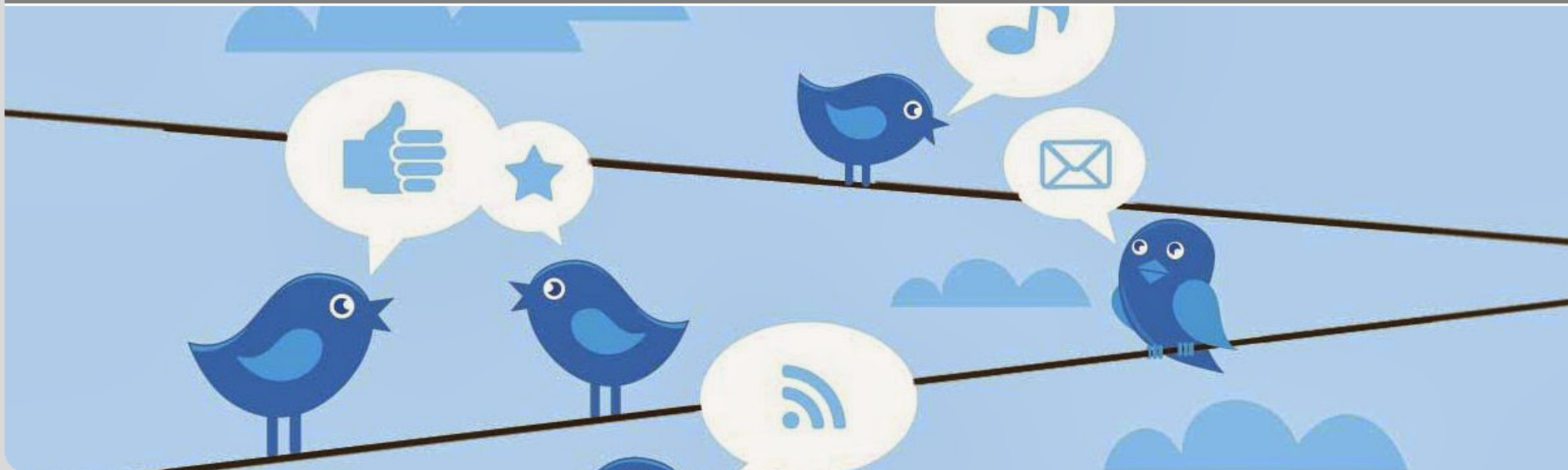


Praxis der Software Entwicklung: Visualizing Trends. Was verrät uns Twitter?

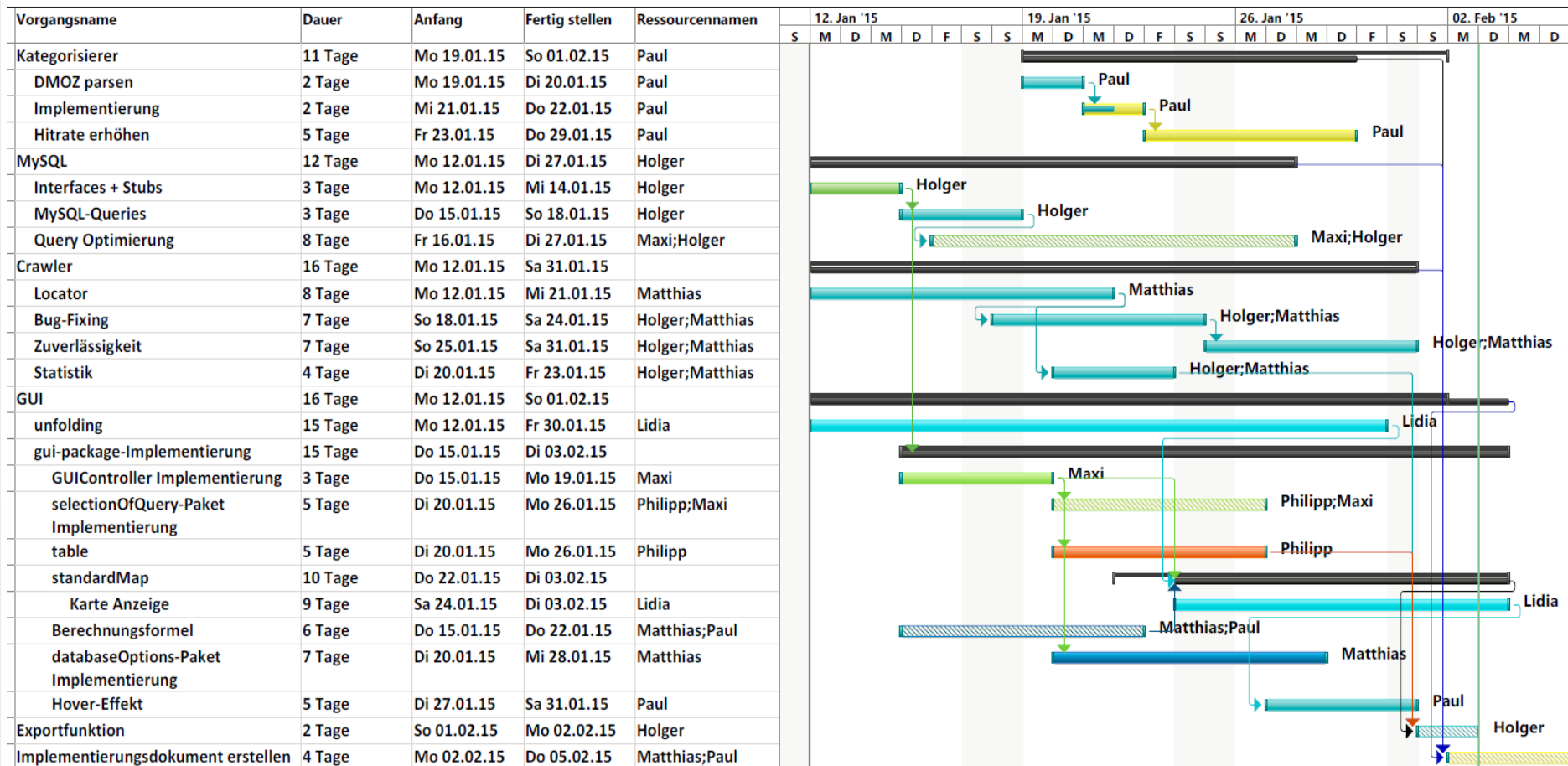
Präsentation Implementierung, 5. Februar 2015



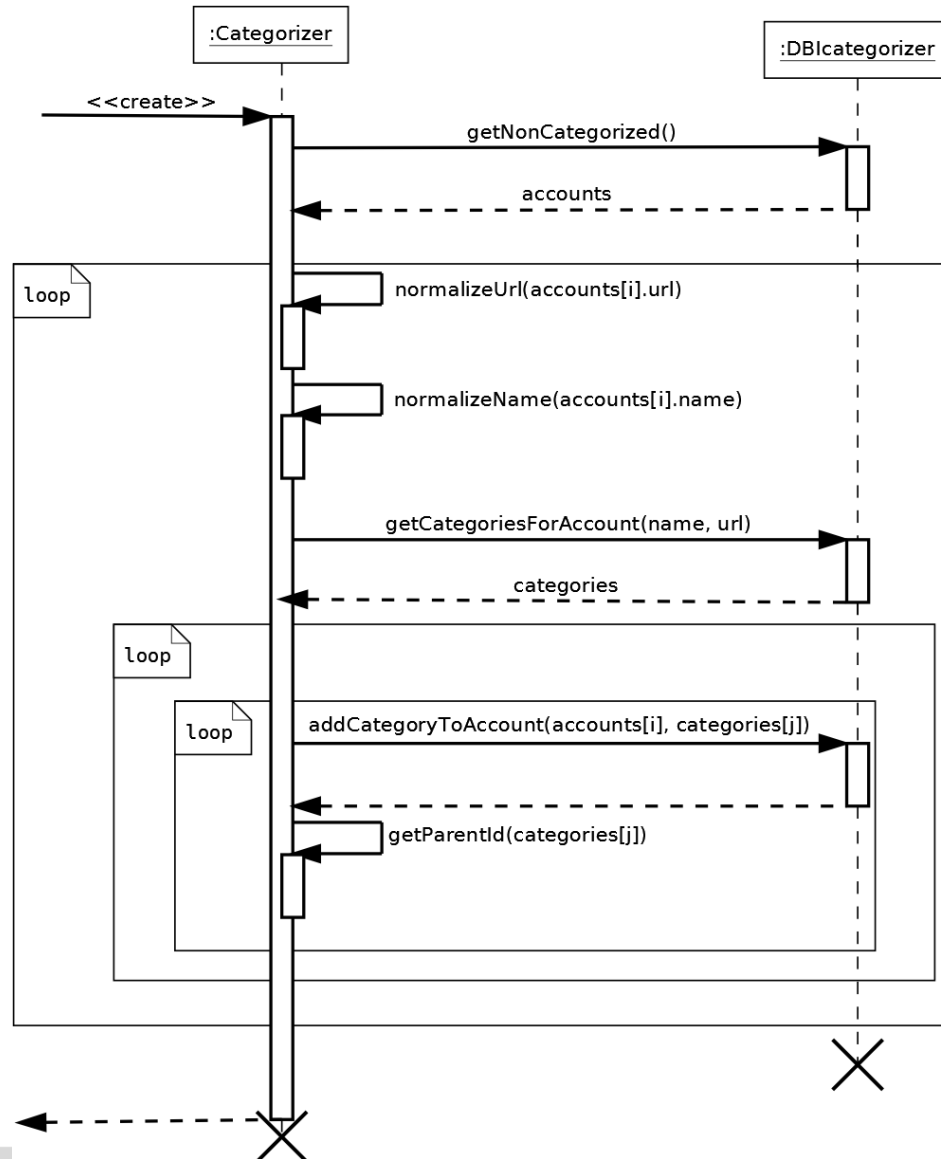
Struktur

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

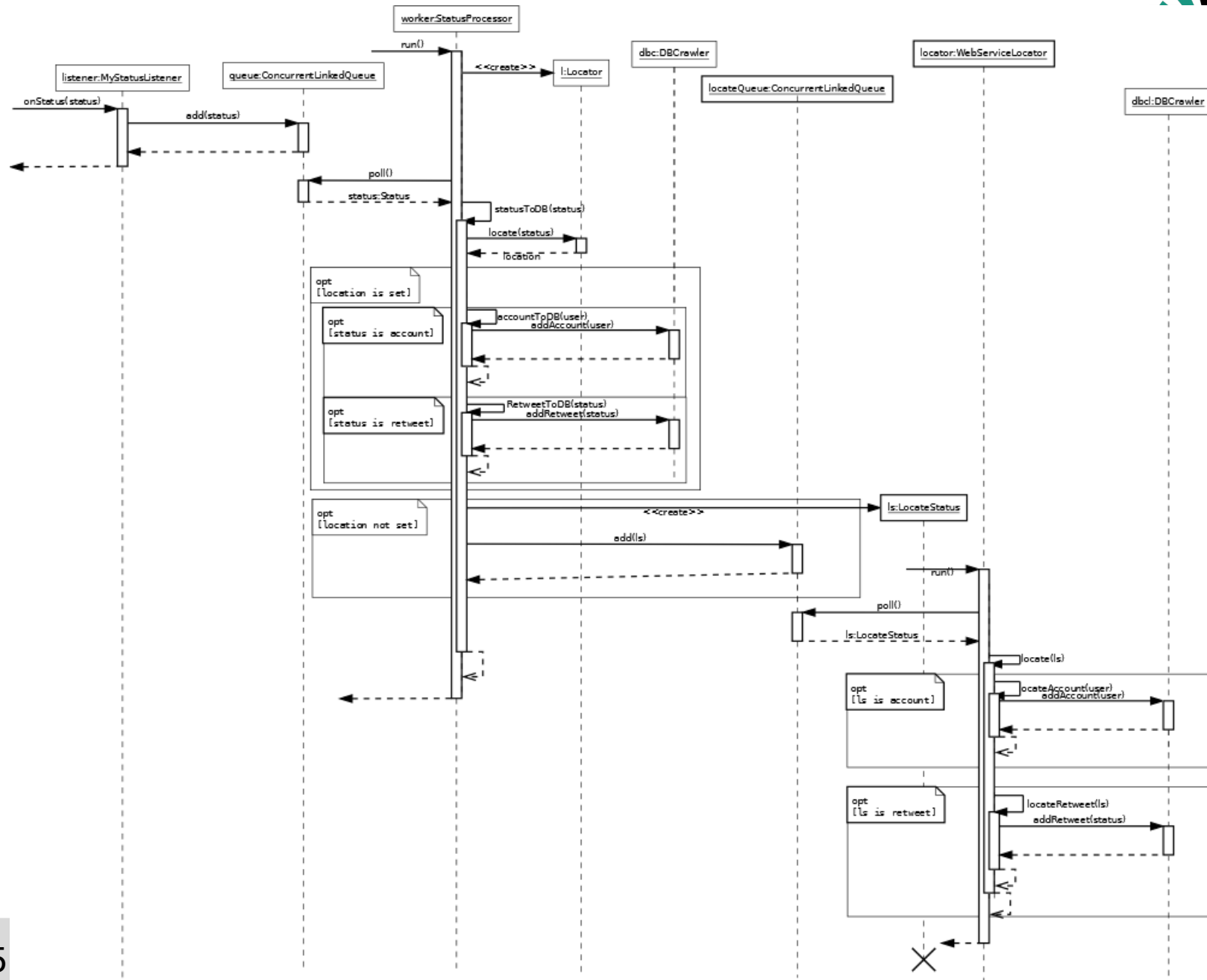
Ablauf der Implementierungsphase



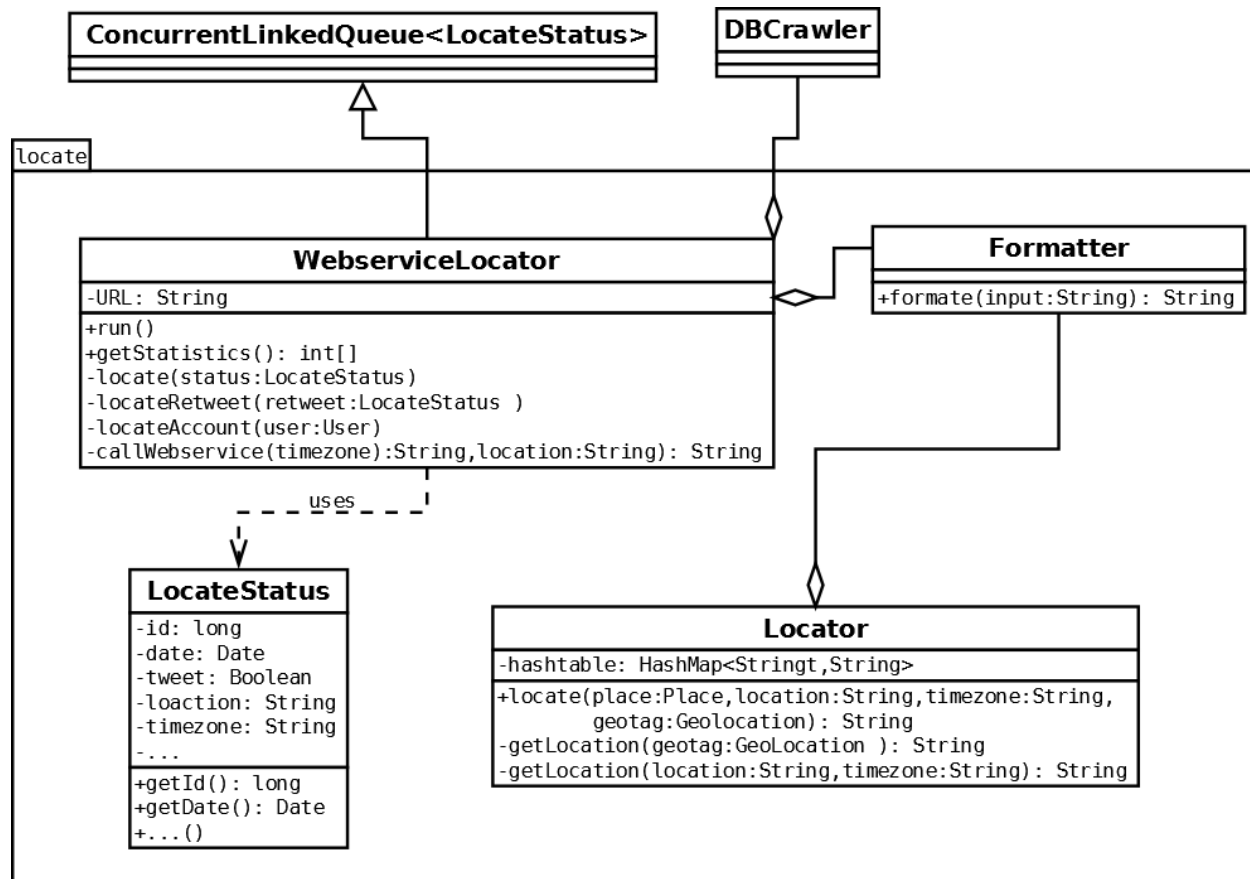
Änderung des Kategorisierers



Änderungen am Crawler



Locator



- 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>
</tabPane>
```

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

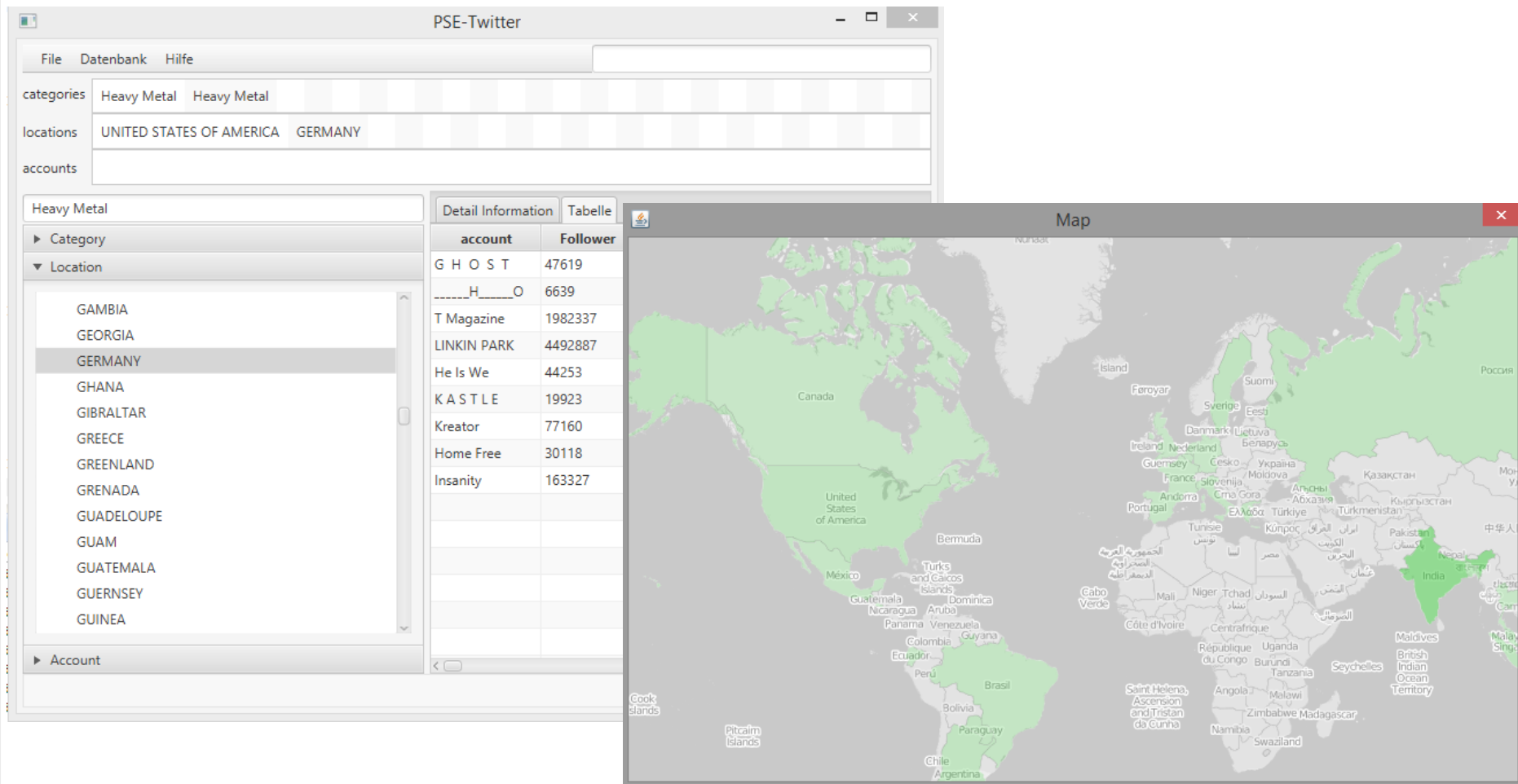
■ InfoRunnable

- Statusnachrichten unten, verwendet RunnableParameter

Unterpaket „databaseOpts“

- Geringfügige Änderungen
 - Private Methoden, z.B. `buildTree()`
- Action 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

- Tabelle `wordLocation` zum Speichern der HashTable zur Lokalisierung
- Package MySQL
 - Einige weitere Abfrage- und Änderungsmethoden
 - z.B. `addLocationString(String, String)`,
`setCategorized(int)`

■ GUI

- Junit-Tests für den `GUIController`
- z.B. `testGetCategories()`, `testGetAccounts()`

■ In Unterpaketen von „gui“: Tests von Hand

- Hauptsächlich Abfangen der richtigen Events

■ Crawler

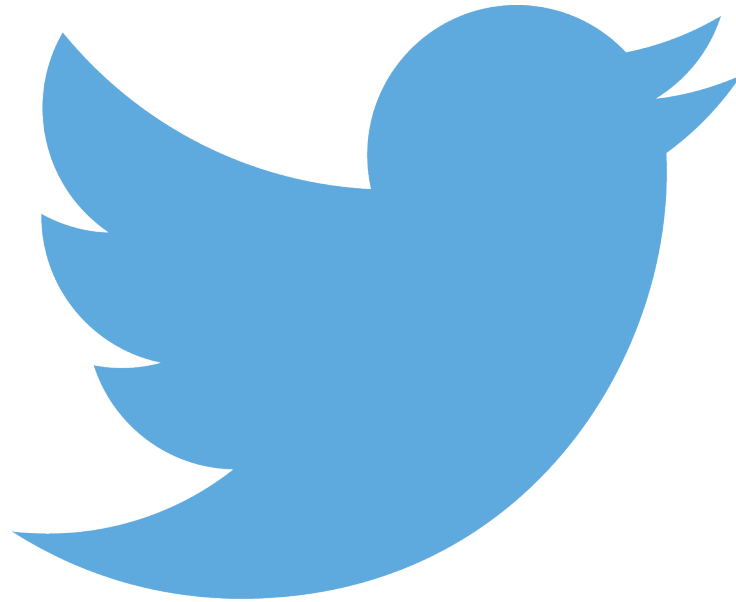
- Protokollieren des Programmablaufs

■ 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?