ExPort: Detecting and Visualizing API Usages in Large Source Code Repositories

Evan Moritz, Mario Linares-Vásquez,
Denys Poshyvanyk, Mark Grechanik,
Collin McMillan, Malcom Gethers

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
            PreparedStatement updateSales = null;
45
46          String updateString =
                                                  here COF_NAME = ?";


                                                  tement) con.prepareStatement(updateString);

                                                  eger> e : salesForWeek.entrySet()) {
                                                  e.getValue().intValue());
                                                  2, e.getKey());
                                                  ate();


60          }
61      }
```

How should I update a table using JDBC ?

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
44          PreparedStatement updateSales = null;
45
46          String updateString =
47                  "update COFFEES "
48                  + "set SALES = ? where COF_NAME = ?";
49
50
51          con.setAutoCommit(false);
52          updateSales = (PreparedStatement) con.prepareStatement(updateString);
53
54          for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
55              updateSales.setInt(1, e.getValue().intValue());
56              updateSales.setString(2, e.getKey());
57              updateSales.executeUpdate();
58
59              con.commit();
60          }
61      }
```

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
44          PreparedStatement updateSales = null;
45
46          String updateString =
47                  "update COFFEES "
48                  + "set SALES = ? where COF_NAME = ?";
49
50
51          con.setAutoCommit(false);
52          updateSales = (PreparedStatement) con.prepareStatement(updateString);
53
54          for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
55              updateSales.setInt(1, e.getValue().intValue());
56              updateSales.setString(2, e.getKey());
57              updateSales.executeUpdate();
58
59              con.commit();
60          }
61      }
```

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
            PreparedStatement updateSales = null;
45
            String updateString =
                    "update COFFEES "
                    + "set SALES = ? where COF_NAME = ?";
49
50
            con.setAutoCommit(false);
51          updateSales = (PreparedStatement) con.prepare
52
53
            for (Map.Entry<String, Integer> e : salesForW
54              updateSales.setInt(1, e.getValue().intVal
55              updateSales.setString(2, e.getKey());
56              updateSales.executeUpdate();
57
58              con.commit();
59          }
60      }
61
```

How should I initialize the Connection?

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

Limitations of existing approaches

Alternative uses

Dependencies

High-level functionality

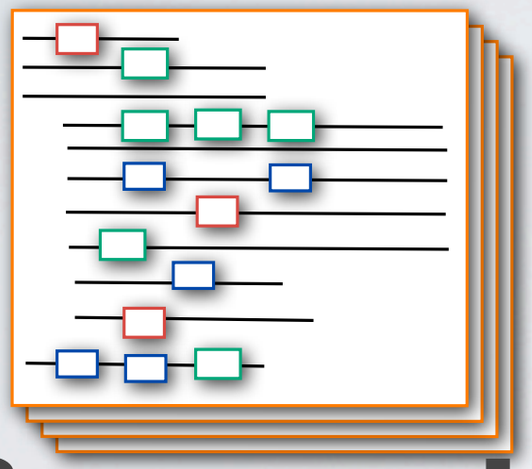
Single files

ExPort:

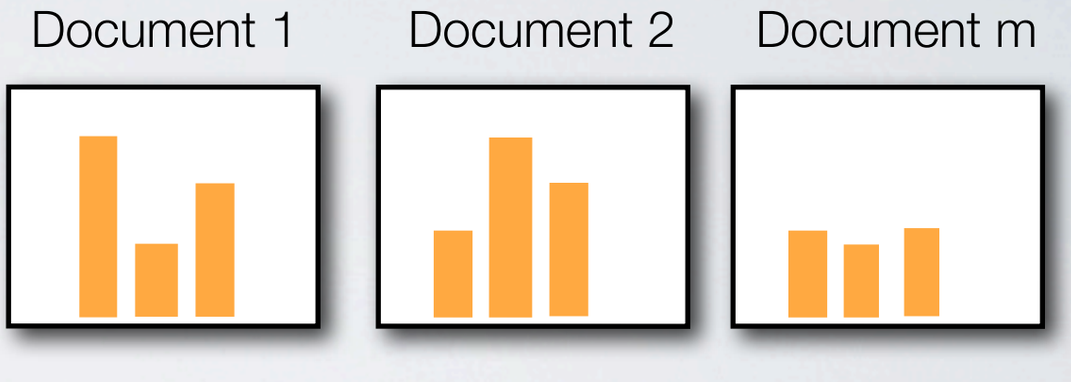
Detecting API usages in several files/ functions by using Relational Topic Models (RTM)

Latent Dirichlet Allocation

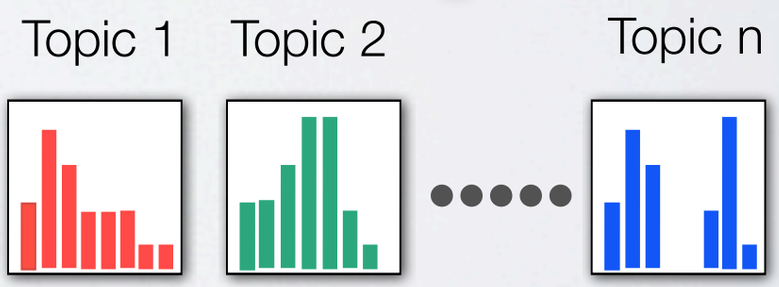
[Blei *et al.* 03]



Source code



Words x Documents

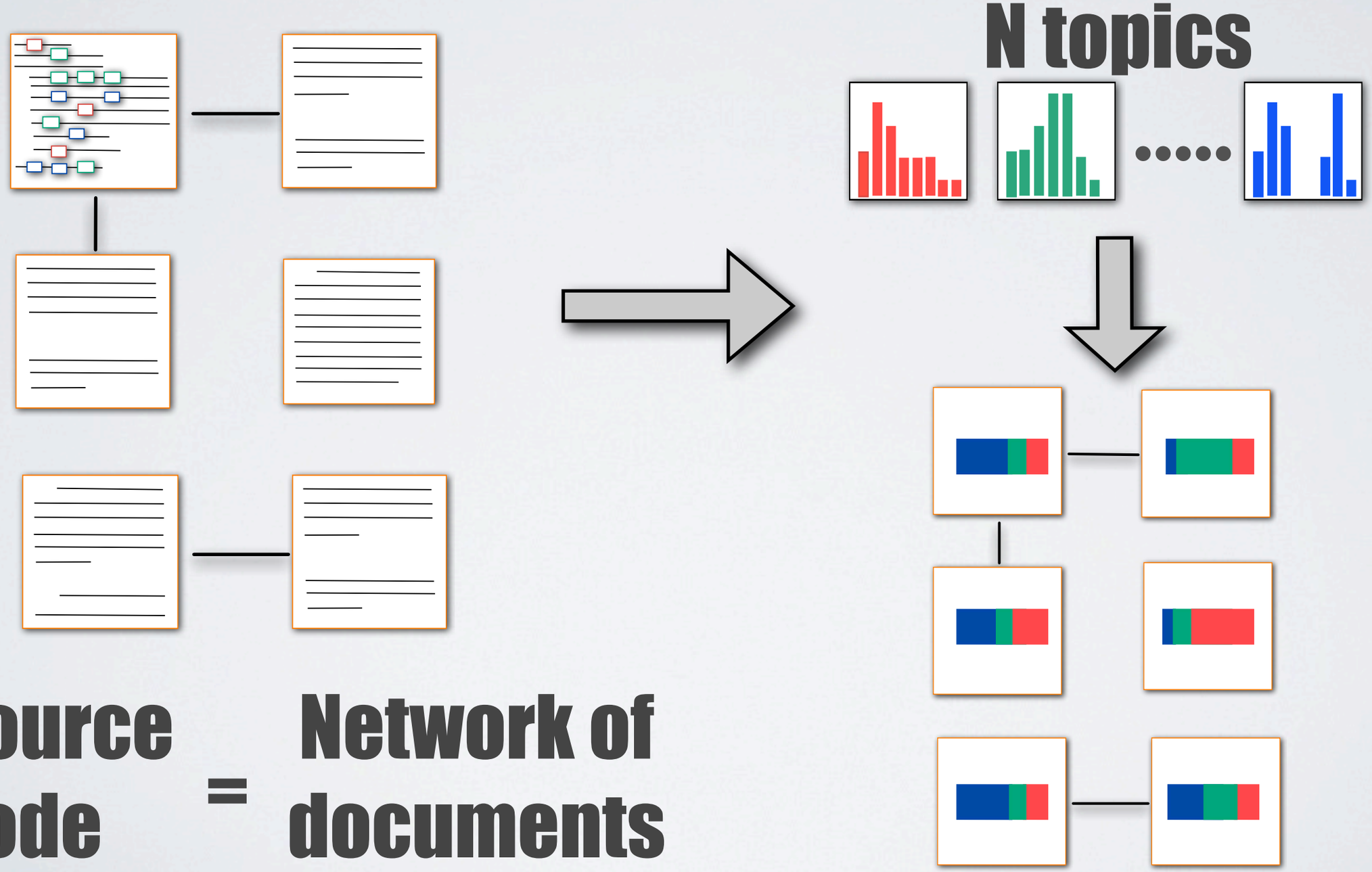


Words x Topics

Topics x Documents

Relational Topic Model

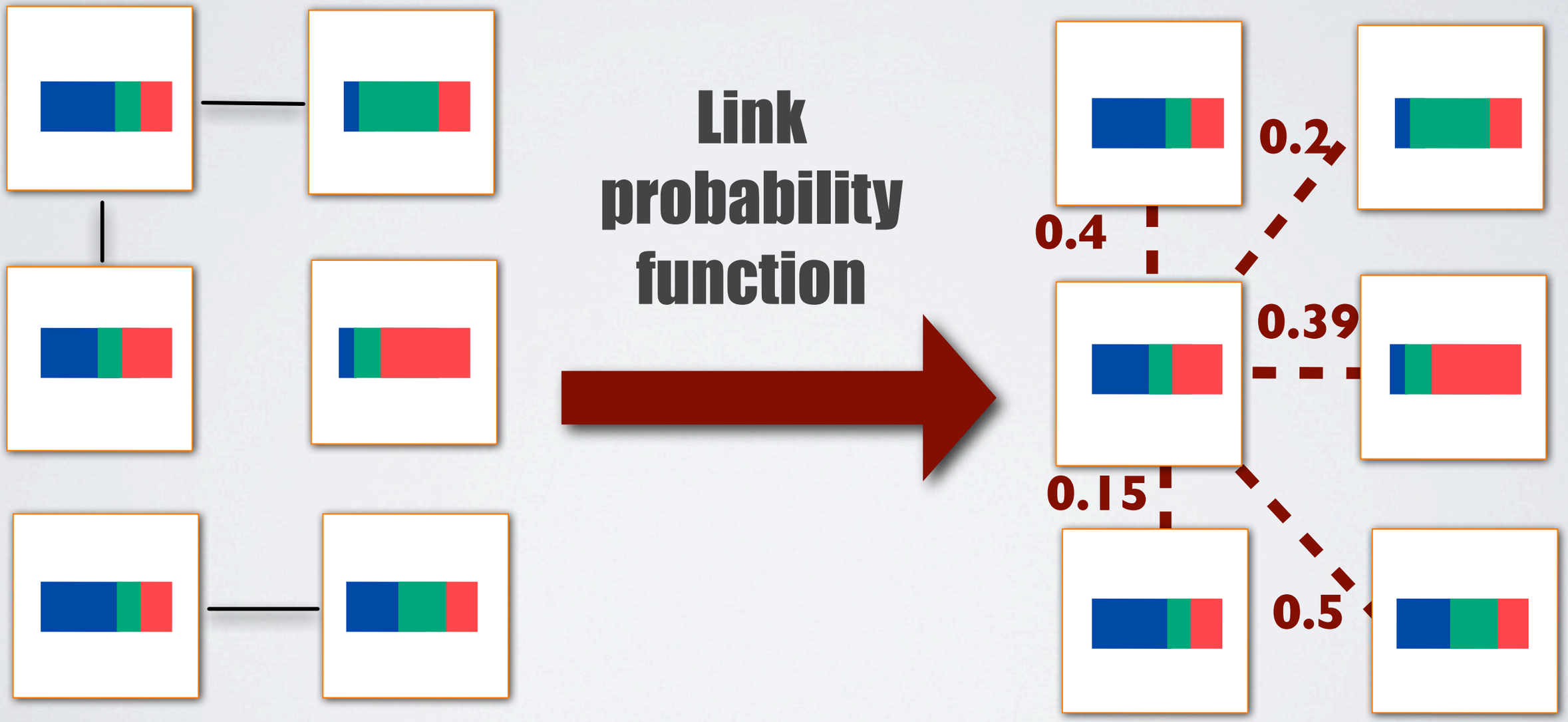
[Chang and Blei 10]



Source code = Network of documents

Relational Topic Model

[Chang and Blei 10]



RTM in ExPort

Documents: Functions

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

Documents: Functions

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

Words: API calls

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

Documents: Functions

Words: API calls

Links: Function A calls Function B

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

```java
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```java
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
            "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Target API method

`API.executeUpdate()`

```java
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
            "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

```java
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //.....
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```java
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
            "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Links between documents

```java
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

```java
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```java
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
            "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Target API method

`API.executeUpdate()`

```java
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

```java
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

```java
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
            "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Link probability (0.54)

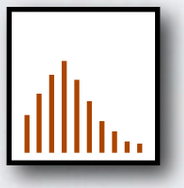
```java
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

RTM in ExPort

```java
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //......
        Connection  con = DBConnection.getInstance("POS-db");
        //....
        updateCoffeeSales(salesForWeek, con);
        //....
    } catch (Exception ex) {
        //Do something
    }
}
```

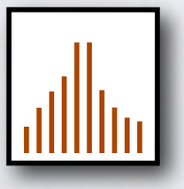
Similar API method

`API.getConnection()`



Target API method

`API.executeUpdate()`



```java
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
            "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

```java
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

ExPort:

Visualizing API usages by allowing browsing and navigation

Browsing software repositories

- Browsing is more effective when following relationships

- Browsing helps to understand high level concepts

Automatically Inferring Concern Code from Program Investigation Activities

Martin P. Robillard and Gail C. Murphy
Department of Computer Science
University of British Columbia
2366 Main Mall, Vancouver, BC
Canada V6T 1Z4

Abstract

small subset of the source code as part of a program evolu-

When perf

Browsing and Searching Software Architectures

Susan Elliott Sim† Charles L.A. Clarke* Richard C. Holt‡ Anthony M. Cox‡
†Computer Science *Electrical and Computer Engineering ‡Computer Science
University of Toronto University of Toronto University of Waterloo
simsuz@cs.utoronto.ca clclarke@eecg.utoronto.ca {holt, amcox}@plg.uwaterloo.ca

tion. Browsing is
and exploratory
such as flipping

NavTracks: Supporting Navigation in Software Maintenance

Janice Singer
Institute for Information Technology
National Research Council Canada
janice.singer@nrc-cnrc.gc.ca

Robert Elves
Department of Computer Science
University of Victoria
relves@uvic.ca

Margaret-Anne Storey
Department of Computer Science
University of Victoria
mstorey@uvic.ca

In this paper,
supports browsin

Portfolio: Finding Relevant Functions and Their Usages

Collin McMillan
College of William & Mary
Williamsburg, VA 23185
cmc@cs.wm.edu

Mark Grechanik
Accenture Technology Lab
Chicago, IL 60601
mark.grechanik@accenture.com

Denys Poshyvanyk
College of William & Mary
Williamsburg, VA 23185
denys@cs.wm.edu

Qing Xie, Chen Fu
Accenture Technology Lab
Chicago, IL 60601
{qing.xie,chen.fu}@accenture.com

ABSTRACT

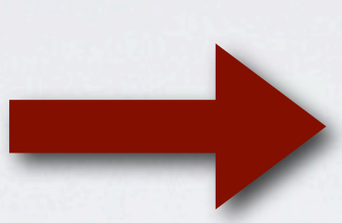
Different studies show that programmers are more interested in

implement high-level requirements. Second, programmers must understand how a function is used in order to use it themselves. Third, programmers must see the chain of function invocations in

Browsing software repositories

Target API method

`API.method()`

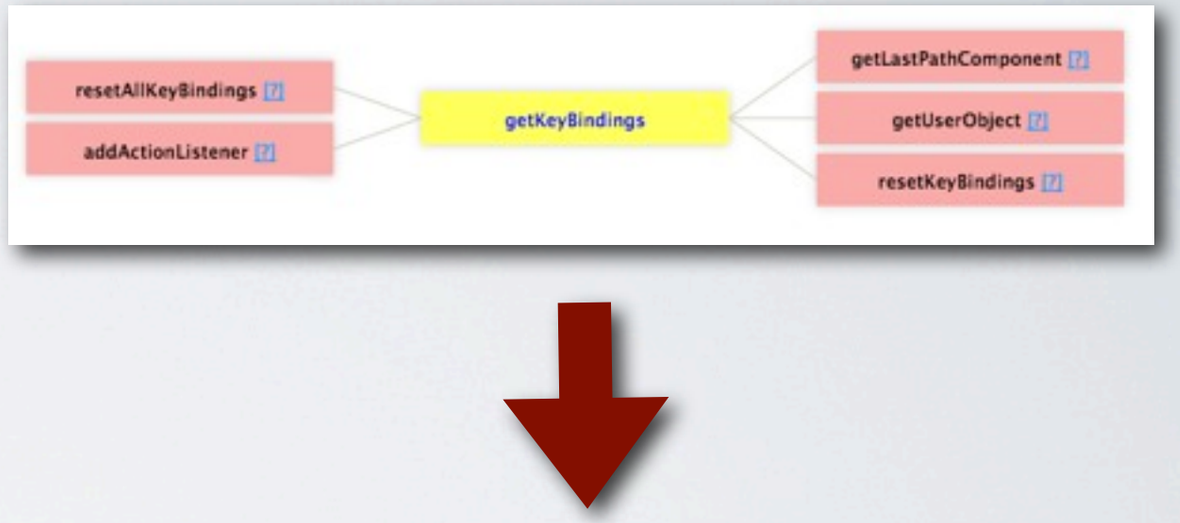


Similar API methods

Browsing software repositories

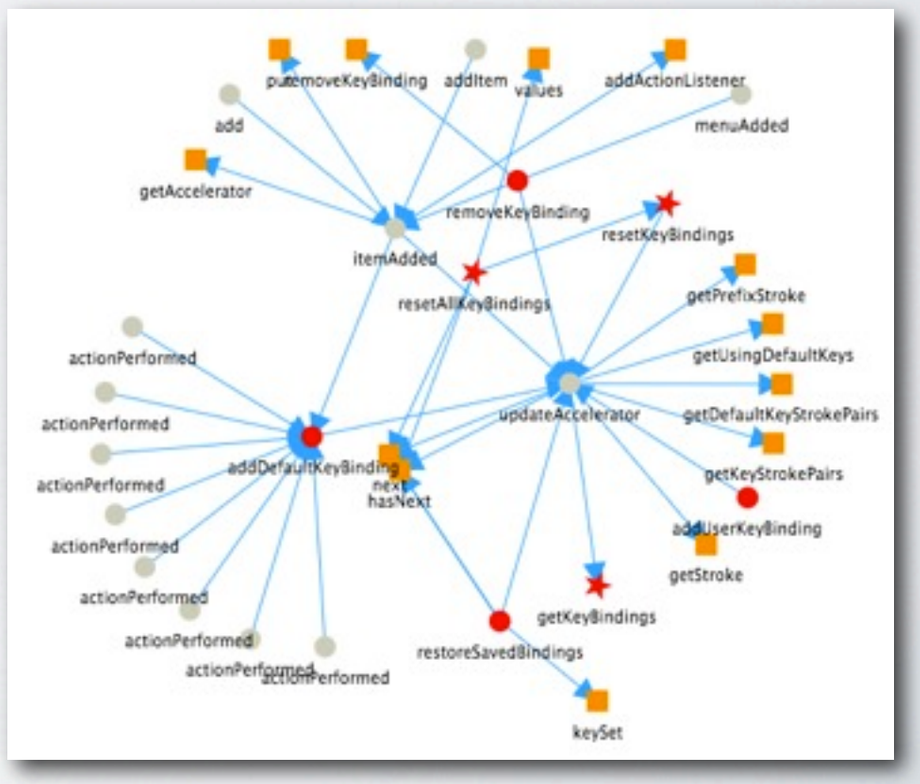
Target API method

`API.method()`



Call graph

- Functions
- API methods
- Calls



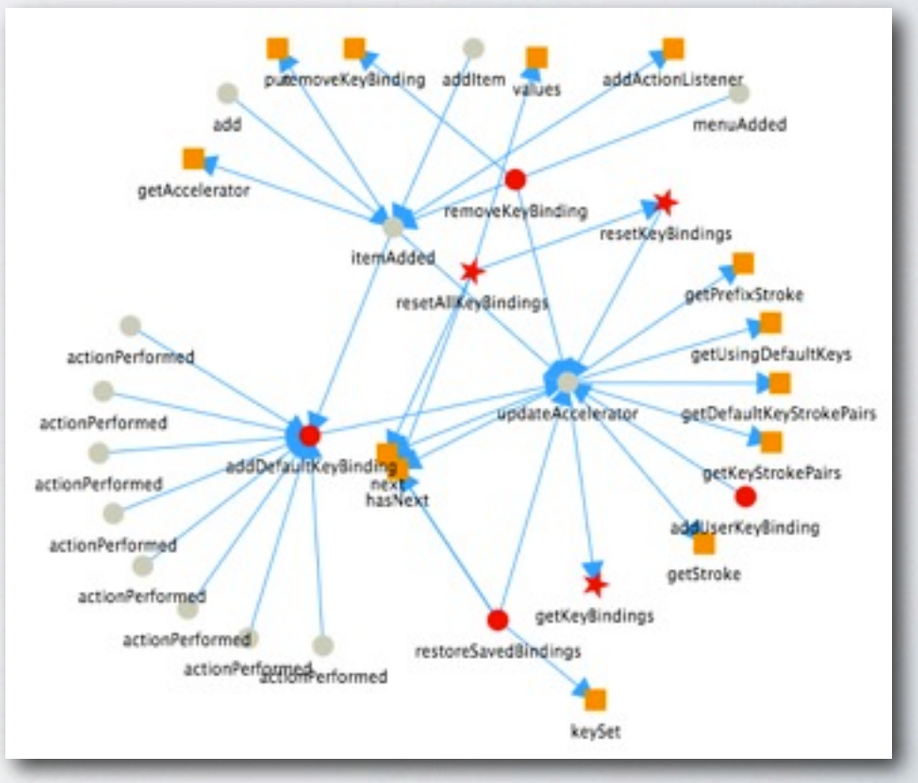
Browsing software repositories

Target API method

`API.method()`

API Usage example

```
public void updateAccelerator(String actionDesc) {
    // get valid key binding, update menus with it
    KeyBindings bindings = menuBarGroup.keyBindingManager.getKeyBindir
    KeyStroke accelerator = null;
    if (bindings != null) {
        Iterator it;
        if (bindings.getUsingDefaultKeys()) it = bindings.getDefaultKe
            it = bindings.getKeyStrokePairs();
        while (it.hasNext()) {
            KeyStrokePair pair = (KeyStrokePair)it.next();
            if (pair.getPrefixStroke() != null) continue;          // s
            accelerator = pair.getStroke();
            break;
        }
    }
    // update menu items
    synchronized(menuBarGroup) {
        ArrayList list = (ArrayList)menuBarGroup.menuItems.get(actionD
        if (list != null) {
            for (Iterator it = list.iterator(); it.hasNext(); ) {
                JMenuItem m = (JMenuItem)it.next();
                m.setAccelerator(accelerator);
            }
        }
    }
}
```



ExPort

http://www.cs.wm.edu/semeru/export

ExPort Demo

API Visualization Tool

Info about ExPort

k Browse

Conclusions

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
44          PreparedStatement updateSales = null;
45
46          String updateString =
47                  "update COFFEES "
48                  + "set SALES = ? where COF_NAME = ?";
49
50
51          con.setAutoCommit(false);
52          updateSales = (PreparedStatement) con.prepare
53
54          for (Map.Entry<String, Integer> e : salesForW
55              updateSales.setInt(1, e.getValue().intVal
56              updateSales.setString(2, e.getKey());
57              updateSales.executeUpdate();
58
59              con.commit();
60          }
61      }
```

How should I initialize the Connection?

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

Conclusions

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
44          PreparedStatement updateSales = null;
45
46          String updateString =
47                  "update COFFEES "
48                  + "set SALES = ? where COF_NAME = ?";
49
50
51          con.setAutoCommit(false);
52          updateSales = (PreparedStatement) con.prepare
53
54          for (Map.Entry<String, Integer> e : salesForW
55              updateSales.setInt(1, e.getValue().intVal
56              updateSales.setString(2, e.getKey());
57              updateSales.executeUpdate();
58
59              con.commit();
60          }
61      }
```

How should I initialize the Connection?

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

RTM in ExPort

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //.....
        Connection  con = DBConnection.getInstance("POS-db");
        //.....
        updateCoffeeSales(salesForWeek, con);
        //.....
    } catch (Exception ex) {
        //Do something
    }
}
```

```
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
                        "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Links between documents

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

Conclusions

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
44          PreparedStatement updateSales = null;
45
46          String updateString =
47                  "update COFFEES "
48                  + "set SALES = ? where COF_NAME = ?";
49
50
51          con.setAutoCommit(false);
52          updateSales = (PreparedStatement) con.prepare
53
54          for (Map.Entry<String, Integer> e : salesForW
55              updateSales.setInt(1, e.getValue().intVal
56              updateSales.setString(2, e.getKey());
57              updateSales.executeUpdate();
58
59              con.commit();
60          }
61      }
```

How should I initialize the Connection?

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

RTM in ExPort

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //.....
        Connection  con = DBConnection.getInstance("POS-db");
        //.....
        updateCoffeeSales(salesForWeek, con);
        //.....
    } catch (Exception ex) {
        //Do something
    }
}
```

```
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
                        "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Links between documents

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";


    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

Browsing software repositories

Target API method

`API.method()`

API Usage example

Conclusions

Mining API usage examples

```
41      public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
42              throws SQLException {
43
44          PreparedStatement updateSales = null;
45
46          String updateString =
47                  "update COFFEES "
48                  + "set SALES = ? where COF_NAME = ?";
49
50
51          con.setAutoCommit(false);
52          updateSales = (PreparedStatement) con.prepare
53
54          for (Map.Entry<String, Integer> e : salesForW
55              updateSales.setInt(1, e.getValue().intVal
56              updateSales.setString(2, e.getKey());
57              updateSales.executeUpdate();
58
59              con.commit();
60          }
61      }
```

How should I initialize the Connection?

Based on http://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html

RTM in ExPort

```
public void consolidate(HashMap<String, Integer> salesForWeek){
    try {
        //.....
        Connection con = DBConnection.getInstance("POS-db");
        //.....
        updateCoffeeSales(salesForWeek, con);
        //.....
    } catch (Exception ex) {
        //Do something
    }
}
```

```
public static Connection getInstance(String schema) throws Exception {
    Properties connectionProps = new Properties();
    connectionProps.put("user", "root");
    connectionProps.put("password",
                        "*******");
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    return DriverManager.getConnection("jdbc:mysql://localhost:3306/"
            + schema, connectionProps);
}
```

Links between documents

```
public void updateCoffeeSales(HashMap<String, Integer> salesForWeek, Connection con)
        throws SQLException {

    PreparedStatement updateSales = null;

    String updateString =
            "update COFFEES "
            + "set SALES = ? where COF_NAME = ?";

    con.setAutoCommit(false);
    updateSales = (PreparedStatement) con.prepareStatement(updateString);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();

        con.commit();
    }
}
```

Browsing software repositories

Target API method

`API.method()`

API Usage example

ExPort

http://www.cs.wm.edu/semeru/export