Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

3D-Visualisierung von Zeitreihen im Energiekontext

Bachelorarbeit

Leipzig, TODO, 2016

vorgelegt von

Bechert, Stefan
Studiengang Informatik
Bachelor of Science

Betreuender Hochschullehrer:
Dr. Stefan Kühne
Fakultät für Mathematik und Informatik
Institut für Informatik

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Zeitreihen sind für alle heutigen wissenschaftlichen Disziplinen von großer Relevanz. Sei es zur Auswertung von Smart Meter Daten im Energiebereich oder der Analyse und Vorhersage von Aktienkursen an der Börse, Zeitreihen eignen sich immer dann wenn eine zeitliche Entwicklung empirisch betrachtet wird. [vgl. 1, S. X]

Im Rahmen der Auswertung von Zeitreihen sind verschiedene Darstellungsformen möglich, die jeweils ihre eigenen Vorteile mit sich bringen. Denkbar sind zweidimensionale oder dreidimensionale Darstellungen. In Ersterer ist ein Kennwert immer genau durch einen Zeitpunkt indentifizierbar.

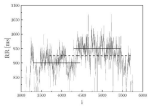

Abbildung 1: Zweidimensionale Zeitreihendarstellung[1]

Bei der dreidimensionelle Darstellung hingegen, kann man die Zeitreihe in Abhängigkeit von zwei zeitlichen Dimensionen auswerten. In Abbildung zwei zum Beispiel wird ein Kennwert nur jeweils durch ein Tupel aus Zins und Spalte eindeutig indentifiziert.

Ein Vorteil den diese Darstellung mit sich bringt, ist die Möglichkeit einzelne Spalten gegneinander zu vergleichen. So sieht man exemplarisch in Abbildung zwei, dass der Zins Anfang des Jahres 2014 in Spalte 10 bedeutend höher ist als Ende des Jahres.

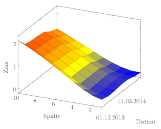

Abbildung 2: Dreidimensionale Zeitreihendarstellung[2]

Ein weiterer Vorteil, der in Industrie und Wissenschaft große Bedeutung hat, ist, dass Ausschläge in den Daten in vielen Fällen sofort ins Auge fallen. Darunter könnten sich potentiell auch Fehler in den Daten befinden.

In Abbildung drei ist eine Zeitreihe zu sehen, bei der die Extrema in der Mitte der Daten deutlich sichtbar sind. Vergleiche hinsichtlich dem Unterschied zweidimensionale und dreimdimensionale Darstellung werden in Abschnitt 2.1 angeführt.

[1] `http://www.max-schaldach-stiftung.de/.include/images/Stationaer.jpg`, 07.03.2016
[2] `http://texwelt.de/wissen/upfiles/test_75.png`, 07.03.2016

1.2 Zielsetzung

Ziel dieser Arbeit ist es vorhandene Webframeworks, die für die Darstellung von Zeitreihen entwickelt wurden und open source zugänglich sind, zu evaluieren und auf Basis eines davon eine Webkomponente zu bauen, die eine einfachere Auswertung von äquidistanten Zeitreihendaten ermöglicht.

Analysis: Bereits existierende Frameworks für Zeitreihendarstellung sollen hinsichtlich den geforderten Zielen dieser Bachelorarbeit auf ihren Funktionsumfang geprüft werden. Anhand dessen soll aufgezeigt werden welches Framework sich am ehesten für diese Arbeit eignet und welche Schritte dann in der Implementierung noch von nöten sind um die gewünschten Funktionalitäten umzusetzen.

Functionality: Die Komponente soll dem Benutzer ein Interface bereitstellen mit der die zeitlichen Dimensionen beliebig angepasst werden können. Um das zu realisieren müssen vordefinierte Aggregatsfunktionen bereitgestellt werden. Desweiteren soll durch eine anpassbare Farbskala auch die Auswertung der Daten in der dritten Dimension, der Höhe, vereinfacht werden.

Reusability: Die Webkomponente soll ein in sich abgeschlossenes Modul werden, dass sich in verschiedenen Webapplikationen integrieren lässt. Aus diesem Grund soll die Software auf Basis von AngularJS [3] entwickelt und schließlich mit bekannten Web-Build-Tools wie bower [4] oder npm [5] ausgeliefert werden.

[3] `https://angularjs.org/`
[4] `http://bower.io/`
[5] `https://www.npmjs.com/`

1.3 Gliederung

Die Arbeit ist in sechs Bestandteile gegliedert. Zu Beginn steht die Einleitung in der die Motivation für diese Arbeit erläutert und die Ziele, die erfüllt werden sollen, gesetzt werden.

Darauf folgt der Analyse-Abschnitt. Relevante Anwendungsszenarien werden simuliert und ausgewertet. Ziel ist es sowohl nochmals den Mehrwert, den diese Arbeit bietet, genauer herauszuarbeiten als auch Ziele der entstehenden Software zu spezifizieren. Desweiteren werden Frameworks, die als Grundlage dienen können, analysiert und zum Schluss eine Machbarkeitsstudie durchgeführt.

Den dritten Teil dieser Arbeit bildet ein Grundlagenkapitel. Hier werden wichtige Begrifflichkeiten definiert und die logischen Überlegungen, auf denen die Software aufgebaut seien wird, erläutert.

Im folgenden Kapitel, dem Konzept, werden Vorüberlegungen getroffen nach welchem Muster entwickelt werden soll. Wird ein Vorprojekt entwickelt? Wird test-first Development eingesetzt? Welche Programmierparadigmen werden zu Grunde gelegt? Diese Fragen werden beantwortet.

Danach folgt die Dokumentation des eigentlichen Implementierungsprozesses. Wichtige Fragen und Entscheidungen in der technischen Umsetzung werden hier diskutiert.

Zu guter Letzt folgt eine Auswertung der Arbeit. Es wird evaluiert ob die geforderten Ziele erreicht wurden und welche weiteren Ideen auf Basis der hier entwickelten Software perspektivisch umgesetzt werden könnten.

2 Analyse

In der Analyse werden Anwedungsszenarien vorgestellt anhand deren die genauen Anforderungen an die Software formuliert werden. Desweiteren werden Webframeworks auf die ermittelten Anforderungen hin untersucht und auf Basis dessen, eines davon als Grundlage für die Implementierung ausgewählt. Zum Schluss folgt eine Machbarkeitsstudie.

2.1 Anwendungsszenarien

2.1.1 Smart Meter Strom Zeitreihe

Viele Energielieferanten nutzen schon seit über zwei Jahrzehnten sogenannte „intelligente“ Zähler. Was früher hauptsächlich für Großunternehmen eingesetzt wurde findet heute auch immer mehr Gebrauch in Privathaushalten.

Während herkömmliche Stromzähler lediglich die Summe des verbrauchtes Stroms messen, ermöglichen es „intelligente“ Zähler, englisch Smart Meter genannt, auch Informationen darüber zu geben, wann wie viel Strom verbraucht wurde. Durch die Einbindung der Smart Meter in Kommunikationsnetze wird außerdem häufiges Auslesen der Daten für den Energeilieferanten kostengünstiger. Auf diese Weise entstehen Zeitreihen, die den Verbrauch an einem Anschlusses mit einem gegeben Takt in einem bestimmten Zeitraum widerspiegeln.

Auf Basis dieser Daten kann der Energielieferant für sich wichtige Auswertungen vornehmen. Tarife können verbrauchsspezifisch auf den Endkunden angepasst werden. [6] Oder durch die gewonnenen Daten können Verbesserung am Stromnetz, wie zum Beispiel die Optimierung von Stromspeichern, durchgeführt werden.

Im Folgenden soll ein Szenario betrachtet werden, bei dem eine Smart Meter Zeitreihe eines Endkunden in 1-Stunden Intervallen über ein Jahr gegeben sei. Mögliche Fragen, die ein Analyst bei der Auswertung dieser Daten für sein Energieunternehmen stellen könnte, wären:

1. Wie viel Strom wurde im gesamten Jahr verbraucht?
2. In welchem Monat wurde durchschnittlich am meisten/wenigsten Strom verbraucht?
3. Haben die Daten Fehler?
4. Gibt es ein starkes Verbrauchsgefälle zwischen Arbeitstagen und Wochenenden?
5. Bleibt der Verbrauch über einen Monat hin nahezu konstant?
6. Wie sieht der Verbrauch in der ersten Woche genau aus?

[6] vgl. https://www.bmwi.de/BMWi/Redaktion/PDF/Publikationen/Studien/kosten-nutzen-analyse-fuer-flaechendeckenden-einsatz-intelligenterzaehler,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf

2.1.2 Emissionsfaktoren der Stromerzeugung - 14 Jahre

`http://www.iinas.org/tl_files/iinas/downloads/GEMIS/2007_Emi-Daten_AGEEStat-ZSW.pdf`

2.2 Anforderungsanalyse

Auf Basis der vorgestellten Szenarien werden hier funktionale-, wie auch nicht-funktionale Anforderungen festgehalten.

2.2.1 Funktionale Anforderungen

Funktionale Anforderungen beschreiben in der Sprache des Systems die Funktionen, die ein System bereitstellen soll [vgl. 2, S. 29]. Im Folgenden werden use cases erstellt, die die Abläufe aus den oben genannten Szenarien verallgemeinern sollen. Das System (bzw. die Software) wird spezifischer als *Komponente* betitelt. Die Durchnummerierung dient dem besseren Bezug in der Auswertung am Ende der Arbeit.

1. Dreidimensionale Darstellung einer Zeitreihe.

   Die Komponente soll eine Zeitreihe in einem von ihr vordefinierten Format annehmen und dreidimensional darstellen können.

2. Bewegung im dreidimensionalen Raum.

   Die Komponente soll dem Nutzer ermöglichen die Kamera in der Darstellung zu bewegen. Dazu zählt sowohl Rotation als auch Verschiebung auf allen drei Achsen.

3. Anpassung der Farbskala.

   Die Komponente soll dem Nutzer ermöglichen die Farbskala aus einer großen Auswahl von Skalen auszuwählen. Desweiteren soll eine Möglichkeit bereit gestellt werden, die Unter- und Obergrenze beliebig zu setzen.

4. Verändern der Zeitachsen.

   Beide Zeitachsen sollen vom Nutzer im Rahmen, den die gegebene Zeitreihe vorgibt, angepasst werden können. Für diese Anforderung wird 5. benötigt.

5. Bereitstellen von Aggregationsfunktionen.

   Die Komponente soll dem Nutzer Aggregationsfunktionen für Summe, Durschnitt, Maximum und Minimum zur Verfügung stellen.

2.2.2 Nicht-Funktionale Anforderungen

Nicht-Funktionale Anforderungen, oder auch Qualitätsanforderungen, legen die qualitativen Eigenschaften, die das System leisten soll, fest. Sie beziehen sich auf die oben genannten funktionalen Anforderungen und besitzen die Eigenschaft nur schwer spezifizierbar, bzw. testbar zu sein [vgl. 2, S. 29 - 30]. Folgende nicht-funktionale Anforderungstypen werden unterschieden [vgl. 3, S. 249 - 250]:

- Aussehen und Handhabung
- Benutzbarkeit
- Sicherheit
- Leistung
- Wartbarkeit
- Übertragbarkeit
- Politische oder Kulturelle Umgebungsbedingungen
- Gesetzliche Vorgaben

Für die Komponente sind allerdings nur die Folgenden von Wichtigkeit:

Aussehen und Handhabung

Aussehen und Handhabung befassen sich nicht direkt mit den Details des Designs sondern legen viel mehr fest, welchen Eindruck die Komponente vermitteln soll [vgl. 3, S. 250].

Bei der zu entwickelnden Komponente wird der Hauptfokus im Bereich Aussehen und Handhabung auf ein schlichtes Design gesetzt, das es später ermöglicht, die Komponente in verschiedene Webanwendungen zu integrieren ohne deren Grunddesign zu stören. Desweiteren, da die Aufgabe dieser Komponente im Bereich der Visualisierung liegt, ist es ebenfalls wichtig, dem Interface, das die Komponente mit sich bringt, den Eindruck zu verleihen, die Komponente wäre nach dem aktuellen Stand der Technik (engl. state of the art [vgl. 3, S. 250]) designt worden.

Benutzbarkeit

Benutzbarkeit ist eine der Schlüsseleigenschaften, die entscheiden, ob die Software später von Benutzern, für die sie entwickelt wurde, genutzt wird [vgl. 3, S. 253]. Eine schlechte Benutzbarkeit kann trotz vollstäniger Funktionalität und gegebener Korrektheit zur Ablehnung der Software führen.

Da die zu entwickelnde Komponente Open Source ausgeliefert werden soll, um in verschiedenen Webapplikationen Anklang zu finden, ist es wichtig, die Benutzbarkeit der Komponente für verschiedene Endbenutzer sicherzustellen. Interesse an dieser Software haben hauptsächlich Benutzer, welche Zeitreihen auf verschieden Aspekte hin auswerten wollen. Das fordert zwar inhaltliches Wissen über die Auswertung, setzt

allerdings nur die üblichen Kompetenzen im Umgang mit Computern vorraus. Insofern soll die Komponente für gerade diese Benutzer einfach zu lernen sein.

Skalierbarkeit

Skalierbarkeit beschreibt, wie performant Software mit Änderungen in der Datengröße umgehen kann. Hinsichtlich der Tatsache, dass Zeitreihen beliebige Größenausmaße annehmen können, muss auch die Komponente mit solchen Änderungen umgehen können. Da die Komponente allerdings im Rahmen einer Webapplikation auf der Hardware des Endbenutzers laufen wird, lassen sich an dieser Stelle keine konkreten Angaben zu der Größe und der Zeit, in der die Operatioenn ablaufen, treffen. TODO (vlt Vergleichssystem wählen oder mit Komplexität festsetzen: doppelte Datengroesse = doppelte Zeit?)

Übertragbarkeit

Wie oben erwähnt, soll die Komponente Open Source für andere Webapplikationen zur Verfügung stehen. Im Rahmen dessen muss die Komponente die geforderten Umgebungsbedingungen, der Webapplikationen erfüllen. Ziel ist es, die Komponente in jedem Browser, der WebGL unterstützt, verwenden zu können. Desweiteren soll die Übertragbarkeit verbessert werden, indem die Komponente für Webentwickler über öffentliche Paket Manager ausgeliefert wird.

Korrektheit

Zu guter Letzt spielt die Korrektheit als nicht-funktionaler Anforderungstyp eine große Rolle für die Komponente. Die erfolgreiche und korrekte Auswertung einer Zeitreihe ist nur dann möglich, wenn die Daten richtig verarbeitet und visualisiert werden. Fehler in der Darstellung oder der Umrechnung würden zu falschen Schlussfolgerungen führen und könnten schwerwiegende Konsequenzen für den Endbenutzer mit sich bringen.

2.3 Webframeworks zur Zeitreihendarstellung

2.3.1 dygraphs

2.3.2 Plotly

2.3.3 x3dom

2.3.4 Fazit

2.4 Machbarkeitsstudie

3 Entwurf

Welche theoretischen und praktischen Grundlagen sind zur Umsetzung dieser Arbeit erforderlich?

3.1 Zeitreihen, Repräsentationen und Operationen

3.1.1 Definition Array

Sei $n \in \mathbb{N}$. Ein **Array** $arr = [a_0, a_1, ..., a_{n-1}]$ ist eine endliche Abfolge von Werten, die ueber ihren Index i referenziert werden. Es gilt $arr[i] = a_i$, wobei $i \in \{0, 1, ..., n-1\}$.

Die **Länge** eines Arrays $|arr|$ entspricht der Anzahl der Werte in arr, also $|arr| = n$.

Sei arr ein Array. Falls ein $i \in \{0, 1, ..., |arr|-1\}$ existiert sodass $arr[i]$ selbst ein Array ist so heisst arr **geschachtelt** und $arr[i]$ Unterarray von arr. Ansonsten heisst arr **nicht geschachtelt**.

Sei arr ein Array. Die Menge $\boldsymbol{\chi(arr)}$ ist die Menge aller Arrays in arr und deren rekursiv erreichbaren Unterarrays, die selbst nicht geschachtelt sind. (Eigentlich muesste man das besser ausdruecken)

3.1.2 Definition Zeitreihe

Eine **äquidistante Zeitreihe** ist ein Tripel

$$D^2 = (V, i, s) \tag{1}$$

wobei

$$V = [v_0, v_1, ..., v_{n-1}] \tag{2}$$

ein Array aus n Kennwerten darstellt. i beschreibt den zeitlichen Abstand zwischen zwei Kennwerten in Millisekunden und s repräsentiert den Starzeitpunkt der Zeitreihe in Millisekunden seit dem 01.01.1970.

3.1.3 Repräsentationen

2D - Repräsentation Die Repräsentation nach 3.1.2 ist eine 2-dimensionale Darstellung der Kennzahlen, da nur ein Index benoetigt wird um jede Kennzahl eindeutig zu referenzieren.

3D - Repräsentation Jede Zeitreihe kann auch in Abhängigkeit von zwei unterschiedlichen Zeitdimensionen angegeben werden.

Sei $D^2 = (V, i, s)$ eine Zeitreihe in 2D-Repräsentation. So lässt sich diese Zeitreihe auch darstellen in 3D-Repräsentation als

$$D^3 = (V', i, s) \tag{3}$$

mit

$$V' = [[v_0, v_1, ..., v_{t_1-1}]_0, [v_0, v_1, ..., v_{t_2-1}]_1, ..., [v_0, v_1, ..., v_{t_m-1}]_m] \tag{4}$$

wobei

$$\sum_{k=1}^{m} t_k = |V|. \tag{5}$$

Falls

$$t_1 = t_2 = ... = t_m \tag{6}$$

gilt, so nennt man V' eine **homogene**, ansonsten eine **inhomegene** 3D-Repräsentation von D.

Folgerung Homogen 3D Sei D^3 als homogen gegeben. So folgt:

$$\sum_{k=1}^{m} t_k = t * m = |V|. \tag{7}$$

nD - Repräsentation, Ueberlegung Allgemein kann man jede Zeitreihe in n vielen Dimensionen darstellen (da beliebig kleine Zeitschritte), allerdings erreicht man dadurch irgendwann keine neuen Informationen mehr sondern ausschliesslich Kennwertduplizierung.

3.1.4 Operationen - allgemein

Sei K^n die Menge aller Zeitreihen in nD-Repraesentation.

Unterteilung $\forall n \geq 2$ gilt:
Sei $D^n = (V, i, s)$ eine Zeitreihe in nD-Repraesentation.
Eine Unterteilung $\delta = (f, g)$ ist ein Tupel, wobei f eine Abbildung

$$f : K^n \to K^{n+1}. \tag{8}$$

darstellt mit

$$f(D^n) = f((V, i, s)) = (g(V), i, s) \tag{9}$$

und

$$g(V) = g.replace(arr \in \chi(V), g(arr)). \tag{10}$$

Sei $D^n = (V, i, s)$ homogen und $\delta(D^n) = D^{n+1} = (V', i, s)$. Eine Unterteilung heisst **homogen** falls ein $n \in \mathbb{N}$ exisitiert, sodass

$$\forall arr \in \chi(V') : |arr| = n. \tag{11}$$

Aggregation $\forall n \geq 3$ gilt:
Sei $D^n = (V, i, s)$ eine Zeitreihe in nD-Repraesentation.

Eine Aggregation ϕ ist eine Abbildung

$$\phi : K^n \to K^{n-1}. \tag{12}$$

Aggregationsregeln Sei $D^n = (V, i, s)$ eine Zeitreihe in nD-Repraesentation.

1. Eine Zeitreihe ist genau dann aggregierbar, wenn $n \geq 3$.
2. Sei $D^n = (V, i, s)$ homogen. Fuer
$$\phi(D^n) = D^{n-1} = (V', i', s) \tag{13}$$
gilt
$$i' = |\chi(D^n)| * i \tag{14}$$
.
3. Sei $D^n = (V, i, s)$ inhomogen. Fuer
$$\phi(D^n) = D^{n-1} = (V', i', s) \tag{15}$$
gilt
$$i' = [l_0, l_1, ..., l_{|\chi(V)|-1}] \tag{16}$$
wobei $i'[0] = l_0 = |\chi(V)[0]|$.

Unterteilung	**Aggregation**
D^2	D^2
$\downarrow$	$\uparrow$
D^3	D^3
$\downarrow$	$\uparrow$
...	...
$\downarrow$	$\uparrow$
D^{n-1}	D^{n-1}
$\downarrow$	$\uparrow$
D^n	D^n

4 Implementierung

Die Definition in 3.1.2 gilt im folgenden als die initiale Repräsentation einer Zeitreihe in der Anwendung. (Passt hier eig nicht, sollte eher zu Implementierung)

Tatsaechliche Umsetzung: Probleme, Entscheidungen,

Wichtige Forderungen TODO aber merken Dass die Teilarrays in der 3D Repraesentation die gleich Groesse haben ist fuer den spaeteren Render-Prozess von hoher Wichtigkeit, da andernfalls Daten in der Darstellung verloren gehen. Nichts destotrotz kann eine Zeitreihe auch in ungleiche Teile zerteilt werden (z.B. Monate), die allerdings zur Darstellung auf jeden Fall aggregiert werden muessen um die Forderung einzuhalten.

5 Zusammenfassung und Evaluation

5.1 Auswertung

Wurden die Ziele erfuellt? Wenn ja zeigen. Wenn nein, wieso nicht? Zeitpunkt: Nach der Fertigstellung der software.

5.2 Ausblick

Was koennte man aufbauend auf dieser Arbeit noch machen?

Literaturverzeichnis

[1] Schlittgen, Rainer / Streitberg, Bernd H. J.: Zeitreihenanalyse. 9. unwesentlich veränderte. München: Oldenbourg Verlag, 2001.

[2] Ebert, Christof: Systematisches Requirements Engineering : Anforderungen ermitteln, spezifizieren, analysieren und verwalten. dpunkt.verlag, 2011.

[3] Robertson, Susanne / Robertson, James: Mastering the Requirements Process: Getting Requirements Right. 3. Auflage. Amsterdam: Addison-Wesley, 2012.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann

Ort Datum Unterschrift

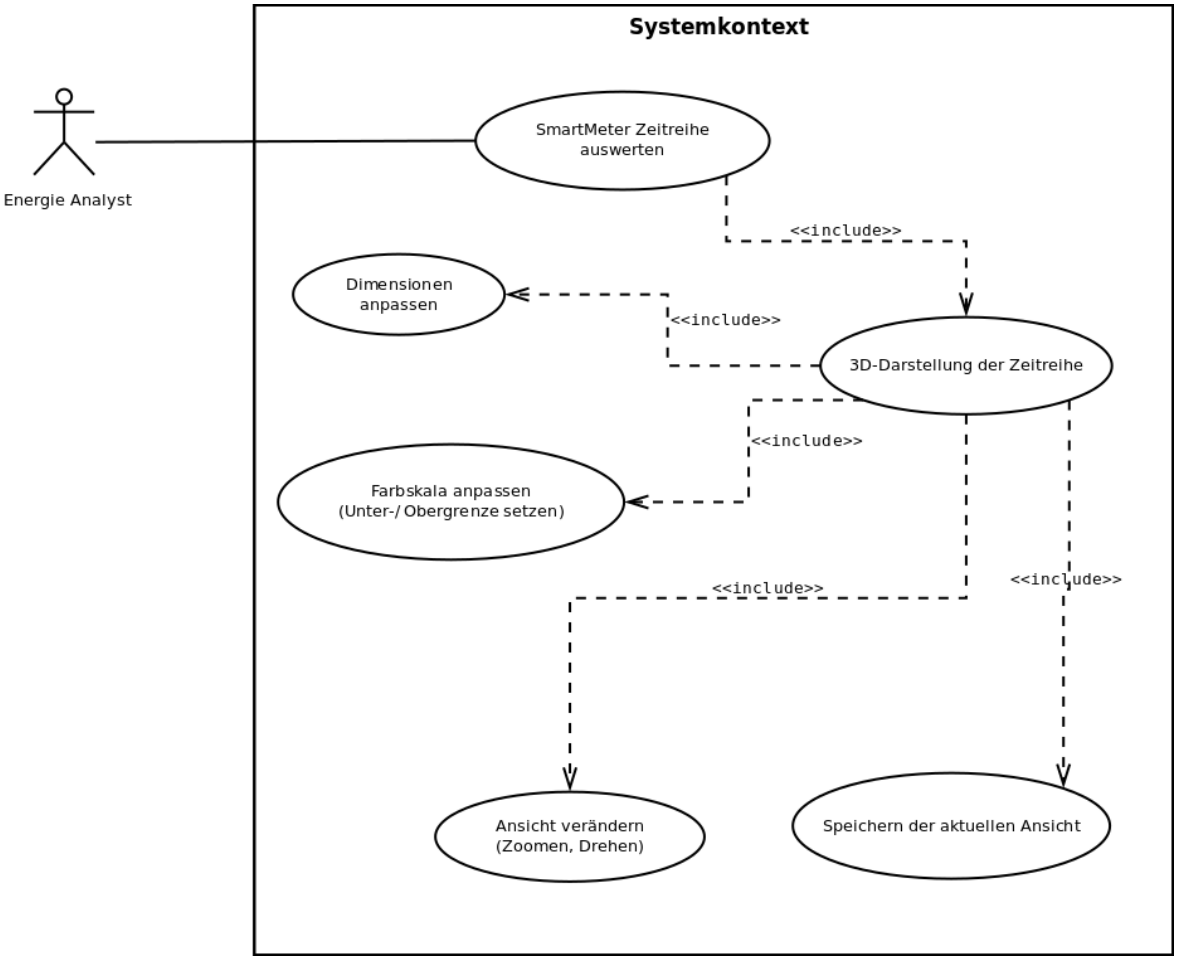

Abbildung 4: Anwendungsfall SmartMeter