwenden. Dieser sollte laut Datenblatt bei ungefähr 5k-Ohm liegen. Die konkret verwendete Verkabelung zwischen RPI und den Sensoren über das Breadboard ist in Abbildung 9.1 zu sehen.

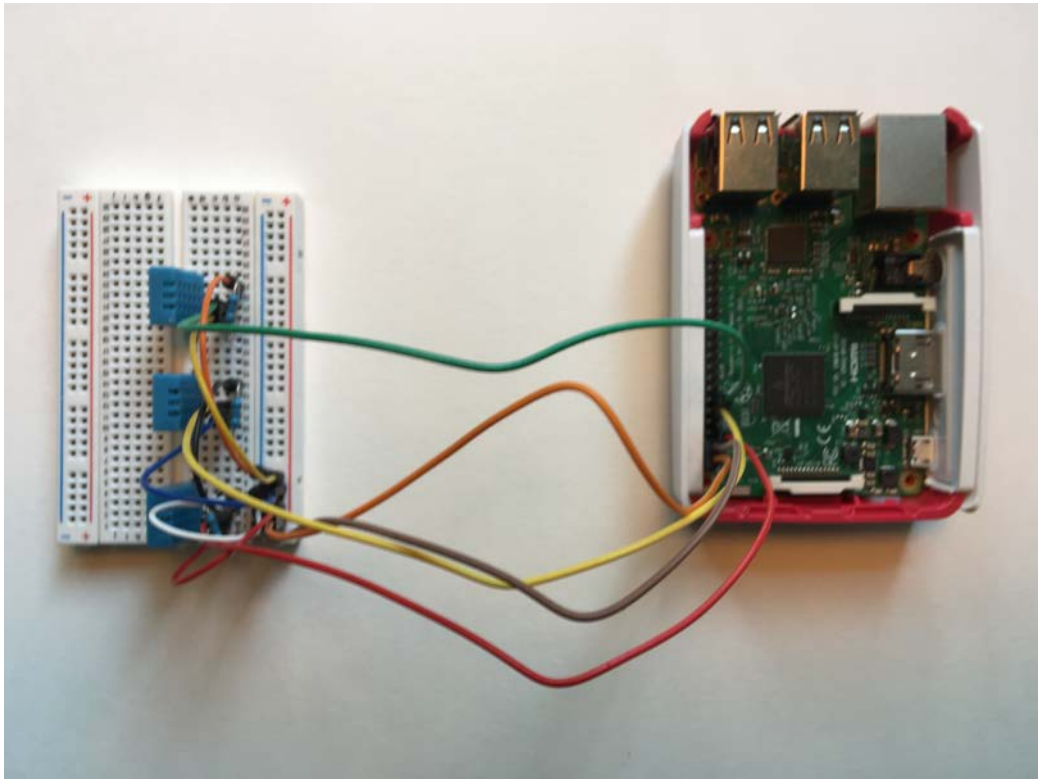


Abbildung 9.1: Sensornetz der entwickelten Anwendung bestehend aus einem Raspberry Pi und drei Sensoren des Typs DHT11

Kapitel 10

Zentraler Server und Web-Platform

Der Code des Hauptservers der Anwendung basiert auf einem Anwendungsbeispiel, welches über die Forge-Website und auf Github zum Download angeboten wird[1]. Dieses Gerüst stellt die Basisversion eines Viewers für CAD-Modelle dar und eignet sich hervorragend als Ausgangspunkt für Anwendungen, die den Viewer von Forge integrieren. Die Anwendungsvorlage besteht dabei aus dem Backend Server Code mit allen notwendigen Funktionen sowie dem client-side User Interface mit der Hypertext Markup Language (HTML) und Cascading Style Sheets (CSS) Code für die Website sowie dem für den Viewer notwendigen client-side Java Script (JS). Im Rahmen des Projektes wurde dieses Ausgangsgerüst modifiziert und um zahlreiche Funktionen erweitert, um den definierten Anforderungen gerecht zu werden und die gewünschten Funktionalitäten zu liefern. Das Forge Sample ist in den Sprachen Node.js, .NET, Go, PHP und Java verfügbar.
Für das vorliegende Projekt wurde Node.js zum Programmieren des Servers verwendet. Clientside wird JS verwendet. Dies bringt den Vorteil mit sich, dass Backend sowie Frontend der Anwendung bis auf kleine Unterschiede in der gleichen Syntax gestaltet werden können. Die Zeit, die zum Erlernen der für das Projekt notwendigen Programmiersprachen aufgebracht werden muss, lässt sich dadurch verkürzen.
Als Web-Framework wurde wie schon bei dem Server auf dem RPI *express* verwendet. *Express* ermöglicht die einfache Einrichtung der notwendigen Strukturen, z.B. das Routings, auf dem Server. Das Routing beschreibt, wie der Server auf HTTP Anfragen eines bestimmten Typs zu einem bestimmten Endpoint reagiert. Es wurde dabei Wert darauf gelegt, den Server als REST Web Service zu gestalten. Representational State Transfer (REST) definiert eine Reihe grundlegender Prinzipien für die Architektur von Webanwendungen. Diese sollen die Interoperabilität zwischen verschiedenen Anwendungen, eine gute Skalierbarkeit und eine intuitive Handhabung gewährleisten. Insbesondere handelt es sich dabei um folgende vier Designgrundsätze (Rodriguez, 2008):

[1] https://github.com/Autodesk-Forge/learn.forge.viewmodels/tree/nodejs, Stand: 20.05.2020

1. **Explizite Verwendung der HTTP-Methoden:**. Verwendet werden sollen dabei die vier Methoden POST, GET, PUT und DELETE, die mit den Create, Read, Update, Delete (CRUD)-Operationen korrespondieren. Für die Erstellung von Ressourcen soll PUT verwendet werden. Um Ressourcen vom Server abzufragen soll GET verwendet werden. Der Status oder Zustand einer bereits bestehenden Ressource soll mit der PUT-Methode verändert werden. Das Löschen von Ressourcen soll mit DELETE durchgeführt werden.

2. **Zustandslosigkeit:** die Antwort der Anwendung auf eine HTTP-Request soll nicht von einem vorhergegangenen Zustand abhängig sein. D.h. alle Informationen, die die Anwendung zur Erstellung einer Antwort benötigt, müssen komplett in der Anfrage enthalten sein. Zwischen zwei Anfragen werden keine Informationen auf dem Server gespeichert und wiederverwendet. Anfragen sind damit unabhängig und zustandslos. Diese Eigenschaft ist insbesondere dann von Bedeutung, wenn die Anwendung skaliert wird und z.B. die Kapazität durch die Nutzung weitere Server und Proxys erweitert werden soll. Damit jeder Server jede Anfrage, unabhängig von vorangegangenen Anfragen des Nutzers an andere Server des Clusters, bearbeiten kann, muss diese Unabhängigkeit in der Architektur gewährleistet sein.

3. **Konsistente und verständliche Addressierbarkeit der Ressourcen:** Diese syntaktische Anforderung soll die Benutzerfreundlichkeit der Anwendung steigern. Adressen, die zu bestimmten Ressourcen führen, sollen logisch und einer Ordnerstruktur gleich angelegt werden. Endpoints werden dabei über URLs angesteuert, welche dabei so benannt werden, dass es sich nicht nur um einen mit „/„ getrennten Namen handelt, sondern eine hierarchische Struktur der Ressourcen suggeriert wird. So wird in der vorliegenden Anwendung beispielsweise das Abfragen des Werteverlaufs des Sensors mit der ID 7 durch eine GET-Anfrage an die folgende URL abgefragt: <http://localhost:3000/sensor/data/history?sensorid=7>

4. **Übertragung von Extensible Markup Language (XML) oder JSON Dateien:** Daten die zwischen Nutzern und Server übertragen werden, sollen in den Dateiformaten XML oder JSON oder beiden zusammen ausgetauscht werden.

Diese Grundsätze wurden beim Aufsetzen des Servers so gut wie möglich beachtet und umgesetzt. Der Quellcode der gesamten Anwendung (d.h. sowohl der Code des Servers auf dem RPI als auch der Code des Hauptservers) sind der Arbeit in der digitalen Ausführung als Anhang beigefügt.

10.1 Serverarchitektur

Die Funktionsweise des Servers kann in zwei zentrale Aufgaben aufgeteilt werden. Zum einen stellt er das zentrale Steuerelement zum Initiieren des Auslesens der Sensoren, für die Datenverarbeitung und die Weiterleitung der Daten an die Datenbank dar. Zum anderen stellt er die Website für Nutzer (momentan über ein lokales Netzwerk) zur Verfügung. Im Folgenden wird die Server Architektur und Funktionsweise komponentenweise vorgestellt.

10.1.1 Abfragen der Sensorwerte

Der Hauptserver ist der zentrale Knoten, an dem die Werte aller Sensoren zur Überprüfung zusammen getragen werden bevor sie an die Datenbank zur Speicherung übergeben werden. Hierzu wird zunächst eine Liste aller registrierten Sensoren aus der Datenbank angefragt. Diese enthält neben Sensor ID und Netzwerkaddresse auch die vom Nutzer spezifizierte Auslesefrequenz. Das Ergebnis dieser Abfrage könnte folgendermaßen aussehen:

sensor_id	address	pin	type	component_id	status	intervall
1	pi1:8080	4	dht11	3542	1	10
2	pi1:8080	17	dht11	3574	1	10
3	pi1:8080	27	dht11	3527	1	10

Dieses Ergebnis wird in einem Array gespeichert und einer Funktion übergeben, die das Auslesen der Sensoren in der angegebenen Frequenz veranlasst. Dazu wird eine For-Schleife über alle Sensoren durchlaufen und jeder Sensor der unten stehenden Funktion als Parameter übergeben. Diese Funktion wird ausgeführt, solange der Server selbst läuft. Der Server sendet dazu in den definierten Intervallen HTTP GET-Requests an die entsprechende Netzwerkadresse, d.h. die Adresse des RPI an dem der Sensor montiert ist. Die ID des Sensors, welcher ausgelesen werden soll, wird in der URL als Query Parameter mit übergeben. Soll z.B. der erste Sensor aus dieser Tabelle ausgelesen werden, muss ein HTTP GET-Request an die URL <http://pi1:8080/data?sensorid=1> gesendet werden.
Als Antwort sendet der RPI die Messwerte zusammen mit der Sensor ID und einem Zeitstempel zurück. Die genaue Funktionsweise des Servers auf dem RPI wird im 9.1.1 beschrieben. Die Sensorwerte können dann auf Grenzwertüberschreitungen oder andere Faktoren hin überprüft werden. Im Fall einer Überschreitung kann eine Warnung ausgegeben werden. Die Sensormesswerte werden dann an die Datenbank zur Speicherung weitergeleitet. Dieses Vorgehen einer zentralisierten Steuerung des Auslesevorgangs wurde der einfacheren Handhabung wegen für dieses Projekt gewählt. Es hat Vorteile aber auch Nachteile. Auf der einen Seite wird der Ausfall eines Sensors oder eines RPI sofort im zentralen Server registriert. Bekommt der Server eine Fehlermeldung oder gar keine Antwort auf die GET-Request vom RPI so kann der Status des Sensors oder Systemteils direkt in der Datenbank aktualisiert werden und eine Warnung

an den Betreiber ausgegeben werden. Ein weiterer Vorteil ist die einfache Anpassbarkeit der Sensorabfragen über den Nutzer. Die Sensorauslesung efolgt über HTTP-Anfragen in den definierten Intervallen. Allerdings können auch zwischendurch jederzeit Auslesungen veranlasst werden, ohne den Betrieb zu unterbrechen. Konfigurationsänderungen, die der Nutzer vornimmt, wie das Einfügen eines neuen Sensors, werden sofort wirksam und nicht erst nach der Synchronisation des Servers auf dem RPI. Nachteil ist, dass bei einem Ausfall des zentralen Servers kein Sensor mehr ausgelesen wird.
Alternative könnte das Auslesen der Sensoren über eine Schleife direkt auf dem RPI zu veranlasst werden. Dabei kann es allerdings vorkommen, dass bei einem Ausfall kein Statusupdate beim zentralen Server ankommt. Um dem zu begegnen müsste dann eine zusätzliche Statusabfrage implementiert werden, die vom Hauptserver aus periodisch durchgeführt wird und überprüft, ob alle Teile des Systems noch planmäßig arbeiten. Optimalerweise sollte eine Kombination beider Möglichkeiten für ein Monitoring-System dieser Art in Betracht gezogen werden. Die Messwerte sollten auch bei einem Ausfall des zentralen Servers weiterhin abgefragt und gespeichert werden. Bei erneutem Betrieb des Servers können die Daten dann im Block an den Server und die Datenbank weitergeleitet werden. Zusätzlich sollte es weiterhin möglich sein, das Auslesen der Sensoren auch unabhängig der festgelegten Intervalle vom zentralen Server aus zu veranlassen. Im Rahmen dieser Arbeit kommt es jedoch für den Rest des Systems nicht darauf an, auf welche Art und Weise die Daten auf den Hauptserver und in die Datenbank gelangen. Daher wurde auf die Implementierung einer aufwändigeren Lösung verzichtet und die beschriebene einfache Variante gewählt.

10.1.2 Hosten der Website

Die zweite Aufgabe des Servers ist die Bereitstellung der Website, die das Herzstück der Anwendung darstellt. Diese besteht aus dem Backend Code des Servers auf der einen Seite und dem Frontend mit den HTML, CSS und Client-Side JS Dateien auf der anderen Seite. Es werden eine Reihe an APIs von Autodesk Forge eingebunden. Diese werden verwendet, um die Modelle in ein vom Browser darstellbares Datenformat umzuwandeln, die hochgeladenen Modelle in der Cloud zu speichern und vor allem die Modelle zu rendern. Mehr über die Funktionsweise und den Workflow der APIs im Abschnitt 10.3. Die Benutzeroberfläche und Funktionen der Website sind in den Abschnitten 10.4 bzw. 10.4 näher vorgestellt.

10.2 Datenbank

Die Auswahl einer geeigneten Speicherlösung für die Sensordaten ist eine zentrale Anforderung an jedes SHM-Projekt. Die gesammelten Sensordaten müssen zuverlässig und effizient gespeichert werden und sollten bei Bedarf leicht abgerufen werden können (vgl. Kapitel **??**). Verwendet wurde für dieses Projekt eine relationale Datenbank. Als Relational Database

Management System (RDBMS) wurde MariaDB verwendet. MariaDB ist durch eine Abspaltung von MySQL entstanden und wurde als „drop-in-replacement“ für MySQL entwickelt und um neue Optionen und Erweiterungen ergänzt. D.h. die Kompatibilität mit MySQL ist in der Regel vollständig gegeben und die Vor- und Nachteile einer MySQL Datenbank treffen grundsätzlich auf für MariaDB zu [2]. Dabei ist sie open source, sodass keine Lizenzgebühren zur Nutzung fällig werden. MySql (damit auch MariaDB) ist relativ schnell und findet gerade zur Speicherung von großen Mengen an Sensordaten große Verwendung. Einige der in Abschnitt 5 vorgestellten Projekte verwenden daher MySql als RDBMS.

> *„MySQL is one of the most popular database management systems currently in use. MySQL focuses on the features most people need. In other words, MySQL has fewer features than the other competitors, but has a small-sized and faster implementation being able to run modest desktop systems. It is easy to install without a lot of difficult and sophisticated configuration which is attractive to beginners of RDBMS. Also, MySQL is easy to build interfaces to other software written in C, PHP, Perl, Python, Ruby, and Microsoft .NET languages. For aspect of licensing, MySQL is an open-source code project […].“* (Koo *et al.*, 2011)

Für Datenbank-Operationen, wie z.B. das Importieren, Ändern oder Abfragen von Daten, wird die Datenbanksprache SQL verwendet. Die Syntax der Abfragesprache ist damit genau die gleiche wie für MySQL Datenbanken. Für die Kommunikation zwischen Node.js Server und Datenbank kann aufgrund der vollständigen Kompatibilität das Modul mysql über NPM eingebunden werden.
In der Datenbank müssen zum einen die Messwerte der Sensoren abgespeichert werden, und zum anderen Informationen über die Sensoren selbst. Diese Informationen beinhalten die geometrische Lage im Modell bzw. im Bauwerk sowie die Erreichbarkeit der Sensoren im Netzwerk und stellen damit die Verknüpfung zwischen den Messwerten im Bauwerk und den Sensoren im Modell dar. Die Datenbank enthält dafür folgende Tabellen:

- Eine Tabelle zur Speicherung der Sensoren. Hier werden alle im Netzwerk installierten Sensoren mit ID, Lage im Bauwerk, Netzwerkadresse und anderen Metadaten abgelegt.
- Eine Tabelle pro verwendetem Sensortyp zur Speicherung der Messwerte.
- Eine Tabelle, die jedem Sensortypen eine Messwertetabelle zuordnet.

Das Entity-Relationship-Diagramm der Datenbank findet sich in Abbildung 10.1. Für jeden neuen Sensortyp, der im Sensornetz verwendet wird, muss eine neue Tabelle für die Messwerte in der Datenbank angelegt werden. Da die Datentypen und die Anzahl der Parameter pro Auslesung zwischen verschiedenen Sensortypen sehr unterschiedlich sein können, wurde diese Lösung gewählt, um die Spaltenzahl pro Tabelle und die Anzahl der leeren Zellen zu

[2] https://mariadb.org, Stand 20.07.2020

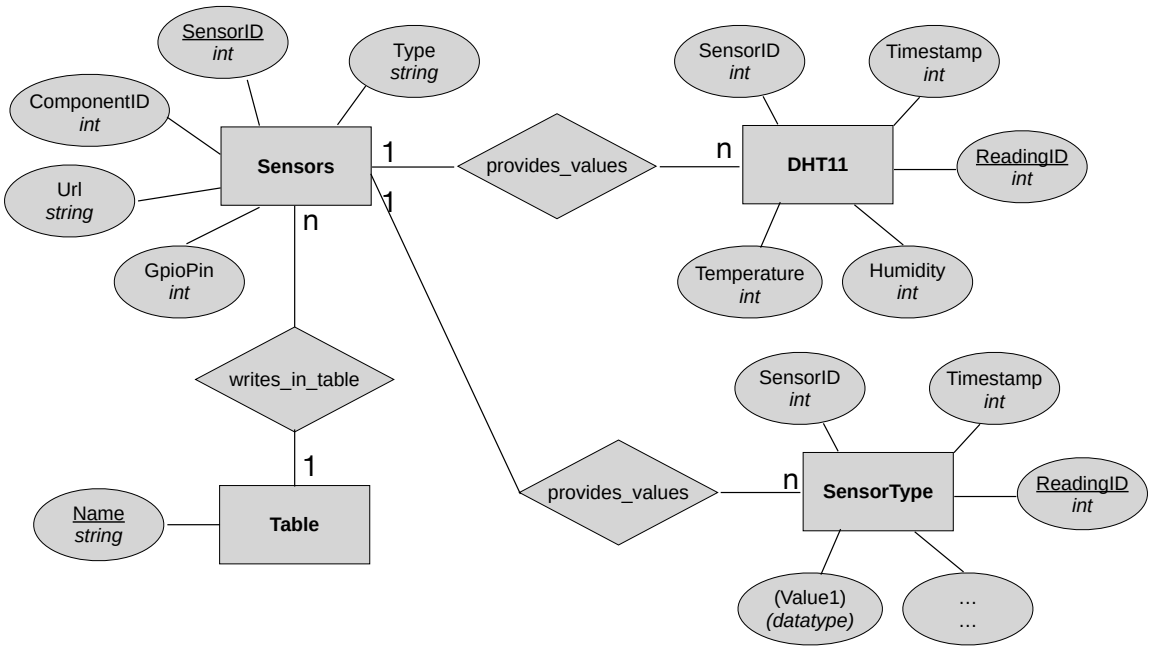


Abbildung 10.1: Entity-Relationship Diagramm der Datenbank

minimieren. Wird für die Messwerte aller Sensortypen eine einzige Tabelle verwendet, muss für jeden Werttyp, den ein Sensor liefern kann, eine eigene Spalte angelegt werden. In jeder Zeile werden dann nur die dem jeweiligen Sensor zugehörigen Zellen ausgefüllt, alle anderen bleiben frei, dadurch geht Speicherplatz verloren. Eine Tabelle für einen neuen Sensortyp kann mit folgendem SQL-Code erzeugt werden:

Erstellen der Messwertetabelle für einen neuen Sensortyp

```
1 CREATE TABLE IF NOT EXISTS <sensortyp>(
2     reading_id INT AUTO_INCREMENT PRIMARY KEY,
3     timestamp INT,
4     sensor_id INT,
5     <parameter> <DATATYPE>,
6 );
```

Der Name der Tabelle sowie die Parameter, die der Sensor liefert, müssen dabei abhängig vom Sensortyp definiert werden. Es können dabei beliebig viele Parameter-Datentyp-Paare angelegt werden. Alle anderen Parameter bleiben für jeden Sensortyp gleich. In Abbildung 10.2 ist die Tabelle zur Speicherung der Sensoren zu sehen mit allen Parametern und den zugehörigen Datentypen. Über *SensorID* und *ComponentID* lässt sich ein Sensor eindeutig einem Bauteil zuordnen und damit die Lage im Bauwerk festlegen. Die Parameter *Url* und *GpioPin* legen fest, über welchen GPIO-Pin der Sensor mit welchem RPI verbunden ist und wie er über das Netzwerk ansprechbar ist. Der Typ beschreibt den Sensor und legt zusätzlich fest, in welche Tabelle die Messwerte des Sensors geschrieben werden. *Status* ist ein Parameter,

sensors	
sensor_id	INT
address	VARCHAR
pin	TINYINT
type	VARCHAR
component_id	INT
status	BOOLEAN
intervall	INT

Abbildung 10.2: Datenbanktabellenschema für die Speicherung der Sensoren

der den Zustand des Sensors beschreibt. Hierfür ist ein logischer Wert vorgesehen, der anzeigt, ob der Sensor funktioniert oder nicht.

10.3 Forge APIs

Autodesk Forge bietet eine Reihe verschiedener APIs zur Nutzung an. Um diese verwenden zu können, muss ein Account bei Autodesk Forge eingerichtet werden. Für die akademische Nutzung ist dies kostenfrei. In dem Account können beliebig viele Apps registriert werden. Dazu muss der Name der App und eine Callback URL angegeben werden. Außerdem muss ausgewählt werden, welche Forge APIs von der App verwendet werden. Bei der Registrierung der App erzeugt Autodesk Forge dann Client ID und Client Secret. Diese können dazu verwendet werden, die angegebenen APIs in einer beliebigen Anwendung zu verwenden. Die Nutzung ist dabei für einige APIs kostenfrei, für andere wird je nach angefragter Leistung eine Gebühr fällig. Bei der Erstellung eines Accounts wird Nutzern ein Guthaben gutgeschrieben. Damit ist es möglich, ohne Kosten mit den APIs zu experimentieren und Anwendungen zu entwickeln. Von den vorhandenen APIs werden im Rahmen dieser Arbeit vier genutzt. Diese sind in den nachfolgenden Abschnitten vorgestellt. Eine ausführliche Dokumentation aller verfügbaren APIs findet sich bei Autodesk-Forge (2020).

10.3.1 Authentification

Die Verwendung der meisten Forge APIs verlangt einen gültigen Access Token. Dieser muss bei HTTP-Anfragen im Request Header übergeben werden. Damit wird sichergestellt, dass

die Anwendung bzw. der Nutzer tatsächlich die Berechtigung hat, die angefragte Ressource zu nutzen. Die App erhält einen solchen Token über die Authentification API, der Ablauf ist dabei folgendermaßen:

1. Die Anwendung sendet einen HTTP-Request an den OAuth REST Endpoint von Autodesk-Forge und übermittelt dabei die Credentials.
2. Als Antwort bekommt die Anwendung einen Access Token versehen mit einer Gültigkeitsdauer.
3. In folgenden Requests der Anwendung an andere APIs wird der Token im Request Header übergeben.

Die Credentials, bestehend aus Client Secret und Client ID, werden beim Registrieren der App auf der Autodesk Forge Platform erzeugt. Scopes sind Berechtigungen, die an die Tokens geknüpft werden. Wird mit einem Token ein Endpoint aufgerufen, so wird überprüft, ob die Berechtigung zur Nutzung durch die beinhalteten Scopes gegeben ist. Ist dem nicht so, wird die Nutzung verweigert. Das ist nützlich, wenn z.B. Endnutzern andere Berechtigungen gegeben werden sollen als dem Betreiber des Servers. In der entwickelten Anwendung werden deshalb Client-Side andere Scopes verwendet als Server-Side:

Unterschiedliche Scopes Server- und Client-Side

```
1 scopes: {
2         internal: ['bucket:create', 'bucket:read', 'data:read', 'data:
              create', 'data:write'],
3         public: ['viewables:read']
4     }
```

Mit einem durch den Authentification Flow erhaltenen Token können dann andere APIs genutzt werden, solange die benötigten Scopes vorhanden sind. Der Vorgang ist noch einmal im unten stehenden Flussdiagramm dargestellt.
Der vorgestellte Ablauf zeigt die von der entwickelten Anwendung verwendete Methode der 2-legged Authentication. D.h der Authentication Prozess findet nur zwischen Server und Autodesk-Forge Plattform statt. Der Endnutzer baut keinen direkten Kontakt mit Forge auf, sondern kommuniziert ausschließlich mit dem Server der Anwendung. Sollen auch Daten direkt zwischen Autodesk-Forge und dem Endnutzer ausgetauscht werden, muss eine 3-legged Authentication verwendet werden.

10.3.2 Model Derivative

Die nächste von der Anwendung verwendete API ist die Model Derivative API. Sie konvertiert Source Files verschiedenen Dateityps in sogenannte Derivatives, d.h. Output Files. Das

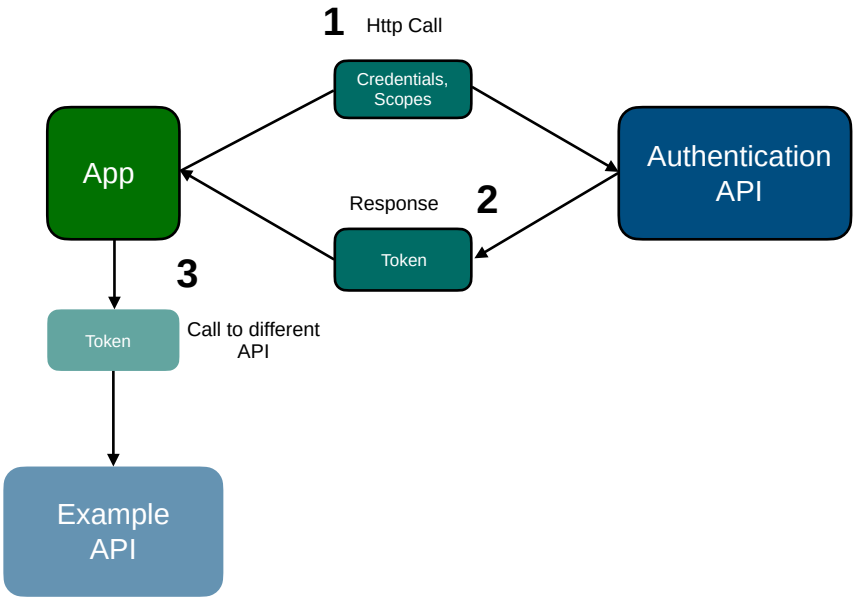


Abbildung 10.3: Authentication Flow

wichtigste Output Format ist dabei SVF, da dieser Dateityp vom Forge Viewer im Browser dargestellt werden kann. Es sind aber auch andere Dateitypen als Output möglich. So können aus dem Source File OBJ und STL Dateien, sowie Thumbnails verschiedener Größe erzeugt werden. Thumbnails können beispielsweise dazu verwendet werden, in Menüs eine Vorschau des Modells darzustellen. Als Input werden dabei ca. 80 verschiedene Formate akzeptiert. Neben den Autodesk eigenen Formaten wie .rvt, .dwf oder .dwg fallen darunter auch offene Dateiformate wie IFC. Die Model Derivative API von Forge ist in der Lage IFC Dateien in das SVF Format zu konvertieren. Allerdings werden zum aktuellen Zeitpunkt nur Versionen bis IFC 2x3 unterstützt. Die Erweiterung IFC-Bridge, die eine standardisierte Beschreibung von Brückenbauwerken in IFC erlaubt, existiert seit der Version 4x2 und lässt sich daher im Forge-Viewer momentan nicht anzeigen.

10.3.3 Data Management

Die Data Management API hat vier Funktionalitäten, von denen im Rahmen dieser Arbeit nur eine verwendet wird, nämlich der Object Storage Service (OSS). Dieser ermöglicht Upload, Download und Speichern von Dateien in der Cloud. Die Modelle müssen damit nicht jedes Mal beim Aufrufen der Anwendung neu hochgeladen und konvertiert werden, sondern können als Objekte in sogenannte Buckets abgelegt werden. Diese Buckets sind Cloudspeicher und eindeutig einer App zugewiesen. Dabei kann, für jeden Bucket gesondert, festgelegt werden, wie lange die Daten gespeichert bleiben, bevor sie automatisch gelöscht werden. Die API ermöglicht es auch Buckets zu erzeugen oder zu löschen.

10.3.4 Viewer

Der Viewer ist das Kernstück des Frontends der Anwendung. Es handelt sich dabei um eine WebGL basierte Bibliothek, die es ermöglicht Zweidimensional (2D) und Dreidimensional (3D) Modelle des Dateityps SVF im Browser zu rendern. Modelle, die mit CAD-Programmen erstellt wurden, lassen sich so ohne Installation der entsprechenden Software nach Konvertierung in das Format SVF im Browser darstellen. Die Konvertierung übernimmt dabei die Model Derivative API wie in Kapitel 10.3.2 dargestellt. Der Vorgang ist schematisch in Abbildung 10.4 dargestellt.

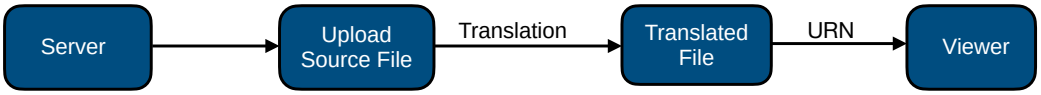


Abbildung 10.4: Workflow zur Bereitstellung des Modells an den Viewer

Neben herkömmlichen Navigationsfunktionalitäten, die es ermöglichen das Modell aus allen Blickrichtungen und in verschiedenen Detailgraden zu betrachten, bietet der Viewer noch weitere Möglichkeiten, mit dem Modell zu interagieren. So lassen sich Schnittebenen anlegen oder das Modell „explodieren“, um auch innen gelegene Bauteile sehen zu können. Da der Viewer alle Modellelemente auch als einzelne Objekte repräsentiert, ist es möglich einzelne Bauteile auszuwählen und zugehörige Daten zu extrahieren. So lassen sich geometrische Daten, wie Position und Ausdehnung aber auch Metadaten sowie eindeutige IDs zu den einzelnen Bauteilen, gewinnen. Diese Funktion wird im Fall der entwickelten Anwendung auch genutzt, um Sensoren in das Netz einzufügen und eindeutig einem Bauteil zuzuweisen. Der Viewer akzeptiert Erweiterungen, sogenannte Extensions, die bei Initialisierung des Viewers oder zu einem späteren Zeitpunkt geladen werden können. Damit ist es möglich dem Viewer eigene Funktionalitäten hinzuzufügen. Modular können so UI-Elemente wie Toolbars oder Buttons angepasst oder hinzugefügt und mit eigenen Methoden versehen werden. Die Extensions werden als Child der Autodesk.Viewing.Extension Klasse angelegt und erben deren Methoden. Dieses Vorgehen wurde für die Implementierung der meisten Funktionalitäten der Website verwendet. Mit dem unten stehenden Code lässt sich z.B. eine neue Toolbar mit einem neuen Button zur bestehenden Toolbargruppe des Viewers hinzufügen. Auf ähnliche Weise lassen sich auch die anderen in dieser Arbeit verwendeten UI-Elemente, wie z.B. Docking Panels, erzeugen.

Erstellen eines Buttons innerhalb des Forge-Viewers

```
this._group = new Autodesk.Viewing.UI.ControlGroup('NewToolbar');
            this.viewer.toolbar.addControl(this._group);
this._button = new Autodesk.Viewing.UI.Button('NewButton');
this._button.onClick = (ev) => {
            ...
        }
this._group.addControl(this._button);
```

10.4 Graphical User Interface und implementierte Funktionen

Die Benutzeroberfläche der Website ist in drei Bereiche unterteilt: Den Objekt Browser, den Forge-Viewer und ein Panel zur Darstellung der Sensorwerte (vgl. 10.5). Es wurde dabei der Versuch unternommen, die in Abschnitt 3.7 vorgestellten Prinzipien so weit wie möglich umzusetzen und die wichtigen Datenkategorien zu visualisieren. Der Objektbrowser erlaubt

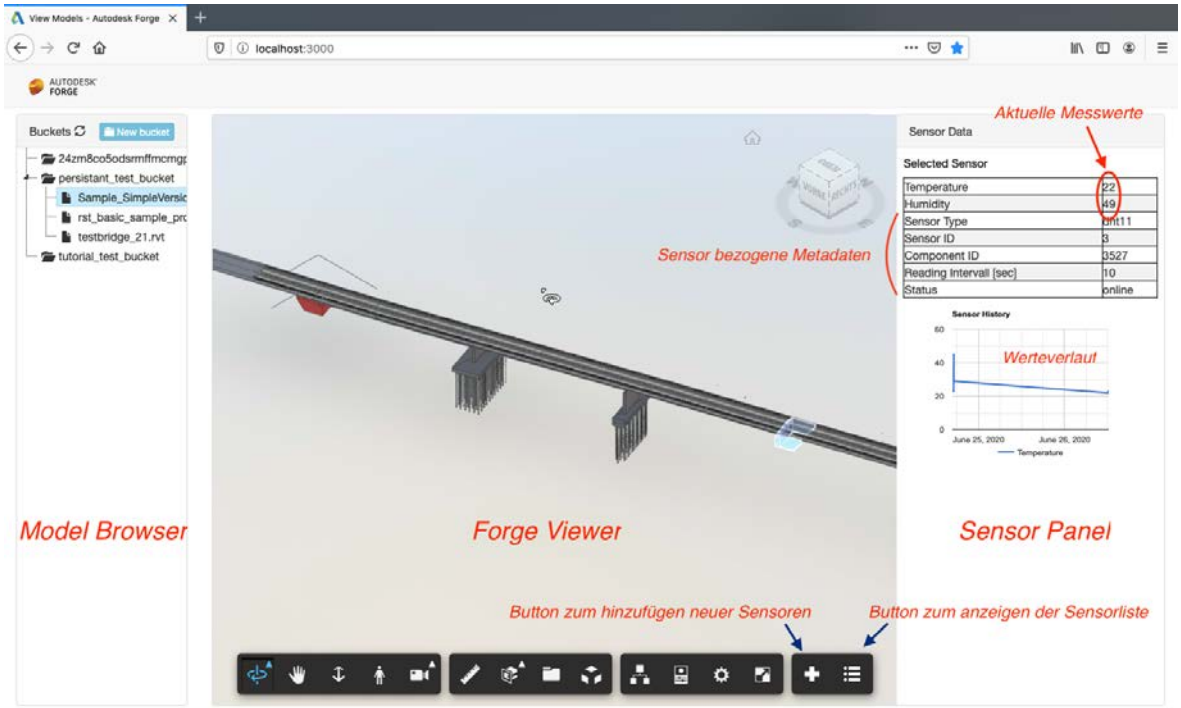


Abbildung 10.5: Benutzeroberfläche der Website

den Upload und die Verwaltung beliebiger Bauwerksmodelle. Hier kann zwischen den hochgeladenen Modellen navigiert werden. Mit einem Klick auf das Modell in der Ordnerstruktur wird es in den Viewer geladen und gerendert. Zusätzlich ist es möglich, neue Ordner in Form von Buckets anzulegen und zu benennen. Die Aufgabe des Forge-Viewers ist die Darstellung des Modells. Er bringt dabei einige tools mit, die die Naviagation im Modell sowie weitere

Funktionen ermöglichen wie z.B. die Erzeugung von Schnittebenen oder die Abfrage von Bauteildaten wie Abmessung und Material. Für diese Arbeit wurde der Viewer um eine Toolbar mit zwei Buttons erweitert, mit denen die Integration der Sensordaten bewerkstelligt wird. Das Sensorpanel schließlich bietet allen sensorbezogenen Daten Platz. Neben den aktuellen Messwerten und dem Werteverlauf fallen darunter auch Metadaten und Konfigurationsparameter. Neben den erwähnten Funktionalitäten des Forge-Viewers, die durch die Anwendung übernommen werden, wurden weitere Funktionen implementiert, die die Integration der Sensordaten und die Interaktion mit diesen ermöglichen.

10.4.1 Einfügen neuer Sensoren in das Netzwerk

Der Button Add Sensor öffnet ein Formular zum Hinzufügen eines neuen Sensors in das Netzwerk (Abbildung). Die Voraussetzung dafür ist, dass der Sensor bereits physisch im Netzwerk existiert, d.h. mit einem im Netzwerk registrierten RPI verbunden ist. Die Netzwerkadresse des RPI sowie die Nummer des GPIO-Pin muss dabei bekannt sein. Das Formular übernimmt die ID des aktuell ausgewählten Modellelements als Component ID. Dies stellt die Verknüpfung zwischen Sensor und Modell dar. Um sicherzustellen, dass nur bereits bekannte Sensorentypen in das Netz integriert werden, muss dieser über ein Dropdown ausgewählt werden. Alle weiteren Parameter, d.h. Netzwerkadresse des RPI, Nummer des verwendeten GPIO-Pin und die Auslesefrequenz müssen vom Nutzer eingegeben werden. Sind alle Felder ausgefüllt, kann das Formular abgesendet werden. Dabei werden die Daten über ein HTTP POST-Request an den Hauptserver geschickt. Nach Überprüfen der übermittelten Daten und Verifizieren der Existenz des spezifizierten Sensors wird ein neuer Eintrag in der Datenbank erstellt und eine Erfolgsnachricht an die Website zurückgegeben. Damit die Änderungen in der Datenbank wirksam werden und der Sensor ausgelesen wird, ist momentan noch ein Neustart des Servers notwendig.
Wenn kein Element ausgewählt wurde, wird eine Fehlermeldung ausgegeben. Momentan werden die Sensoren auf diese Weise mit Bauteilen verknüpft. Das ist eine schnelle Lösung, die Lage eines Sensors annäherungsweise darzustellen und abzuspeichern. Damit wird angezeigt, für welches Bauteil der Messwert des Sensors Relevanz hat. Für einige Anwendungsfälle bzw. Sensortypen kann dieses Vorgehen ausreichend sein. Es wird aber schnell klar, dass es hier zu Problemen kommen kann. Insbesondere wenn ein Sensor Werte für mehrere Bauteile liefert, kann dies in der aktuellen Implementierung der Sensorzuweisung und der Datenbank nicht abgebildet werden. Auch wenn an einem Bauteil mehrere Sensoren montiert werden, müssen bei der Implementierung der Darstellung der Sensorwerte Änderungen vorgenommen werden. Hier wird momentan bei der Auswahl eines Bauteils der korrespondierende Sensor mit dem bisherigen Werteverlauf angezeigt. Dieses Problem ließe sich aber leichter lösen. Dennoch könnte es im allgemeinen wesentlich sinvoller sein, die Sensoren selbst in das Modell zu integrieren, d.h. sie ebenfalls zu modellieren. Das bringt mehrere Vorteile und umgeht die

genannten Probleme. Eine Zuordnung von realem Sensor zu Modellelement ist hier eindeutig und besitzt eine 1:1 Beziehung. Da die Sensoren selbst Elemente des Modells sind, wird jeder Sensor genau einem Element im Modell zugewiesen. Weder können einem Element mehrere Sensoren zugewiesen werden, noch kann ein Sensor mehreren Modellelementen zugewiesen werden, was beide Probleme behebt. Welcher Sensor für welches Bauteil relevante Werte liefert, kann dann in einer eigenen Tabelle abgebildet werden, sodass auch diese Komponente abgedeckt ist.
Die geometrische Lage sowie die Eigenschaften des Sensors sind eindeutig festgelegt. Theoretisch können Grenzwerte für die einzelnen Sensoren aus dem Modell abgeleitet werden. Aus einem Building Information Model lassen sich z.B. kritische Belastungszustände berechnen und davon ableiten, welche Grenzwerte bei einem speziellen Sensor an einer definierten Stelle kritisch sind. Es ließen sich damit automatisch für jeden Sensor Grenzwerte ableiten, bei deren Überschreiten eine Warnung ausgegeben wird. Dieser Vorschlag geht allerdings weit über den Rahmen dieser Arbeit hinaus und bleibt Aufgabe für zukünftige Forschungsprojekte. Über die Integration von typischen Sensoren bei Brückenmonitoring in BIM wird in Abschnitt BIM näher eingegangen. Die Verknüpfung der realen Sensoren mit Sensoren im Modell scheint aus den genannten Gründen eine elegante Lösung zu sein. Auch wenn das über workarounds bereits möglich ist (vgl. BIM), ist die Integration von Sensoren in BIM was das Brückenmonitoring betrifft noch nicht ausgereift. Aus diesem Grund wurde für diese Arbeit folgende Lösung gewählt: Sensoren werden mit Elementen des Modells über Bauteil ID und Sensor ID verknüpft. Dabei werden momentan nur 1:1 Beziehungen bei der Implementierung berücksichtigt. Dieses Vorgehen ist nicht optimal, könnte für den Fall, dass die Sensoren selbst modelliert werden, genau so beibehalten werden. Aus oben genannten Gründen ist dies ohnehin erstrebenswert.

10.4.2 Hervorheben der Bauteile mit assoziierten Sensoren

Um die räumliche Lage der Sensoren im Bauwerk darzustellen, werden die Elemente des Modells, die mit Sensoren verknüpft wurden, farblich hervorgehoben. Dies ermöglicht einen schnellen Überblick der Position der Sensoren und erleichtert dem Nutzer die Auswahl dieser Elemente. Bei Initialisierung des Viewers wird über den Hauptserver eine Datenbankabfrage nach allen für dieses Modell vorliegenden Sensoren durchgeführt. Das Ergebnis wird vom Server in Form eines JSON Arrays als Antwort an die Website gesendet. Hier werden in einer For-Schleife alle Elemente, die über die Bauteil ID mit einem Sensor verknüpft wurden eingefärbt:

Einfärben der Bauteile

```
1 for (var sensor in window.sensorData){
```

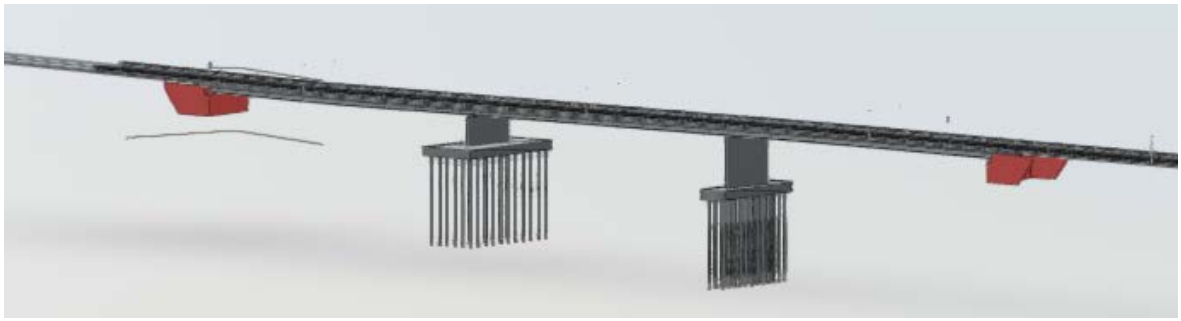


Abbildung 10.6: Bauteile mit Sensoren werden farblich hervorgehoben

```
2         this.viewer.setThemingColor(window.sensorData[sensor].component_id,
              red);
3     };
```

10.4.3 Darstellung aller Sensoren in Listenform

Der Sensortoolbar wurde ein weiterer Button hinzugefügt, der die Darstellung aller für dieses Modell registrierten Sensoren in Listenform ermöglicht. Diese Liste beinhaltet die Sensor ID

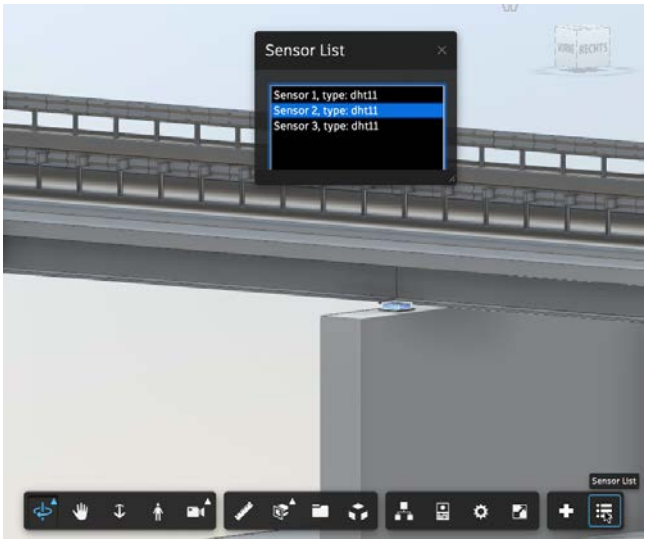


Abbildung 10.7: Darstellung und Auswahl der Sensoren über eine Liste

sowie den Typ der Sensoren und ermöglicht die Auswahl einzelner Sensoren. Als Container für die Liste wurde ein Docking Panel als Erweiterung des Viewers erzeugt. Es lässt sich verschieben und in der Größe verändern, um während der Nutzung der Liste die Sicht auf das Modell weiterhin zu gewährleisten.

10.4.4 Selektion eines Sensors

Die Auswahl eines Sensors und die damit verbundene Anzeige der Sensordaten ist auf zwei Arten möglich. Entweder über Selektion eines Modellelements mit assoziiertem Sensor, oder über Auswahl eines Sensors aus der beschriebenen Sensorliste. Bei Auswahl eines Bauteils im Viewer wird überprüft, ob Sensorwerte für dieses Element vorliegen. Dazu wurde eine weitere Extension nach dem oben genannten Muster angelegt. Verwendet wird die onSelectionEvent Methode des Viewers. Diese wird ausgeführt, sobald sich die Elementauswahl im Viewer ändert. Bei Laden des Modells werden hierzu von der Datenbank alle Sensoren abgefragt, die für das Modell registriert sind. Diese Liste wird in dem window Objekt gespeichert und ist damit global verfügbar. In der Extension wird dann in einer For-Schleife überprüft, ob die ID des ausgewählten Elements mit einer Bauteil ID eines der Sensoren übereinstimmt. Falls dies der Fall ist, werden die Daten des Sensors aus der Datenbank geladen und im Sensorpanel visualisiert. Das Auffinden der korrekten Modellelemente kann trotz der farblichen Hervorhebung mühsam sein und es besteht die Gefahr, dass Sensoren mit dieser Methode nicht leicht ausgewählt werden können. Aus diesem Grund wurde die Möglichkeit hinzugefügt, die Sensoren, wie oben beschrieben, ebenfalls aus einer Liste wählen zu können. Die Auswahl eines Sensors in der Liste bewirkt dabei die Auswahl des entsprechenden Modellelements. Da sich dadurch die Elementauswahl im Modell ändert, wird ebenfalls die beschriebene onSelectionEvent Methode ausgeführt und die Sensordaten werden dadurch angezeigt. Ein weiterer Vorteil ist, dass durch die Selektion des Modellelementes und die damit verbundene Markierung direkt die Lage des gewählten Sensors angezeigt wird.

10.4.5 Anzeige der Sensordaten

Die Anzeige der Sensordaten kann in zwei Bereiche unterteilt werden. Einerseits werden die aktuellen Messwerte des Sensors gemeinsam mit Metadaten wie Sensor ID, Bauteil ID oder Ausleseintervall in Form einer Tabelle angezeigt. Andererseits wird der Verlauf der Messwerte als Liniengraph geplottet. Bei Auswahl eines Sensors entweder durch Selektion des entsprechenden Modellelements oder durch Auswahl des Sensors aus der Liste werden die aktuellen Sensordaten geladen und angezeigt. Die Echtzeitdaten werden alle drei Sekunden erneuert und überschrieben, um dem Nutzer jederzeit die aktuellen Werte zu liefern. Dazu wird eine HTTP GET-Request mittels fetch an den /sensor/data Endpoint des Servers gesendet. Als Query Parameter wird die ID des Sensors übergeben. Auf dem Server wird dann eine Datenbankabfrage nach den Messwerten dieses Sensors durchgeführt. Dabei wird nur der aktuellste Eintrag aus der Datenbank mit folgendem SQL Code entnommen:

SQL-Code zur Abfrage des letzten Messwertes

```
1 'SELECT * FROM dht11 WHERE sensor_id=' + sensorId + ' ORDER BY reading_id
      DESC LIMIT 1'
```

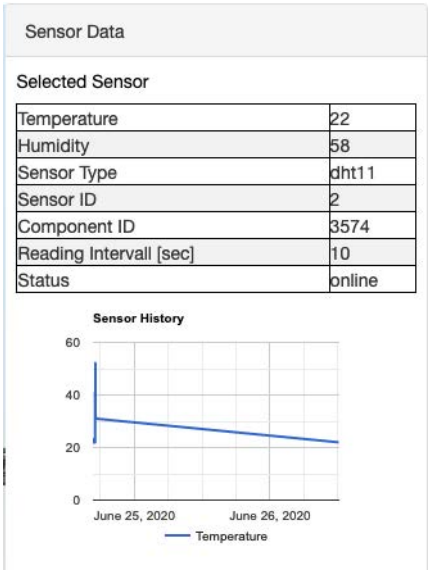


Abbildung 10.8: Darstellung der Messwerte und Sensordaten als Tabelle und Diagramm

Das Ergebnis der Query wird der Website als Antwort auf die Anfrage zurückgegeben. Der Verlauf der Messwerte wird auf ähnliche Art und Weise vom Server angefordert. Ein Unterschied besteht allerdings darin, dass der Verlauf anders als die aktuellen Messwerte nicht periodisch erneuert, sondern nur einmal bei Auswahl des Sensors abgefragt und geplottet wird. Für die Erzeugung des Diagramms wurde die Google Charts API verwendet [3]. Die Anzeige wird so lange beibehalten, bis der Nutzer eine Änderung der Element Selektion im Modell vornimmt, d.h. bis ein anderes Element (oder kein Element) ausgewählt wird.

[3] https://developers.google.com/chart, Stand: 22.07.2020

Kapitel 11

Einbindung weiterer im Web verfügbaren Ressourcen

Die Einrichtung neuer Monitoring Systeme ist immer mit Aufwand und Kosten verbunden. Es müssen Sensoren verbaut und neue Infrastruktur für die Überwachung angelegt werden. Aus diesem Grund kann es sehr vorteilhaft sein, auf bereits vorhandene und möglicherweise sogar frei verfügbare Ressourcen zurückzugreifen. Da es sich bei der im Rahmen dieser Arbeit entwickelten Anwendung ohnehin um eine Webanwendung handelt, die bereits bestimmte APIs einbindet, ist die Verwendung weiterer bereits verfügbarer APIs bezogen auf den entstehenden Nutzen möglicherweise mit relativ wenig Aufwand verbunden. An dieser Stelle sollen zwei mögliche Dinge genannt werden, deren Einbindung für das Brückenmonitoring je nach Ausrichtung sehr nützlich sein könnten.

1. APIs zur Einbindung von Verkehrsdaten
   Es gibt eine Reihe kleinerer und größerer Anbieter, die APIs für Verkehrsdaten anbieten. Allen voran ist hier die Google Maps API zu nennen. Diese stellt nicht nur Karten bereit, sondern ermöglicht es darüber hinaus, auch Echtzeit-Verkehrsdaten zu integrieren. Diese Angebote sind aber in der Regel nicht frei verfügbar, sondern kostenpflichtig. Verkehrsdaten sind nicht unbedingt aus Gründen der Standsicherheit relevant, können aber für Brückenbetreiber durchaus von Interesse sein. Je nach Ausrichtung und Ziel des Monitoring Systems kann eine Einbindung der aktuellen Verkehrslage, ohne diese selbst erheben zu müssen, daher sinnvoll und günstig sein.

2. Verkehrskameras
   Das Bayrische Staatsministerium für Wohnen, Bau und Verkehr stellt auf der Website bayerninfo.de aktuelle Bilder von Verkehrskameras zur Verfügung. Auch in anderen Bundesländern oder anderen Staaten finden sich ähnliche Angebote von unterschiedlichen Quellen. In der Regel handelt es sich dabei um Standbilder, die in unterschiedlichen

Intervallen erneuert werden, seltener werden Videostreams angeboten. Die Bilder sind dabei häufig frei über eine URL abrufbar. Bei günstiger Lage einer solchen Kamera wäre auch die Einbindung in ein SHM System zur Brückenüberwachung denkbar, um entweder den allgemeinen Zustand, die Witterungsbedingungen oder die momentane Verkehrsauslastung schnell einschätzen zu können.

Kapitel 12

Ausblick

Das sensorgestützte Monitoring gewinnt gerade für Brückenbauwerke immer mehr an Bedeutung. Zahlreiche Forschungsarbeiten beschäftigen sich mit der Suche nach geeigneten Methoden zur automatisierten Schadenserkennung und den besten Lösungen für Erfassung, Übertragung und Speicherung von Messwerten. Mit dieser Arbeit wurde der Versuch unternommen, einen Beitrag für die Integration von Bauwerksmodellen in diesenThemenkomplex zu leisten und die Ergebnisse eines solchen Monitorings nutzerfreundlich verfügbar zu machen. Die dabei entstandene Anwendung stellt einen simplen Prototypen dar und zeigt, dass die Verknüpfung von Modell und Sensordaten in Echtzeit mit den aktuell verfügbaren Mitteln gut realisierbar ist. Ein kleines Sensornetz liefert Messwerte, die visualisiert und auf der Website in das Bauwerksmodell integriert werden.
Mit der Forge Plattform von Autodesk existiert eine reiche Auswahl an Werkzeugen für die Darstellung und den Umgang mit Bauwerksmodellen, die dazu verwendet werden kann, SHM-Systeme zu implementieren und vor allem die visuelle Aufbereitung leicht verständlich zu gestalten. Die gemeinsame Darstellung des Bauwerks und der daran verbauten Sensoren könnte die Überwachung und Einschätzung des Zustandes eines Bauwerks wesentlich intuitiver und übersichtlicher werden lassen. Die Anwendung setzt nicht alle Anforderungen an ein SHM-System optimal um, zumal kein reales Bauwerk Ziel des Monitorings ist, auf welches das System ausgerichtet werden könnte. Die konkrete Konfiguration eines Monitoring Systems ist abhängig vom Bauwerk und dem Ziel der Überwachung, und stellt aufgrund der Einzigartigkeit von Bauwerken immer einen Einzelfall dar. Dennoch konnte mit der Anwendung ein Grundgerüst erstellt werden, das die Implementierung weiterer Komponenten wie z.B. Algorithmen zur Schadenserkennung ohne weiteres erlaubt und sich damit auf verschiedene Projekte anpassen lässt.
Auch die Forge Platform bietet noch viele Möglichkeiten, die im Rahmen dieser Arbeit nicht ausgeschöpft wurden und die Inhalt weiterer Forschungsprojekte bieten können. Insbesondere die Verwaltung verschiedener Projekte und die Interaktion mit den Bauwerksmodellen bietet noch viel Potenzial für die Installation von SHM-Systemen. So könnte die Möglichkeit

zur Konfiguration des Netzwerks über die Website weiter ausgebaut und um Funktionen ergänzt werden, die z.B. das Einfügen neuer Sensortypen oder das Entfernen von Sensoren ermöglichen. Auch die Datendarstellung kann interaktiver gestaltet werden, z.B. durch die Auswahlmöglichkeit verschiedener Anzeigezeiträume oder Datenfiltern. Eine offensichtliche Verbesserung für die Anwendung wäre wie bereits erwähnt die Modellierung der Sensoren als Teil des Bauwerksmodells. Die 1:1 Verknüpfung der Sensoren mit Bauteilen für ein reales Projekt bringt eine Reihe von Problemen mit sich und ist im allgemeinen zu ungenau. Es ist dabei denkbar, die Sensoren direkt über die Webplattform in das Modell einzufügen, auch wenn hierfür weitere APIs Verwendung finden müssten.
Die Integration von Sensordaten in Bauwerksmodelle wird bis jetzt relativ selten eingesetzt und bietet ein weites Feld für künftige Forschung. Mit der rasanten Entwicklung in den Bereichen IoT und BIM ist zu erwarten, dass sich die Art und Weise, wie der Status von Bauwerken festgestellt und überwacht wird, in Zukunft ändern wird. Die Verknüpfung der beiden Bereiche IoT und BIM wird dabei wahrscheinlich eine große Rolle spielen.

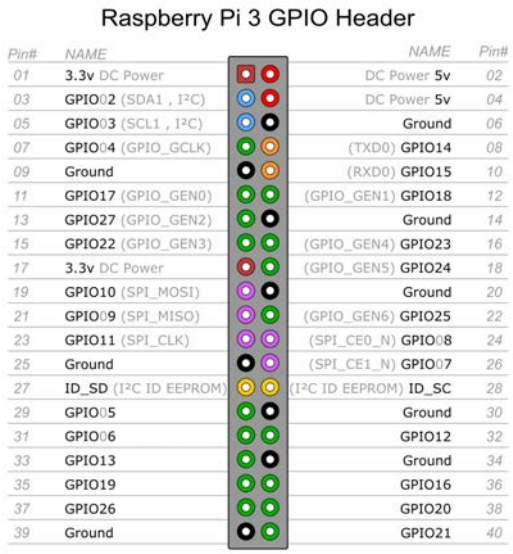
Anhang A

Bezeichnung des Anhangs

A.1 Quellcode der Anwendung

Der Quellcode der Entwickelten Anwendung stellt einen Teil dieser Abschlussarbeit dar und ist für die Dauer des Bewertungsvorgangs über folgenden Link einsehbar: https://syncandshare.lrz.de/getlink/fiKBKLWVTxUsCywEMU7qG1BL/

A.2 Raspberry Pi GPIO-Pinout



Quelle: https://raspitorials.blogspot.com/2016/03/hier-die-belegung-der-gpio-pins-fur-den.html, Stand: 20.06.2020

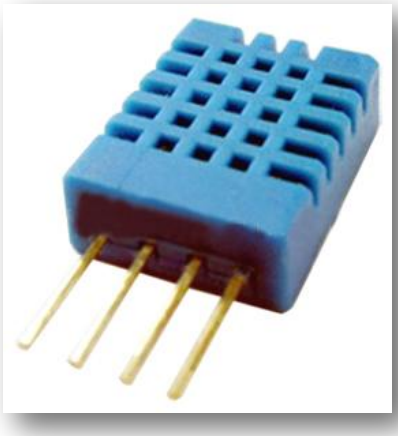
A.3 DHT11 Technisches Datenblatt

Das Technische Datenblatt wurde über den Link https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf abgerufen (Stand: 20.07.2020)

DHT 11 Humidity & Temperature Sensor

1. Introduction

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.



Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

3. Typical Application (Figure 1)

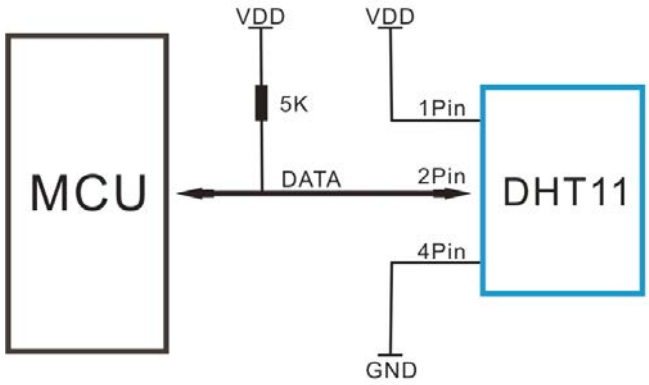


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

4. Power and Pin

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is **40bit**, and the sensor sends **higher data bit** first.
Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

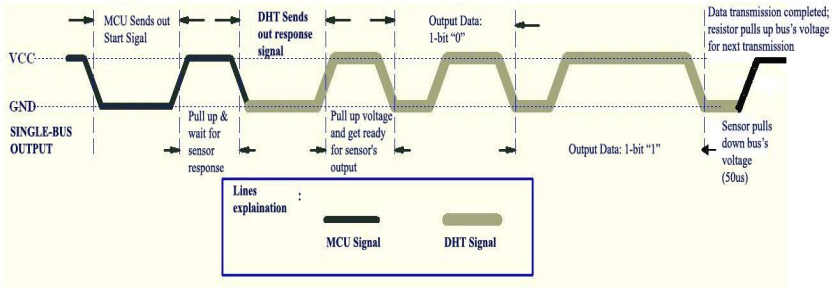


Figure 2 Overall Communication Process

5.2 MCU Sends out Start Signal to DHT (Figure 3, below)

Data Single-bus free status is at high voltage level. When the communication between MCU and DHT11 begins, the programme of MCU will set Data Single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.

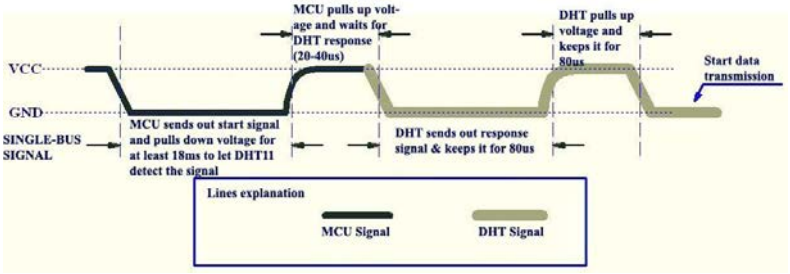


Figure 3 MCU Sends out Start Signal & DHT Responses

5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80us. Then the programme of DHT sets Data Single-bus voltage level from low to high and keeps it for 80us for DHT's preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80us and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50us low-voltage-level and the length of the following high-voltage-level signal determines whether data bit is "0" or "1" (see Figures 4 and 5 below).

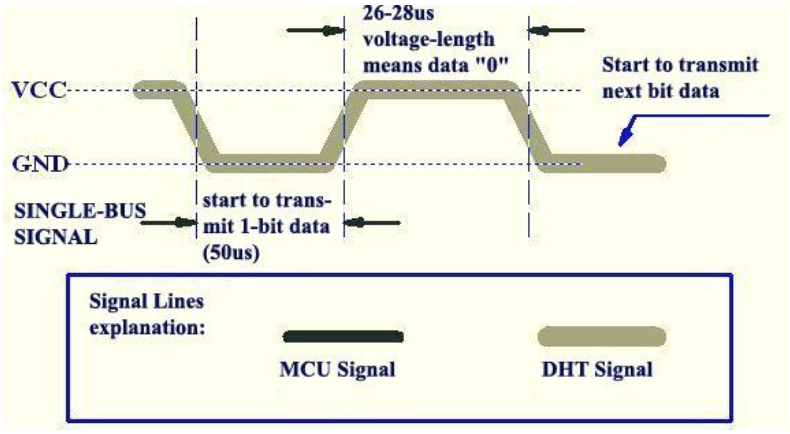


Figure 4 Data "0" Indication

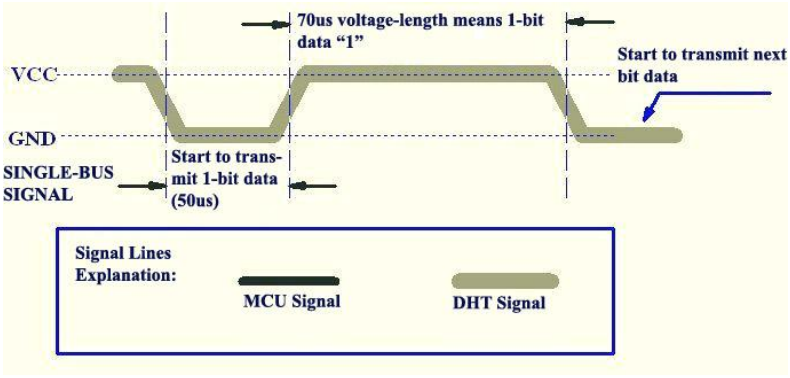


Figure 5 Data "1" Indication

If the response signal from DHT is always at high-voltage-level, it suggests that DHT is not responding properly and please check the connection. When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50us. Then the Single-Bus voltage will be pulled up by the resistor to set it back to the free status.

6. Electrical Characteristics

VDD=5V, T = 25℃ (unless otherwise stated)

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

Note: Sampling period at intervals should be no less than 1 second.

7. Attentions of application

(1) Operating conditions

Applying the DHT11 sensor beyond its working range stated in this datasheet can result in 3%RH signal shift/discrepancy. The DHT11 sensor can recover to the calibrated status gradually when it gets back to the normal operating condition and works within its range. Please refer to (3) of

this section to accelerate its recovery. Please be aware that operating the DHT11 sensor in the non-normal working conditions will accelerate sensor's aging process.

(2) Attention to chemical materials

Vapor from chemical materials may interfere with DHT's sensitive-elements and debase its sensitivity. A high degree of chemical contamination can permanently damage the sensor.

(3) Restoration process when (1) & (2) happen

Step one: Keep the DHT sensor at the condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;
Step two:K keep the DHT sensor at the condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Temperature Affect

Relative humidity largely depends on temperature. Although temperature compensation technology is used to ensure accurate measurement of RH, it is still strongly advised to keep the humidity and temperature sensors working under the same temperature. DHT11 should be mounted at the place as far as possible from parts that may generate heat.

(5) Light Affect

Long time exposure to strong sunlight and ultraviolet may debase DHT's performance.

(6) Connection wires

The quality of connection wires will affect the quality and distance of communication and high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius and contact should take less than 10 seconds.
* Avoid using the sensor under dew condition.
* Do not use this product in safety or emergency stop devices or any other occasion that failure of DHT11 may cause personal injury.
* Storage: Keep the sensor at temperature 10-40℃, humidity <60%RH.

Disclaimer

This is a translated version of the manufacturer's data sheet. OSEPP is not responsible for the accuracy of the translated information.

Literaturverzeichnis

Abdelgawad, A. & Yelamarthi, K. (2017). Internet of things (IoT) platform for structure health monitoring. *Wireless Communications and Mobile Computing* 2017.

Agarwal, Y., Upadhyay, P., Majgankar, R., Gaikwad, S. & Wairagade, V. (2019). Conditional assessment of bridges case study: Kund-Mala Bridge.

Alavi, A. H., Hasni, H., Jiao, P., Aono, K., Lajnef, N. & Chakrabartty, S. (2019). Self-charging and self-monitoring smart civil infrastructure systems: current practice and future trends. In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019*, Volume 10970, S. 109700W. International Society for Optics and Photonics.

An, Y., Chatzi, E., Sim, S.-H., Laflamme, S., Blachowski, B. & Ou, J. (2019). Recent progress and future trends on damage identification methods for bridge structures. *Structural Control and Health Monitoring* 26(10), S. e2416.

Aono, K., Hasni, H., Pochettino, O., Lajnef, N. & Chakrabartty, S. (2019). Quasi-self-powered piezo-floating-gate sensing technology for continuous monitoring of large-scale bridges. *Frontiers in Built Environment* 5, S. 29.

Aono, K., Lajnef, N., Faridazar, F. & Chakrabartty, S. (2016). Infrastructural health monitoring using self-powered Internet-of-Things. In: *2016 IEEE international symposium on circuits and systems (ISCAS)*, S. 2058–2061. IEEE.

Autodesk-Forge (2020). Forge APIs. https://forge.autodesk.com/en/docs/. Stand: 2020-6-28.

Bakht, B. & Mufti, A. (2015). *Bridges: Analysis, design, structural health monitoring, and rehabilitation*. Springer.

Borrmann, A., König, M., Koch, C. & Beetz, J. (2015). *Building Information Modeling: Technologische Grundlagen und industrielle Praxis*. Springer-Verlag.

Bundesregierung (2019). Umsetzungsstrategie Digitalisierung. https://www.bundesregierung.de/breg-de/themen/digital-made-in-de/building-information-modeling-bim--1547084. Stand: 2019-9-10.

Dhakal, D., Neupane, K., Thapa, C. & Ramanjaneyulu, G. (2013). Different techniques of structural health monitoring. *Research and Development (IJCSEIERD)* 3(2), S. 55–66.

Farrar, C. R. & Jauregui, D. A. (1998). Comparative study of damage identification algorithms applied to a bridge: I. Experiment. *Smart materials and structures* 7(5), S. 704.

Farrar, C. R., Park, G., Allen, D. W. & Todd, M. D. (2006). Sensor network paradigms for structural health monitoring. *Structural Control and Health Monitoring: The Official Journal of the International Association for Structural Control and Monitoring and of the European Association for the Control of Structures* 13(1), S. 210–225.

Farrar, C. R. & Worden, K. (2007). An introduction to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 365(1851), S. 303–315.

Fraser, M., Elgamal, A.-W., He, X. & Conte, J. (2010). Sensor Network for Structural Health Monitoring of a Highway Bridge. *Journal of Computing in Civil Engineering* 24, S. 11–24.

Glisic, B., Inaudi, D. & Casanova, N. (2010). SHM process as perceived through 350 projects. In: *Smart Sensor Phenomena, Technology, Networks, and Systems 2010*, Volume 7648, S. 76480P. International Society for Optics and Photonics.

Glisic, B., Yarnold, M. T., Moon, F. L. & Aktan, A. E. (2014). Advanced visualization and accessibility to heterogeneous monitoring data. *Computer-Aided Civil and Infrastructure Engineering* 29(5), S. 382–398.

Hasni, H., Jiao, P., Lajnef, N. & Alavi, A. H. (2018). Damage localization and quantification in gusset plates: A battery-free sensing approach. *Structural Control and Health Monitoring* 25(6), S. e2158.

Koo, K. Y., Battista, N. D. & Brownjohn, J. M. (2011). SHM data management system using MySQL database with MATLAB and web interfaces. In: *5th International Conference on Structural Health Monitoring of Intelligent Infrastructure (SHMII-5), Cancún, México*, S. 589–596.

Kuchekar, C. & Deshpande, U. (2017, mar). Visual Inspection of Concrete Bridge. *International Journal Of Innovations In Engineering Research And Technology [Ijiert]* 4, S. 125–127.

Kumar, R. & Rajasekaran, M. P. (2016). An IoT based patient monitoring system using raspberry Pi. In: *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, S. 1–4. IEEE.

Lapp, A. (2017). Big Data, SQL und NoSQL – eine kurze Übersicht. https://www.bigdata-insider.de/big-data-sql-und-nosql-eine-kurze-uebersicht-a-602249/. Stand: 2020-6-28.

Lee, S. & Kalos, N. (2015). Bridge inspection practices using non-destructive testing methods. *Journal of Civil Engineering and Management* 21(5), S. 654–665.

Mahmud, M. A., Bates, K., Wood, T., Abdelgawad, A. & Yelamarthi, K. (2018). A complete internet of things (IoT) platform for structural health monitoring (shm). In: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, S. 275–279. IEEE.

Meier, A. (2007). *Relationale und postrelationale Datenbanken*. Springer-Verlag.

Meier, A. (2018). *Werkzeuge der digitalen Wirtschaft: Big Data, NoSQL & Co*. Springer.

Phares, B. M., Washer, G. A., Rolander, D. D., Graybeal, B. A. & Moore, M. (2004). Routine highway bridge inspection condition documentation accuracy and reliability. *Journal of Bridge Engineering* 9(4), S. 403–413.

Rio, J., Ferreira, B. & Martins, J. P. P. (2013). Expansion of IFC model with structural sensors.

Rodriguez, A. (2008). Restful web services: The basics. *IBM developerWorks* 33, S. 18.

Sargsyan, N., Sargsyan, G. & Sargsyan, A. (2019, dec). Bridge health monitoring: the Davtashen bridge example in Yerevan. *IOP Conference Series: Materials Science and Engineering* 698, S. 077009.

Steiner, R. (2009). *Grundkurs Relationale Datenbanken*, Volume 7. Springer.

Tong, X., Yang, H., Wang, L. & Miao, Y. (2019). The development and field evaluation of an IoT system of low-power vibration for bridge health monitoring. *Sensors* 19(5), S. 1222.

Webb, G., Vardanega, P. J. & Middleton, C. R. (2015). Categories of SHM deployments: technologies and capabilities. *Journal of Bridge Engineering* 20(11), S. 04014118.

Zhang, H., Guo, J., Xie, X., Bie, R. & Sun, Y. (2013). Environmental effect removal based structural health monitoring in the internet of things. In: *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, S. 512–517. IEEE.

Abbildungsverzeichnis

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelor-Thesis selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ich versichere außerdem, dass die vorliegende Arbeit noch nicht einem anderen Prüfungsverfahren zugrunde gelegen hat.

München, 26. Juli 2020

Nepomuk Wolf

Nepomuk Wolf

[redacted]