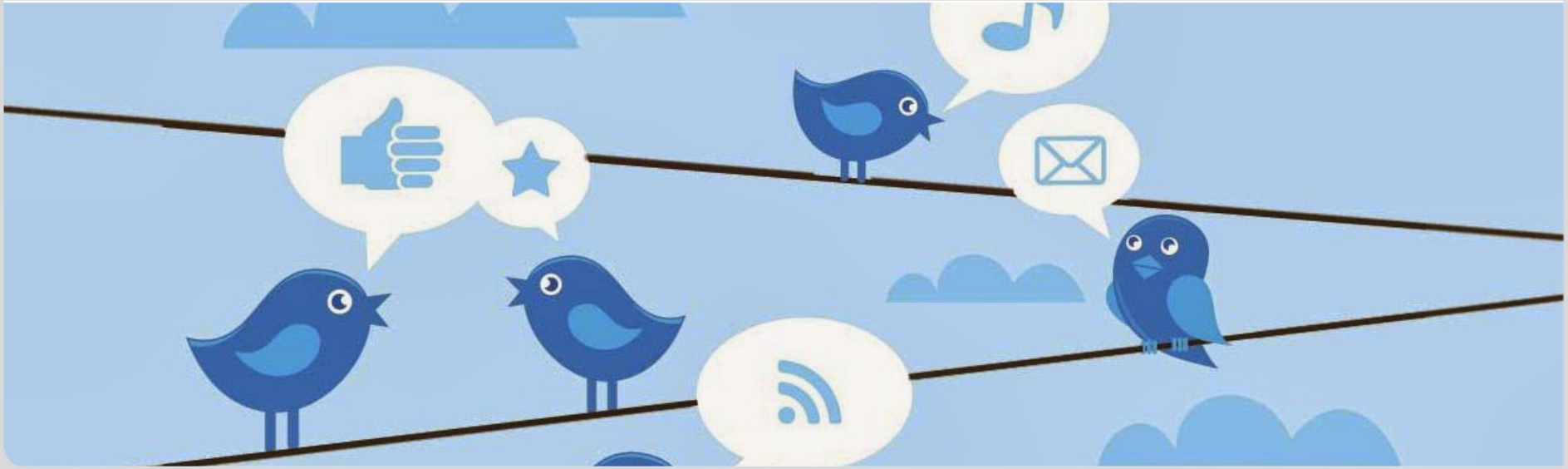
Karlsruher Institut für Technologie

Praxis der Software Entwicklung: Visualizing Trends. Was verrät uns Twitter?

Präsentation Implementierung, 5. Februar 2015



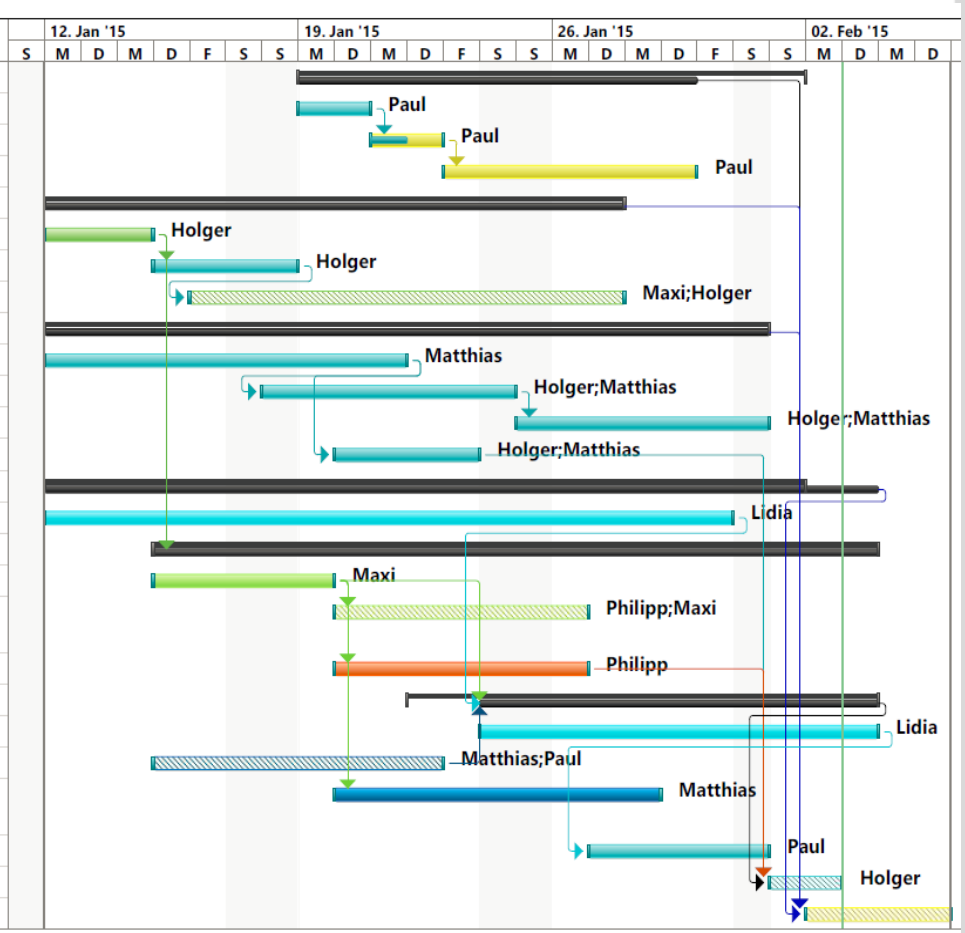
Maximilian Awiszus, Holger Ebhart, Lidia Grigoriev, Paul Jungeblut, Philipp Kern und Matthias Schimek

Struktur

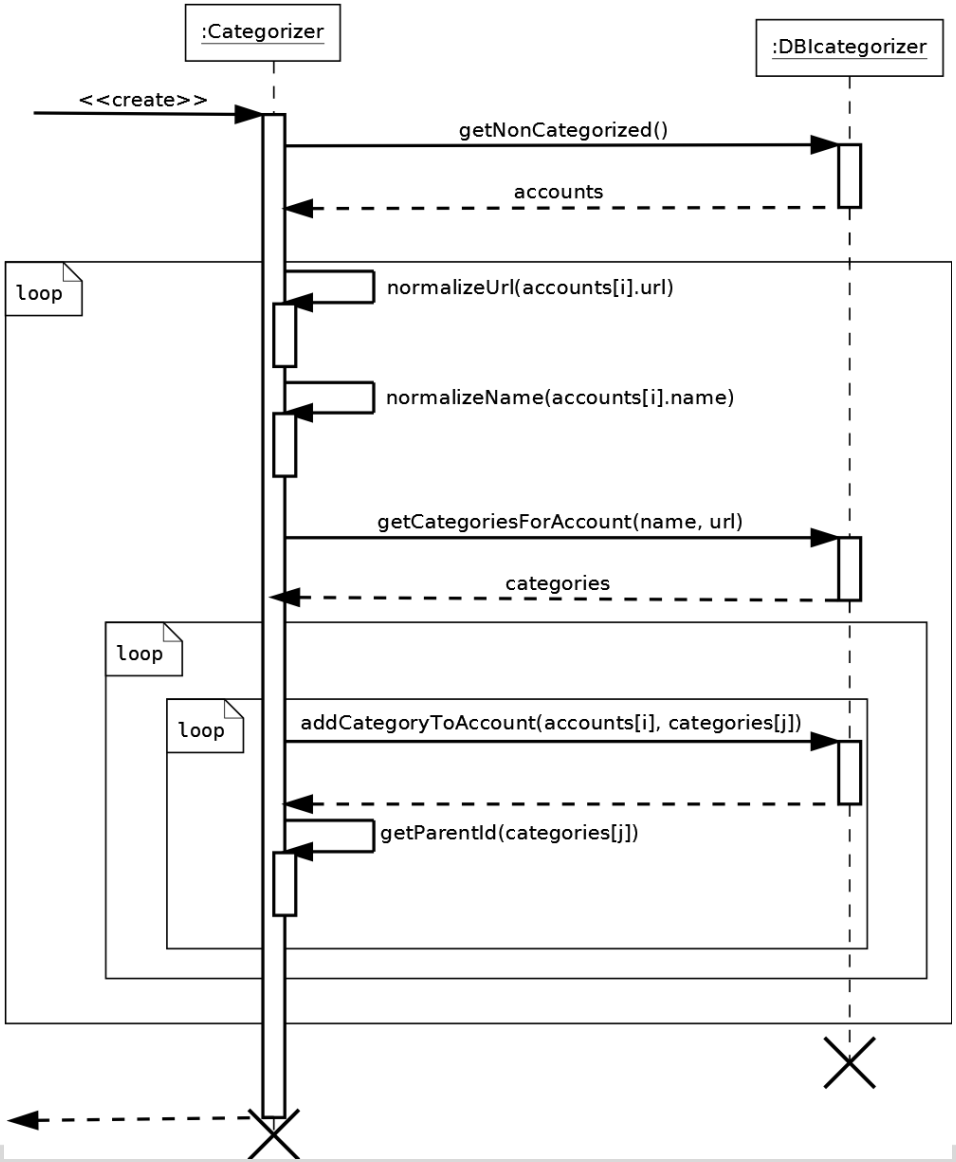
- Ablauf: Vergleich Planung – Umsetzung
- Änderungen am Entwurf
- Komponententests
- Fragen
- Vorführung

Ablauf der Implementierungsphase

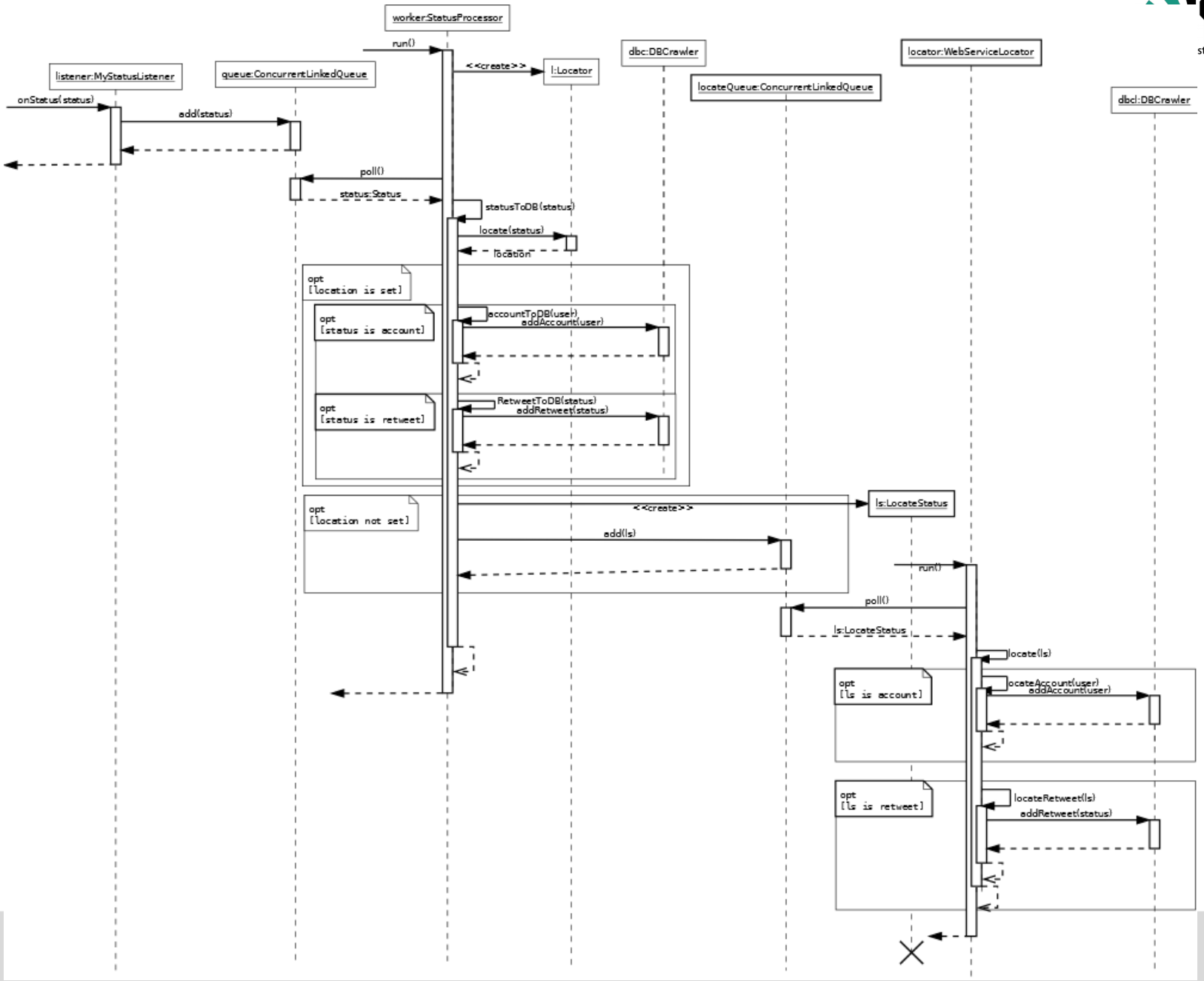
Vorgangsname	Dauer	Anfang	Fertig stellen	Ressourcennamen
Kategorisierer	11 Tage	Mo 19.01.15	So 01.02.15	Paul
DMOZ parsen	2 Tage	Mo 19.01.15	Di 20.01.15	Paul
Implementierung	2 Tage	Mi 21.01.15	Do 22.01.15	Paul
Hitrate erhöhen	5 Tage	Fr 23.01.15	Do 29.01.15	Paul
MySQL	12 Tage	Mo 12.01.15	Di 27.01.15	Holger
Interfaces + Stubs	3 Tage	Mo 12.01.15	Mi 14.01.15	Holger
MySQL-Queries	3 Tage	Do 15.01.15	So 18.01.15	Holger
Query Optimierung	8 Tage	Fr 16.01.15	Di 27.01.15	Maxi;Holger
Crawler	16 Tage	Mo 12.01.15	Sa 31.01.15	
Locator	8 Tage	Mo 12.01.15	Mi 21.01.15	Matthias
Bug-Fixing	7 Tage	So 18.01.15	Sa 24.01.15	Holger;Matthias
Zuverlässigkeit	7 Tage	So 25.01.15	Sa 31.01.15	Holger;Matthias
Statistik	4 Tage	Di 20.01.15	Fr 23.01.15	Holger;Matthias
GUI	16 Tage	Mo 12.01.15	So 01.02.15	
unfolding	15 Tage	Mo 12.01.15	Fr 30.01.15	Lidia
gui-package-Implementierung	15 Tage	Do 15.01.15	Di 03.02.15	
GUIController Implementierung	3 Tage	Do 15.01.15	Mo 19.01.15	Maxi
selectionOfQuery-Paket Implementierung	5 Tage	Di 20.01.15	Mo 26.01.15	Philipp;Maxi
table	5 Tage	Di 20.01.15	Mo 26.01.15	Philipp
standardMap	10 Tage	Do 22.01.15	Di 03.02.15	
Karte Anzeige	9 Tage	Sa 24.01.15	Di 03.02.15	Lidia
Berechnungsformel	6 Tage	Do 15.01.15	Do 22.01.15	Matthias;Paul
databaseOptions-Paket Implementierung	7 Tage	Di 20.01.15	Mi 28.01.15	Matthias
Hover-Effekt	5 Tage	Di 27.01.15	Sa 31.01.15	Paul
Exportfunktion	2 Tage	So 01.02.15	Mo 02.02.15	Holger
Implementierungsdokument erstellen	4 Tage	Mo 02.02.15	Do 05.02.15	Matthias;Paul



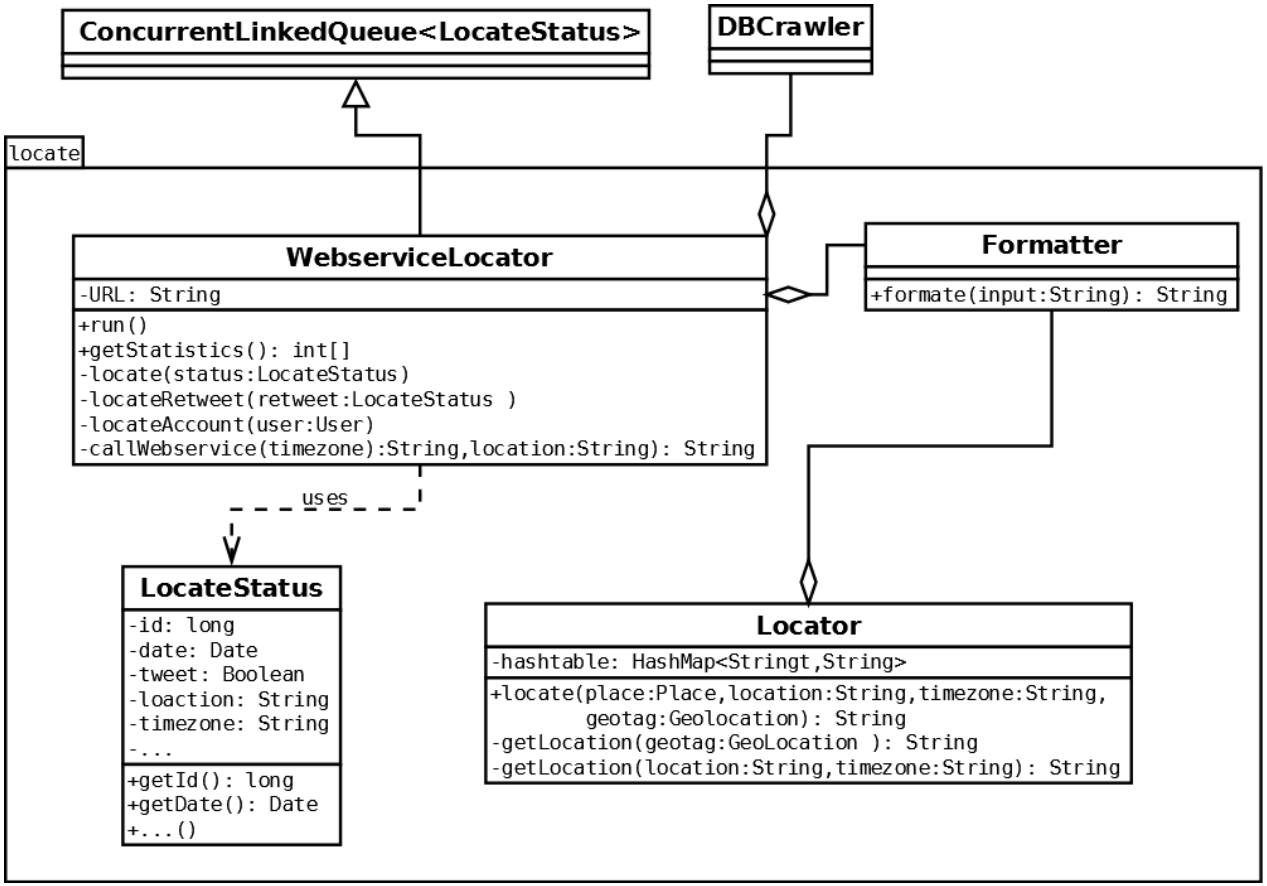
Änderung des Kategorisierers



Änderungen am Crawler



Locator


ConcurrentLinkedQueue<LocateStatus>
DBCrawler
locate
WebserviceLocator
-URL: String
+run()
+getStatistics(): int[]
-locate(status:LocateStatus)
-locateRetweet(retweet:LocateStatus)
-locateAccount(user:User)
-callWebservice(timezone):String,location:String): String
Formatter
+formate(input:String): String
uses
LocateStatus
-id: long
-date: Date
-tweet: Boolean
-loaction: String
-timezone: String
-...
+getId(): long
+getDate(): Date
+...()
Locator
-hashtable: HashMap<Stringt,String>
+locate(place:Place,location:String,timezone:String,
geotag:Geolocation): String
-getLocation(geotag:GeoLocation): String
-getLocation(location:String,timezone:String): String

GUI: View-Klassen - FXML

- Trennung von Darstellung und Programmlogik
- Visuelle Komponenten werden durch FXML-Dateien beschrieben
- Logisch gegliedert in einige kleine FXMLs

```
:tabs>
  <Tab fx:id="tab_Acc_tab1" text="Account hinzufügen">
    <content>
      <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="180.0" prefWidth="200.0">
            <children>
              <TextField fx:id="txtField_Acc_tab1" layoutX="14.0" layoutY="60.0" />
              <Label fx:id="l_Acc_searchAccount" layoutX="14.0" layoutY="23.0" text="Suche Account" />
              <ListView fx:id="list_Acc_tab1" layoutX="300.0" layoutY="60.0" prefHeight="256.0" prefWidth="272.0" />
              <Label fx:id="l_Acc_twitterAccounts" layoutX="300.0" layoutY="23.0" text="Twitter Accounts" />
              <Button fx:id="b_Acc_tab1_schliessen" layoutX="507.0" layoutY="316.0" mnemonicParsing="false" text="Weiter" />
              <Button fx:id="b_Acc_tab1_hinzufuegen" layoutX="395.0" layoutY="316.0" mnemonicParsing="false" text="hinzufügen" />
              <Button fx:id="b_Acc_tab1_suchen" layoutX="14.0" layoutY="106.0" mnemonicParsing="false" text="Suchen" />
            </children></AnchorPane>
    </content>
  </Tab>
:/tabs>
:abPane>
```

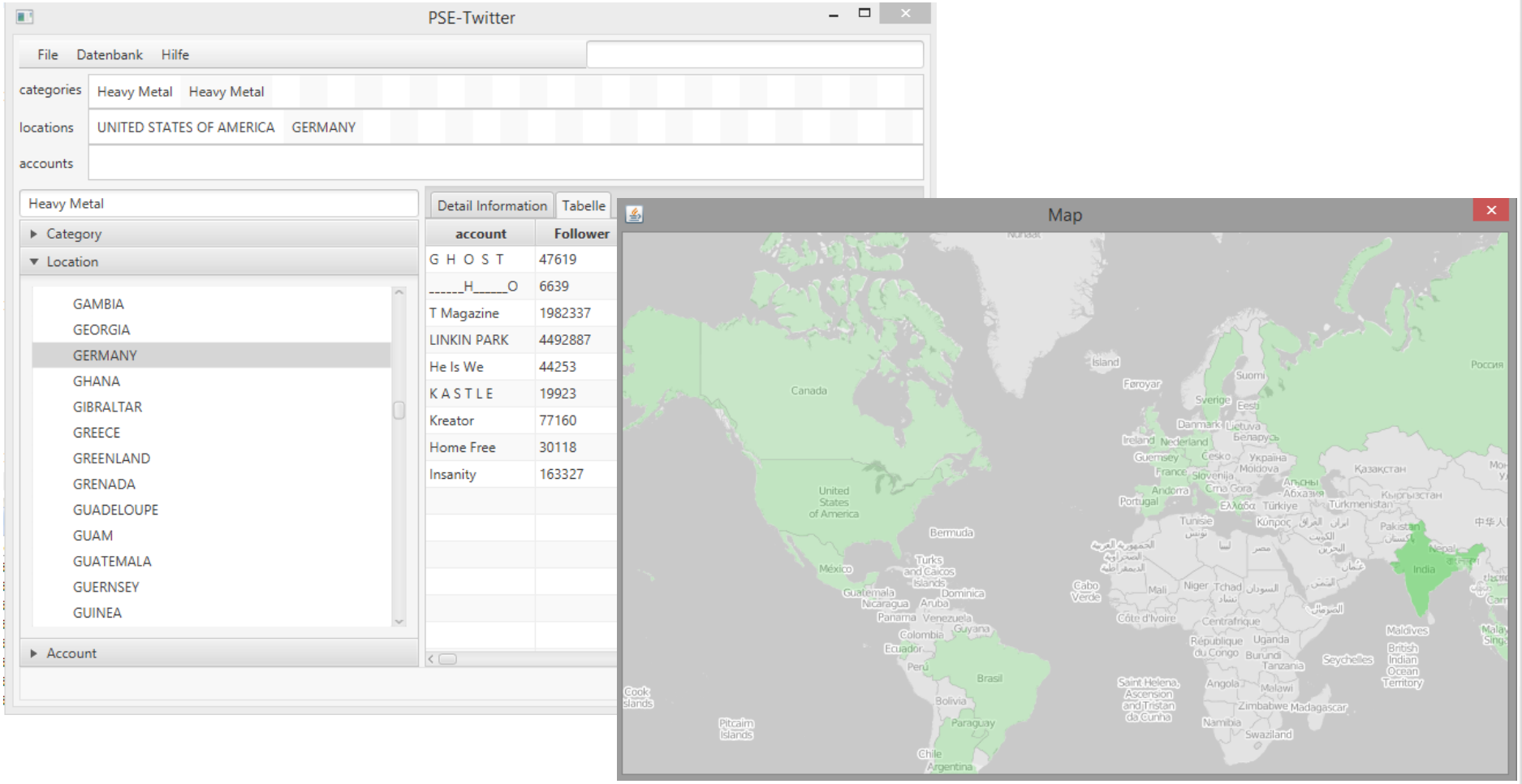
Paket „gui“

- GUIController
  - Einige Methoden wurden hinzugefügt
  - z.B. `getSelectedAccounts()`, `getCategoryRoot(int[])`
- SelectionHashList<T>
  - Effiziente Anfrage, ob Accounts/Kategorien zur aktuellen Abfrage gehören
- Labels
  - Sprachflexibilität
- RunnableParameter
  - Java-Runnables, denen ein Parameter übergeben werden kann
- InfoRunnable
  - Statusnachrichten unten, verwendet RunnableParameter

Unterpaket „databaseOpts“

- Geringfügige Änderungen
  - Private Methoden, z.B. `buildTree()`
- Event Handler
  - Innere Klassen, zusätzlich zum Entwurf
  - `MyActionEventHandler`
  - `MyLocEventHandler`
  - `MyAccEventHandler`
  - `MyCatEventHandler`

Änderung der Karte



Paket „standardMap“

- Inkompatibilität von „unfolding“ mit JavaFX
- Neue Klasse `StandardMapDialog` implementiert `JFrame`
- Eine Karte für StandardMap und TimeSliderMap
  - StandardMap nimmt implizit maximalen Zeitraum
  - TimeSlider wurde zu DatePickern
  - Ohne große Änderungen am Entwurf möglich

Unterpaket „unfolding“

- „unfolding“ Map als Singleton implementiert
  - Alle Kartendarstellungen benötigen nur eine Map
  - „unfolding“ weiß nichts von verschiedenen Berechnungen im `GUIController`
- Detailinformationen per Mausklick
  - Anzahl der Retweets in diesem Land insgesamt
  - Anzahl der Retweets in diesem Land zur gewählten Kombination

Änderungen der Datenbank

- Tabelle `wordLocation` zum Speichern der HashTable zur Lokalisierung

- Package MySQL
  - Einige weitere Abfrage- und Änderungsmethoden
  - z.B. `addLocationString(String, String)`, `setCategorized(int)`

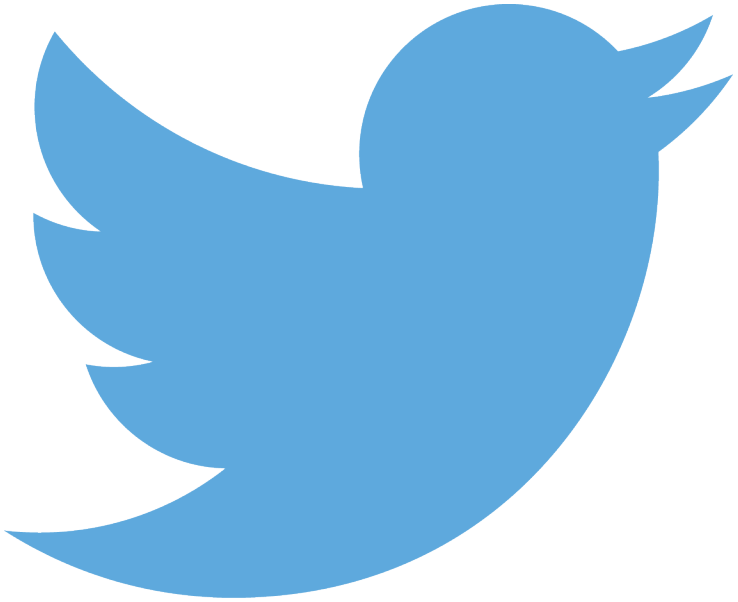
Tests

- GUI
  - Junit-Tests für den `GUIController`
  - z.B. `testGetCategories()`, `testGetAccounts()`
- In Unterpaketen von „gui“: Tests von Hand
  - Hauptsächlich Abfangen der richtigen Events
- Crawler
  - Protokollieren des Programmablaufs
- MySQL
  - JUnit-Tests für `DBcrawler`, `DBcategorizer`, `DBgui`

Beispieltest – Retweet einfügen

```
/**
 * test to add a retweet
 */
@Test
public void test2AddRetweet() {
    boolean res1 = dbc.addDay(date);
    boolean[] res2 = dbc.addRetweet(9999, null, date);
    cleaner.sql("DELETE FROM retweets WHERE 1;");
    cleaner.sql("DELETE FROM day WHERE day = \"" +
        dateFormat.format(date) + "\";");
    assertTrue(res1);
    assertTrue(res2[0]);
    assertTrue(!res2[1]);
}
```

Fragen



Fragen?