



Hochschule Neubrandenburg
University of Applied Sciences

Hochschule Neubrandenburg

Studiengang Geoinformatik

Webbasierte dynamische Visualisierung gewässerbezogener Informationen für linear referenzierte Daten

Bachelorarbeit

vorgelegt von: *Martin Neumann*

Zum Erlangen des akademischen Grades

„Bachelor of Engineering“ (B.Eng.)

Erstprüfer: Prof. Dr.-Ing. Andreas Wehrenpfennig

Zweitprüfer: Dipl.- Ing. André Steinhäuser

Eingereicht am 06.06.2017

URN: urn:nbn:de:gbv:519-thesis2017-0319-4

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Neubrandenburg, den 06.06.2017

Unterschrift

Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, um mich bei allen zu bedanken, die mich während der Entstehung dieser Arbeit unterstützt haben. Ein spezieller Dank richtet sich an Herrn Prof. A. Wehrenpfennig und Herrn Dipl.- Ing. A. Steinhäuser für die hervorragende Betreuung meiner Bachelorarbeit. Mit ihren Fragen und Anmerkungen zu meiner Arbeit haben sie mir sehr geholfen.

Ein besonderer Dank geht an meine Familie und meine Freunde für das Interesse an meinen Fortschritten und der ständigen Motivation.

Kurzfassung

Durch die Erfindung des Computers wurde das Sammeln großer Datenmengen erst ermöglicht. Aus dem Auswerten und Vergleichen dieser Datenmengen entwickelte sich die Aufgabe diese auch ansprechend zu präsentieren.

Hierbei gewinnen Webapplikationen aufgrund von verbesserter Bandbreite, Browserprogrammen und Webtechnologien immer mehr an Bedeutung, um als Plattform zu dienen, die Daten bereitzustellen, um sie zu analysieren und weiterzuverarbeiten.

Im Rahmen dieser Bachelorarbeit wurde ein Konzept entworfen, dass es ermöglicht den Längsschnitt eines Fließgewässers darzustellen in Bezug auf seine ökologischen und hydromorphologischen Eigenschaften.

Abstract

The collecting of big data amounts was only allowed by the invention of the computer. From evaluating and comparisons of these data amounts developed the job to present this also attractively.

On this occasion, web applications win on the basis of improved range, browser programmes and web technologies more and more in meaning, to serve as a platform, to provide the data to analyse them and to process.

Within the scope of this bachelor's work a draught was sketched that it enables to show the profile of assembly-line waters concerning his ecological and hydromorphologischen components.

Inhaltsverzeichnis

1	Einleitung und Ziele der Bachelorarbeit.....	7
2	Fachliche Grundlagen.....	8
3	Anforderungsanalyse.....	14
3.1	Funktionale Anforderungen.....	14
3.2	Nicht funktionale Anforderungen.....	16
4	Webanwendung.....	18
4.1	Aufbau.....	18
4.2	Auswahl eines Webservers.....	19
4.3	Fazit.....	21
5	Bootstrap.....	22
6	Datenbanken.....	24
7	Skriptsprache.....	25
7.1	Erweiterung des Webservers um eine Skriptsprache.....	25
7.2	Konfiguration des Webservers.....	27
8	Framework.....	29
8.1	Auswahlkriterien.....	29
8.2	Darstellungs- und Diagrammoptionen.....	29
8.3	Auswahl eines Frameworks.....	30
8.3.1	mit Grundfunktionalität.....	30
8.3.2	Frameworks mit erweiterter Funktionalität.....	31
8.4	Beurteilung der Frameworks.....	32
8.4.1	Statische Diagrammoptionen.....	33
8.4.2	Dynamische Diagrammoptionen.....	33
9	Datenanalyse.....	33
10	Entwurf.....	39
10.1	Systemumgebung im LUNG.....	39
10.2	Architekturentwurf.....	42
10.3	Komponentendiagramm.....	43
10.4	Schnittstellenspezifikation.....	45
10.5	Klassendiagramm.....	46
10.6	Aktivitätsdiagramm.....	50
10.7	Datenflussdiagramme.....	52
10.8	Feinentwurf.....	55

10.8.1	Datenselektion und Speicherung	55
10.8.2	Datenabfrage.....	62
10.8.3	Datenaufbereitung	65
10.9	Erweiterung des Systems um neue Attribute	71
11	Prototypische Umsetzung	72
12	Ausblick.....	79

1 Einleitung und Ziele der Bachelorarbeit

Die Aufgabe im Pflichtpraktikum war es, sich einen Überblick zu verschaffen wie linear referenzierte gewässerbezogene Daten dargestellt werden können. Das Ergebnis war eine Anwendung, die den Längsschnitt eines Flussabschnittes zur Laufzeit generiert und Daten zur Gewässerstrukturgüte in einem speziellen Flächendiagramm darstellte. Die zu Grunde liegenden Werte zur Erzeugung des Diagramms befinden sich in der WRRL-Datenbank des Landesamtes für Umwelt, Naturschutz und Geologie.

Auf der Grundlage basiert das Thema dieser Abschlussarbeit. Das Konzept, um dynamisch Diagramme zu erzeugen, die zur Laufzeit vom Nutzer individuell angepasst werden können, besteht aus drei Teilen: einer Datengrundlage die in einem Datenbankmanagementsystem (DBMS) vorliegt, einer Anwendung oder Bibliothek, die Daten für Diagramme aufbereitet und schließlich dem Diagramm selbst, welches webbasiert dem Nutzer angezeigt wird.

Der Kern des Konzeptes ist ein Webframework, welches benötigte Daten aus der Datenbank einliest und die Werte erzeugt für das Diagramm. Diese Ergebnisse werden webbasiert dargestellt. Untersucht werden die Anforderungen an das zu erzeugende Diagramm und welches Framework dafür in Frage kommen.

Als eigentliches Ergebnis des Gesamtkonzeptes steht ein interaktives Diagramm mit kombinierbarem Inhalt im Vordergrund, mit dem man alle stationiert vorliegenden Gewässerdaten wie z.B. die Strukturgüte, Flächennutzung oder Bauwerke in einem Diagramm darstellen kann.

2 Fachliche Grundlagen

Das Landesamt für Umwelt, Naturschutz und Geologie (LUNG) ging am 01.01.1999 aus den Fachbehörden Landesamt für Umwelt und Naturschutz(LAUN) und dem geologischen Landesamt (GLA) hervor und seitdem beziehen sich die Aufgaben des LUNG auf die Erfassung, Dokumentation und Bewertung des Umweltzustandes, sowie der Erarbeitung von Grundlagen für die Planung und Umsetzung landesweiter Schutzmaßnahmen.

Die Beratungsfunktion gegenüber den staatlichen Ämtern, der Landesregierung oder den Kommunen ist auch ein wesentlicher Bereich im Aufgabenspektrum des LUNG.

Die Organisationsstruktur des Landesamtes schließt 5 Abteilungen mit ein:

- 1) Allgemeine Abteilung
- 2) Naturschutz und Naturparke
- 3) Geologie (geologischer Landesdienst), Wasser und Boden
- 4) Immissionsschutz und Abfallwirtschaft
- 5) Umweltanalytik und Strahlenschutz

In der Abteilung 3 befindet sich u.a. das Dezernat EG-Wasserrahmenrichtlinie.

Mit dem Erlass der europäischen Richtlinie 2000/60/EG zur Schaffung eines Ordnungsrahmens für Maßnahmen der Gemeinschaft im Bereich der Wasserpolitik wurde die EG-Wasserrahmenrichtlinie oder kurz WRRL ins Leben gerufen und verfolgt seitdem folgende Ziele:

- den Zustand der Gewässerökosysteme und der unmittelbar von ihnen abhängenden Landökosysteme und Feuchtgebiete zu schützen und zu verbessern,
- eine nachhaltige Wassernutzung zu fördern,
- die Einleitung und Freisetzung sogenannter prioritärer Stoffe und prioritärer gefährlicher Stoffe in die Gewässer zu reduzieren oder einzustellen,
- die Verschmutzung des Grundwassers zu verringern und
- die Auswirkungen von Überschwemmungen und Dürren zu mindern

- Einzugsgebietsbezogene Bewirtschaftung der Gewässer

Unter einem Einzugsgebiet ist ein Gebiet zu verstehen, aus dem der gesamte Oberflächenabfluss an einer einzigen Flussmündung in das Meer gelangt. Der Einzugsgebietsansatz fasst somit das Einzugsgebiet, die zugehörigen Oberflächengewässer sowie das zugeordnete Grundwasser als eine Einheit zusammen. Da sich Flussgebiete nicht an Verwaltungsgrenzen halten, dürfen diese auch der Bewirtschaftung im Ganzen nicht entgegenstehen. M-V ist in vier Flussgebietseinheiten unterteilt, wie in Abbildung 1 zu sehen ist:



Abbildung 1 - Flussgebietseinheiten M-V¹

Bei Fließgewässern in einer Flussgebietseinheit führen Unterschiede in der Substratbeschaffenheit, der Fließgeschwindigkeit und den physiko-chemischen Eigenschaften des Gewässers (z.B. Temperatur, pH-Wert, Leitfähigkeit, Sauerstoffgehalt und Nährstoffverhältnisse) zur Ausbildung unterschiedlicher Lebensgemeinschaften von Pflanzen und Tieren (Biozönosen).

¹ Wasserrahmenrichtlinie M-V

Hinzu kommen faunistische und floristische Unterschiede zwischen den verschiedenen Ökoregionen (z. B. Alpengebiet, Tiefland), in denen sich die Gewässer befinden. Diese Unterschiede werden bei der typspezifischen Bewertung des ökologischen Zustandes nach der WRRL berücksichtigt.

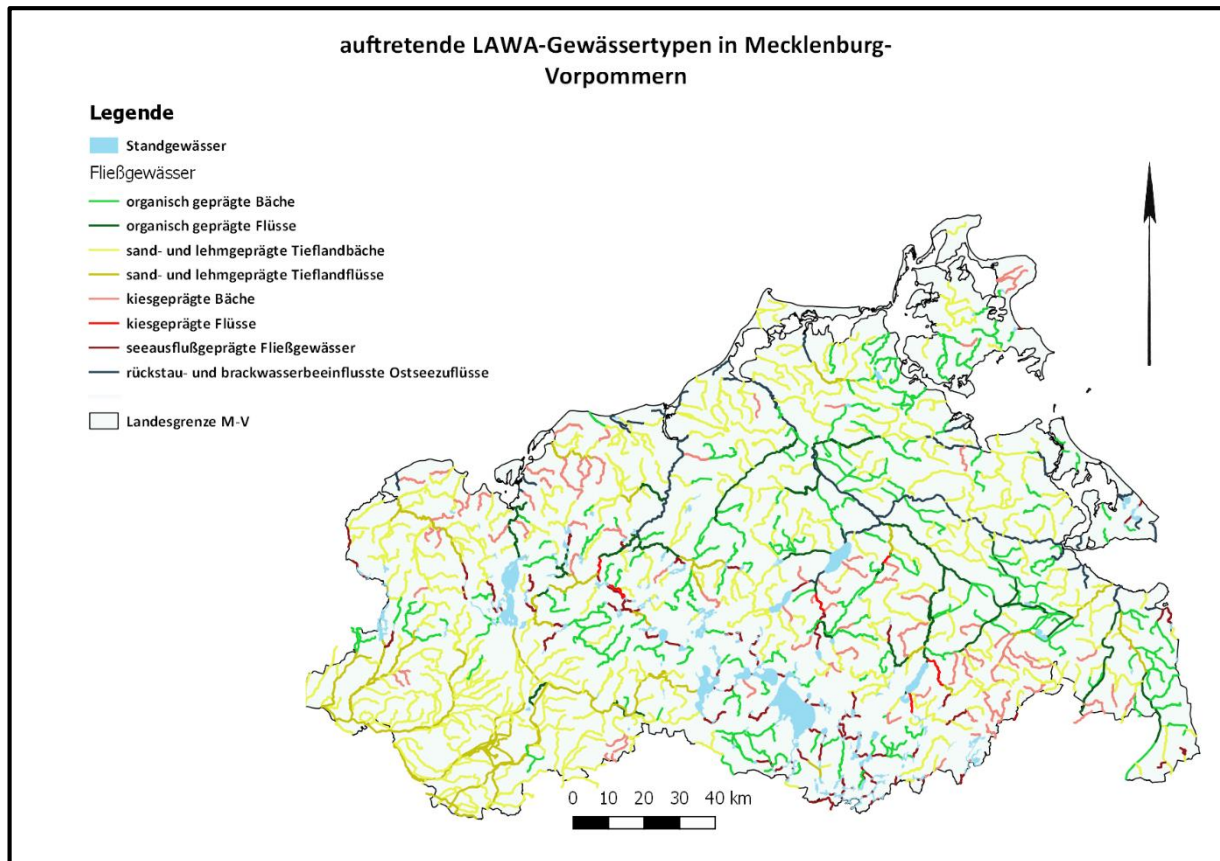


Abbildung 2 - in M-V auftretende Fließgewässertypen

Hierbei dienen die Fließgewässertypen als Referenz in biologischer, chemischer und hydromorphologischer Hinsicht. Ein Gewässertyp stellt immer einen „Idealfall“ dar, der in der Realität individuell ausgestaltet auftritt.

Die Bewertung der Fließgewässer bzw. Fließgewässer-Wasserkörper erfolgt hinsichtlich ihres ökologischen und ihres chemischen Zustandes. Der ökologische Zustand wird dabei im Wesentlichen aus den biologischen Qualitätskomponenten sowie unterstützenden Komponenten wie der Hydromorphologie abgeleitet.

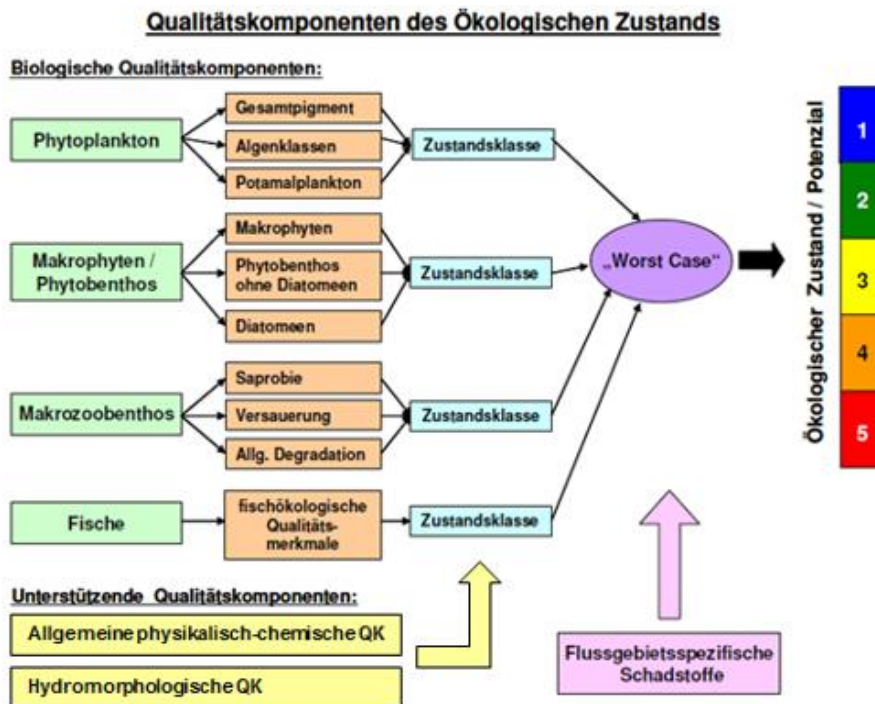


Abbildung 3 - biologische Fließgewässerbewertung in M-V ²

Die LAWA (2000) sieht grundsätzlich zwei Verfahren zur Gewässerstrukturkartierung vor: das Übersichtsverfahren und das Vor-Ort Verfahren, welche im Folgenden erläutert werden.

Das Übersichtsverfahren stützt sich bei der Erfassung von Parametern und deren Bewertung auf Topografische Karten im Maßstab 1:25000 oder Geologischen Karten. Zusätzlich werden Informationen von Ortskundigen herangezogen, sodass Vor-Ort Begehungen eine Ausnahme darstellen. Erfasst werden in der Regel 1 Kilometer Abschnitte oder ein Mehrfaches.

Im Vor-Ort Verfahren hingegen werden die Daten zumeist direkt an den Gewässern erhoben und in 100m Abschnitten bewertet.

Auf der Basis dieser Verfahrensarten wurde in Zusammenarbeit mit dem Institut biota GmbH und Intevation Hannover ein ähnliches Verfahren entwickelt, welches in M-V angewandt wird. Dabei erfolgt die Bewertung von Fließgewässern bis zu einer Breite von 25m anhand von 6 Hauptparametern, die 28 Merkmale für die Beurteilung des ökologisch-morphologischen Zustandes enthalten.

² Vortrag in der AL-Runde Biologische Bewertung von Fließgewässern

Diese Parameter lassen sich den drei Gewässerkompartimenten „Sohle“, „Ufer“, „Land“ zuordnen. Es wird für dieses Verfahren ein 5-stufiges Bewertungssystem verwendet, welches sich am natürlichen Zustand des Gewässers orientiert.

Die Berechnung wird automatisch über die Fachschale „Fließgewässerstrukturkartierung“ als Bestandteil der WRRL-Datenbank vorgenommen.³

Durch die so gewonnenen Datensätze zu den einzelnen Flussabschnitten lässt sich ein Längsschnitt erzeugen, der sowohl die strukturelle Bewertung einer Gewässerroute darstellt, als auch ergänzende Informationen stationiert wiedergibt.

Eine mögliche Art diese Daten zu visualisieren, ist die X-Achse mit der Stationierung von der jeweiligen Gewässerroute zu belegen und im positiven Bereich der Y-Achse die FGSK-Güteklassen abzubilden. Im negativen Bereich werden zusätzliche stationiert hinterlegte Informationen und Belastungen wie Querbauwerke oder Zuläufe von Kläranlagen dargestellt.

Der folgende skizzenhafte Aufbau einer Website ist in Abbildung 4 zusehen.



Abbildung 4 – mögliche Darstellung von Güte- und Belastungsinformationen

Auf diese Weise bietet die zu entwickelnde Web-Applikation einen Gesamtüberblick über einzelne Wasserkörper und unterstützt die Maßnahmenplanung durch die Identifizierung von Bereichen mit Handlungsbedarf mittels Überlagerung von Güte- und Belastungsinformationen.

3 Anforderungsanalyse

Dieses Kapitel befasst sich mit den zentralen funktionalen und nichtfunktionalen Anforderungen an die zu entwickelnde Web-Anwendung. Unter Zuhilfenahme der Ergebnisse dieser Anforderungsanalyse werden dann im nächsten Abschnitt die funktionalen Komponenten der Anwendung mittels des konzeptionellen Aufbaus erläutert.

3.1 Funktionale Anforderungen

Ziel ist die Entwicklung einer Anwendung, die die folgenden Spezifikationen und Schnittstellen, unter Berücksichtigung der Rahmenbedingungen (siehe Abschnitt nicht-funktionale Anforderungen) implementiert:

Auswahl einer Wasserkörper-Route

Mithilfe eines Formulars wird vom Nutzer eine korrekte Eingabe erwartet, zu einem Fließgewässer-Wasserkörper der in der WRRL-Datenbank des LUNG enthalten ist. Zur Unterstützung wird ein PDF-Dokument der Website beigefügt, das eine vollständige Liste der Wasserkörper Bezeichnungen und der jeweiligen Abkürzung enthält.

Aufgrund der Tatsache, dass ein Wasserkörper mehrere Routen enthalten kann, wird nach der Benutzereingabe eine Tabelle erzeugt, die alle Routen auflistet zum gesuchten Wasserkörper.

Um sich einen Gesamtüberblick über alle Routen zu einem Wasserkörper zu verschaffen, wird in der Tabelle zu jeder Route ein Link automatisch generiert, der auf das jeweilige Diagramm verweist. Auf diese Weise kann sich der Nutzer in verschiedenen Browser Fenstern jede einzelne Route anzeigen lassen.

Erzeugen von Diagrammen

Ein Ergebnis der Web-Anwendung ist das dynamisch erzeugte Diagramm für Fließgewässer-Wasserkörper Routen. In der Grundform besitzt das Diagramm zwei Achsen. Auf der x-Achse wird die Stationierung abgetragen, die y-Achse enthält Güte- und Belastungsinformation.

Als Diagrammart wird ein Flächendiagramm genutzt, unter der Bedingung, die lückenlose Stationierung einer Route wiederzugeben.

Bearbeiten von Diagrammen

Im Mittelpunkt der Interaktion zwischen Anwendung und User steht die Auswahl verschiedener linear referenzierter Informationen die angezeigt bzw. auch wieder entfernt werden können über eine Check-Box, durch das Aktualisieren der Internetseite.

Diagrammoptionen sollen den Nutzer unterstützen, indem sie ihm eine schnelle und komfortable Möglichkeit bietet, die gewünschten Informationen zu entnehmen. Das sind zum einen eine Zoomfunktion auf der x-Achse, um nahe beieinanderliegende Gewässerabschnitte in einer geeigneten Auflösung darzustellen. Über Tooltips werden zusätzliche Informationen zum einzelnen Datensatz angezeigt, wie die Abschnittsnummer oder die exakte Stationierung.

Speichern und Importieren einer Attributauswahl

Um dem Nutzer, trotz einer wachsenden Liste an darstellbaren Parametern, eine effiziente und schnelle Methode bieten zu können, immer wiederkehrende Fragenstellungen in der Maßnahmenplanung von Fließgewässern zu beantworten, soll es eine Schnittstelle geben in der sowohl das Speichern, als auch Importieren von Konfigurationsfiles möglich ist.

In diesen Textdateien wird die aktuell dargestellte Attributauswahl gespeichert und kann zu einem beliebigen Zeitpunkt wieder importiert werden.

Exportieren von Diagrammen

Es soll die Möglichkeit bestehen, fertige Diagramme sowohl als im PDF-Format als auch im Raster-Format zu speichern.

Die wichtigsten Funktionen der zu entwickelnden webbasierten Anwendung werden im folgenden Anwendungsfalldiagramm zusammengefasst.

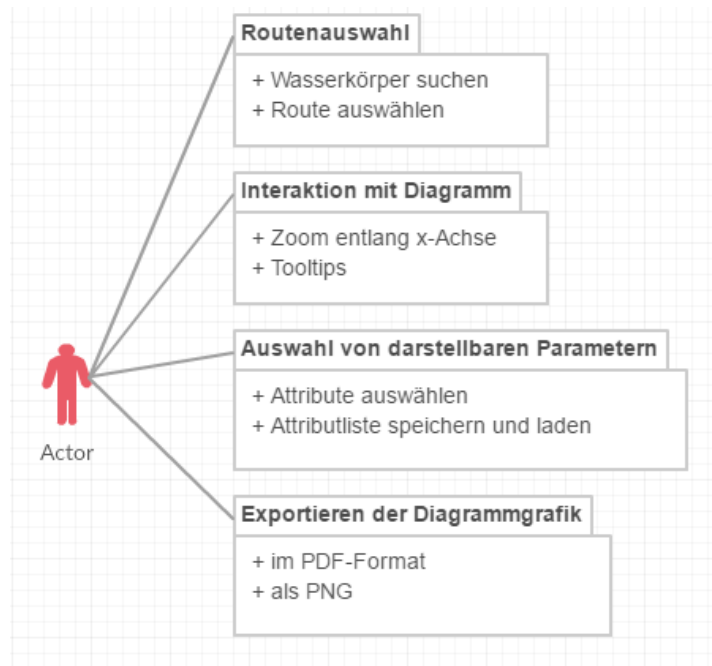


Abbildung 5 - Use-Case-Diagramm

3.2 Nicht funktionale Anforderungen

Technische Anforderungen

Die Anwendung soll ohne eine Benutzer-Authentifizierung und frei von zusätzlich zu installierender Software abrufbar sein und ohne proprietäre Komponenten auskommen.

Einzig ein JavaScript Plugin muss im Webbrowser aktiviert sein, als Voraussetzung für das Framework das die Diagramme erzeugt.

Auf Grund der Tatsache, dass es sich um eine webbasierte Anwendung handelt, stellt ein Webserver die Dateien für die Erzeugung der Website bereit und bearbeitet clientseitige Anfragen. Für ein einheitliches und modernes Website-Layout wird Bootstrap in der Version 3.0 verwendet.

Die Datenspeicherung aller gewässerbezogenen Daten erfolgt in einer Datenbank und eine Abfrage der Datensätze kann mithilfe einer serverseitigen Skriptsprache und dem SQL-Standard, als Schnittstelle zwischen Webserver und Datenbank, erfolgen.

Eine parametrisierbare URL wird benötigt, um einzelne Websites der Anwendung, wie die Routentabelle, oder die Präsentationsseite die das Diagramm beinhaltet, dynamisch aufzurufen. Im LUNG kann je nach Anwendungsfeld eine URL mit den vorhandenen Daten

zusammengesetzt werden und ohne eine direkte Benutzereingabe Diagramme mit fachspezifischen Attributen erzeugt und dargestellt werden.

Arbeitsumgebung

Die Anwendung soll vorrangig in Büros genutzt werden, die mit Desktop-PCs und Monitoren, Laptops oder Tablet PCs ausgestattet sind, daher wird die Verwendung auf einem Smartphone nicht berücksichtigt. Der Überblick über die Datenmenge würde eine zu große Herausforderung für die kleinen Displays bedeuten. Zudem ermöglichen Touch-Umgebungen keine Mouseover Funktionen zum Anzeigen von zusätzlichen Informationen im Diagramm.

Anwender

Alle Nutzer von Geofachdaten, im besonderen Mitarbeiter des LUNG im Dezernat der WRRL sind als Anwender vorgesehen. Sie bringen i.d.R. gute Kenntnisse im Umgang mit Datenbankbasierten Web-Anwendungen mit und durch ein sehr schlichtes und intuitives Userinterface ist die Bedienung der Anwendung leicht zu erlernen. Über einen Hilfe-Button in der Navigationsleiste der Website sind im speziellen die Interaktionen zwischen Nutzer und Diagramm erklärt.

Erweiterbarkeit

Die Attributtabelle zur Auswahl der darzustellenden Güte- und Belastungsinformationen soll um neue Parameter ergänzt werden können, indem klar definierte Schnittstellen in den Klassen und Methoden angepasst werden.

4 Webanwendung

„Eine Web-Anwendung ist ein Softwaresystem, das auf Spezifikationen des World Wide Web Consortium (W3C) beruht und Web-spezifische Ressourcen wie Inhalte und Dienste bereitstellt, die über eine Benutzerschnittstelle, den Web-Browser, verwendet werden.“⁴

4.1 Aufbau

Statische und dynamische Inhalte

Zunächst wird die prinzipielle Funktionsweise des Internets betrachtet: Das World Wide Web funktioniert gemäß dem Client-Server-Prinzip, wonach der Client dem Server eine Anfrage überstellt, welche dieser bearbeitet und dem Client eine Antwort übermittelt. Beim Anfordern einer HTML-Seite lädt der Server diese aus seinem Dateiverzeichnis und sendet sie an den Client, dieser zeigt die Seite schließlich im Browserfenster an. Solche Webdokumente bezeichnet man auch als statisch.

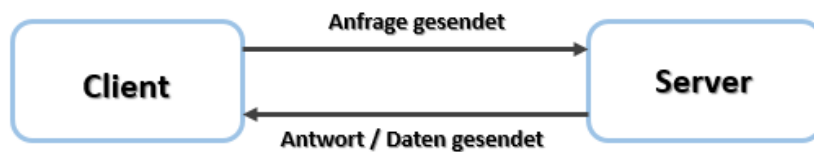


Abbildung 6 – statisches Client-Server Prinzip

Wünschenswert für eine Vielzahl von Anwendungen ist jedoch ein Mechanismus, der es erlaubt, den Inhalt der Webdokumente in Abhängigkeit von Benutzeranfragen zu modifizieren. Solche Webdokumente werden als dynamisch bezeichnet.

Damit der Server die Anfragen des Clients qualifiziert beantworten kann, ist eine serverseitige Dynamik erforderlich. Anfragen, die der Client dem Server stellt, werden von serverseitigen Programmen für den Client bedarfsgerecht aufbereitet.

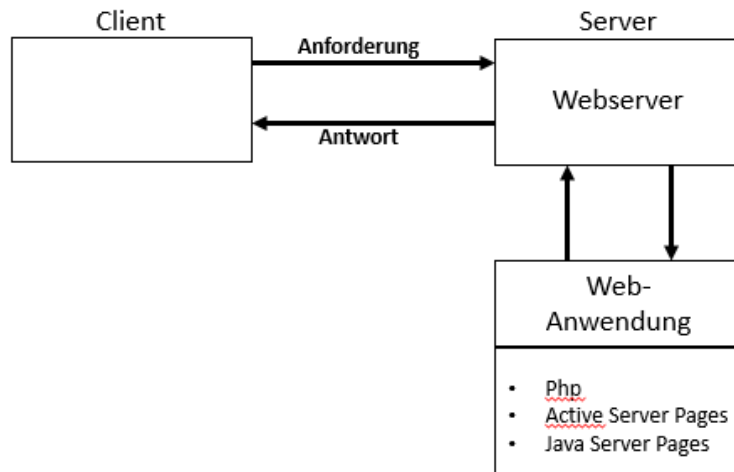


Abbildung 7 - dynamisches Client-Server Prinzip

Für die Datenbankbasierte Web-Anwendung ist den Anforderungen nach ein Webserver auszuwählen, der das dynamische Client-Server Prinzip unterstützt. Dazu wird im Folgenden die Funktionsweise und die wichtigsten Aufgaben eines Webservers aufgezeigt. Im Anschluss daran, werden mehrere Vertreter vorgestellt und einer ausgewählt der für den Systementwurf entsprechend konfiguriert wird.

Grundsätzlich ist unter einem Webserver ein Serverdienst zu verstehen, der das HTTP-Protokoll unterstützt. Der Server belegt typischerweise den tcp-Port 80 und beantwortet alle Anfragen die im HTTP-Protokoll formuliert sind. Darüber hinaus sollte er auch in der Lage sein, mehrere scheinbar gleichzeitig gestellte Anfragen zu bearbeiten und häufig angefragte Informationen zu puffern für einen schnelleren Zugriff. Zu den ergänzenden Aufgaben eines Webservers gehören das Verwalten und Prüfen von Berechtigungen, wie Datei- oder Verzeichnisrechte, die Reaktion auf Fehler, die Anpassbarkeit an unterschiedliche Sprachen, sowie eine Unterstützung von Sicherheitsprotokollen.

4.2 Auswahl eines Webservers

Zu den zwei wichtigsten Webserver- Anbietern gehören neben dem Apache HTTP Server, auch nginx. Diese werden zu verschiedenen Punkten miteinander verglichen und am Ende ein Fazit gezogen, welcher Server für diese Anwendung besser geeignet ist.

Der Apache HTTP Server ist laut w3techs ⁵ der am meisten eingesetzte Webserver, gehört zur Apache Software Foundation und ist sowohl quelloffen als auch kostenfrei. Die Ursprünge des Webserver reichen bis ins Jahr 1995 zurück, als der Code des National Center for Supercomputing Applications (NCSA) Webserver optimiert werden sollte. Grundsätzlich ist die Software mit allen gängigen UNIX und Windows Distributionen kompatibel. Durch die Apache Portable Runtime verfügt der Server ab der Version 2.0 über zahlreiche Multiprocessing-Module, welche die jeweiligen Stärken der Betriebssysteme hervorhebt. ⁶

Der große Vorteil von Apache, liegt in der Architektur. Das Modulkonzept erlaubt es vor dem Installieren der Software aus dem Quellcode heraus, dass verschiedene Module hinzugelinkt werden können. Zusätzlich sind weitere Module im Internet verfügbar ⁷.

Dynamische Websites können mithilfe verschiedener Skriptsprachen wie PHP oder Perl erstellt werden, indem die entsprechenden Module in den Konfigurationsdateien nachträglich aktiviert werden.

Zwei weitere Anforderungen an die Konfigurierbarkeit sind die Änderung des tcp-Portes auf dem der Webserver die eingehenden Anfragen annimmt und die Anpassung des Pfades für das Wurzelverzeichnis. Der Port ist in der Konfigurationsdatei httpd.conf anpassbar und die Pfadangabe des Rootverzeichnisses in der Datei 000-default. Über verschiedene htaccess-Dateien sind weitere individuelle Einstellungen möglich und der Webserver kann auf diese Weise entsprechend an seine Systemumgebung gut angepasst werden.

Die NGINX-Server-Software wurde 2004 vom russischen Softwareentwickler Igor Sysoev veröffentlicht und wird laut w3techs ⁴ von von 33,9% alle Websites weltweit genutzt. Angesichts der wachsenden Beliebtheit und Verbreitung des Webserver gründete Sysoev 2011 NGINX Inc. Das Unternehmen entwickelt die Software seitdem weiter und bietet zusätzlich zur kostenfreien Version mit NGINX Plus, auch kostenpflichtigen Support und Zusatzfeatures an.

Dieser Webserver hebt sich von Apache durch die ressourcenschonende Verarbeitung von einer hohen Zahl an Nutzeranfragen ab, mit einer gleichzeitig hohen Performance. Während Apache für jede Client-Anfrage einen neuen Thread oder Prozess öffnet, arbeitet der NGINX-Server ereignisorientiert. Somit kann er die Anfragen asynchron bearbeiten, was

Arbeitsspeicher und Zeit spart. Die Arbeitsweise des NGINX-Servers ist darauf ausgelegt statische HTML-Dokumente an den Client auszuliefern. Dynamische Anfragen werden über eine Proxy-Schnittstelle an die entsprechende Software weitergeleitet.

NGINX ist auf allen gängigen Windows, Linux und Mac OS Betriebssystemen installierbar und besitzt, wie der Apache Webserver, einen modularen Aufbau. Diese Module müssen allerdings bereits bei der Kompilierung explizit in das Hauptprogramm eingebettet werden. Somit ist dieser Server weniger für Anwendungen geeignet, die häufiger konfiguriert werden müssen.⁸

4.3 Fazit

Für diese Webanwendung ist für den Webserver das Erstellen von dynamischen Websites und einer Kommunikation mit einer Datenbank von Bedeutung. Es muss eine leichte Konfigurierbarkeit möglich sein, um mit geringen Aufwand auch zu einem späteren Zeitpunkt die Skriptsprache und den Austausch von Daten individuell einstellen zu können. Für den Einsatzfall, dass die Anwendung Stand-Alone in die Systemumgebung des LUNG integriert werden sollte, handelt es sich um eine reine Fachanwendung und somit sind die gleichzeitigen Anfragen, die der Webserver beantworten soll, so gering dass der Apache HTTP Webserver den Anforderungen genügt.

Sollte die Applikation für gewässerbezogene Daten in einem größeren Umfeld eingesetzt werden und die Last für den Apache zu hoch sein, ist auch eine Kombination aus NGINX und Apache möglich, wobei ersterer dann die statischen Webseiten Inhalte ausliefert und die dynamischen Anfragen, welche einen Datenbankzugriff erfordern, an den Apache weiterleitet.

5 Bootstrap

Bei der Entwicklung von Web-Anwendungen, ist stets auf die Trennung von Inhalt und Design zu achten. Während Content-Elemente wie Texte, Bilder oder Diagramme mit der Markup-Sprache HTML ausgezeichnet werden, sind grafische Gestaltungen in Form von Stylesheets festzulegen.

Bei der Wahl eines, auf modernen Webtechnologien basierendes CSS Front-End Framework, wurde Bootstrap ausgewählt, weil es gewisse Vorzüge gegenüber regulären HTML-Elementen für die Gestaltung von Websites aufweist.

Bootstrap wurde 2011 von Twitter als Open-Source Projekt veröffentlicht und wurde für eine zentrale Front-End Bibliothek der internen Analyse-und Verwaltungswerkzeuge des Twitter-Dienstes entwickelt. Die Weboberflächen wurde bis dahin mithilfe verschiedener Bibliotheken entwickelt, was zu einer geringen Skalierbarkeit, Inkonsistenz und einem hohen Wartungsaufwand führte.⁹ Diese Nachteile werden mit dem Einsatz eines einzigen Frameworks weitestgehend eliminiert und aus diesem Grund wurde sich auch bei der Entwicklung der Web-Anwendung, die Gegenstand dieser Abschlussarbeit ist, für Bootstrap entschieden.

Bootstrap war von Beginn an für die Unterstützung von HTML5 und CSS3 konzipiert. Basierend auf diesen Technologien bietet Bootstrap Hilfsmittel wie ein Grid-System zur Definition einer grundlegenden Struktur der Website an, sowie Vorlagen für die Gestaltung von Webseiten über eine einheitliche Typografie und Textformatierung. Mit den von Bootstrap bereitgestellten CSS-Stylesheets, erfolgt eine einheitliche Stildefinition für die wichtigsten HTML-Elemente, wie beispielsweise Tabellen, Formularelemente und Buttons.

Bootstrap stellt neben den regulären HTML-Elementen auch Schablonen für komplexere Layout-Komponenten bereit. Dazu zählen beispielsweise gruppierte Buttons, Navigationsleisten oder Buttons mit Dropdown-Menüs.

Die Grundfunktionalitäten des Frameworks können durch das Einbinden des zentralen CSS-Stylesheets in das entsprechende HTML-Dokument genutzt werden. Dadurch wird bereits die Stildefinition für reguläre HTML-Elemente, wie Buttons oder Überschriften-Tags, übernommen.

Aus der Anforderungsanalyse geht hervor, dass zusätzlich auch eine Tabelle, ein Eingabeformular, sowie eine Navigationsleiste benötigt werden.

Auf der Internetseite von Bootstrap ¹⁰ werden Layout-Vorlagen bereitgestellt und mit Code-Beispielen ergänzt, sodass bei der Entwicklung der Website zum Großteil auf den Inhalt geachtet werden kann und weniger auf das Design, da die vorgefertigten Elemente nur noch um den Content erweitert werden müssen.

6 Datenbanken

Zu Beginn dieses Kapitels werden, mithilfe einer geeigneten Literaturquelle, die Aufgaben und der Aufbau eines Datenbanksystems erläutert und im Anschluss die Anforderungen an dieses System aufgelistet. Anhand dieser Kriterien wird ein System ausgewählt, das die geforderten Merkmale erfüllt.

Im Buch „*Einführung Wirtschaftsinformatik*“¹¹ werden die grundlegenden Aufgaben eines Datenbanksystems aufgezeigt und aus welchen Elementen es sich zusammensetzt.

Den technischen Anforderungen nach, ist für das System ein objektrelationales Datenbanksystem zu verwenden, das unter einer Open-Source-Lizenz läuft. Die verwendete Datenbank muss zu den elementaren Eigenschaften, noch folgenden Kriterien gerecht werden:

- Unterstützt den SQL-Standard, zur standardisierten Abfrage der Daten
- Besitzt eine Erweiterung, um geometrische Datentypen zu speichern
- Verarbeitung paralleler Verbindungen

Es wird, resultierend aus den Anforderungen heraus, PostgreSQL als Datenbanksystem verwendet.

PostgreSQL ist ein frei verfügbares, objektrelationales Datenbankmanagementsystem und wurde als universitäres Projekt an der University of California at Berkeley entwickelt und läuft seit der Veröffentlichung unter einer freien Open-Source Lizenz¹².

Anfragen, die mithilfe von syntaktisch korrekten SQL-Statements an die Datenbank gestellt werden, werden bearbeitet und entsprechende Ergebnisse als neue Sicht zurückgeliefert. Mit PostGIS steht eine Erweiterung bereit, die in Kombination mit der relationalen Datenbank PostgreSQL eine Geodatenbank bildet. Diese wird für die Verarbeitung von linearreferenzierten Datensätzen nicht benötigt, da nur ein indirekter Raumbezug der Daten vorliegt. Nur wenn die Web-Anwendung im Kontext von GIS-Systemen eingesetzt wird, ist eine Speicherung von geometrischen Datentypen notwendig.

Für die Verwaltung von parallelen Verbindungen ist der sogenannte „postmaster“ zuständig, der einen neuen Prozess pro Verbindung startet.

7 Skriptsprache

Damit eine Website in Abhängigkeit von Benutzeranfragen erstellt werden kann, reicht reiner HTML-Code nicht mehr aus. Mit einer Programmiersprache wird serverseitig der Inhalt einer Website generiert und die zusammengesetzte Seite anschließend an den Client ausgeliefert.

Eine Skriptsprache erweitert den Webserver und ermöglicht den Zugriff vom Webclient über einen HTTP-Request auf eine Ressource des Webservers, die sich in einer Datenbank befindet. Die Kommunikation zwischen Server und Datenbanksystem erfolgt nicht über den Kern des Webservers, sondern über die Webservererweiterung.

7.1 Erweiterung des Webservers um eine Skriptsprache

In den Analyseabschnitten zu Webservern und Datenbanksystemen, wurde sich für den Apache HTTP Server und PostgreSQL entschieden. Daraus ergeben sich zwei Auswahlkriterien für die Skriptsprache: Es soll sowohl als Modul im Apache enthalten sein, als auch einen Datenbanktreiber für die Postgres Datenbank besitzen.

Die erste Möglichkeit den Apache Webserver zu erweitern, ist ein Common Gateway Interface einzusetzen. Hierfür bietet sich die Programmiersprache Perl an, mit der man über das Modul Database Interface (DBI) Datenbankfragen programmieren kann. Die einzelnen Funktionen und Treiber für den Datenbankzugriff liegen in der Schnittstelle gekapselt vor. Auf diese Weise muss im Perl-Programm lediglich das Datenbanksystem PostgreSQL angegeben werden und anschließend können die Funktionen z.B. zum Verbindungsauf- und Abbau vollständig genutzt werden.¹²

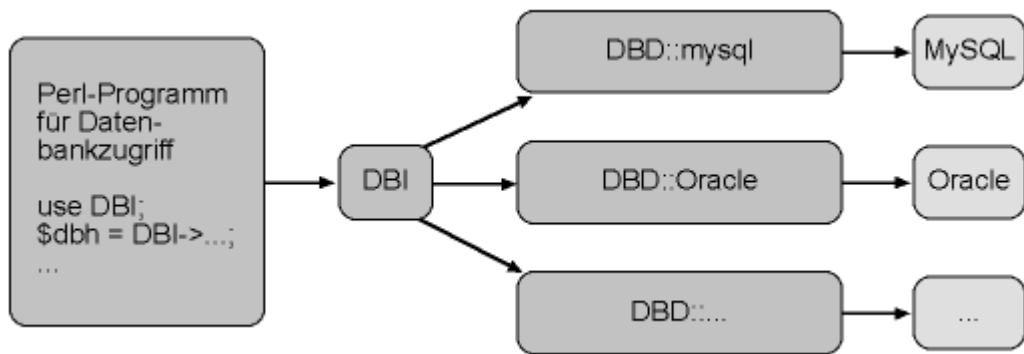


Abbildung 8 - Funktionsweise DBI ¹³

Auf Seiten des Webserver steht das Modul *mod_perl* bereit, für die Skriptsprache Perl. Zusätzlich müssen noch die Module *mod_cgi* und *mod_alias* aktiviert werden, damit Apache in der Lage ist, die von CGI-Skripten erzeugten Inhalte bereitzustellen.

Eine zweite Möglichkeit, den Webserver zu erweitern, bietet eine, in HTML-Dokumente, eingebettete Programmiersprache, wie PHP. Es ist ebenfalls als Modul im Apache vorhanden und mithilfe der Abstraktionsebene PHP Data Objects (PDO) stehen viele Funktionen zur Verfügung, um auf verschiedene Datenbanksysteme zuzugreifen, darunter auch PostgreSQL.

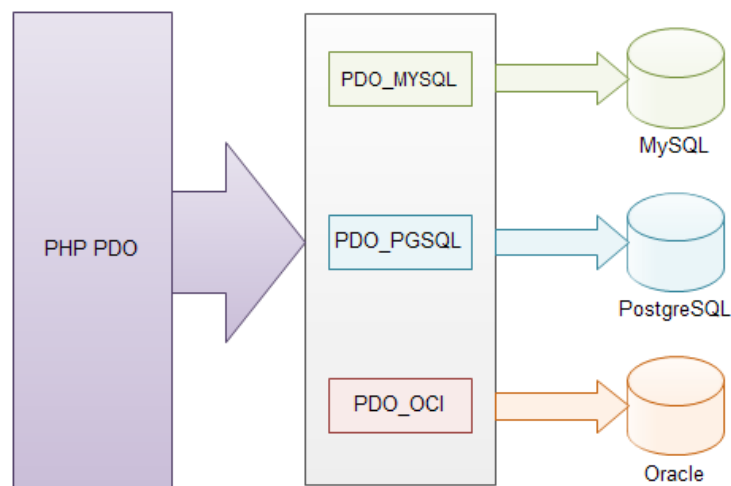


Abbildung 9 - Funktionsweise PDO ¹⁴

Obwohl beide Technologien die Auswahlkriterien erfüllen und mit den Schnittstellen DBI und PDO auch eine Unabhängigkeit gegenüber den Datenbanksystemen besitzen, hat das CGI einen entscheidenden Nachteil. Die mit Perl entwickelten Skripte führen den gesamten HTML-Code aus. Somit ist eine Trennung zwischen Inhalt, Layout und Programmlogik nicht möglich.

Anders verhalten sich eingebettete Skriptsprachen. Befinden sich in HTML-Dokumenten spezielle Auszeichnungen in der Programmiersprache PHP, so werden diese vom Interpreter erkannt und verarbeitet. Auf diese Weise kann mit HTML und CSS-Elementen das Design der Website erstellt werden und die Inhalte und die zugrundeliegende Programmlogik, werden dynamisch mit PHP-Code zur Laufzeit erzeugt. Daher wurde sich für die Skriptsprache PHP als Erweiterung des Webservers entschieden.

7.2 Konfiguration des Webservers

Wird der Apache auf einem Windows-basierten Betriebssystem in der XAMPP-Kombination installiert, werden im Vorfeld die zu installierenden Komponenten ausgewählt:

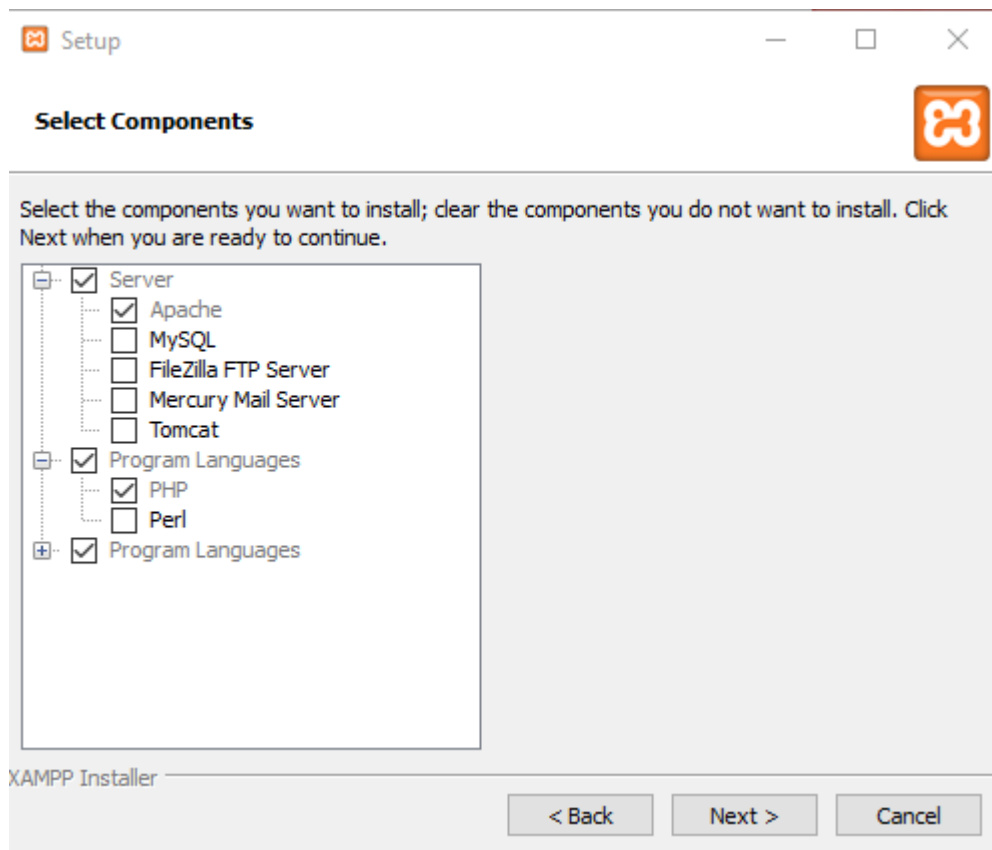


Abbildung 10 - Komponenten des Webservers

Für einen Funktionstest von PHP mit einem lokal installierten Webserver, wird der Programmcode aus Abbildung 11 in dieser Datei abgespeichert:

`\c\xampp\htdocs\phpinfo.php` .

```
1 <?php
2 phpinfo();
3 ?>
```

Abbildung 11 - Funktionstest PHP

Anschließend wird im Browser folgende URL mit der jeweiligen Server-Domain aufgerufen:

<http://Server-Domain/phpinfo.php>.

Damit der PHP-Interpreter über das PDO mit der Datenbank kommunizieren kann, müssen in der Konfigurationsdatei php.ini zwei Erweiterungen aktiviert werden:

- extension=php_pdo_pgsql.dll
- extension=php_pgsql.dll

8 Framework

8.1 Auswahlkriterien

In der Anforderungsanalyse wurden auch Aspekte der Visualisierung näher erläutert. Bei der Auswahl eines geeigneten Webframeworks stehen diese Anforderungen konkret im Fokus. So ist es wünschenswert, dass viele der Anforderungen schon im Funktionsumfang der Script-Bibliothek verankert sind.

All jene Anforderungen, die nicht unmittelbar aus dem Webframework hervorgehen, sollen aufgrund einer guten Dokumentation leicht integrierbar sein. Der Aufbau des Frameworks soll möglichst modular sein, sodass die selbst implementierten Funktionalitäten auch nach einem Update des Webframeworks nicht verloren gehen.

Es werden nur Frameworks untersucht die unter einer OpenSource Lizenz zu erwerben sind, was zwei Gründe hat: Zum einen entfallen die Kosten für den Erwerb einer kommerziellen Software. Zudem könnten Änderungen an kommerziellen Frameworks nur über den Hersteller erfolgen und die Folgekosten der Weiterentwicklung erhöhen. Über ein quelloffenes Softwareprojekt gibt es genau die Möglichkeit der Spezialisierung, sich an die Bedürfnisse des LUNG anzupassen.

8.2 Darstellungs- und Diagrammoptionen

Statische Optionen

- Kartesisches Koordinatensystem mit einer x und y-Achse
- Verschiedene Diagrammarten, wünschenswert ist ein Flächendiagramm
- Farbuweisung von einzelnen Datenreihen
- Exportfunktion der Diagramm-Grafik (z.B. für Berichte) im PDF und PNG Format

Dynamische Funktionen

- Nachladen und Entfernen von Attributen und anschließendes Aktualisieren des Graphen
- Dynamischer Startwert der x-Achse, da ein Routengenbeginn ungleich 0 sein kann
- Direkter Zugriff auf Stationierungswerte über Tooltips (exakten Wert und zusätzliche Informationen anzeigen)
- Hervorheben einer Datenreihe die mit der Maus überfahren wird (Mouseover)
- Interaktive Legende
- Zoomfunktion für Werte auf der x-Achse

8.3 Auswahl eines Frameworks

Im Folgenden wird eine Auswahl verschiedener Frameworks vorgestellt und auf die Anforderungen hin untersucht, die im Vorfeld in Bezug auf das Diagramm festgelegt wurden.

8.3.1 mit Grundfunktionalität

Diese Webframeworks beherrschen grundsätzlich das Erstellen von Diagrammen, allerdings müssen die meisten Zusatzfunktionen selbst eingepflegt werden, da diese Art von Webframeworks ein Komplettpaket zum Erstellen von ganzheitlichen Webapplikationen sind und die Visualisierung von Daten über Diagramme nur einen kleinen Anteil innerhalb der Frameworks einnehmen. Somit beinhalten diese Bibliotheken keine Zoommöglichkeit oder die direkte Interaktion mit einzelnen Messpunkten. Aus diesen Gründen werden nur einige Beispiele kurz erläutert.

Ample SDK

Das Ample SDK ist ein JavaScript User Interface Framework und bietet neben dem geforderten Flächendiagramm unter anderem auch Balkendiagramme, Tortendiagramme und Netzdiagramme. Eine Übersicht findet man auf der Beispielseite ¹⁵ der Ample SDK Homepage.

Das Webframework besitzt zwei interaktive Features. Zum einen hebt es Datenreihen beim Überfahren mit der Maus hervor, zum anderen gibt es die Möglichkeit, Datenreihen mit einem Click zu deaktivieren.

Ext JS Charting

Ext JS ist ein sehr umfangreiches JavaScript-Framework-Paket. Auch das Charting Modul von Ext JS hat viele unterschiedliche Diagrammartentypen im Funktionsumfang, so auch ein Flächendiagramm. Tooltips oder das hervorheben von Datensätzen sind integriert, Funktionen darüber hinaus sind selbst zu integrieren. Eine gute Dokumentation ist hier zu finden: ¹⁶

YUI Library

Das JavaScript und CSS Framework YUI von Yahoo versteht sich als Komplettpaket zur Erstellung von interaktiven Webapplikationen. Es bietet ebenfalls ein Flächendiagramm und besitzt wenige Funktionalitäten, wie z.B. das De- und Aktivieren von Datenreihen und das Anzeigen von Tooltips.

Eine Dokumentations- und Beispielseite ¹⁷ zeigt gut dokumentiert, wie man mit YUI Diagramme erstellen kann.

8.3.2 Frameworks mit erweiterter Funktionalität

Bibliotheken die sich auf Visualisierung von Infografiken spezialisiert haben, stehen im folgenden Abschnitt im Vordergrund, da sie eine große Auswahl an Features und Schnittstellen besitzen, die eine intensive Interaktion zwischen User und Graphen ermöglichen. Gerade aus diesem Grund sind in der späteren Beurteilung der Frameworks, hinsichtlich ihres modularen Aufbaus und der Funktionsweise, zu berücksichtigen.

jqPlot

Das Plugin jqPlot ist eine Visualisierungsbibliothek für das jQuery Webframework und unter der MIT-Lizenz frei erhältliches Open Source Projekt. Neben Linien-, Balken- und Tortendiagrammen gibt es ein Flächendiagramm, sowie weitere Darstellungsarten- und optionen, die auf der Demo-Seite des Projekts ¹⁸ eingesehen werden können. Die Diagrammausgabe wird mit HTML5 mittels Canvas-Objekt realisiert und die Werteübergabe erfolgt pro Datensatz Tupelweise in einem Array.

Dygraphs

Dygraphs wurde auf die zeitbasierte Darstellung von Daten mit hoher Dichte ausgerichtet, um den Benutzern zu ermöglichen, diese zu analysieren und zu interpretieren. Flächendiagramme, Tortendiagramme und andere Arten der Visualisierung werden nicht mit angeboten, nur reine X-Y-Graphen stehen im Mittelpunkt.

Daher ist Dygraphs zwar für die Darstellung von zeitbasierten Daten von Vorteil, es wird aber dem Anforderungskatalog für diese Projekt aber genau aus diesem Grund nicht gerecht, da die geforderten Diagramm Typen selber implementiert werden müssen bzw. über extra Funktionen bereit zu stellen sind.

HTML5 mit dem Canvas-Objekt ist auch in dieser Bibliothek die Technologie zur Ausgabe der Diagramme. Eine Galerie ¹⁹ zeigt einige Möglichkeiten des Frameworks, ein Testverzeichnis ²⁰ vervollständigt die Beispielauswahl. Die Dokumentation ist gut gegliedert in die Menüstruktur der Homepage eingebunden.

Dygraphs erwartet die anzuzeigenden Daten entweder als Array oder als CSV-Datei. Die Daten sind nicht als einzelnes Array pro Messkanal zu übergeben, sondern in einer gemeinsamen Array-Tabelle.

Highcharts

Highcharts ist eine JavaScript Bibliothek zum Erstellen von interaktiven Diagrammen. Es gibt eine große Auswahl an Diagrammtypen, auch das zu untersuchende Flächendiagramm wird auf einer Demo-Website angeboten ²¹. Im Gegensatz zu den bis hierhin vorgestellten Frameworks setzt Highcharts auf den SVG-Standard. Das bringt den Vorteil mit sich, dass die erzeugten Grafiken beliebig skalierbar sind und so in einem fluiden oder responsiven Layouts auf der ganzen Bildschirmbreite darstellbar sind. Die Datenübergabe erfolgt pro Datensatz, wahlweise als Array von Werten oder in Intervallform mit einer Schrittweite.

Alle Optionen die das Framework bietet sind in einer API sehr gut dokumentiert ²².

8.4 Beurteilung der Frameworks

Auf der Basis der vorangestellten Anforderungen, wird in diesem Abschnitt eine Übersicht erstellt, um zu vergleichen welches Framework dem Anforderungskatalog am ehesten gerecht wird. In der Tabelle sind alle erfüllten Anforderungen mit einem „x“ markiert, die nicht bereitgestellten Features werden mit „---“ gekennzeichnet.

8.4.1 Statische Diagrammoptionen

---	Area Chart	Farbzuweisung	Export in PDF oder PNG	Statische Legende
Ample SDK	x	--	--	x
Ext JS Charting	x	x	x	x
YUI Library	x	x	---	x
Jq Plot	x	x	x	x
Dygraphs	---	x	---	x
Highcharts	x	x	x	x

8.4.2 Dynamische Diagrammoptionen

---	Deaktivieren und aktivieren von Datenreihen	Zoomfunktion x-Achse	Tooltips	Hervorheben von Datensätzen
Ample SDK	x	---	---	---
Ext JS Charting	x	---	---	x
YUI Library	---	---	x	x
Jq Plot	---	x	---	x
Dygraphs	x	x	x	x
Highcharts	x	x	x	x

Aus dem Vergleich von Diagrammoptionen, wurde Highcharts als Framework-Bibliothek ausgewählt, da es die Anforderungen am besten erfüllt.

9 Datenanalyse

Um die vorhandenen gewässerbezogenen Daten für den Systementwurf nutzbar zu machen, ist es notwendig die Datensätze zu kategorisieren. Betrachtet werden die Kategorien Güte- Belastungs- und Sonstige Informationen und zusätzlich die Attribute zur Stationierung eines Bewertungsabschnittes.

Das digitale Gewässernetz MV (DLM25W)²³ führt das LUNG im FIS-Gewässer und umfasst u.a. alle Fließgewässer in M-V. Die "LAWA-Richtlinie zur Verschlüsselung von Gewässern"²⁴ vergibt bundeseinheitliche Gewässerschlüssel für Fließgewässer und Seen. Die Gewässerkennzahl besitzt maximal 10 Ziffern und hat folgenden Grundaufbau: Die erste Ziffer gibt das

Stromgebiet an zu dem das Einzugsgebiet gehört, jede weitere Ziffer unterteilt die Teileinzugsgebiete weiter, welche durch Wasserscheiden begrenzt werden.

„Mit dem Quellgebiet bei Ziffer 1 beginnend, werden bis zur Mündung grundsätzlich neun Teileinzugsgebiete festgelegt. Einzugsgebiete der Nebenfließgewässer erhalten die geraden Ziffern 2,4,6 und 8 und die Einzugsgebiete entlang des Hauptfließgewässers(Zwischengebiete) die ungeraden Ziffern 3,5,7 und 9.“²⁴

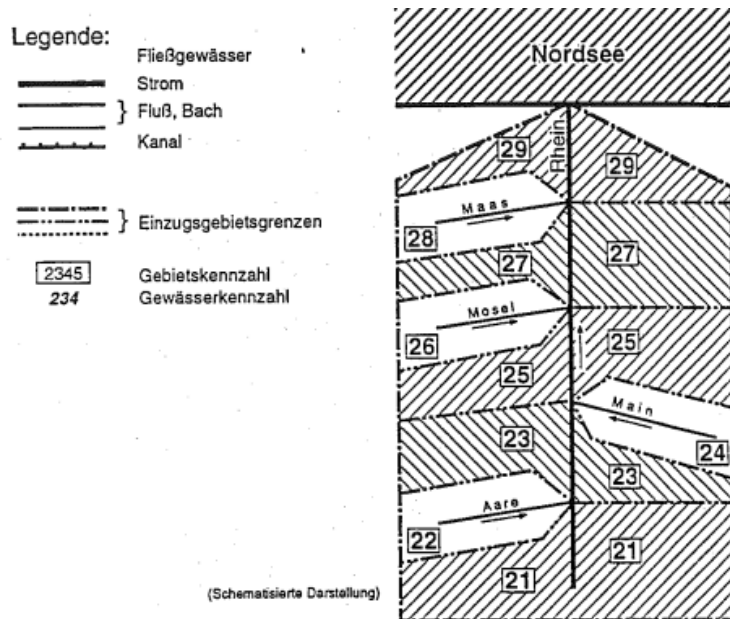


Abbildung 12 - Vergabe von Gewässerkennzahlen²⁴

Jede Gewässerroute besitzt eine eindeutige Gewässerkennzahl und ist einheitlich von der Mündung zur Quelle gerichtet. So entsteht ein lückenfreies Routensystem in M-V, dass in Verwaltungseinheiten, die sogenannten Wasserkörper, unterteilt ist.

Gewässerbezogene Informationen werden statt einer Positionsangabe in X-Y Koordinaten, mit linear referenzierten Daten erhoben. Diese beschreiben die räumliche Verortung von Daten bezüglich der relativen Position auf einer bereits bestehenden Linie. So wird zu jedem Beginn und Ende eines Bewertungsabschnittes mithilfe eines GPS-Empfängers der Standort ermittelt und daraus die relative Position auf der Gewässerroute abgeleitet. So entstehen lineare Features, also Punktpositionen entlang einer Linie. Mehrere Attributwerte können auf diese Weise einem Teil eines Features zugeordnet werden. In der folgenden Abbildung werden verschiedene Bewertungskriterien beispielhaft als Ereignisse einer linearen Referenzierung entlang einer Gewässerroute behandelt.

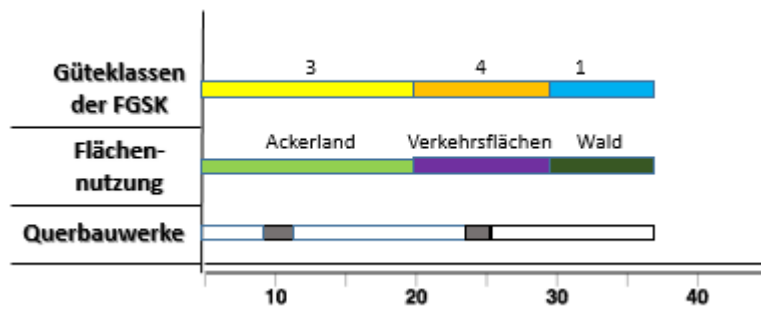


Abbildung 13 - Ereignisse einer linearen Referenzierung²⁵

Die Abbildung „Ereignisse einer linearen Referenzierung“ stellt drei Bewertungsabschnitte dar mit den Ereignis-Kategorien auf der y-Achse. Die einzelnen Bewertungsabschnitte auf einer Route besitzen die Attribute „stat_von“ für den Abschnittsbeginn und „stat_bis“ gleichbedeutend mit dem Abschnittsende, womit ihre Position auf der Gewässerroute angegeben wird. Durch das lückenlose Routennetz ist nur der Beginn eines Abschnittes von Bedeutung für das Abbilden der Stationierung.

Die Kartierabschnitte sind zwischen 50 m und 400 m lang und sollen einen in sich homogenen Bereich aufweisen, der sich nur in wenigen unmaßgeblichen Einzelparametern unterscheidet.³

Beginnend bei der Quelle eines Fließgewässers werden die Routen zur Mündung hin fortlaufend mit einer Abschnittsnummer versehen.

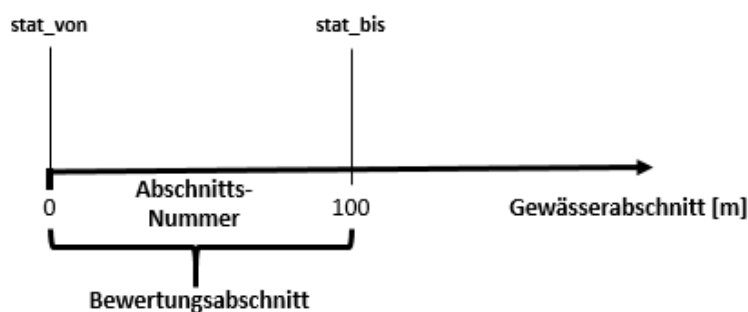


Abbildung 14 - Aufbau einer Gewässerroute

Die Gewässerkompartimente Land, Ufer und Sohle werden pro Abschnitt auf ihre bewertungsrelevanten Merkmale hin untersucht und entsprechende Zustandsklassen gebildet. Die so ermittelten Güteklassen stellen die Grundlage für die Güteinformationen im Längsschnitt einer Route dar.

Eine weitere Güteinformation bildet das biologische Monitoring. Es wird eingesetzt, um Zusammenhänge zwischen Belastungen im Gewässer und dem Zustand der biologischen Qualitätskomponenten zu erkennen. Die einzelnen Biokomponenten stellen Indikatoren für verschiedene Arten der Belastungen dar.

In einem Messnetz, das über M-V verteilt ist, werden rund 50% der Wasserkörper mittels biologischer Qualitätskomponenten untersucht und so könnten diese Messstellen ebenfalls als Güteinformation im Diagramm dargestellt werden. Allerdings sind in den aktuellen Datensätzen, die für den Entwurf des Systems herangezogen werden, keine linearreferenzierten Attribute vorhanden, sondern nur Rechts- und Hochwerte als Koordinaten zu einer Messstelle angegeben. Deshalb wird im Systementwurf auf eine Darstellung dieses Attributes abgesehen.

Am Gewässer vorkommende Bauwerke werden während einer Geländebegehung erfasst und die aktuelle Durchgängigkeit für Fische, Wasserwirbellose und den Fischotter eingeschätzt.

Über die Angabe zur Stationierung kann die Breite des Bauwerks berechnet werden und unter den Belastungsinformationen im Diagramm dargestellt werden. Eine farbliche Unterscheidung wird anhand der Durchgängigkeit vorgenommen.

Das Gewässerumfeld wird über die Flächennutzung als zusätzliche Information in den Längsschnitt integriert. Zu jedem Bewertungsabschnitt wird bis zu einer Entfernung von 50m links und rechts vom Gewässer die angrenzende Nutzung erfasst. Mögliche Nutzungsarten sind Acker –und Grünland, Laub –und Nadelwälder, Lagerflächen oder Verkehrsflächen.

Letztere Nutzungsart zählt zu den schädlichen Umfeld Strukturen und wird im Diagramm ähnlich wie die Querbauwerke zu den Belastungsinformationen gezählt, da befestigte Verkehrsanlagen nur eine geringe oder keine Versickerung des Grundwassers zulassen. Beeinträchtigungen des Gewässers werden auch durch Abtragungen, wie z.B. Kiesgruben, Müllablagerungen oder Hochwasserschutzbauwerken hervorgerufen.

Mit einem Entity-Relationship-Modell werden die im Vorfeld vorgestellten Attribute in Beziehung gesetzt und in ER-Diagrammen dargestellt.



Abbildung 15 - ER-Diagramm für die Beziehung Gewässerroute - Gewässerkennzahl

Das ER-Diagramm aus Abbildung 15 beschreibt folgende Beziehung: Eine Gewässerroute hat eine Gewässerkennzahl.



Abbildung 16 - ER-Diagramm für die Beziehung Wasserkörper und Gewässerroute

Der Fließgewässer-Wasserkörper kann sich aus einer oder mehrerer Gewässerrouten zusammensetzen und bildet einen ökologisch funktionsfähigen Raum, der eine sinnvoll zu bewirtschaftende Verwaltungseinheit darstellt, für die Maßnahmenpläne effektiver erstellt werden können als für einzelne Routen. ²⁶

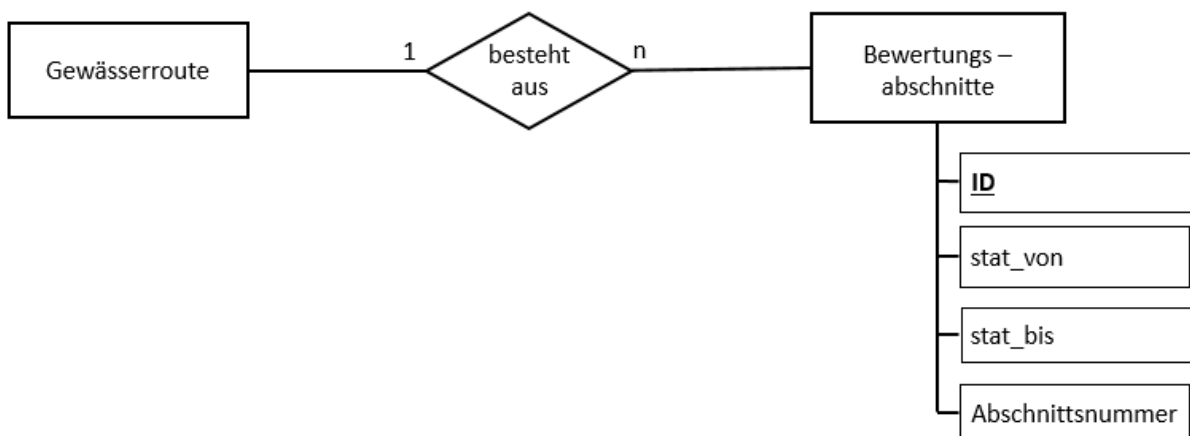


Abbildung 17 - ER-Diagramm für die Beziehung zwischen Gewässerroute und Bewertungsabschnitte

Eine Gewässerroute besteht aus mehreren Bewertungsabschnitten. Diese haben eine eindeutige Identifikationsnummer. Weiterhin besitzen die Bewertungsabschnitte drei Attribute, welche die relative Lage des Abschnittes auf der Gewässerroute beschreiben: den Abschnittsbeginn(stat_von), das Abschnittsende(stat_bis) und eine Abschnittsnummer.

Hinzu kommen für jeden Bewertungsabschnitt die spezifischen Attribute zur Beschreibung des ökologischen Zustandes und zu den Belastungsinformationen. Für die FGSK ist es die Gesamtbewertung, welche in Güteklassen 1-5 angegeben wird. Bei der Flächennutzung ist zwischen der linken und rechten Gewässerseite zu unterscheiden und daher sind zwei Attribute erforderlich, wovon aber nur jeweils eine Seite im Diagramm angezeigt werden kann. Bei den Querbauwerken ist es das Attribut Durchgängigkeit-Gesamt(DGK_gesamt) das unterschiedlich eingefärbt wird, je nachdem wie stark ein Bauwerk einen Einfluss auf die Umwelt ausübt.

10 Entwurf

10.1 Systemumgebung im LUNG

Im Dezernat der EG-Wasserrahmenrichtlinie werden verschiedene Fachinformationssysteme genutzt, um Angaben über Wasserkörper, Belastungen, Maßnahmen/Projekte, Querbauwerke, sowie qualifizierte Daten aus anderen Dezernaten der Abteilung Wasser, zu erhalten bzw. fortzuschreiben. Daraus ergibt sich ein breites Feld an möglichen Anwendungsgebieten für die Web-Applikation, die im folgenden Abschnitt erläutert werden.

Die Hauptanwendung bildet das Fachinformationssystem Wasserrahmenrichtlinie (FIS-WRRL). Zusätzlich können webbasiert Daten zur Maßnahmenplanung über das WebGIS-Framework kvwmap abgefragt werden unter der URL: <http://www.fis-wasser-mv.de/kvwmap/index.php?gast=35>

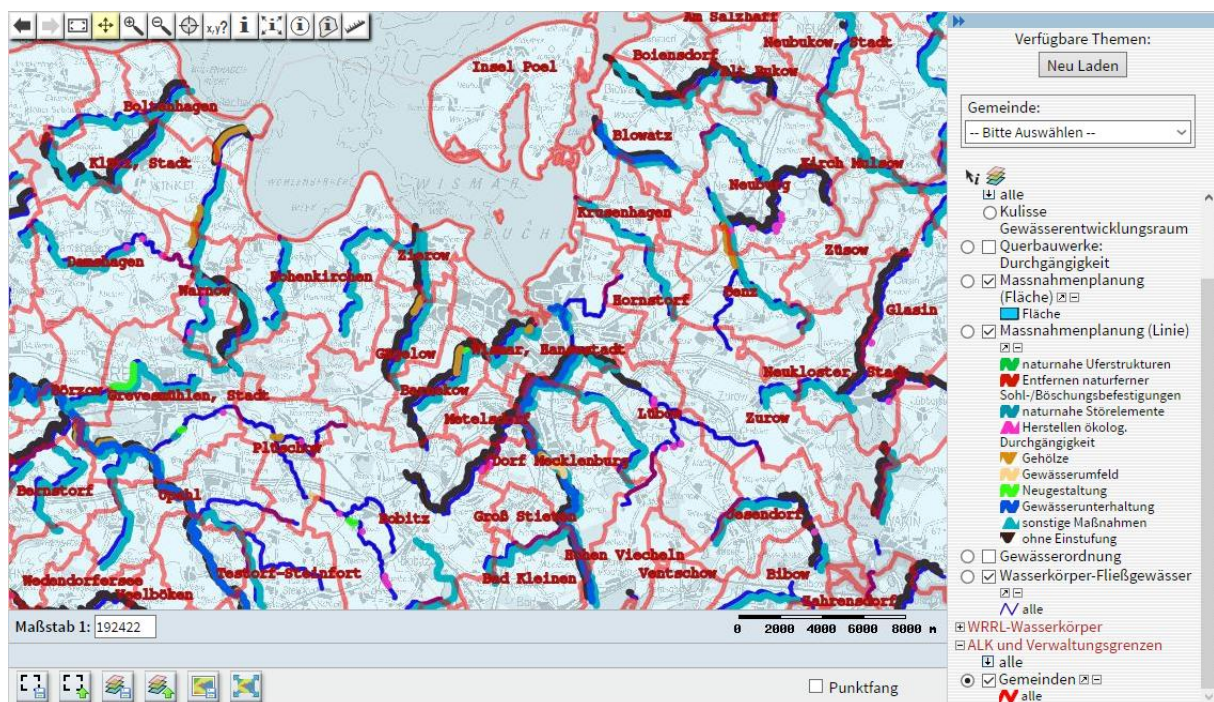


Abbildung 18 - WebGIS kvwmap(FIS-WRRL light) ²⁷

Die Anwendung zur Visualisierung von gewässerbezogenen linear referenzierten Daten wird über die Internetseite www.fis-wasser-mv.de aufrufbar sein. Als Ergebnis des Pflichtpraktikums im Vorfeld der Bachelorarbeit, ist bereits eine Prototypische Anwendung entstanden und kann als Fachanwendung über diesen Link gestartet werden: http://www.fis-wasser-mv.de/charts/formular_1.php.

Damit die Applikation für weitere Anwendungsfelder, wie eine Feature Info-Abfrage in Webdiensten oder eine Verknüpfung mit Wasserkörpersteckbriefen genutzt werden kann, muss sie über einen parametrisierbaren Aufruf gestartet werden können. Eine mögliche Umsetzung dieser Anforderung an das System ist im WebGIS FIS-WRRL light zu sehen:

FIS-WRRL	
Wasserkörper-Kürzel	EMES-2100
Wasserkörpername	Müritz-Elde-Wasserstraße
StALU	StALU WM
Gewässerordnung	1
WBV	2 # Untere Elde
Kreis	LLP
Gemeindeanteile	Dömitz(12%), Eldena(28%), Grabow(30%), Malk Göhren(13%), Malliß(6%), Neu Kaliß(10%)
Wasserkörper-Steckbrief	EMES-2100.pdf
FGSK-Längsschnitt	http://www.fis-wasser-mv.de/charts/formular_2.php?Suchfeld=EMES-2100&Submit=Suchen

Abbildung 19 – Feature Info Abfrage um Attribut FGSK-Längsschnitt erweitert ²⁷

In der Abbildung 19 wird deutlich das die Feature Info Abfrage für WRRL-Fließgewässer um das Attribut FGSK-Längsschnitt erweitert wurde und somit ist für jedes Fließgewässer, dass der Wasserrahmenrichtlinie unterliegt, ein dynamischer Link generiert worden mit dem Key-Value Parameter „Suchfeld=Wasserkörperkürzel“, der als Applikationseinstieg die Routentabelle aufruft.

The screenshot shows a web browser address bar with the URL: www.fis-wasser-mv.de/charts/formular_2.php?Suchfeld=EMES-2100&Submit=Suchen. Below the browser, there is a table with the following content:

	ID	Wasserkörper	Routenname
○	1	EMES-2100	Müritz-Elde-Wasserstraße

Below the table is a button labeled "absenden".

Abbildung 20 - Aufruf der Routentabelle über einen dynamisch erzeugten Link

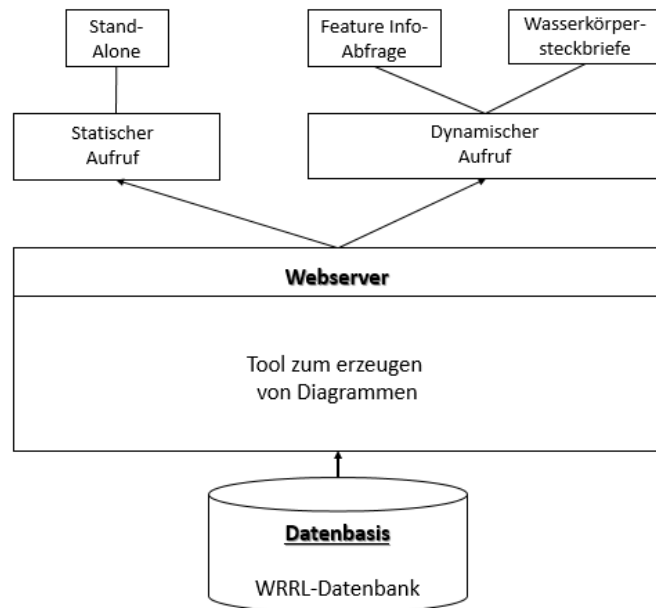


Abbildung 21 - Einsatzgebiete der Web-Anwendung

In der Abbildung 21 werden drei mögliche Anwendungsgebiete zusammengefasst, in denen der Einsatz des Systems möglich wäre. Es sind zwei verschiedene Aufrufmöglichkeiten der Anwendung vorgesehen: Über den statischen Zugriff auf das Tool zum Erzeugen von Diagrammen greift der Benutzer auf die Eingabemaske für Wasserkörper zu und kann die Anwendung ohne weiteren Content nutzen. Mit dem dynamischen Aufruf wird der URL der Wasserkörper als Parameter übergeben und somit kann die Anwendung innerhalb von Fachinformationssystemen oder Wasserkörpersteckbriefen genutzt werden, um die gewässerbezogenen Daten mithilfe der erzeugten Diagramme zu analysieren.

Das Gesamtsystem setzt sich aus einer Datenbasis, der WRRL-Datenbank, einem Webserver, in dessen Subsystem läuft die Anwendung, und einem Client, in dem die Daten präsentiert und abgerufen werden können, zusammen.

Der modulare Aufbau der Anwendung macht es möglich, dass die URL parametrisiert aufrufbar ist und auf diese Weise flexibel den Einsatzgebieten und dem vorhandenen Datenbestand angepasst werden kann. Die zugrundeliegenden Entwurfsentscheidungen werden im folgenden Abschnitt zunächst in einem Grobentwurf, dann in einem Feinentwurf detailliert beschrieben.

10.2 Architekturentwurf

Aufgrund der Tatsache, dass in der WRRD-Datenbank alle erforderlichen Datensätze für die Erzeugung der Diagramme linear referenziert vorliegen, wird der Architekturentwurf auf ein drei Schichten Modell reduziert, basierend auf dem relationalen Datenbanksystem PostgreSQL und die Nutzung der räumlichen Daten vernachlässigt.

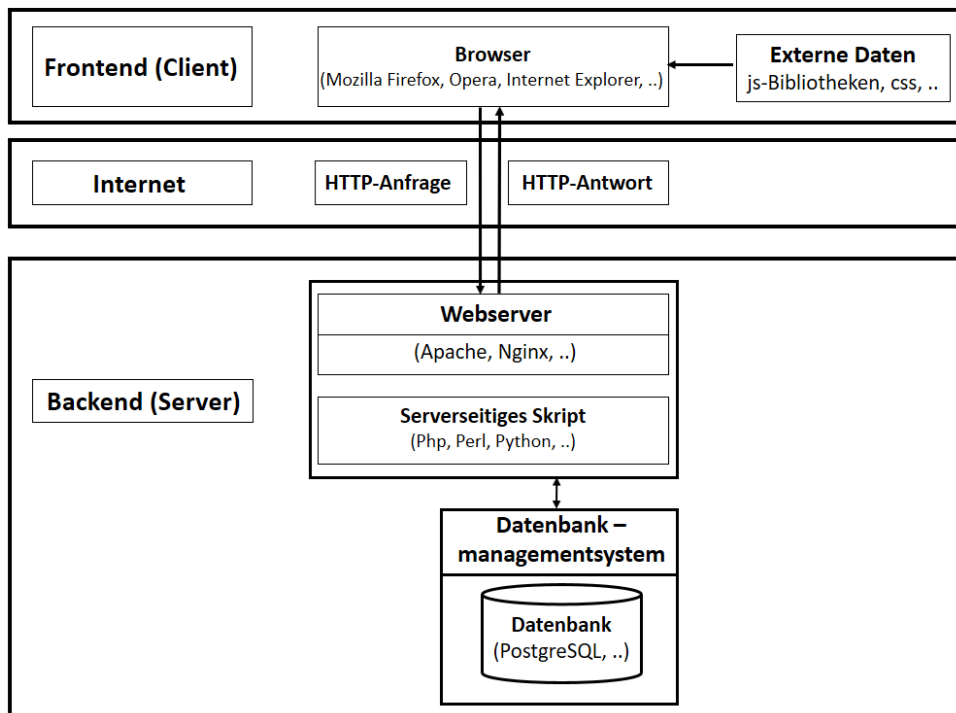


Abbildung 22 - Architekturentwurf

Der Architekturentwurf besteht aus einer Client-Server-Architektur. Das Frontend gibt über einen Webbrowser die grafische Benutzeroberfläche wieder und bindet in dieser Schicht auch externe Daten ein, wie das JavaScript-Framework für die Datenvisualisierung und das CSS-Framework Bootstrap, das für die Darstellung und Strukturierung der Websites verantwortlich ist. Die Kommunikation zwischen Client und Server erfolgt über das Internet und dem Übertragungsprotokoll HTTP sowie einem Webserver, der über eine serverseitige Skriptsprache, Anfragen an die Datenbank stellen kann. Die Datenquelle der Anwendung ist eine Datenbank, da an dieser Stelle die gewässerbezogenen und linear referenzierten Fachdaten gespeichert werden.

10.3 Komponentendiagramm

Aus der Client-Server Architektur werden im Folgenden Diagramm die Komponenten abgeleitet, damit sie dem im Vorfeld beschriebenen Architekturf Entwurf gerecht werden.

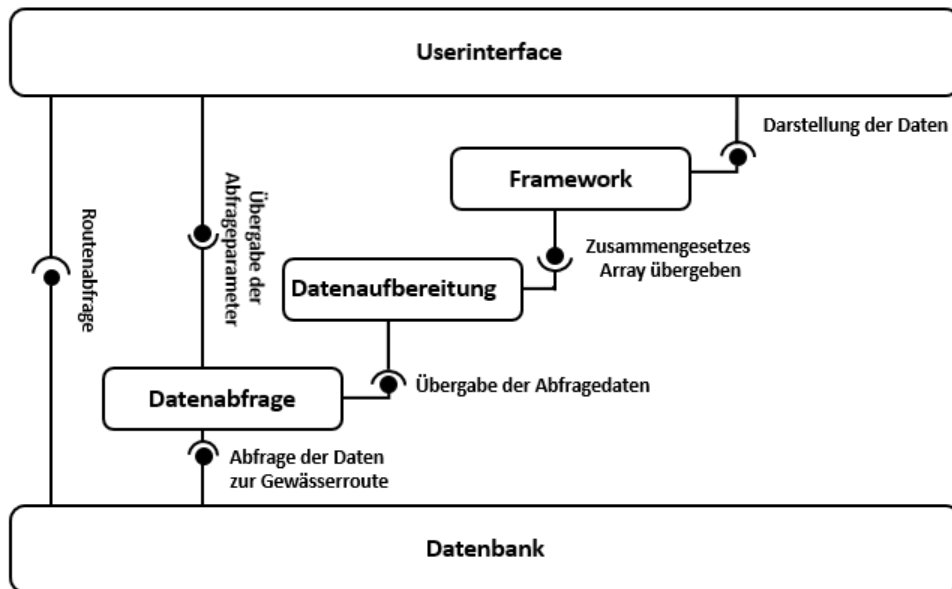


Abbildung 23 – Komponentendiagramm

Das Gesamtsystem besteht aus zwei Hauptschichten. Dem Userinterface, in Form eines Webbrowsers als Präsentations- und Interaktionsschicht und einer Datenbank, die als Datenquelle fungiert. Dazwischen befinden sich drei Verbindungsschichten, um die Nutzereingabe zu verarbeiten und die gewünschten Daten in einem Diagramm zu visualisieren.

Auf der Basis der funktionalen Anforderungen besteht das Userinterface aus drei Teilen:

Eingabemaske für den Wasserkörper

Abbildung 24 - Eingabemaske Wasserkörper

Einer **Tabelle**, um die Gewässerrouten zu dem Wasserkörper aufzulisten und einer Auswahl-Box um Attribute auszuwählen. Die Tabelle besitzt die Attribute ID, Wasserkörper, Routenname, Hyperlink

ID	Wasserkörperkürzel	Routenname	Hyperlink
1	WAOB-0800	Warnow	www.fis-wasser.de

Attributauswahl

Güteinformationen

Belastungsinformationen

Sonstiges

Diagramm zeichnen

Abbildung 25 - Userinterface Routentabelle

Der **Präsentations- und Interaktionswebsite**, in der das erzeugte Diagramm eingebunden wird und ebenfalls eine Attributauswahl vorgenommen werden kann. Im oberen Bereich der Seite wird eine Navigationsleiste zu sehen sein. Mit dem ‚Suche Route‘-Button gelangt der Benutzer zur Routentabelle des zuvor eingegebenen Wasserkörpers zurück. Der Button ‚Neuer Wasserkörper‘ ruft die Startseite der Anwendung auf, um nach einem neuen Wasserkörper zu suchen. Mit dem Button Hilfe wird eine separate Seite geöffnet, in der die Attributauswahl und allgemeine Funktionsweise der Anwendung beschrieben wird. Zusätzlich werden auf dieser Website die Buttons zum Speichern und Laden einer Attributliste angezeigt, sowie zum Exportieren der Diagrammgrafik. Eine Legende wird passend zum dargestellten Content unter dem Diagramm angezeigt.

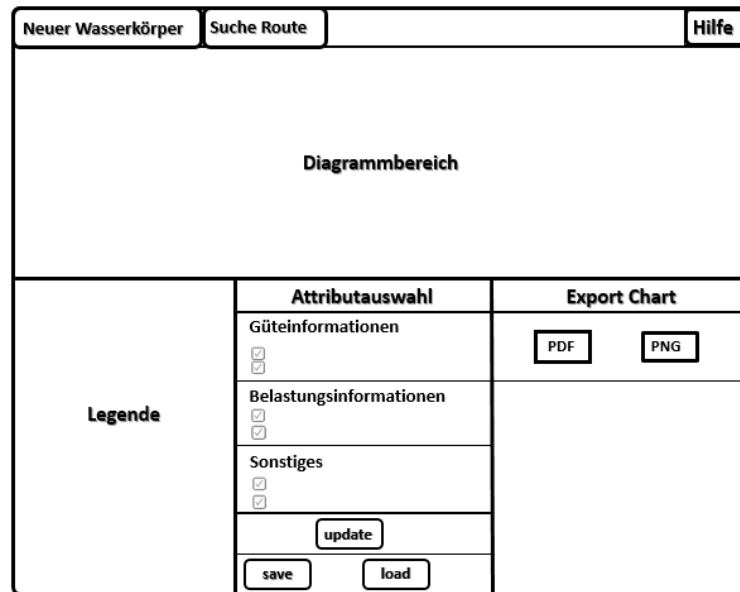


Abbildung 26 - Userinterface Diagrammpräsentation

10.4 Schnittstellenspezifikation

Routenabfrage

Die Abfrage der Gewässerrouten wird mithilfe eines SQL-Statements mit dem eingegebenen Wasserkörper als übergebenen Parameter realisiert.

Übergabe der Abfrageparameter

Nach dem Auswerten des Formulars werden der Datenbankabfrage drei Parameter übergeben:

- Wasserkörper
- Ausgewählte Route
- Ausgewählte Attribute

Abfrage der Daten zur Gewässerroute

Mit einem parametrisierten SQL-Statement werden die erforderlichen Daten zur Diagrammerzeugung aus einer Datenbank abgefragt.

Übergabe der Abfragedaten

Für jedes Attribut werden die abgefragten Daten auf jeweils einem Array gespeichert, die der Komponente Datenaufbereitung übergeben wird.

Zusammengesetztes Array übergeben

Das übergebene Array entspricht der Schnittstelle des Frameworks und enthält alle Diagrammwerte die dargestellt werden sollen.

Darstellung der Daten

Das generierte Diagramm wird als SVG Grafik in einem HTML-Container auf der Website aufgerufen.

10.5 Klassendiagramm

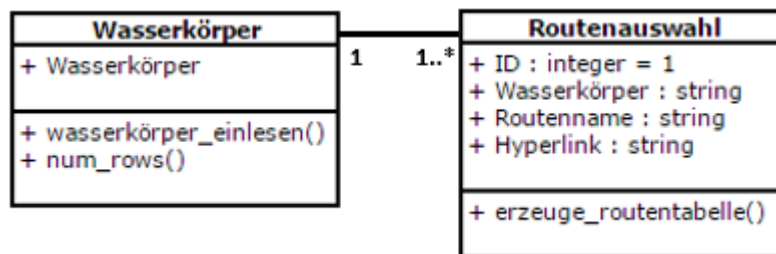


Abbildung 27 - Klassendiagramm Wasserkörper

Nach dem Aufruf der Startwebsite bietet die Klasse **Wasserkörper** über die Funktion `„wasserkörper_einlesen“` dem Nutzer die Möglichkeit eine Wasserkörperbezeichnung einzutragen.

Die folgende Klasse **Routenauswahl** wurde als 1: n Beziehung modelliert, da ein **Wasserkörper** sich aus mindestens einer oder mehrerer Gewässerrouten zusammensetzt.

Aus diesem Grund wird über die Funktion `„erzeuge_routentabelle()“` eine Tabelle mit den Attributen der Klasse **Routenauswahl** generiert.

Das Attribut `ID` besitzt den Startwert 1 und weist so jedem Routenamen eine eindeutige Nummer zu.

Der eingetragene **Wasserkörper** steht in der Tabelle unter dem gleichnamigen Attribut und in der Spalte `Routenname` werden alle Routen aufgelistet die zu einem **Wasserkörper** gehören

Über das Attribut `Hyperlink` ist zu jeder Route ein Link angegeben, womit der User direkt zum Diagrammfenster der jeweiligen Route gelangen kann. Dort ist statisch nur das Attribut `FGSK`

dargestellt, aber durch die eingebundene Klasse Attributauswahl können neue Attribute hinzugefügt und entfernt werden.

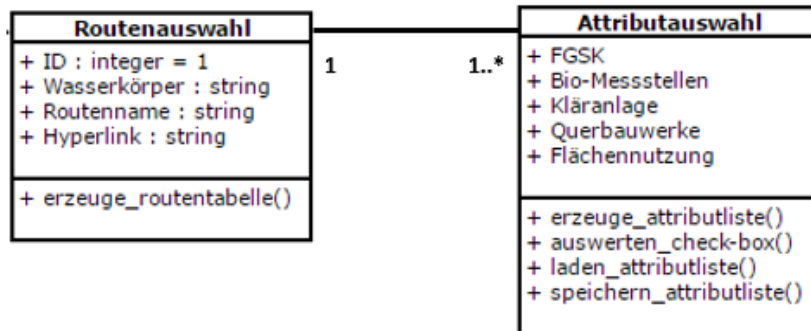


Abbildung 28 - Klassendiagramm Attributauswahl

Die Klasse Routenauswahl wird im nächsten Schritt mit einer 1: n Beziehung um eine Attributauswahl erweitert, da zu einer Route mindestens ein Attribut ausgewählt werden muss, dass im Diagramm dargestellt werden soll.

Mögliche Eigenschaften der Klasse Attributauswahl, die im aktuellen Entwurf berücksichtigt wurden, gehen aus Abbildung 28 hervor und wurden in ihrer fachlichen Bedeutung im Kapitel Datenanalyse näher erläutert.

Mit der Funktion *erzeuge_attributliste()* wird das Array mit den aufgeführten Attributen in einer Schleife durchlaufen und als Liste ausgegeben. Über die Check-Box kann der Benutzer so eine Auswahl an Attributen treffen die dargestellt werden soll. Um diese Auswahl auszuwerten gibt es die passende Funktion *auswerten_check-box()*, die nach dem abschicken des Formulars prüft welche Attributwerte ausgewählt wurden.

Sollte die Anwendung um weitere Attribute ergänzt werden, die darzustellen sind, gewährleisten die Funktionen zum Laden und speichern der Attributliste ein übersichtliches Arbeiten, da der User sich bereits bewährte Kombinationen von Attributen abspeichern und wieder einbinden kann.

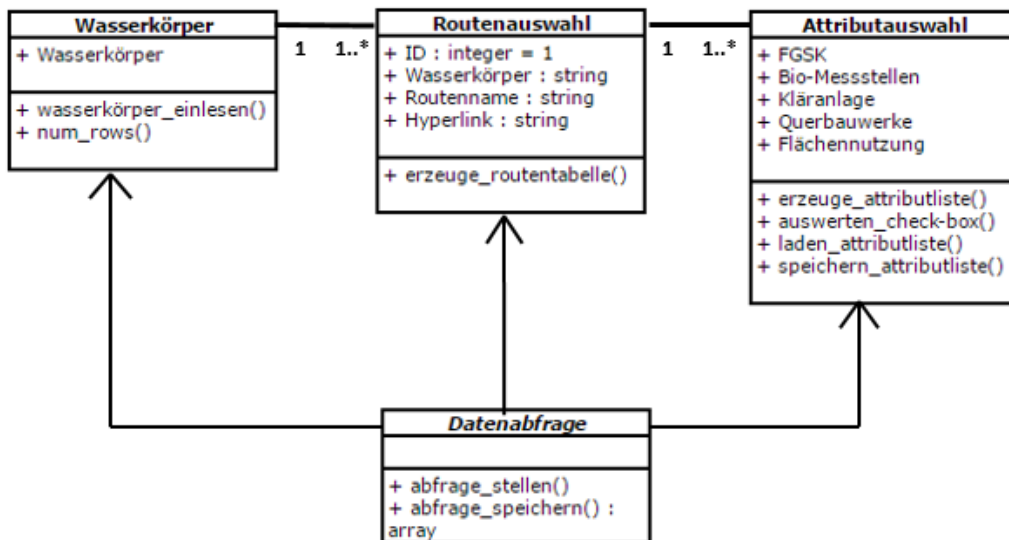


Abbildung 29 - Klasse Datenabfrage

In Abbildung 29 steht die abstrakte Klasse Datenabfrage im Mittelpunkt der Betrachtung. Zwar können in dieser Klasse keine Objekte erstellt werden, dafür existieren aber drei Assoziationen in Richtung der Klassen Wasserkörper, Routenauswahl und Attributauswahl. Dadurch kennt die Klasse Datenabfrage die Instanzen der aufgezählten Klassen und kann so parametrisierte SQL-Statements an die Datenbank schicken.

Es wird über die Funktion ‚*abfrage_stellen()*‘ pro ausgewähltes Attribut eine Abfrage gestellt. Das Ergebnis wird über die Funktion ‚*abfrage_speichern()*‘ jeweils auf einem Array gespeichert.

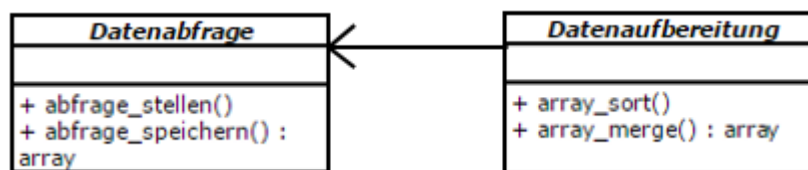


Abbildung 30 - Klassendiagramm Datenaufbereitung

Die Klasse Datenaufbereitung ist ebenfalls abstrakt und besitzt eine Assoziation in Richtung der abstrakten Klasse Datenabfrage. Von dieser Klasse bekommt die Funktion ‚*array_sort()*‘ die gespeicherten Abfrageergebnisse übergeben, um sie entsprechend eines vorgegebenen Musters zu sortieren.

Sind die Arrays in der richtigen Reihenfolge, werden sie mit der Funktion ‚*array_merge()*‘ zu einem Array verbunden.

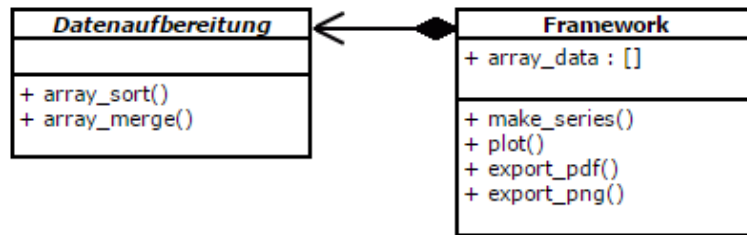


Abbildung 31 - Klassendiagramm Framework

Der Klasse Framework wird das verbundene Array übergeben und ist als Attribut `array_data` hinterlegt. Die Beziehung zwischen beiden Klassen wird als Komposition dargestellt, denn ohne die notwendigen Daten aus dem Array, kann das Framework kein Ergebnis in Form eines Diagramms zurückliefern.

Mit der Funktion `,make_series()'` wird jedes Arrayobjekt in eine Teilfläche umgewandelt, und mit `,plot()'` das Diagramm erzeugt. Innerhalb der Klasse werden ebenfalls die Schnittstellen zum Exportieren des Diagramms in den Formaten PDF und PNG implementiert.



Abbildung 32 - Klassendiagramm Diagrammfenster

Die Klasse Diagrammfenster besitzt eine Assoziation in Richtung der Klasse Framework und kann so auf die Instanz des Highcharts-Objektes zugreifen und auf diese Weise das erzeugte Diagramm über das Attribut `Diagramm_plot` einbinden.

In der Eigenschaft `Legende` sind alle auswählbaren Attribute aufgelistet und in ihrer Symbolik erklärt. Welcher Legendenbereich aber auf der Website dargestellt wird, ist von der Funktion `,legende_sichtbar()'` abhängig. Über eine Navigationsleiste kann der Benutzer u.a. zur Routenauswahl zurückkehren oder einen neuen Wasserkörper eingeben. Durch das einbinden der Instanzen der Klasse `Attributauswahl` über eine Assoziation wird das Auswählen von neuen

Attributen ermöglicht. Im Diagrammfenster werden über die Attribute Export-PDF und Export-PNG auch die Funktionalitäten zum exportieren des Diagramms eingebunden.

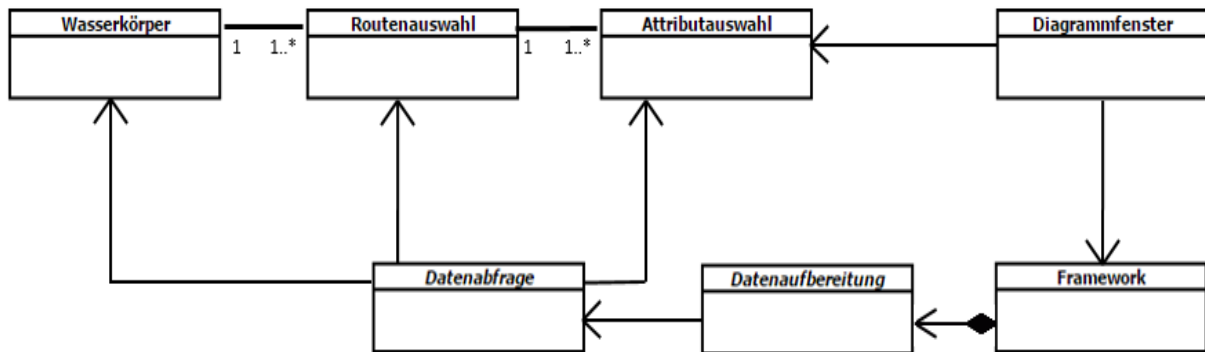


Abbildung 33 - Gesamtübersicht Klassendiagramm

Oben sind alle verwendeten Klassen in einer Gesamtübersicht zu sehen. Da für diese Gesamtübersicht Eigenschaften und Methoden der Klassen nicht relevant sind, wurden sie ausgeblendet. Das Klassendiagramm stellt also lediglich die Zusammenhänge zwischen allen Klassen dar, die für die Entwicklung der Webanwendung benötigt werden.

10.6 Aktivitätsdiagramm

Aktivitätsdiagramme werden verwendet, um den Kontrollfluss einzelner Aktivitäten innerhalb des Systems zu beschreiben.

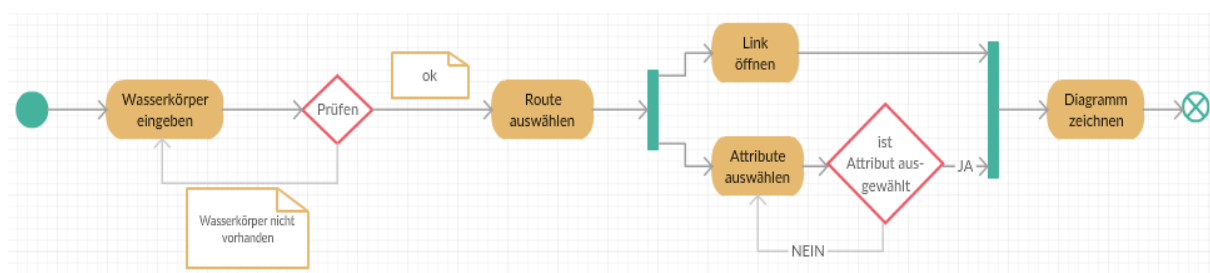


Abbildung 34 - Aktivität erster Diagrammaufruf

Die Abbildung 34 zeigt den ersten Aufruf des Diagramms. Mit dem Start der Anwendung wird der Nutzer aufgefordert in ein Suchfeld eine Wasserkörperbezeichnung einzugeben. Nach einer Prüfung in der Datenbank, ob diese Bezeichnung existiert, wird die Routentabelle erzeugt, in der eine Route ausgewählt werden kann. Ist der gesuchte Wasserkörper nicht in der Datenbank vorhanden wird eine erneute Eingabe erwartet.

Wurde eine Gewässerroute gewählt, verzweigt sich die Aktivität in zwei Threads. Zum einen lässt sich das Diagramm direkt über den Hyperlink öffnen, zum anderen können an dieser Stelle bereits Attribute ausgewählt werden die im Diagramm gezeichnet werden sollen. In diesem Thread muss anschließend aber eine Prüfung stattfinden, ob Attribute ausgewählt wurden.

Im nächsten Schritt läuft die Aktivität in einem Thread weiter, denn in beiden Fällen wird das Diagramm erzeugt, womit das Ablaufende erreicht wurde.

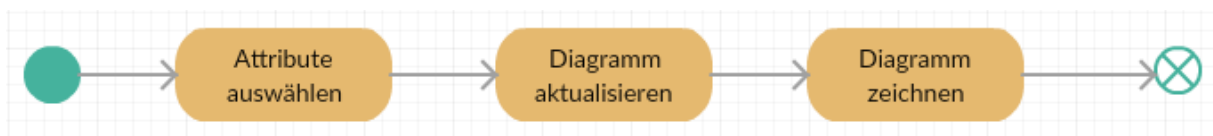


Abbildung 35 - Interaktion im Diagrammfenster

Von der Diagrammseite aus, eröffnet sich dem Benutzer die Möglichkeit eine neue Attributauswahl vorzunehmen. Nach dem Aktualisieren der Website wird das Diagramm mit der neuen Auswahl gezeichnet.

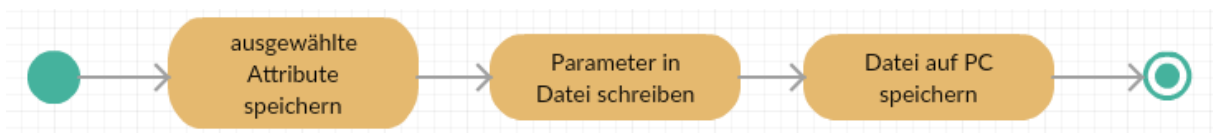


Abbildung 36 - Speichern einer Attributliste

Die Aktivität zum Speichern einer Attributliste beginnt mit dem auswählen von Attributen. Danach wird, durch das Drücken eines Buttons im Userinterface, die Aktion zum Speichern von Attributen ausgelöst. Danach werden die Parameter mit einer Funktion in eine Datei geschrieben und im Anschluss auf dem PC des Nutzers gespeichert.

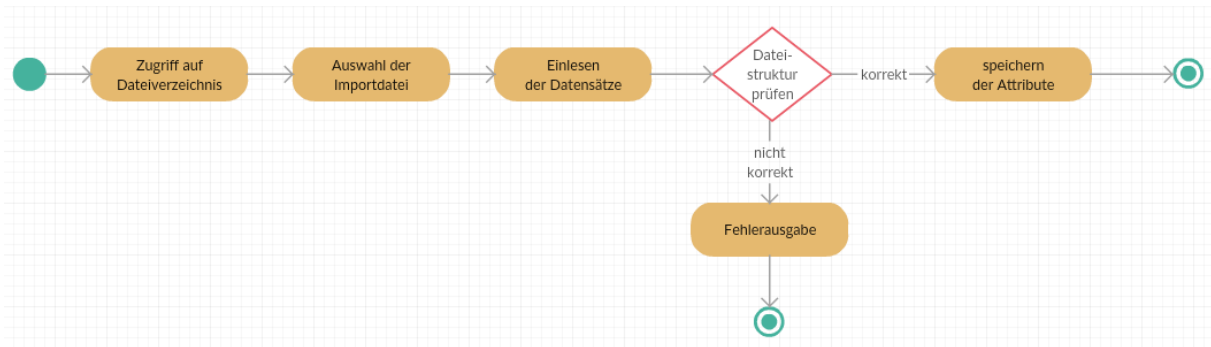


Abbildung 37 - Einlesen von neuen Attributen

Die Aktivität zum ‚laden einer Attributliste‘ wird durch das Drücken eines Buttons im Userinterface ausgelöst. Als erstes muss auf das Dateiverzeichnis des Benutzers zugegriffen werden, damit dort eine Datei ausgewählt werden kann, welche die neue Attributauswahl enthält. Danach wird die Datei im Lesemodus geöffnet und die zeilenweise eingelesenen Datensätze auf ungültige Zeichen geprüft. Ist die Datenstruktur fehlerhaft, wird eine Ausnahmebehandlung ausgelöst und die Aktivität beendet. Enthalten die Parameter nur gültige Zeichen werden sie auf einem Array gespeichert und diese Aktivität ist beendet.

10.7 Datenflussdiagramme

Neben den Aktivitätsdiagrammen, die im Kapitel Anforderungsanalyse die Kontrollflüsse im System beschrieben haben, wird im folgenden Abschnitt die Funktionalitäten und Aktivitäten durch Datenflüsse zu beschreiben und die Datenspeicher im System aufgezeigt. In der UML bietet sich hierfür das Datenflussdiagramm an.

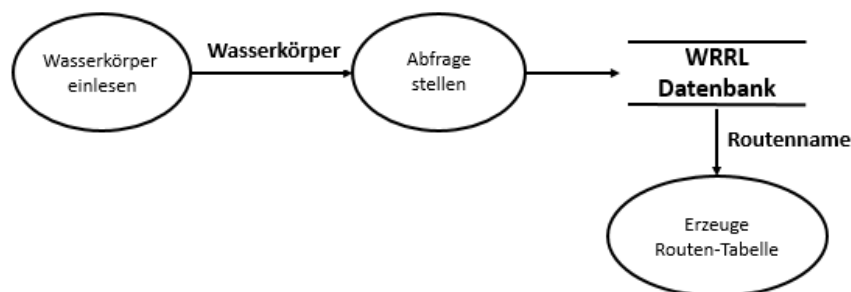


Abbildung 38 - Datenflussdiagramm für das Erzeugen einer Routentabelle

Als erstes wird der Datenfluss für das Erzeugen der Routentabelle dargestellt. Am Anfang wird mit der Funktion ‚Wasserkörper einlesen‘ ein Datensatz im System aufgenommen und im Anschluss wird die Wasserkörperbezeichnung der Funktion ‚Abfrage stellen‘ übergeben, die mithilfe des Parameters Wasserkörper, eine entsprechende Abfrage an den Datenspeicher ‚WRRL-Datenbank‘ stellt. An dieser Stelle werden alle zum Wasserkörper gehörenden Routennamen ermittelt und der Funktion ‚Erzeuge Routen-Tabelle‘ übergeben, wo sie zusammen mit den Attributen ID, Wasserkörper und Hyperlink in einer Routentabelle dargestellt werden.

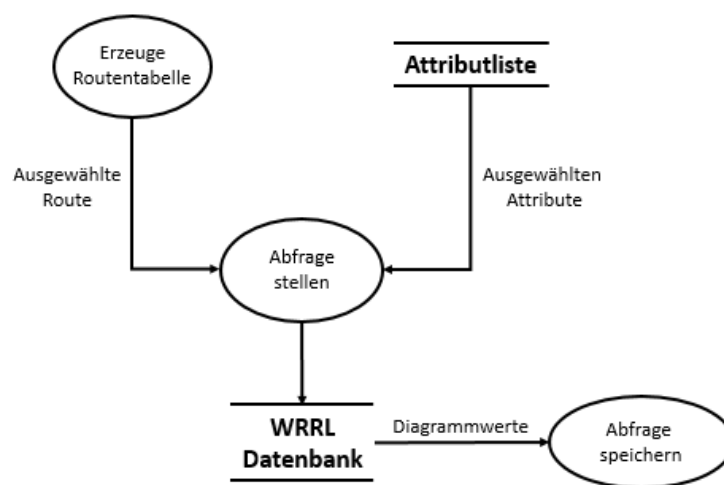


Abbildung 39 - Speichern von Diagrammwerten

Die Funktion ‚Abfrage stellen‘ benötigt für die Transformation der Daten, drei Datenströme. Zum einen die ausgewählte Gewässerroute mit dem zugehörigen Wasserkörper, zum anderen die darzustellenden Attribute. Letztere sind im Datenspeicher Attributliste vorgehalten und werden als Array übergeben.

In der Funktion ‚Abfrage stellen‘ werden die übergebenen Parameter für eine Datenbankabfrage gebraucht, die an den Datenspeicher WRRL-Datenbank gerichtet ist und die konkreten Diagrammwerte ermittelt. Diese werden der Funktion ‚Abfrage speichern‘ übergeben. Dort werden die Daten entsprechend der ausgewählten Attribute auf verschiedenen Arrays abgespeichert.



Abbildung 40 - sequenzieller Aufruf von Funktionen zur Datenaufbereitung

Im Anschluss an die Datenspeicherung auf Arrays, beginnt die Datenaufbereitung, mit dem Ziel ein einzelnes Array zurückzuliefern, dass die im Zuge der Datenselektion ausgewählten Datensätze darstellt.

Abschließend wird der gesamte Datenfluss im System in einem Diagramm übersichtlich dargestellt.

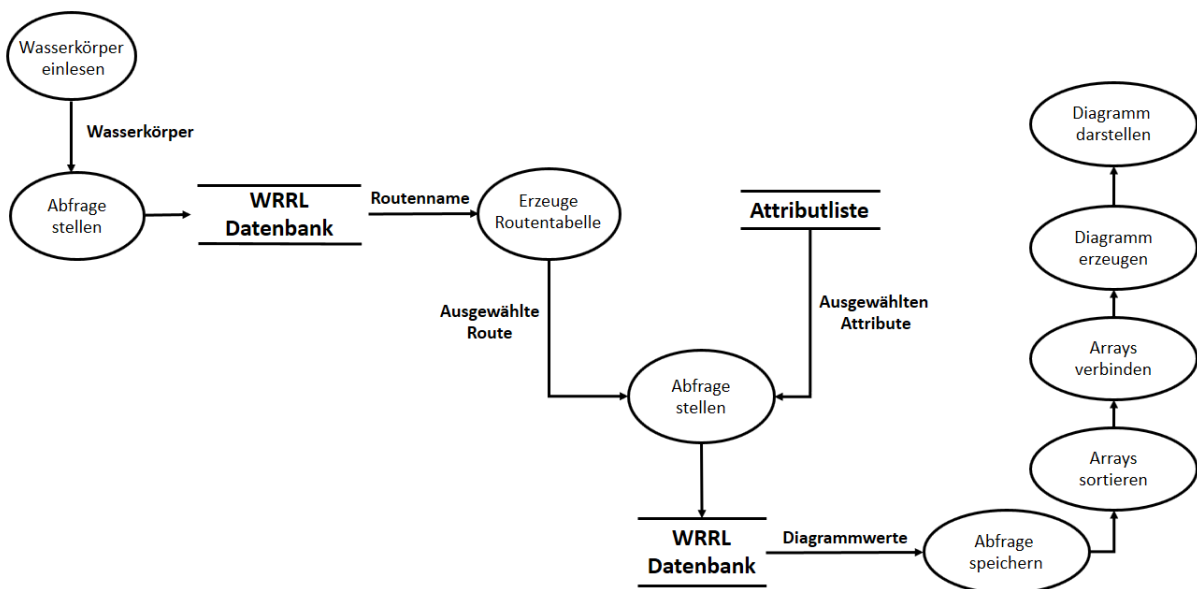


Abbildung 41 - Datenflussmodell der Webanwendung

In der Gesamtdarstellung wird deutlich, dass bis zum Prozess ‚Abfrage speichern‘ eine Selektion der gewässerbezogenen Daten stattfindet und diese gefilterten Datensätze mit den folgenden Prozessen für die Visualisierung aufbereitet werden.

10.8 Feinentwurf

Der detaillierte Entwurf beschreibt die im Klassendiagramm verwendeten Methoden näher und zeigt die Zusammenhänge zwischen einzelnen Programmteilen auf.

10.8.1 Datenselektion und Speicherung

Die Funktion ‚Wasserkörper einlesen‘ besteht aus einem HTML-Formular und enthält ein Eingabefeld von Typ „text“. Es fragt ein Wasserkörperkürzel vom Nutzer ab und prüft nach dem absenden des Formulars die Eingabe. Liefert eine Testabfrage über die Funktion `num_rows()` in der WRRL-Datenbank mindestens eine Gewässerroute zurück, wird der Inhalt mit der Formular-Übertragungsart GET der Klasse Routenauswahl zur Verfügung gestellt.

Besitzt das Abfrageergebnis keinen Datensatz wird eine Fehlermeldung ausgegeben, dass der gesuchte Wasserkörper nicht existiert, da sich jeder Wasserkörper aus mindestens einer Gewässerroute zusammensetzt.

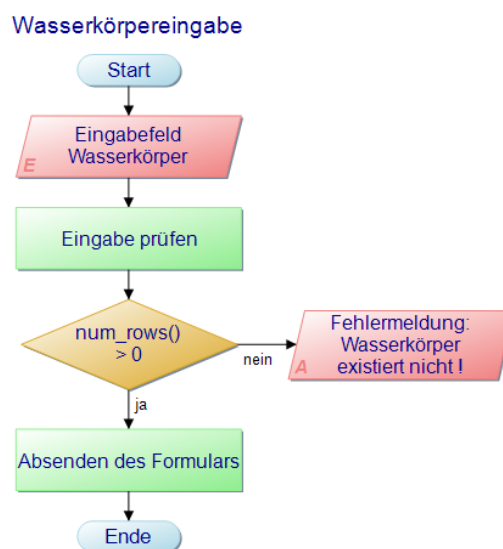


Abbildung 42 - Programmablaufplan zur Wasserkörpereingabe

Mit der PHP Funktion `$_GET[]` wird auf das HTML-Eingabefeld der Klasse Wasserkörper zugegriffen und der Wert des Feldes auf der Instanzvariablen Wasserkörper abgespeichert.

Die Funktion `erzeuge_Routentabelle()` führt ein Formular aus, indem ein SQL- Statement an die Datenbank gesendet wird, um die Gewässerrouten zum Wasserkörper zu ermitteln. Damit die Anwendung dynamisch aufgerufen werden kann, wird in der Abfrage die Variable, auf dem der Name des Wasserkörpers gespeichert ist, aufgerufen.

```
1 SELECT distinct table_wk.wasserkoeper,  
2     ersetze_zeichen(table_gn.gewaessername, ' ', '+', 'g') AS route_neu  
3  
4 FROM table_wk, table_gn.gewaessername  
5  
6 WHERE entferne_zeichen(table_wk.gewaesserkennzahl::text, '0'::text)::numeric = table_gn.lawa_gewaesserkennzahl_kurz  
7  
8 and fgsksktemp.wk_k = 'Wasserkörper'  
9
```

Abbildung 43 - SQL- Statement für Gewässerrouten

Die Datenbankabfrage nutzt zwei Tabellen, um einmal die Wasserkörperbezeichnung in Kurzform auszugeben und als zweites Attribut den oder die Gewässernamen. Da jeder Bewertungsabschnitt einem Datensatz entspricht, werden mit dem Ausdruck *distinct* die doppelten Einträge herausgefiltert. In Zeile 2 werden die Leerzeichen innerhalb der Gewässernamen durch ein + ersetzt, damit die URL korrekt kodiert werden kann, um im Webbrowser ausgeführt zu werden. Dort sind nach der RFC3986 [<https://tools.ietf.org/html/rfc3986>] keine Leerzeichen erlaubt.

In der Where-Klausel werden zwei Bedingungen definiert. Als erstes wird ein gemeinsames Attribut in beiden Tabellen gewählt, wofür in `table_wk` beim Attribut `gewaesserkennzahl` die Nullen von rechts aus abgeschnitten werden müssen. In der Tabelle `table_gn` enthält dann die Spalte `lawa_gewaesserkennzahl_kurz` zu jedem Datensatz dieselbe Gewässerkennzahl wie `table_wk`. Der zweiten Bedingung wird die Variable `Wasserkörper` übergeben.

Als Beispielabfrage würde die Datenbank für den Wasserkörper ‚WAOB-0800‘ folgendes Ergebnis liefern:

	wk_k character varying(9)	route_neu text
1	WAOB-0800	Floßgraben
2	WAOB-0800	Streitgraben
3	WAOB-0800	Warnow

Abbildung 44 - Abfrageergebnis Routentabelle

Zu Beginn der Klasse Routenauswahl werden die Instanzvariablen ID, Routenname, Wasserkörper und Hyperlink angelegt.

Das Abfrageergebnis aus der Datenbank wird auf einem assoziativen Array gespeichert und mit `$_row` in jedem Schleifendurchlauf auf das nächste Array-Element zugegriffen. Auf diese Weise wird der Tabelleninhalt spaltenweise dynamisch erzeugt. Die Statische Tabellenstruktur wird mit HTML-Code erzeugt. Die Variable Hyperlink enthält einen String mit der URL die auf die Diagrammseite der jeweiligen Gewässerroute verweist.

Dieser Link setzt sich zusammen aus der Domain des Webservers, dem Dateipfad zur Klasse Routenauswahl gefolgt von vier übergebenen Parametern. Das sind ein Hilfsattribut zur Formularauswertung, der Wasserkörper, der Routenname und der Submit-Parameter zum Abschicken der Formulardaten. Damit die URL parametrisiert angegeben werden kann, wird für die Parameter Wasserkörper und Routenname jeweils mithilfe von PHP-Code das Attribut aus dem Abfrageergebnis eingesetzt. Auf diese Weise ist die Anwendung dynamisch aufrufbar.

Die gewässerbezogenen Daten werden in einem Formular mithilfe der Klasse Attributliste weiter selektiert. Die Attributliste ist in einem Array vorgehalten, dass in einer while-Schleife durchlaufen wird. In jedem Schleifendurchlauf wird eine Checkbox erzeugt vom Typ Array und nimmt den Wert des jeweiligen Array-Objektes an.

Funktion erzeuge_Attributliste

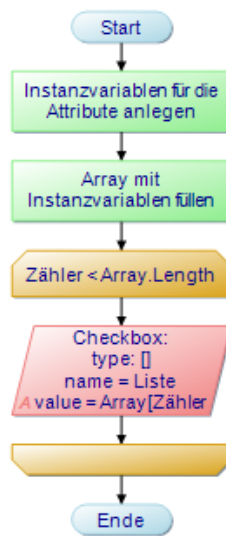


Abbildung 45 - Programmablaufplan erzeuge_Attributliste

Zum Auswerten der Checkbox wird mit einer If-Bedingung abgefragt, welche Array-Elemente markiert wurden und im Anschluss werden mit einer foreach Schleife, alle ausgewählten Attribute in einem neuen Array gespeichert und dieses nach dem Abschicken des Formulars zur Kontrolle ausgegeben.

auswerten_Checkbox

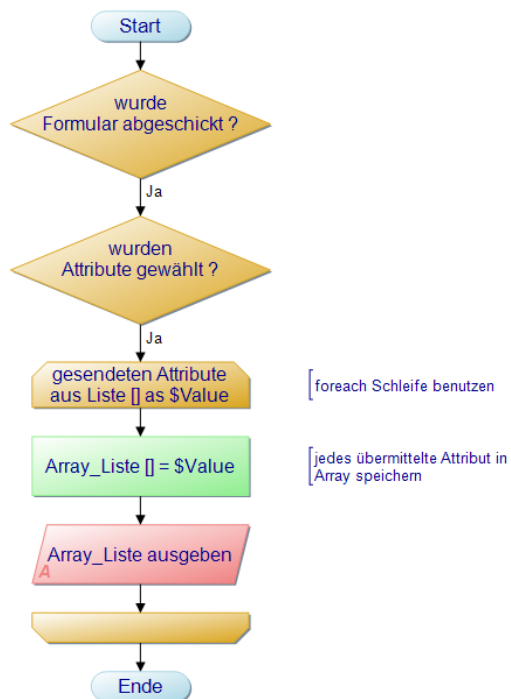


Abbildung 46 - Programmablaufplan auswerten_Checkbox

Eine Attributauswahl kann auf der Präsentations- und Interaktionsseite mit dem „save“-Button gespeichert werden. Durch das Drücken des Buttons wird die aktuell getroffene Auswahl an Attributen übernommen und die Funktion *auswerten der Checkbox* aufgerufen. Als Output kommt ein Array heraus, das nur die ausgewählten Attribute enthält. Diese werden dem Benutzer in einem extra Fenster angezeigt und abgefragt, ob die ausgewählten Attribute in eine txt.-Datei geschrieben werden sollen. Möchte der Anwender die Auswahl nochmals korrigieren, wird die Aktion zum Speichern von Attributen abgebrochen und man gelangt wieder auf die normale Diagrammseite.

Ist die Auswahl korrekt wird eine Datei im angegebenen Zielverzeichnis angelegt und mit Schreib- und Leserechten ausgestattet. In diese Datei wird der Inhalt des erzeugten Arrays aus der Funktion *auswerten der Checkbox* geschrieben. Anschließend wird die Datei geschlossen und der Hinweis ausgegeben, dass das Speichern von Attributen erfolgreich war.

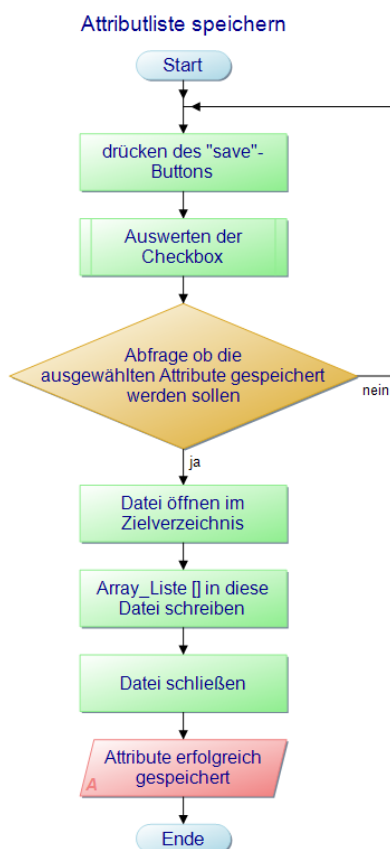


Abbildung 47 - Programmablaufplan Attributliste speichern

Eine Attributauswahl kann auf der Präsentations- und Interaktionsseite mit dem „load“-Button geladen werden. Durch diese Aktion öffnet sich ein Dialogfenster, indem der User eine txt.-Datei auswählen kann, die Attribute enthält. Im Anschluss an die Dateiauswahl, wird auf die Daten im Lesemodus mithilfe der php-Funktion fopen() zugegriffen und mit einer foreach-Schleife und fgets() zeilenweise ein Datensatz eingelesen.

Dieser wird einer Prüffunktion übergeben, die pro Schleifendurchlauf jeden Datensatz zuerst auf ungültige Zeichen prüft und bei einer korrekten Datenstruktur den Wert in einem Array abspeichert. Ist der Datenzeiger am Ende der Datei angekommen, wird der Datenstrom wieder beendet und die Datei mit fclose() geschlossen.

Das erzeugte Array wird anschließend von der Funktion Abfrage stellen() aufgerufen.

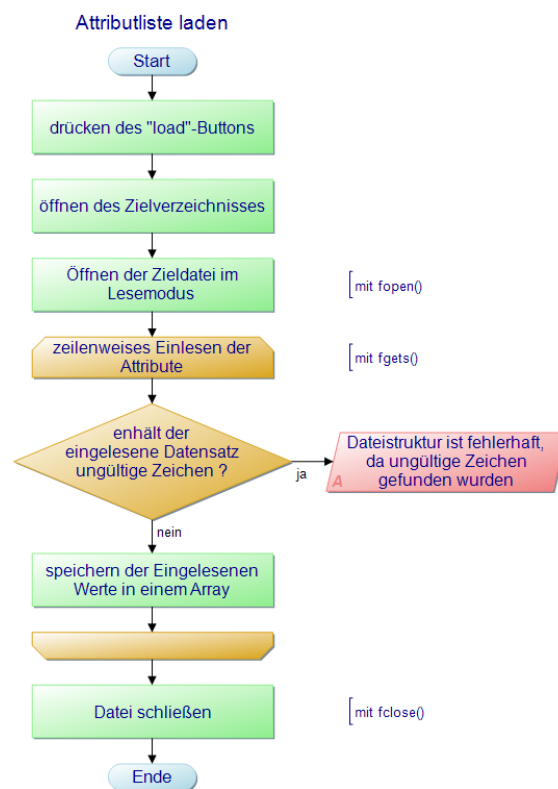


Abbildung 48 - Programmablaufplan Attributliste laden

Entspricht sowohl die ausgewählte Gewässerroute, als auch die getroffene Attributauswahl den Vorstellungen des Benutzers, werden die Daten an die Klasse Datenabfrage geschickt. Mit der Funktion abfrage_stellen, werden zu den ausgewählten Attributen die passenden SQL-Statements ausgeführt. Die Zuordnung von Attribut und SQL-Statement erfolgt über eine If-Bedingung und pro Schleifendurchlauf wird ein Array-Element geprüft, ob es einem

vorgegebenen Attributnamen entspricht. War die Prüfung erfolgreich, wird das SQL-Statement an die Datenbank gesendet.

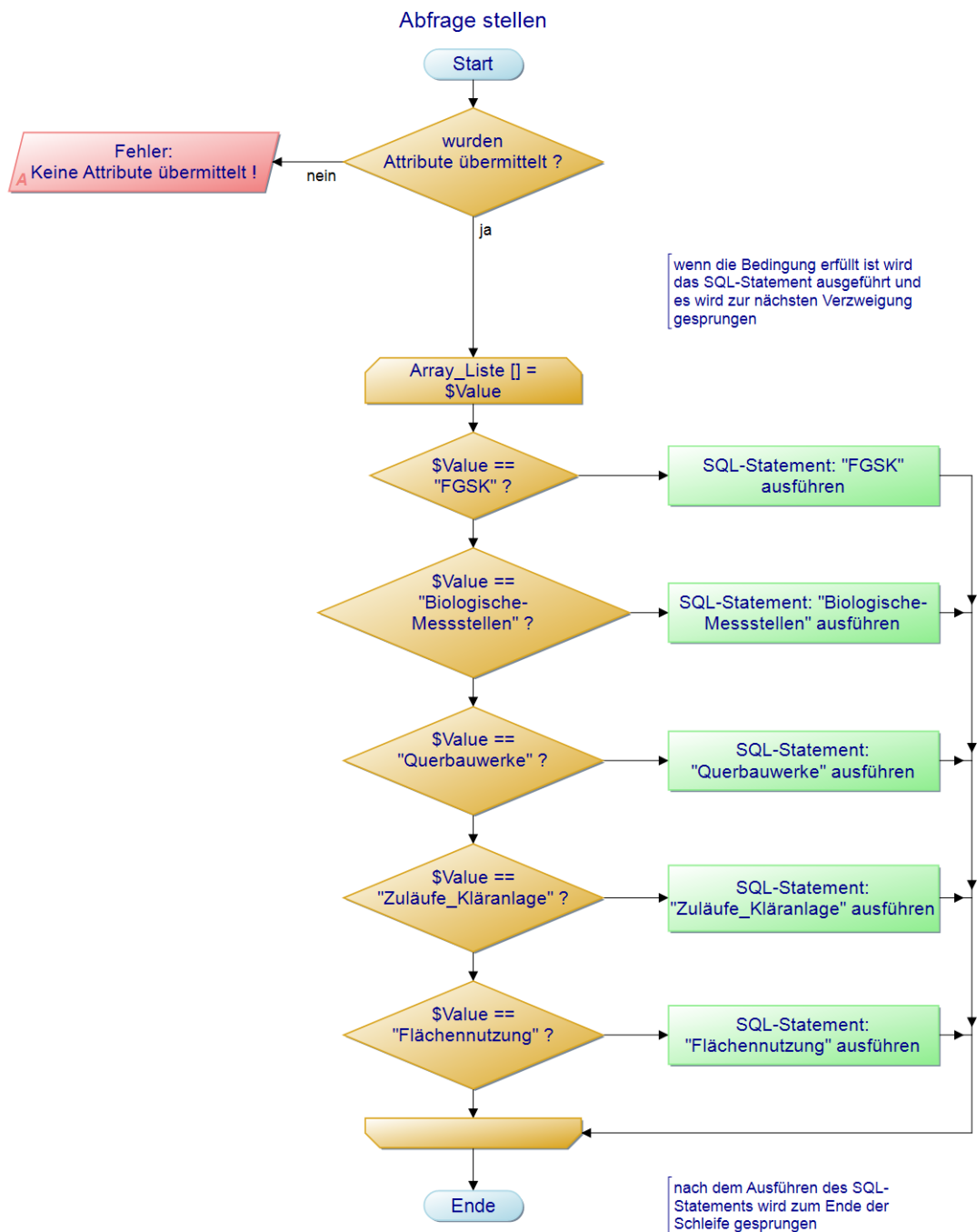


Abbildung 49 - Programmablaufplan Abfrage stellen

10.8.2 Datenabfrage

Die Daten für das Attribut FGSK, setzen sich aus Stationierungs- und Bewertungsrelevanten Daten zusammen. Für die Stationierung werden die Namen des Wasserkörpers und der ausgewählten Route zur Laufzeit als Variable in der Where-Bedingung übergeben. Da sich das Gewässernetz in M-V aus lückenlosen Bewertungsabschnitten zusammensetzt, reicht das Attribut stat_von aus, um alle Abschnitte auf der x-Achse abzutragen. Die Y-Achse enthält die Bewertungsparameter für jeden x-Wert und zusätzlich wird ein Farbwert übergeben. Realisiert werden diese Bedingungen in dem SQL-Statement über eine Fallunterscheidung mittels der CASE - WHEN – THEN- Klausel. So werden über die erste Klausel die y-Werte erstellt, in Abhängigkeit vom Attribut der Gesamtbewertung eines Abschnittes(bew_ges). Die zweite Fallunterscheidung ist für die Generierung der Farbwerte zuständig, ebenfalls in Abhängigkeit von der Gesamtbewertung.

```
1 SELECT distinct table_wk.wasserkoeper, table_gn.gwk_gn, table_wk.bew_ges,
2         table_wk.stat_von, table_wk.sonderfall,
3
4         CASE
5             WHEN table_wk.bew_ges = 1 THEN 1::numeric
6             WHEN table_wk.bew_ges = 2 THEN 2::numeric
7             WHEN table_wk.bew_ges = 3 THEN 3::numeric
8             WHEN table_wk.bew_ges = 4 THEN 4::numeric
9             WHEN table_wk.bew_ges = 5 THEN 5::numeric
10            WHEN table_wk.bew_ges = 0 AND table_wk.sonderfall != 'verrohrt' THEN 0::numeric
11            WHEN table_wk.sonderfall = 'verrohrt' THEN 5::numeric
12        END AS y_werte,
13        CASE
14            WHEN table_wk.bew_ges = 1 THEN 'blue'::text
15            WHEN table_wk.bew_ges = 2 THEN 'green'::text
16            WHEN table_wk.bew_ges = 3 THEN 'yellow'::text
17            WHEN table_wk.bew_ges = 4 THEN 'orange'::text
18            WHEN table_wk.bew_ges = 5 THEN 'red'::text
19            WHEN table_wk.bew_ges = 0 AND table_wk.sonderfall != 'verrohrt' THEN 'gray'::text
20            WHEN table_wk.sonderfall = 'verrohrt' THEN 'black'::text
21        END AS farbe
22
23 FROM table_wk, table_gn
24
25 WHERE entferne_zeichen(table_wk.gewaesserkennzahl::text, '0'::text)::numeric = table_gn.lawa_gewaesserkennzahl_kurz
26 and gwk_gn = 'Routenname' and table_wk.wasserkoeper = 'Wasserkörper'
27
28 ORDER BY stat_von;
```

Abbildung 50 - SQL- Statement für FGSK-Daten

Neben den Vorgaben aus der WRRL, dass die Güteklassen 1-5 die Farben Blau, Grün, Gelb, Orange und Rot annehmen müssen, wurden auch zwei Sonderfälle behandelt. Verrohrte Abschnitte bekommen den y-Wert 5 zugewiesen und nehmen die Farbe schwarz an. Sonstige und nicht bewertete Abschnitte sind mit dem y-Wert 0 versehen und werden grau dargestellt.

Als Ergebnis liefert die Datenbank für die Route Warnow im Wasserkörper WAOB-0100 folgende Sicht zurück, welche ausschnittsweise mit 6 Datensätzen abgebildet wird:

	wasserkörper character varying(9)	routenname character varying(60)	stationierungsbeginn integer	gesamtbewertung integer	sonderfall text	y_werte numeric	farbe text
1	WAOB-0100	Warnow	71116	2	kein	2	green
2	WAOB-0100	Warnow	71174	2	kein	2	green
3	WAOB-0100	Warnow	71224	3	kein	3	yellow
4	WAOB-0100	Warnow	71275	2	kein	2	green
5	WAOB-0100	Warnow	71365	2	kein	2	green
6	WAOB-0100	Warnow	71427	0	sonstiges	0	gray

Abbildung 51 –Ergebnis des SQL-Statements für FGSK-Daten

Das schematische SQL-Statement für die Flächennutzungsdaten ist von der Syntax ähnlich aufgebaut. Der Name des Wasserkörpers und die Routenbezeichnung werden in der Where-Klausel zur Laufzeit übergeben und in der Select-Klausel werden über das Attribut fn_links die linksseitigen Flächennutzungsdaten abgefragt. Mit der ersten Fallunterscheidung wird für die Flächennutzungen Acker –und Grünland, Laub –und Nadelwälder, Lagerflächen und Verkehrsflächen der y-Wert -3 festgelegt. Die zweite Fallunterscheidung legt die Farbwerte fest. Es wurden im SQL-Pseudocode beliebige HTML konforme Farbnamen gewählt. Sollte es wie bei den FGSK-Güteklassen eine standardisierte Legende für die genannten Flächennutzungen geben, sind diese Farben zu wählen.

```

1 SELECT table_fn.wasserkoeper,
2        table_fn.gwk_gn,
3        table_fn.stat_von,
4        table_fn.fn_links,
5
6        CASE
7            WHEN table_fn.fn_links = 'Acker' THEN -3::numeric
8            WHEN table_fn.fn_links = 'Grünland' THEN -3::numeric
9            WHEN table_fn.fn_links = 'Laubwald' THEN -3::numeric
10           WHEN table_fn.fn_links = 'Nadelwald' THEN -3::numeric
11           WHEN table_fn.fn_links = 'Lagerflächen' THEN -3::numeric
12           WHEN table_fn.fn_links = 'Verkehrsflächen' THEN -3::numeric
13        END AS y_werte
14        CASE
15           WHEN table_fn.fn_links = 'Acker' THEN 'yellow'::text
16           WHEN table_fn.fn_links = 'Grünland' THEN 'green'::text
17           WHEN table_fn.fn_links = 'Laubwald' THEN 'darkgreen'::text
18           WHEN table_fn.fn_links = 'Nadelwald' THEN 'springgreen'::text
19           WHEN table_fn.fn_links = 'Lagerflächen' THEN 'gray'::text
20           WHEN table_fn.fn_links = 'Verkehrsflächen' THEN 'red'::text
21        END AS farbe
22
23 FROM table_fn, table_gn
24
25 WHERE entferne_zeichen(table_fn.gewaesserzahl::text, '0'::text)::numeric = table_gn.lawa_gewaesserkennzahl_kurz
26 and table_fn.gwk_gn = 'Routenname' and table_fn.wasserkoeper = 'Wasserkörper'
27
28 order by stat_von
29

```

Abbildung 52 - SQL-Statement für Flächennutzungsdaten

Für die rechtsseitige Flächennutzung ist analog zu der obigen Beschreibung das Attribut fn_links durch fn_rechts zu ersetzen.

Abfragedaten speichern

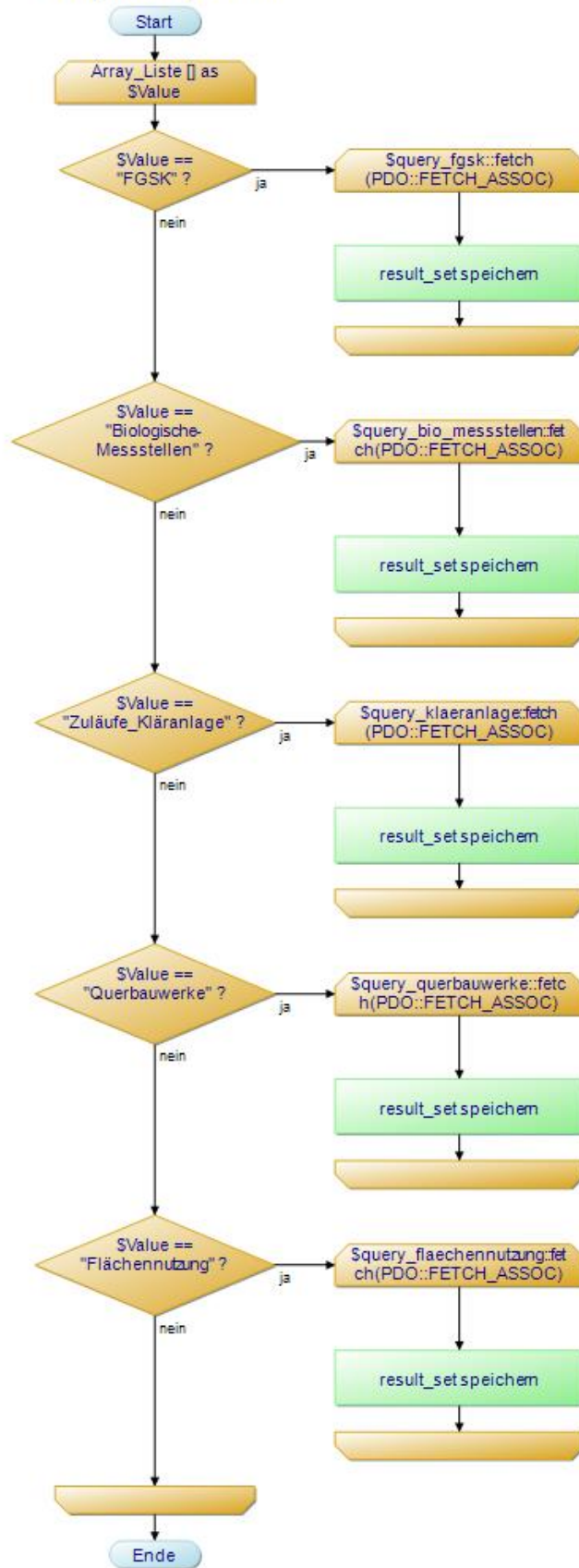


Abbildung 53 - Programmablaufplan Abfrageergebnis speichern

Damit die Datenbankabfrageergebnisse aufbereitet werden können für das Framework, wird in einer foreach-Schleife das Array mit den ausgewählten Attributen durchlaufen und mit einer if bzw. else if-Anweisung geprüft, ob eines der Array-Elemente dem Wert einer Instanzvariablen entspricht, welche die Auswahlattribute enthalten. Liefert die if-Bedingung TRUE zurück wird mit einer While-Schleife das result-set aus der Datenbankabfrage durchlaufen und auf einem Array abgespeichert. So wird für jede ausgeführte SQL-Anweisung ein Array angelegt, das die Werte für das Diagramm enthält.

Der Aufbau ist für alle Arrays gleich. Jeder Bewertungsabschnitt besteht aus drei Elementen und bildet ein Array-Objekt: einem x-Wert, der sich aus dem Stationierungsbeginn ableitet und dem y-Wert, der in Abhängigkeit vom Attribut verschiedene Darstellungsaufgaben übernimmt, z.B. für das Attribut FGSK stellt er die Güteklasse eines Bewertungsabschnittes dar. Zusätzlich wird zu jedem Abschnitt ein Farbwert übergeben. Pro Schleifendurchlauf werden dem assoziativen Array, das mit der php-Funktion `fetch(PDO::FETCH_ASSOC)` erstellt wurde, diese Elemente entnommen.

Eine Besonderheit in der Darstellung der Querbauwerksdaten besteht darin, dass für jedes Bauwerk keine Teilfläche erstellt werden muss, wie bei den Flächennutzungsdaten oder den FGSK-Daten, sondern nur eine Linie die abstrahiert das Bauwerk symbolisiert. Daher werden pro Schleifendurchlauf zwei Array-Objekte erzeugt. Dem ersten Objekt wird die Farbe Weiß zugewiesen, damit die Lücken zwischen den Bauwerken im Diagramm nicht sichtbar sind. Die zweite Farbe ist schwarz, sodass dadurch ein Balken dargestellt wird. Die Breite des Balkens kann über den x-Wert des zweiten Array-Objektes variiert werden.

10.8.3 Datenaufbereitung

Zu Beginn der Datenaufbereitung werden die einzelnen Arrays zu einem multidimensionalen Array verbunden, damit die Möglichkeit besteht die Positionen von den Arrays, die das result-set der Datenbank enthalten, zu vertauschen. Im nächsten Schritt wird das Array der Funktion `merge()` übergeben, damit wieder ein einzelnes Array entsteht, mit allen Diagrammwerten aus den ausgewählten Attributen, das vom Framework erwartet wird.

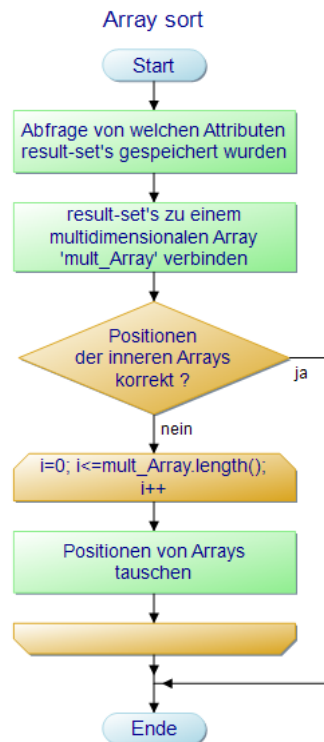


Abbildung 54 - Programmablaufplan Array sort

Bei den, im aktuellen Entwurf, berücksichtigten Attributen ist ein vertauschen der inneren Arrays nötig, denn auf der negativen y-Achse müssen die Arrays so sortiert sein, das der y-Wert mit der geringsten Entfernung zur x-Achse als letztes dem verbundenen Array hinzugefügt wird. So wird verhindert, dass die Attribute mit einem kleineren y-Wert verdeckt werden.

Die folgende Grafik zeigt den schematischen Aufbau, wie die von der Datenbank bereitgestellten Daten verarbeitet werden, bevor das verbundene Array dem Framework übergeben wird.

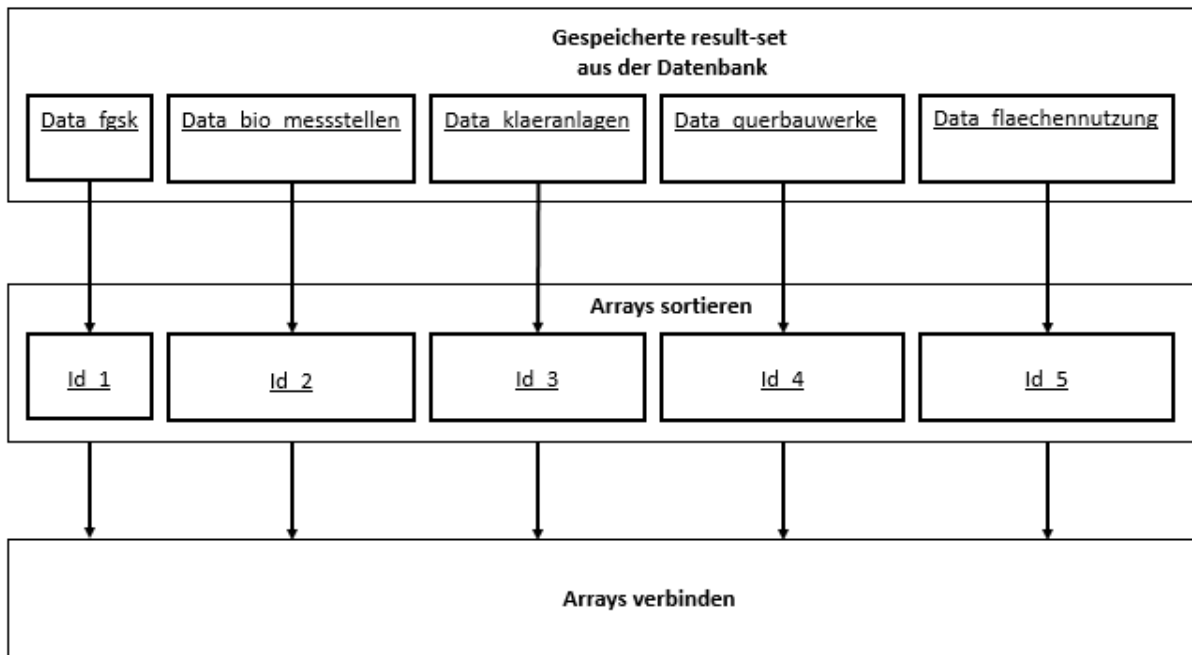


Abbildung 55 - Schnittstelle zwischen Datenspeicherung und Datenaufbereitung

Aus dem Analyseteil ging hervor, dass Highcharts die Anforderungen an das Diagramm am besten erfüllen konnte. Bei diesem Framework erfolgt die Übergabe der Datensätze in einem Series-Objekt, muss dem Datentyp Array entsprechen und sieht in der allgemeinen Form folgendermaßen aus:

```

1  series: [{
2      name: ''
3      data: []
4  }]

```

Abbildung 56 - Series-Objekt in der allgemeinen Form

Eine Übergabe der Datenliste kann auf drei verschiedene Arten erfolgen:

- 1) Eine Liste mit numerischen Werten, wobei diese den y-Wert repräsentieren und der x-Wert automatisch bei 0 beginnend jeweils um 1 erhöht wird.
- 2) Eine Liste von Arrays mit jeweils zwei Werten, die den x- und y-Wert darstellen
- 3) Eine Liste von Objekten die sich innerhalb eines Arrays befinden. In diesem Fall bilden die Objekte jeweils einen Punkt im Diagramm ab, dem drei Parameter zugewiesen werden können die im Abschnitt SQL-Statement und der Datenspeicherung schon erwähnt wurden:
 - x-Wert
 - y-Wert
 - Farbwert

Neben der Bereitstellung von Daten, muss der Diagrammtyp angegeben werden. Den funktionalen Anforderungen nach, soll es ein spezielles Flächendiagramm sein, dass zu jedem Bewertungsabschnitt eine Teilfläche erstellt. Dafür braucht es eine Hilfsfunktion die den Datenstrom von der Klasse Datenaufbereitung entgegennimmt, die Daten umwandelt und ein neues Series-Objekt zurückliefert. Die nachfolgende Grafik veranschaulicht, wie die Funktion die Teilflächen zusammensetzt.

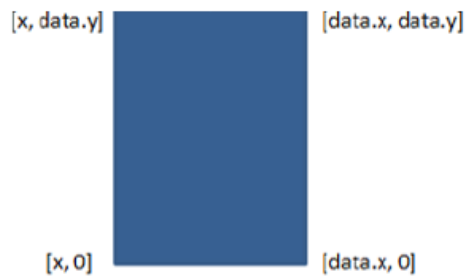


Abbildung 57 - Schema einer Teilfläche vom Diagramm

Jede Teilfläche kann als Rechteck aufgefasst werden und setzt sich aus vier Eckpunkten zusammen, die jeweils einen x und y-Wert besitzen.

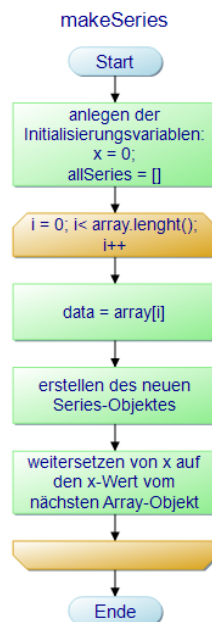


Abbildung 58 - Programmablaufplan für die Hilfsfunktion makeSeries

Als Input der Funktion dient das verbundene Array der Klasse Datenaufbereitung. Es enthält pro Array-Element drei Parameter. Mit einer for-Schleife wird pro Durchlauf ein Array-

Element auf einer Variablen gespeichert und so kann auf jeden Parameter zugegriffen werden. Zu Beginn der Funktion wird der x-Wert, der den jeweiligen Startwert des Rechtecks darstellt, vor Beginn der Schleife auf 0 gesetzt.

Bei jedem Schleifendurchlauf nimmt x den nächsten x-Wert des Datensatzes an. Auf diese Weise wird ein lückenloser Diagrammverlauf sichergestellt, weil der Endwert des vorhergehenden Rechtecks gleichzeitig der Beginn des nächsten Rechtecks ist.

Der y-Wert, und damit obere Grenze jedes Balkens stellt die Güteklasse, oder eine andere Attributeigenschaften dar und in der Funktion wird mit data.y auf den jeweiligen Array-Wert zugegriffen.

Eine mögliche Realisierung der Funktion makeSeries ist in der nächsten Abbildung zu sehen.

```
1 function makeSeries(listOfData) {
2     var allSeries = []
3     var x = 0.0;
4     for (var i = 0; i < listOfData.length; i++) {
5
6         var data = listOfData[i];
7         allSeries[i] = {
8             showInLegend: false,
9             color : data.color,
10            name : data.name,
11            data: [
12                [x, 0], [x, data.y], //Ecke links. unten und links. oben
13                {
14                    x: data.x,
15                    y: data.y,
16                },
17                [data.x, data.y], [data.x, 0] //ecke rechts. oben und rechts.unten
18            ],
19            w: data.x, //Breite(x-achse) des rechtecks
20            h: data.y //Höhe(y-achse) des rechtecks
21        };
22        x = data.x ; //Startwert für jedes Rechteck
23    }
24    return allSeries;
25 }
```

Abbildung 59 - Programmcode für die Funktion makeSeries

Die eigentliche Funktion in dem das Diagramm erzeugt wird ist plot(). Das Highcharts-Objekt besitzt verschiedene Attribute die Angaben zum Diagramm beinhalten können. Die wichtigsten sind:

- title – gibt den Diagrammtitel an
- chart – Angabe vom Diagrammtyp und ob eine Zoomfunktion benutzt werden soll
- x-Axis – Festlegen eines Minimumwertes der x-Achse und eine Achsenbeschriftung kann getroffen werden
- y-Axis – Achsenbeschriftung

- tooltip – Anzeigen von Zusatzinformationen, die allerdings im Series-Objekt enthalten sein müssen
- series – an dieser Stelle wird die Funktion makeSeries aufgerufen und ihr das Array aus der Klasse Datenaufbereitung übergeben
- exporting – Angabe von Höhe und Breite des Diagrammfensters in Pixel

In der Klasse Diagrammfenster sind mehrere Attribute dafür verantwortlich verschiedene Inhalte dem Benutzer zu präsentieren, die im Klassendiagramm in ihrer Funktionalität erklärt wurden. Mit Diagramm_plot wird in einem HTML-Container das Highcharts-Objekt über den Namen aufgerufen und dargestellt. Die Funktionsweise von Legende_sichtbar() wird in einem Struktogramm dargestellt.

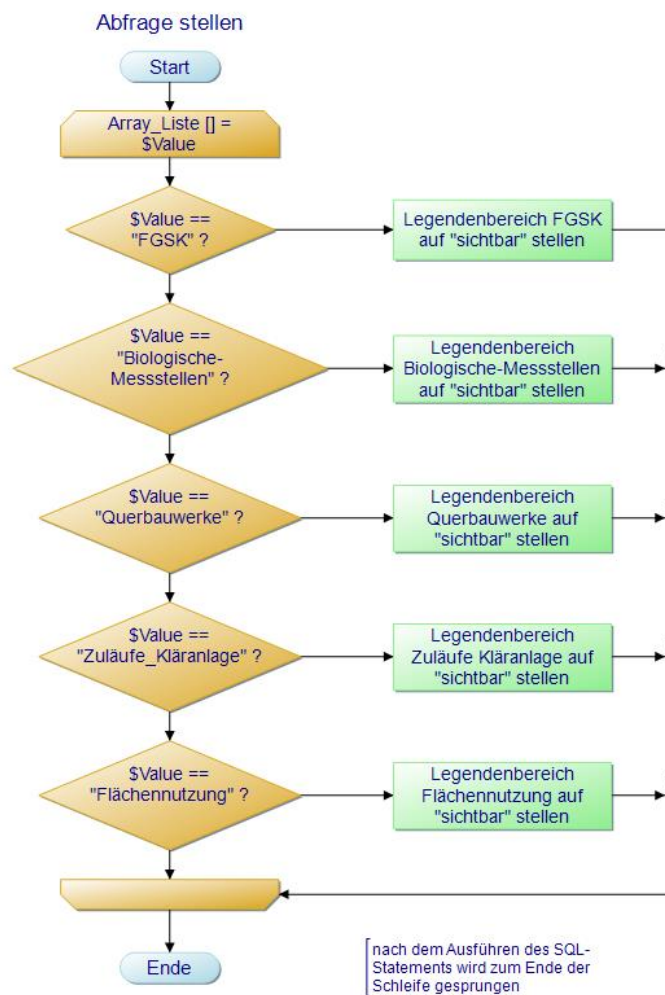


Abbildung 60 - Programmablaufplan für die Funktion Legende_sichtbar

10.9 Erweiterung des Systems um neue Attribute

In diesem Abschnitt von dem Systementwurf werden die Anpassungen aufgezeigt, um neue Attribute im Diagramm zu visualisieren.

In der Funktion `erzeuge_Attributliste` muss der Name des neuen Attributes auf einer Instanzvariablen gespeichert werden, die anschließend dem Array für die Attributliste übergeben wird.

Die Methoden `Abfrage_stellen()` und `Abfrage_speichern()` müssen innerhalb der Schleife um eine neue Verzweigung ergänzt werden, wobei der neue, zu prüfende Ausdruck, dem Namen des neuen Attributes entspricht.

Ein neues SQL-Statement muss erstellt werden, um die passenden Daten abzufragen und diese ist auf einer neuen Variable abzuspeichern, auf die dann in den Funktionen `Abfrage_speichern` und `Array_sort()` zugegriffen werden kann.

In der Methode `Array_sort()` ist die Art der Arrayzusammensetzung zu überarbeiten und eventuell ein neues Tauschschemata zu erstellen, falls der zugewiesene y-Wert bereits für ein anderes Attribut vorgesehen ist.

Ähnlich wie in den Methoden zum speichern und erstellen der Datenbankabfragen muss ebenfalls in der Funktion `Legende_sichtbar()` eine neue If-Bedingung erstellt werden, damit der zum neuen Attribut passende Legendenbereich angezeigt wird auf der Diagrammseite.

11 Prototypische Umsetzung

Nachdem im Abschnitt Entwurf die vollständige Funktionalität des Systems, mithilfe von UML-Diagrammen, Struktogrammen und Pseudocode aufgezeigt wurde, ist in diesem Kapitel die Prototypische Umsetzung der Web-Anwendung beschrieben. Es wird ausschließlich das Einsatzgebiet der Stand-Alone Anwendung beschrieben. Die dynamische Verwendung der Applikation wird nicht berücksichtigt.

Systemumgebung

Die zugrundeliegende Hardware des Prototyps besteht aus einem 64-bit Betriebssystem mit Windows 10 Pro[©], auf dem der Apache Webserver aus dem XAMPP-Paket installiert wurde. Die Installation und Konfiguration des Webserver und der Programmiersprache PHP, folgt den Schritten des Kapitels Skriptsprache.

In den technischen Anforderungen ist beschrieben, dass der verwendete Webbrowser über ein JavaScript-Plugin verfügen muss, um die von Highcharts erzeugten Diagramme darstellen zu können.

Im Prototyp wird Opera in der Version 45.0 als Webbrowser genutzt und in den Einstellungen das Plugin aktiviert.

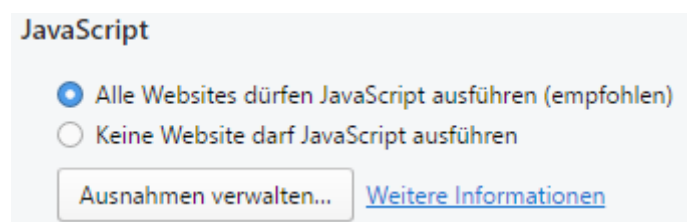


Abbildung 61 - Aktivieren des JavaScript-Plugins

Das zu verwendende Datenbanksystem PostgreSQL ist in der Version 9.5 installiert worden und für die Darstellung von Linearreferenzierten Daten, ohne die PostGIS-Erweiterung. Mit dem Administrations- und Entwicklungstool PgAdmin3 kann eine Datenbank erstellt und Demodatensätze hinzugefügt werden.

Demodatensätze bereitstellen

Für das beispielhafte Erzeugen von Diagrammen, hat das LUNG für den Wasserkörper EMES-2900 die Fachdaten zur Verfügung gestellt. Diese sind auf eine unterschiedliche Art und Weise aufbereitet worden. Für das Attribut FGSK sind die entsprechenden Demo-Daten mithilfe von INSERT-Anweisungen in die Datenbank geschrieben. Mit dem SQL-Statement für FGSK-Daten liefert die Datenbank folgende Sicht zurück, die ausschnittsweise in Abbildung 62 zu sehen ist:

	wk_k character varying(9)	gwk_gn character varying(60)	stat_von integer	bew_ges integer	sonderfall text	y_werte numeric	farbe text
1	EMES-2900	Mühlenbach	0	4	kein	4	orange
2	EMES-2900	Mühlenbach	228	4	kein	4	orange
3	EMES-2900	Mühlenbach	295	3	kein	3	yellow
4	EMES-2900	Mühlenbach	401	3	kein	3	yellow
5	EMES-2900	Mühlenbach	587	4	kein	4	orange
6	EMES-2900	Mühlenbach	727	4	kein	4	orange
7	EMES-2900	Mühlenbach	794	4	kein	4	orange
8	EMES-2900	Mühlenbach	844	4	kein	4	orange
9	EMES-2900	Mühlenbach	926	4	kein	4	orange
10	EMES-2900	Mühlenbach	1009	4	kein	4	orange

Abbildung 62 - Abfrageergebnis zu den FGSK-Daten

Für die Attribute Flächennutzung und Querbauwerke wird jeweils eine Textdatei erstellt, die den Arrayinhalt simulieren, der aus den result-sets erzeugt wird in der Funktion Abfrageergebnis speichern:

```
{color:'green', x:0, y:-3},
{color:'springgreen', x:228, y:-3},
{color:'yellow', x:295, y:-3},
{color:'green', x:401, y:-3},
{color:'white', x:587, y:-3},
{color:'yellow', x:727, y:-3},
{color:'gray', x:794, y:-3},
{color:'yellow', x:844, y:-3},
{color:'green', x:926, y:-3},
{color:'white', x:1009, y:-3},
{color:'darkgreen', x:1123, y:-3},
{color:'green', x:1223, y:-3},
{color:'white', x:1286, y:-3},
{color:'green', x:1345, y:-3},
{color:'green', x:1611, y:-3},
{color:'gray', x:1878, y:-3},
{color:'green', x:2052, y:-3},
{color:'white', x:2157, y:-3},
{color:'green', x:2524, y:-3},
{color:'green', x:2850, y:-3},
{color:'darkgreen', x:3054, y:-3},
{color:'gray', x:3282, y:-3},
{color:'white', x:3705, y:-3},
{color:'green', x:3741, y:-3},
```

Abbildung 63 - Datensimulation für die linksseitige Flächennutzung

```
{color:'white', x:300, y:-5},
{color:'black', x:310, y:-5},
{color:'white', x:780, y:-5},
{color:'black', x:790, y:-5},
{color:'white', x:1200, y:-5},
{color:'black', x:1210, y:-5},
{color:'white', x:1500, y:-5},
{color:'black', x:1510, y:-5},
{color:'white', x:1700, y:-5},
{color:'black', x:1710, y:-5},
```

Abbildung 64 - Datensimulation für Querbauwerke

Eingabefeld Wasserkörper

In diesem HTML-Eingabefeld soll die das Wasserkörperkürzel eingegeben werden. Es ist aber keine Prüffunktion vorhanden, die Nutzer darauf hinweist das der eingegebene Wasserkörpername in der Datenbank nicht vorhanden ist.

Wasserkörper:

Abbildung 65 - Eingabefeld Wasserkörper

Routentabelle

Die Anforderungen an die Routentabelle sind alle erfüllt und umgesetzt. Neben der fortlaufenden ID wird auch der Wasserkörpername in der Tabelle angezeigt. Der Gewässername ist mit dem Hyperlink kombiniert.

#	ID	Wasserkörper	Routenname
<input checked="" type="radio"/>	1	EMES-2900	Mühlenbach
<input type="radio"/>	2	EMES-2900	Graben aus Kremmin

Abbildung 66 – Routentabelle

Navigationsleiste

Über den Eintrag „neuer Wasserkörper“ gelangt der Benutzer bereits wieder zurück zur Auswahlseite, wo die Wasserkörpereingabe erfolgen kann und die Routentabelle abgebildet wird. Hinter den Buttons „Doku“ und „Hilfe“ sind noch keine weiteren Informationen hinterlegt.

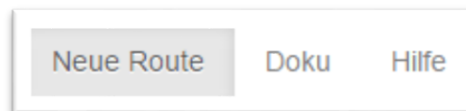


Abbildung 67 – Navigationsleiste

Attributauswahl

Unterhalb der Routentabelle ist das Modul Attributauswahl eingebunden. Es können FGSK-Daten, Querbauwerke und Flächennutzungsdaten selektiert werden, allerdings steht hinter den Radio-Buttons und der Checkbox keine Funktionalität, sodass im Diagramm immer dieselben Werte dargestellt werden.

Attributauswahl

Wählen Sie Attribute aus und aktualisieren anschließend das Diagramm:

Güteinformation

- FGSK-Gesamt
- FGSK-Sohle
- FGSK-Ufer
- FGSK-Land

Belastungsinformation

- Querbauwerke

Sonstiges

- Flächennutzung links
- Flächennutzung rechts

Abbildung 68 – Attributauswahl

Diagrammbereich

Das erzeugte Diagramm bildet den oberen Content Bereich der Website. Der Diagrammtitel wird automatisch erzeugt, indem sowohl der Name des Wasserkörpers, als auch die Gewässerroute dynamisch eingebunden werden. Im aktuellen Diagramm sind die FGSK-Gesamt Daten, die Querbauwerksdaten und die linken Flächennutzungsdaten zum Mühlenbach dargestellt.

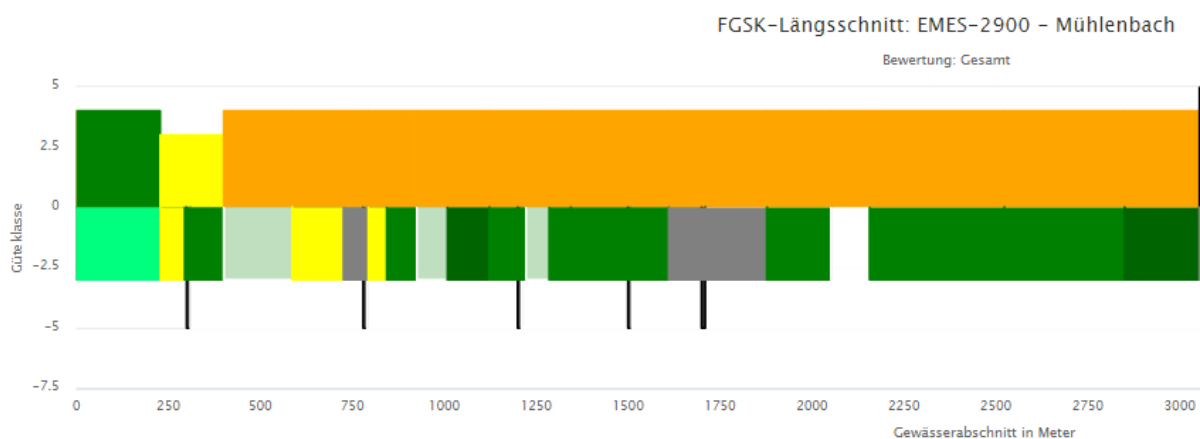


Abbildung 69 - Diagrammbereich

Legende

Die eingebundene Legende wird in der aktuellen Prototypischen Umsetzung noch nicht dynamisch in Abhängigkeit zur den ausgewählten Attributen erzeugt, sondern mithilfe von statischem HTML-Code.

Legende		
FGSK		
Y-Wert	Farbe	Bedeutung
0	grau	nicht kartiert
1	blau	sehr guter Zustand
2	grün	guter Zustand
3	gelb	mäßiger Zustand
4	orange	befriedigender Zustand
5	rot	schlechter Zustand
5	schwarz	verroht
Flächennutzung		
Farbe	Bedeutung	
gelb	Acker	
hellgrün	Nadelwald	
dunkelgrün	Laubwald	
rot	Bebauung mit Freiflächen	
grau	Lagerflächen	
hellgrün	Grünland / Parkanlage	
Querbauwerke		
<hr/>		

Abbildung 70 – Legend FGSK-Daten

Exportieren des Diagramms

Es sind die von Highcharts unterstützten Export-Formate PNG und PDF über jeweils einen Button unterhalb des Diagrammbereiches verfügbar. Die Diagramme werden auf dem Highcharts-Exportserver in das entsprechende Format konvertiert und anschließend im Download-Ordner des Benutzers gespeichert.

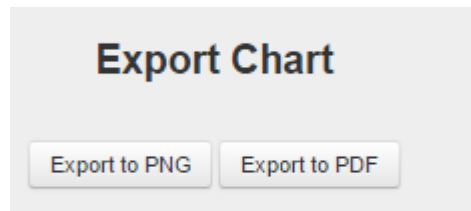


Abbildung 71 - Exportieren des Diagramms

12 Ausblick

Das System bietet bis zur vollständigen Umsetzung des Entwurfes noch einige Möglichkeiten, wie die Web-Anwendung effizienter und interaktiver gewässerbezogene Daten visualisieren kann. Im Folgenden werden Ansätze aufgelistet die ergänzt werden müssen, damit der vorliegende Entwurf umgesetzt wird und sind in ihrer Priorität absteigend sortiert.

Bereitstellen der Datensätze auf Datenbankebene

Der Entwurf stützt sich auf eine datenbankbasierte Abfrage der Daten. Daher sind die Fachdaten, welche in z. B. Excel vorliegen in die Datenbank zu übertragen und entsprechende SQL-Statements zu entwickeln, die eine Datenabfrage gewährleisten.

Funktionalität zur Auswahl von Attributen implementieren

Damit mehrere Attribute gleichzeitig im Diagramm dargestellt werden können ist der Entwurf im Bereich Datenselektion und Darstellung der Daten umzusetzen. In welcher Art und Weise der Datenstring aufgebaut sein muss, der dem Framework übergeben wird, ist prototypisch für die Kombination von FGSK-, Flächennutzungs- und Querbauwerksdaten umgesetzt.

URL Parametrisierung um Attributliste erweitern

Wenn die dynamische Darstellung von den ausgewählten Attributen umgesetzt und funktionsfähig ist, kann die Parametrisierung der URL, um die ausgewählte Attributliste erweitert werden. So kann die Anwendung in den Fachinformationssystemen in vielen Bereichen eingesetzt werden, da auf diese Weise z. B. die Feature-Info Abfrage erweitert kann in Abhängigkeit der dargestellten Attribute.

Abbildungsverzeichnis

Abbildung 1 - Flussgebietseinheiten M-V ¹	9
Abbildung 2 - in M-V auftretende Fließgewässertypen	10
Abbildung 3 - biologische Fließgewässerbewertung in M-V ²	11
Abbildung 4 – mögliche Darstellung von Güte- und Belastungsinformationen.....	13
Abbildung 5 - Use-Case-Diagramm.....	16
Abbildung 6 – statisches Client-Server Prinzip	18
Abbildung 7 - dynamisches Client-Server Prinzip.....	19
Abbildung 8 - Funktionsweise DBI ¹²	26
Abbildung 9 - Funktionsweise PDO ¹³	26
Abbildung 10 - Komponenten des Webservers.....	27
Abbildung 11 - Funktionstest PHP	28
Abbildung 12 - Vergabe von Gewässerkennzahlen ²³	34
Abbildung 13 - Ereignisse einer linearen Referenzierung	35
Abbildung 14 - Aufbau einer Gewässerroute.....	35
Abbildung 15 - ER-Diagramm für die Beziehung Gewässerroute - Gewässerkennzahl	37
Abbildung 16 - ER-Diagramm für die Beziehung Wasserkörper und Gewässerroute	37
Abbildung 17 - ER-Diagramm für die Beziehung zwischen Gewässerroute und Bewertungsabschnitte	37
Abbildung 18 - WebGIS kvwmap(FIS-WRRLLight) ²⁶	39
Abbildung 19 – Feature Info Abfrage um Attribut FGSK-Längsschnitt erweitert ²⁶	40
Abbildung 20 - Aufruf der Routentabelle über einen dynamisch erzeugten Link.....	40
Abbildung 21 - Einsatzgebiete der Web-Anwendung	41
Abbildung 22 - Architekturentwurf	42
Abbildung 23 – Komponentendiagramm	43
Abbildung 24 - Eingabemaske Wasserkörper.....	43
Abbildung 25 - Userinterface Routentabelle.....	44
Abbildung 26 - Userinterface Diagrammpräsentation	45
Abbildung 27 - Klassendiagramm Wasserkörper	46
Abbildung 28 - Klassendiagramm Attributauswahl.....	47
Abbildung 29 - Klasse Datenabfrage	48
Abbildung 30 - Klassendiagramm Datenaufbereitung	48
Abbildung 31 - Klassendiagramm Framework.....	49
Abbildung 32 - Klassendiagramm Diagrammfenster.....	49
Abbildung 33 - Gesamtübersicht Klassendiagramm	50
Abbildung 34 - Aktivität erster Diagrammaufruf.....	50
Abbildung 35 - Interaktion im Diagrammfenster	51
Abbildung 36 - Speichern einer Attributliste.....	51
Abbildung 37 - Einlesen von neuen Attributen	52
Abbildung 38 - Datenflussdiagramm für das Erzeugen einer Routentabelle	52
Abbildung 39 - Speichern von Diagrammwerten	53
Abbildung 40 - sequenzieller Aufruf von Funktionen zur Datenaufbereitung.....	54
Abbildung 41 - Datenflussmodell der Webanwendung	54
Abbildung 42 - Programmablaufplan zur Wasserkörpereingabe.....	55
Abbildung 43 - SQL- Statement für Gewässerrouten.....	56
Abbildung 44 - Abfrageergebnis Routentabelle	57
Abbildung 45 - Programmablaufplan erzeuge Attributliste	58
Abbildung 46 - Programmablaufplan auswerten_Checkbox	58

Abbildung 47 - Programmablaufplan Attributliste speichern	59
Abbildung 48 - Programmablaufplan Attributliste laden	60
Abbildung 49 - Programmablaufplan Abfrage stellen	61
Abbildung 50 - SQL- Statement für FGSK-Daten	62
Abbildung 51 –Ergebnis des SQL-Statements für FGSK-Daten.....	63
Abbildung 52 - SQL-Statement für Flächennutzungsdaten.....	63
Abbildung 53 - Programmablaufplan Abfrageergebnis speichern	64
Abbildung 54 - Programmablaufplan Array sort	66
Abbildung 55 - Schnittstelle zwischen Datenspeicherung und Datenaufbereitung	67
Abbildung 56 - Series-Objekt in der allgemeinen Form	67
Abbildung 57 - Schema einer Teilfläche vom Diagramm	68
Abbildung 58 - Programmablaufplan für die Hilfsfunktion makeSeries.....	68
Abbildung 59 - Programmcode für die Funktion makeSeries	69
Abbildung 60 - Programmablaufplan für die Funktion Legende_sichtbar	70
Abbildung 61 - Aktivieren des JavaScript-Plugins.....	72
Abbildung 62 - Abfrageergebnis zu den FGSK-Daten	73
Abbildung 63 - Datensimulation für die linksseitige Flächennutzung.....	74
Abbildung 64 - Datensimulation für Querbauwerke.....	74
Abbildung 65 - Eingabefeld Wasserkörper.....	74
Abbildung 66 – Routentabelle.....	75
Abbildung 67 – Navigationsleiste	75
Abbildung 68 – Attributauswahl.....	76
Abbildung 69 - Diagrammbereich	76
Abbildung 70 – Legend FGSK-Daten.....	77
Abbildung 71 - Exportieren des Diagramms.....	78

Literaturverzeichnis

- [1] Wasserrahmenrichtlinie M-V – Allgemeines und Organisation der WRRL in M-V, <http://www.wrrl-mv.de/> (Zugriff am 15.02.2017)
- [2] Vortrag im Rahmen der AL-Runde des LUNG vom 10.03.2014, Autor: André Steinhäuser, Titel: Biologische Gewässerbewertung gemäß der EG-Wasserrahmenrichtlinie in M-V
- [3] Endbericht zur Erarbeitung einer Methodik der Fließgewässerstrukturgütekartierung in M-V, Bearbeitung: *liota* - Institut für ökologische Forschung und Planung GmbH, Auftraggeber: Dipl.-Ing. Andreas Küchler (Ansprechpartner und fachliche Betreuung LUNG)
- [4] Kappel/Pröll/Reich/Retschitzegger (Hrsg.): Web Engineering. dpunkt-Verlag, 2003, S.3
- [5] Usage Statistics and Market Share of Web Servers for Websites, May 2017 , https://w3techs.com/technologies/overview/web_server/all (Zugriff am 16.05.2017)
- [6] Apache HTTP Server, https://de.onpage.org/wiki/Apache_HTTP_Server
- [7] Apache HTTP Server, <http://modules.apache.org> (Zugriff am 02.05.2017)
- [8] Welcome to NGINX Wiki! | NGINX, <https://www.nginx.com/resources/wiki/> (Zugriff am 16.05.2017)
- [9] OTTO, Mark: Bootstrap from Twitter. <https://blog.twitter.com/2011/bootstrap-twitter>, 2011 (Zugriff am 16.05.2017)
- [10] Komponenten Bootstrap, <http://holdirbootstrap.de/komponenten/> (Zugriff am 18.04.2017)
- [11] Einführung Wirtschaftsinformatik (2012), Autor: Prof. Dr. Iris Vieweg, Prof. Dr. Dr. Christian Werner, Klaus-P. Wagner, Dr.-Ing. Thomas Hüttl, Dieter Backin, Online ISBN: 978-3-8349-6856-2
- [12] The PostgreSQL Licence (PostgreSQL) | Open Source Initiative, <https://opensource.org/licenses/postgresql> (Zugriff am 12.03.2017)
- [13] Apache - Kommunikation zwischen Webserver (Apache) und Datenbank - TEIA AG, <https://www.teialehrbuch.de/Kostenlose-Kurse/Apache/15486-Kommunikation-zwischen-Webserver-und-Datenbank.html> (Zugriff am 14.03.2017)
- [14] ZenTut - Programming Made Easy, <http://www.zentut.com/wp-content/uploads/2013/04/php-pdo.png> (Zugriff am 14.03.2017)
- [15] Charts examples - Ample SDK, <http://www.amplesdk.com/examples/markup/chart/> (Zugriff am 20.04.2017)
- [16] Ext JS 4.0.7 - Sencha Docs, http://docs.sencha.com/extjs/4.0.7/#!/guide/drawing_and_charting (Zugriff am 20.04.2017)
- [17] Charts - YUI Library, <https://yuilib.com/yui/docs/charts/> (Zugriff am 20.04.2017)
- [18] Examples, <http://www.jqplot.com/examples/> (Zugriff am 21.04.2017)
- [19] Dygraphs Gallery, <http://dygraphs.com/gallery/> (Zugriff am 21.04.2017)
- [20] Dygraphs Tests, <http://dygraphs.com/tests/> (Zugriff am 21.04.2017)
- [21] Highcharts demos | Highcharts, <http://www.highcharts.com/demo/> (Zugriff am 22.04.2017)

[22] Highcharts demos | Highcharts, <http://api.highcharts.com/highcharts> (Zugriff am 22.04.2017)

[23] https://www.umweltkarten.mv-regierung.de/meta/stat_p.pdf (Zugriff am 02.04.2017)

[24]

http://www.lawa.de/documents/Richtlinie_fuer_die_Gebietsbezeichnung_und_die_Verschluesselung_von_Fliessgewaessern_fe8.pdf (Zugriff am 02.04.2017)

[25] Was ist lineare Referenzierung | ArcGIS for Desktop

<http://desktop.arcgis.com/de/arcmap/10.3/guide-books/linear-referencing/what-is-linear-referencing.htm> (Zugriff am 04.04.2017)

[26] Einteilung der Wasserkörper | Nds. Landesbetrieb für Wasserwirtschaft, Küsten- und Naturschutz,

http://www.nlwkn.niedersachsen.de/wasserwirtschaft/flussgebietsmanagement_egwrrl/oberflaeche-ngewaesser/einteilung_wasserkoerper/einteilung-der-wasserkoerper-43983.html

[27] FIS WRRL (light), <http://www.fis-wasser-mv.de/kvwmap/index.php?gast=35>