

Entwurf

Praxis der Softwareentwicklung

Entwicklung einer Software zur Berechnung
der Mandatsverteilung im Deutschen
Bundestag

Gruppe 1

Philipp Löwer, Anton Mehlmann, Manuel Olk, Enes Ördek,
Simon Schürg, Nick Vlasoff



WS 2013 / 14

Inhaltsverzeichnis

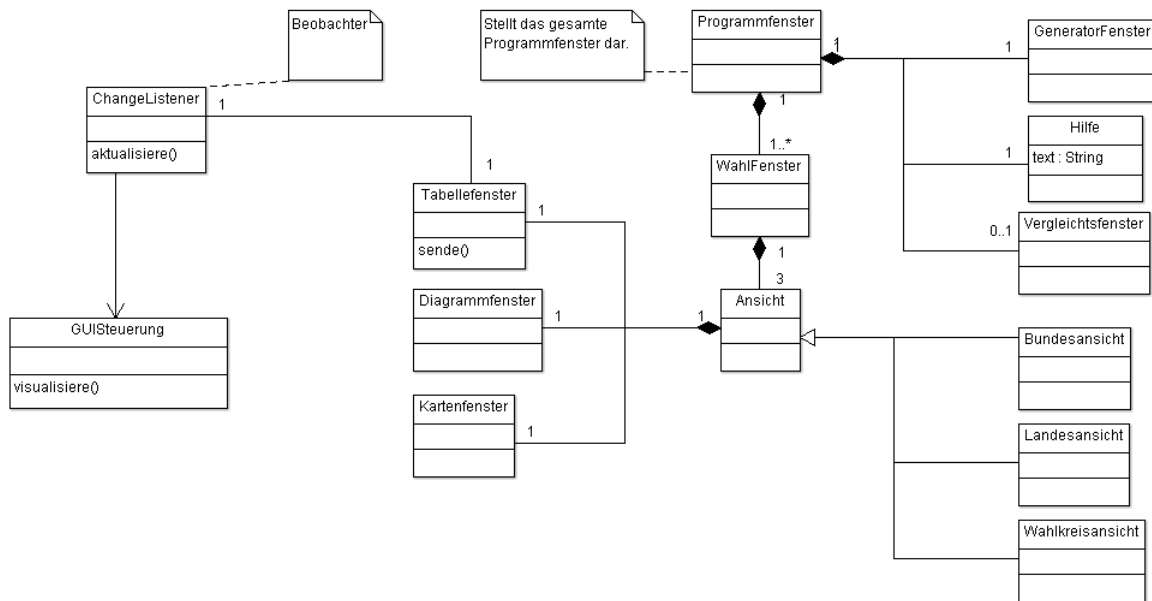
1	Einleitung	3
2	Systemmodell	3
3	Module	3
3.1	GUI	3
3.2	Datenhaltung	4
3.3	Import/ Export	5
3.4	Wahlgenerierung	6
3.5	Chronik	7
4	Anwendungsfälle	8
5	Sequenzdiagramme	8
5.1	Import	8
5.2	Export	9
5.3	Paradoxe Wahlgenerierung und Vergleich	10
5.4	Vergleich	11
5.5	Chronik	12

1 Einleitung

2 Systemmodell

3 Module

3.1 GUI



Dieser Teil des gesamten Klassendiagramms zeigt den Visualisierungsteil.

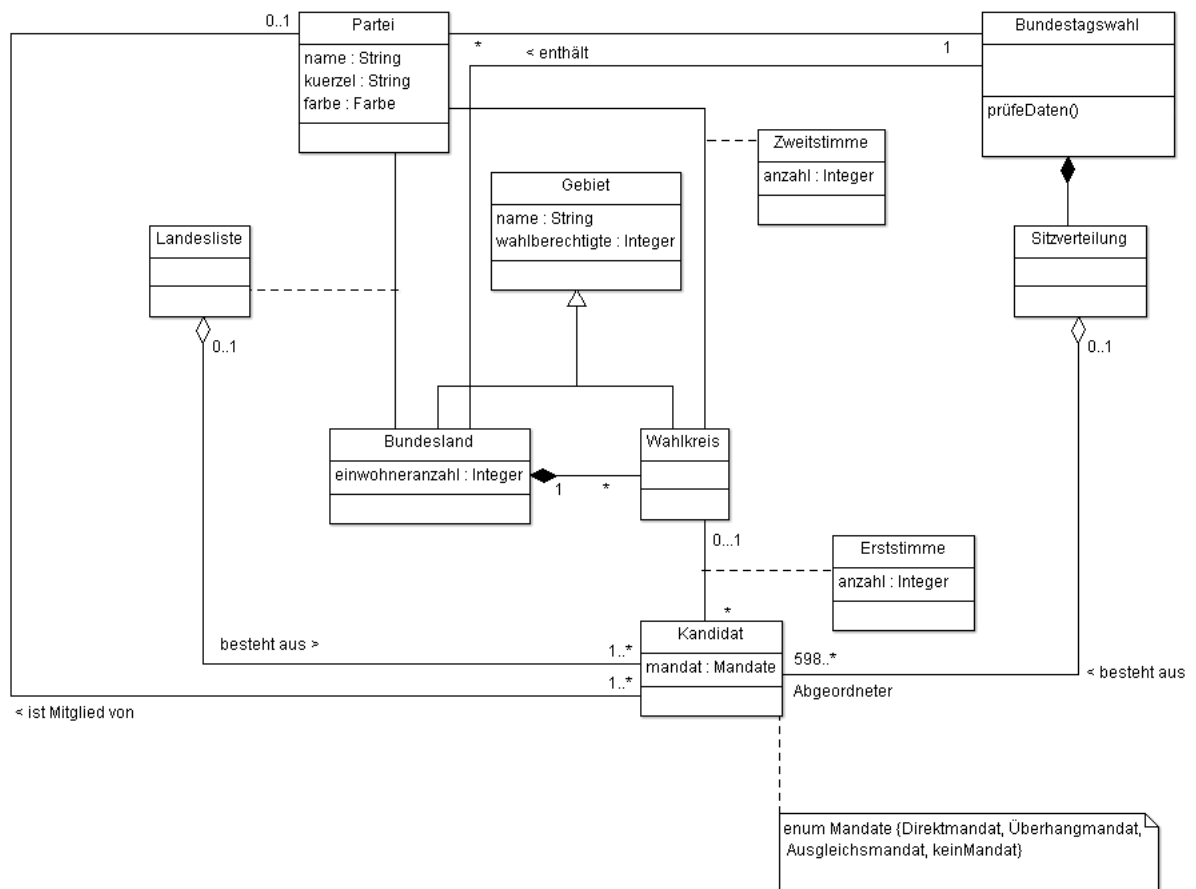
Das Programmfenster besteht aus mehreren Wahlfenstern in Form von Tabs. Weiterhin enthält es ein Generatorfenster, um nicht identifizierten Parteien Stimmen zu geben, die Hilfe, und ein Vergleichsfenster, wenn der Vergleich zweier Wahlen aktiv ist.

Ein Wahlfenster besteht aus genau drei Ansichtsfenstern, diese sind Bundes-, Landes- und Wahlkreisansicht. Zu jeder von diesen gehört jeweils ein Tabellen-, Diagramm- und Kartenfenster.

Das Tabellenfenster, in welchem Erst- und Zweitstimmen geändert werden können, und die dazugehörige ChangeListener-Klasse werden im Beobachter-Prinzip umgesetzt. Der ChangeListener hört das Tabellenfenster ab und wird bei Veränderungen informiert.

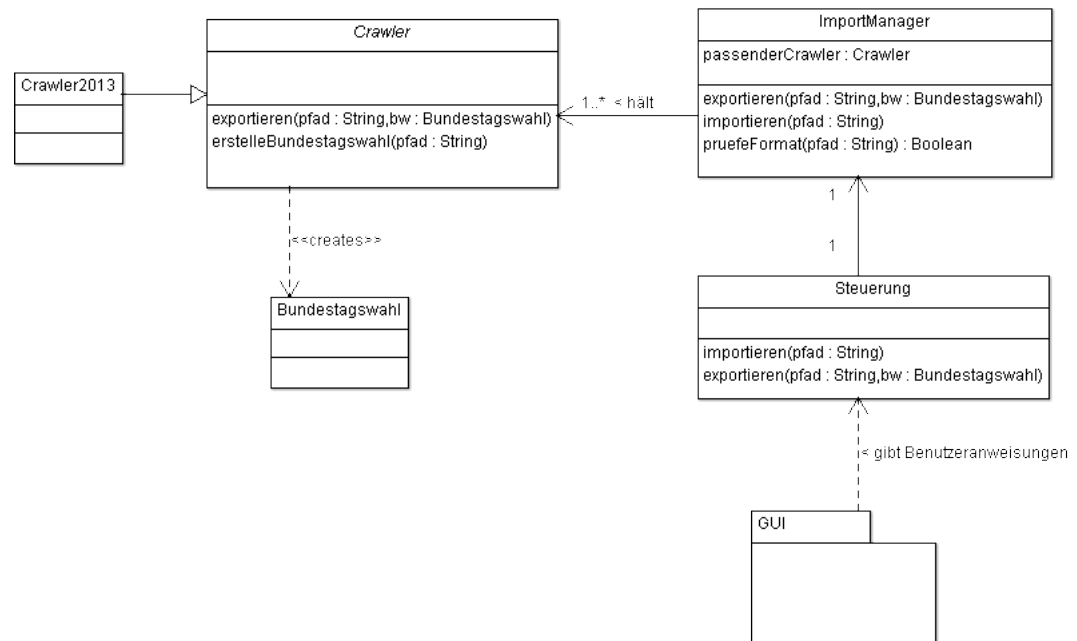
Gesteuert werden alle genannten Klassen durch die GUISteuerungs-Klasse, die alle Tabellen befüllt, Diagramme erstellt und die Kartenfenster füllt. Außerdem werden von ihr auch die Veränderungen, die der ChangeListener bekommt, verwaltet.

3.2 Datenhaltung



Dieser Ausschnitt des Klassendiagramms stellt den Datenhaltungsteil dar.

3.3 Import/ Export



Ausschnitt.png

In diesem Klassendiagramm sieht man den Aufbau des Import-/Exportmoduls. Zur Übersichtlichkeit werden die zum Importieren/Exportieren nicht notwendigen Methoden in *Steuerung* und die genaue Struktur hinter *Bundestagswahl* und der GUI ausgeblendet.

Mit unserem Programm wird nur ein vorimplementierter Crawler mitgegeben, der .csv-Dateien, die dem Format der .csv-Datei zur Bundestagswahl 2013 der Bundeswahlleiter-Webseite entsprechen, auswerten kann - dies ist *Crawler2013*. Um jedoch die Möglichkeit zu garantieren, nachträglich weitere Crawler hinzuzufügen, haben wir uns dafür entschieden, eine abstrakte Oberklasse *Crawler* zu verwenden, von der *Crawler2013* erbt, und alle vorhandenen Crawler von der Klasse *ImportManager* halten zu lassen. Ausgelöst wird der ganze Import- bzw. Exportvorgang durch eine Benutzerinteraktion (z.B. Betätigen des Laden- Knopfs im Menü), worauf die *GUISteuerung* die *Steuerung* anwählt, welche wiederum über den *ImportManager* den geeigneten Crawler bestimmt, der die Informationen ausliest die eigentliche Arbeit ausführt.

Methoden

exportieren(pfad : String, bw : Bundestagswahl)

Schreibt die Bundestagswahl bw in die .csv-Datei, deren Pfad in **pfad** gegeben ist.

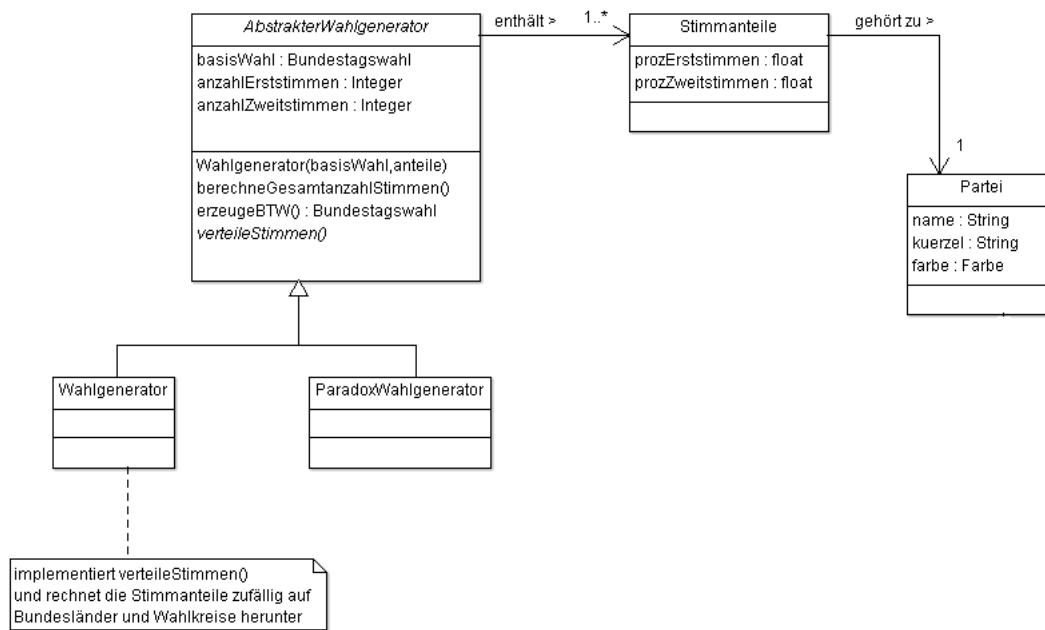
importieren(pfad : String)

Liest die .csv-Datei, deren Pfad in pfad gegeben ist, ein, erstellt ein Bundestagswahlobjekt und füllt es.

pruefeFormat(pfad : String) : Boolean

Prüft das Format der Datei, deren Pfad in pfad gegeben ist. Wenn es sich um eine .csv-Datei handelt, die durch einen vorhandenen Crawler erfasst werden kann, wird dieser Crawler als passenderCrawler gesetzt und true zurückgegeben. Falls kein geeigneter Crawler gefunden wird, da die Datei keine .csv-Datei ist oder ihre Struktur nicht korrekt ist, wird false zurückgegeben.

3.4 Wahlgenerierung

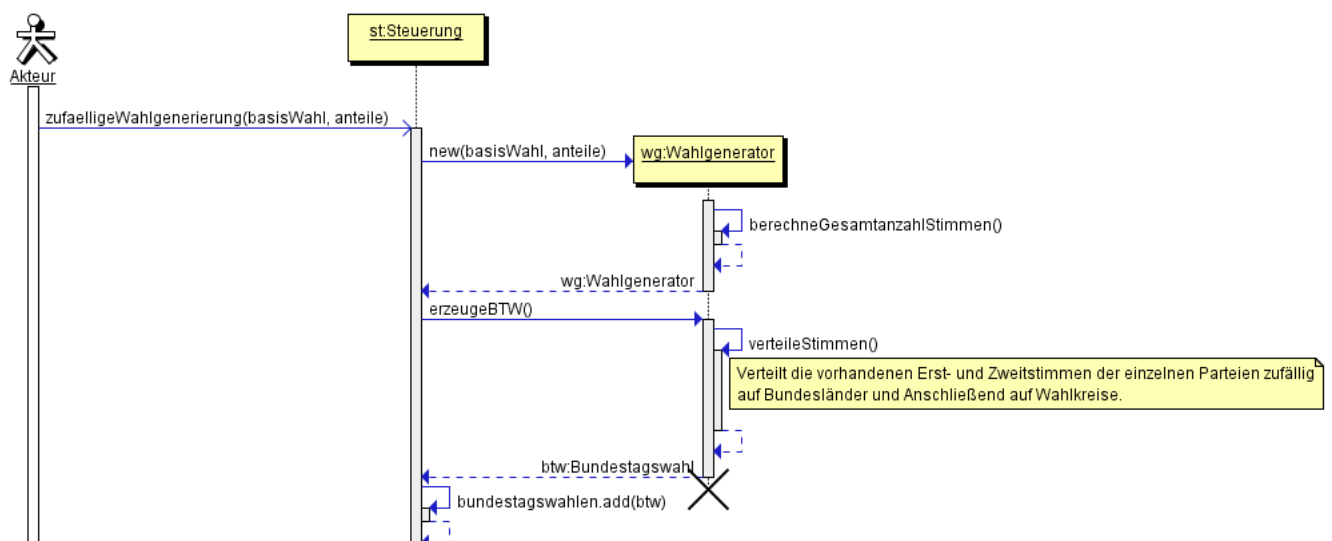


Der obige Ausschnitt des Klassendiagramms zeigt das Wahldatengenerierungs-Modul.

Mit diesem Modul können Bundestagswahlen anhand vorher definierten Stimmanteilen auf Bundesebene generiert werden. Bei diesen Stimmanteilen handelt es sich um eine Liste aller Parteien mit prozentualen Anteilen der Erst- und Zweitstimmen auf Bundesebene. Des weiteren benötigt der Wahlgenerator eine Basis-Bundestagswahl um Daten wie beispielsweise Bundesländer, Wahlkreise und Wahlberechtigte zur Verfügung zu haben.

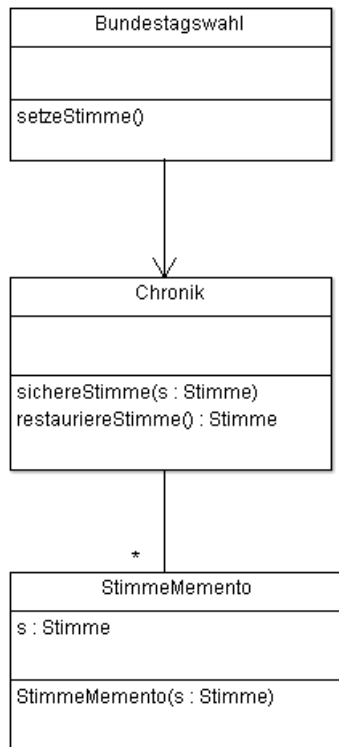
Beim generieren einer Bundestagswahl werden die Stimmanteile mithilfe der Anzahl aller Wahlberechtigten in absolute Zahlen für Erst- und Zweitstimmen umgerechnet. Diese Stimmen werden dann zufällig, erst auf alle Bundesländer und dann auf die einzelnen Wahlkreise verteilt.

Dieser Vorgang ist im folgenden noch einmal als Sequenzdiagramm aufgezeigt.



Neben der zufälligen Verteilung von den Stimmen auf alle Bundesländer und Wahlkreise gibt es noch einen Wahlgenerator der Stimmverteilungen erzeugt, die nach dem Wahlgesetz von 2009 zu negativem Stimmgewicht durch Überhangsmandate führen. Diese werden benötigt um in einer Vergleichsansicht das negative Stimmgewicht zu verdeutlichen.

3.5 Chronik

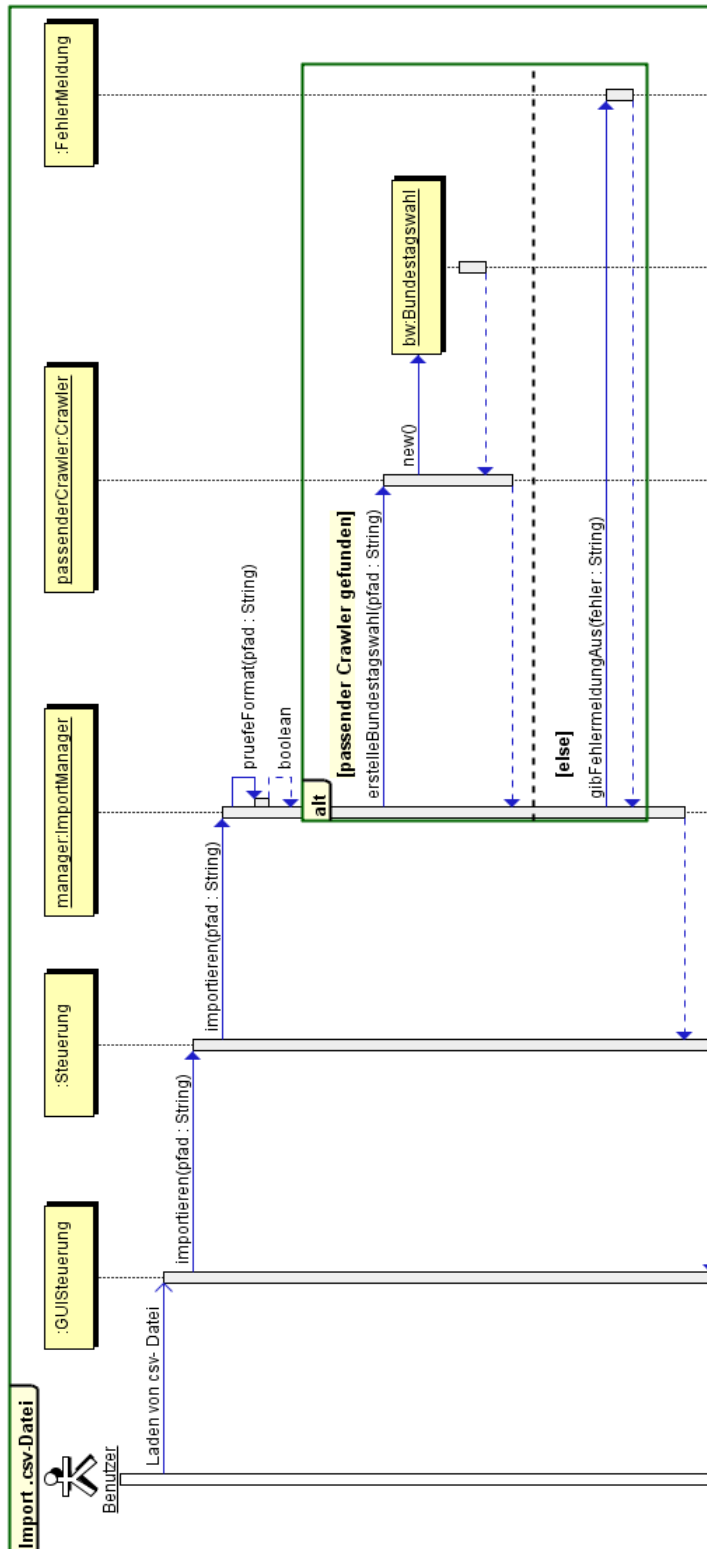


Die Klasse Chronik" gibt dem Programm die Funktionalität, Veränderungen an den Stimmen rückgängig zu machen. Jede "Bundestagswahl" hat hierbei eine eigene Chronik. Chronik" wird in dem Konstruktor von "Bundestagswahl" erzeugt, und ist daher in jedem "BundestagswahlObjekt" enthalten. Es besitzt eine Menge von SStimmenMementoObjekten. Bei jeder Veränderung wird ein neues SStimmenMementoObjekt angelegt, was die Veränderung widerspiegelt. Die Methode sichereStimme" wird von dem "BundestagswahlObjekt" bei jedem Aufruf von setzeStimme" aufgerufen.

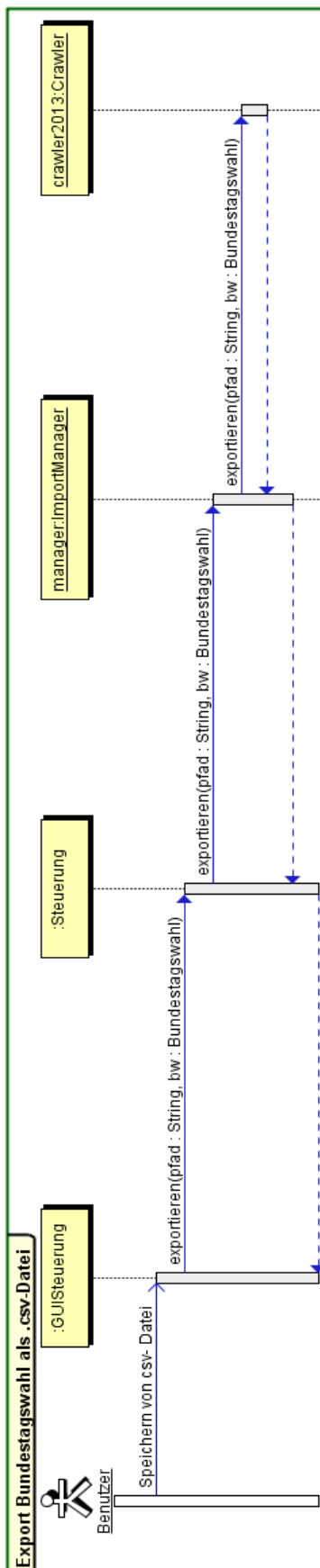
4 Anwendungsfälle

5 Sequenzdiagramme

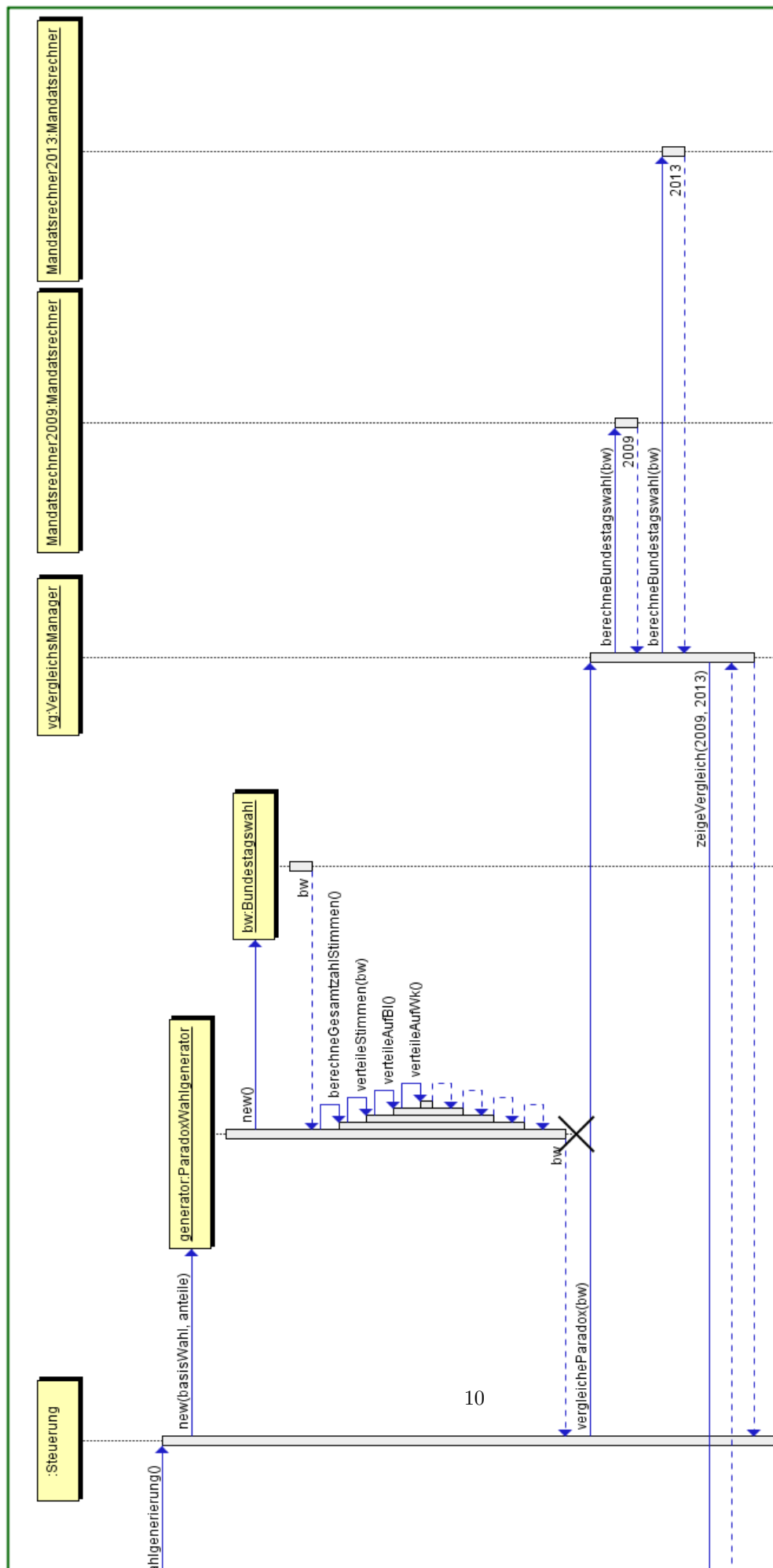
5.1 Import



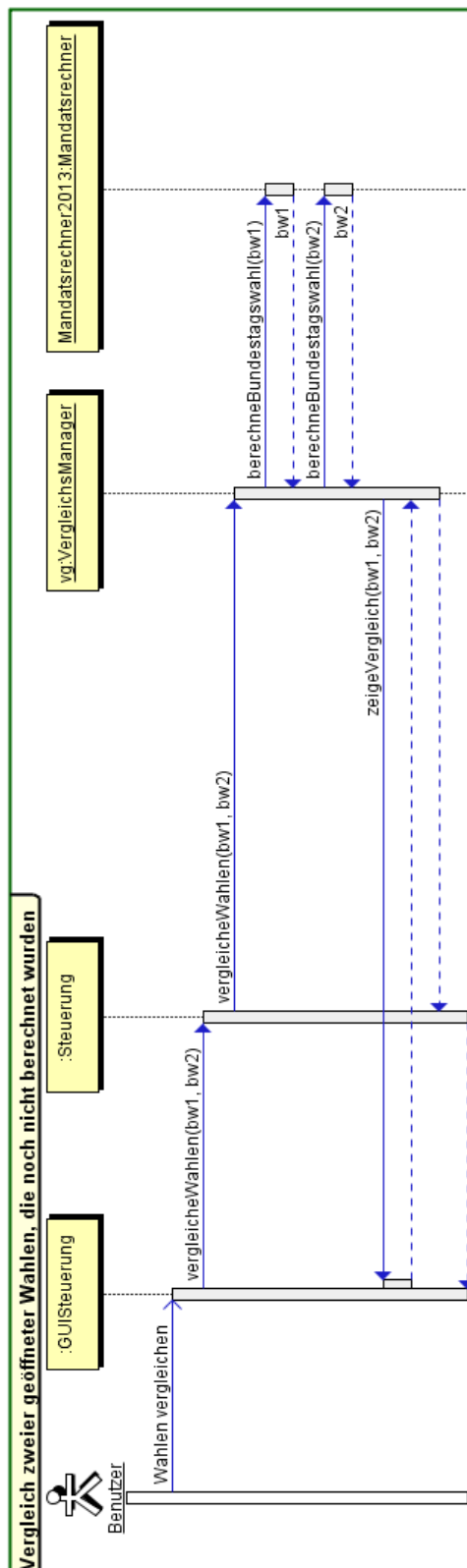
5.2 Export



5.3 Paradoxe Wahlgenerierung und Vergleich



5.4 Vergleich



In diesem Sequenzdiagramm wird gezeigt, was passiert, wenn der Benutzer zwei Tabs geöffnet hat und einen Vergleich starten will.

5.5 Chronik

