Anforderungen: OdysseusDashboard

Stand: 06.06.2012

Von Timo Michelsen

**Dashboard**: Grafische Darstellung von (multiplen) Daten aus mehreren Datenströmen. Sie besteht aus mehreren Dashboard Elementen. Jedes Element kann im Dashboard frei positioniert und skaliert werden. Dashboards sollen einen Großteil der Oberfläche im Endprodukt von Odysseus ausmachen.

**Dashboard Element (Instanz):** Darstellung/grafische Zusammenfassung einer zugrunde liegende Anfrage aus mehreren Datenströmen auf Basis eines Elementtyps. Es können mehrere Elemente eines Typs existieren, jedoch unterschiedlich konfiguriert. Ein Dashboard Element ist eine Instanz eines Elementtyps. Dashboard Elemente werden zu Dashboards zusammengestellt.

**Dashboard Elementtyp:** Grundlegende Beschreibung eines Dashboard Elements (bswp. Klasse). Beinhaltet die typischen Verhaltensweisen, Darstellungen und welche Einstellungen pro Instanz vorgenommen werden können. Beispiele sind Graph, Tabelle und Kreisdiagramm.

**Start-Dashboard:** Ein Dashboard, welches zum Programmstart von Odysseus angezeigt wird (anstatt der normalen RCP-Oberfläche).

**Zugrunde liegende Anfrage:** Eine Anfrage in Odysseus, die speziell für eine Visualisierung erstellt, übersetzt und ausgeführt wird.

1. Der Nutzer kann ein **Dashboard Element (Instanz) erstellen.**
   1. Es muss angegeben werden, zu welchem **Elementtyp** eine Instanz erstellt werden soll.
   2. Zu jeder Instanz muss in Abhängigkeit des Typs **spezifische Einstellungen** vorgenommen werden können.
      1. Es können optionale Einstellungen existieren.
      2. Es können nicht-optionale Einstellungen existieren, d.h. der Nutzer muss vor der Erstellung des Elements einen Wert für diese Einstellungen angeben.
      3. Für nicht-optionale Einstellungen können Standardwerte vorhanden sein.
      4. Ist für eine nicht-optionale Einstellung ein Standardwert vorhanden, wird dieser zu Beginn der Erstellung des Dashboard Elements als ersten Wert eingetragen.
      5. Die Einstellungen können auf den Standardwert zurückgesetzt werden.
         1. Hat eine Einstellung keinen Standardwert, besitzt die Einstellung nach dem Zurücksetzen keinen Wert.
   3. Erst wenn **alle nicht-optionalen Einstellungen korrekt** sind, darf das Dashboard Element erzeugt werden. Dann gilt die Instanz als erstellt.
   4. Es muss zu einem Element die **zugrunde liegende Anfrage** angegeben werden.
      1. Die Anfrage darf aus höchstens einem Statement (bspw. CQL) bestehen.
      2. Die Anfrage muss gültig sein, um ein Dashboard Element zu erzeugen.
      3. Eine Anfrage ist gültig, wenn sie von Odysseus ausgeführt werden kann.
      4. Ist die Anfrage nicht gültig, darf das Element nicht erzeugt werden.
   5. Ein erstelltes Element wird **nicht automatisch angezeigt.**
2. Der Nutzer kann ein **Dashboard Element speichern.**
   1. Es muss ein Dateinamen angegeben werden.
   2. Die Endung muss del sein.
3. Der Nutzer kann ein **Dashboard Element laden.**
   1. Der Nutzer muss ein Dateinamen angeben.
   2. Ein Dashboard Element, welches unter dem Dateinamen existiert, muss geladen werden können.
      1. Kann es nicht geladen werden, muss eine Fehlermeldung ausgegeben werden.
      2. Dem Nutzer muss klar sein, was genau der Fehler ist, damit einfach Korrekturen vorgenommen werden kann.
      3. Die Fehlermeldung wird protokolliert.
   3. Erfolgreich geladene Elemente werden nicht automatisch angezeigt.
4. Der Nutzer kann ein **erstelltes Dashboard Element anzeigen.**
   1. Beim Anzeigen des Elements wird dieser zuvor geladen.
   2. Die zugrunde liegende Anfrage wird genau dann gestartet, wenn das Element angezeigt wird.
   3. Wird die zugrunde liegende Anfrage ausgeführt, wird das Element **automatisch aktualisiert.**
      1. Der Nutzer kann die **Aktualisierung** **unterbrechen.**
      2. Der Nutzer kann eine unterbrochene **Aktualisierung** **fortsetzen.**
   4. Entsteht bei der Anzeige eines Dashboard Elements ein Fehler, wird der Fehler protokolliert.
      1. Eine Textnachricht wird dem Nutzer mitgeteilt, dass ein Fehler aufgetreten ist.
      2. Die Meldung sollte so aussagekräftig sein, dass der Nutzer Korrekturen vornehmen kann.
      3. Die Mitteilung wird dort angezeigt, wo sonst das Element angezeigt werden würde.
      4. Es wird weiterhin versucht, die zugrunde liegende Anfrage auszuführen, d.h. sie wird nicht gestoppt.
   5. Der Nutzer kann die Anzeige des **Elements beenden.**
      1. Wird die Anzeige geschlossen, wird die zugrunde liegende Anfrage beendet und aus dem Ausführungsplan von Odysseus entfernt.
      2. Beim Schließen der Anzeige wird das Element gespeichert.
5. Der Nutzer kann ein erstelltes **Dashboard Element bearbeiten.**
   1. Der Typ soll nicht verändert werden können.
   2. Die zugrunde liegende Anfrage soll nicht verändert werden können.
   3. Einige Einstellungen in Abhängigkeit zum Typ können bearbeitet werden.
   4. Änderungen von Einstellungen müssen gespeichert werden.
   5. Änderungen von Einstellungen muss das Element sofort übernehmen.
   6. Einige Einstellungen sollen nicht modifizierbar sein.
6. Der Nutzer kann ein erstelltes **Dashboard Element löschen.**
   1. Der Nutzer kann entscheiden, ob die zugrunde liegende Anfrage ebenfalls gelöscht werden soll.
7. Der Nutzer kann ein **Dashboard erstellen.**
   1. Zu Beginn ist das Dashboard leer (bspw. Weiße Seite).
   2. Der Nutzer kann darin Dashboard Elemente platzieren.
      1. Jedes Dashboard Element hat eine Position.
         1. Die Position kann vom Nutzer geändert werden.
      2. Jedes Dashboard Element hat eine Größe.
         1. Die Größe kann vom Nutzer geändert werden.
      3. Mehrere Elemente können sich überlappen.
   3. Ein Dashboard Element kann mehrfach platziert werden.
      1. Jede Kopie kann unterschiedlich eingestellt werden (s. 5.).
   4. Platzierte Elemente können vom Nutzer entfernt werden.
      1. Das Element wird nicht vollständig gelöscht.
   5. Geänderte Dashboard Elemente (Größe und Position) werden gespeichert.
8. Der Nutzer kann ein **erstelltes Dashboard anzeigen.**
9. Der Nutzer kann ein **erstelltes Dashboard löschen.**
   1. Die darin platzierten Dashboard Elemente werden nicht gelöscht.
   2. Der Nutzer wird vor der tatsächlichen Löschung mindestens einmal gefragt..
10. Der Nutzer kann ein **erstelltes Dashboard speichern.**
    1. Der Nutzer muss ein Dateinamen angeben.
       1. Die Endung muss dbd sein.
       2. Der Dateiname muss eindeutig sein.
       3. Wird ein Dateiname angegeben, welcher bereits existiert, muss der Nutzer gefragt werden, ob die Datei überschrieben werden soll.
    2. In der Datei wird die Position der Dashboard Elemente gespeichert.
    3. In der Datei wird die Größe der Dashboard Elemente gespeichert.
    4. In der Datei wird eine Referenz zu den Dateien der Elemente gespeichert.
    5. Ansonsten wird nichts gespeichert.
11. Der Nutzer kann ein **gespeichertes Dashboard laden.**
    1. Der Nutzer muss einen Dateinamen angeben.
       1. Ist die Datei mit gegebenen Namen vorhanden, muss sie geladen werden.
          1. Bei einem Fehler muss eine Fehlermeldung ausgegeben werden.
          2. Die Fehlermeldung wird protokolliert.
          3. Die Meldung muss aussagekräftig genug sein, damit der Nutzer entsprechende Korrekturen vornehmen kann.
       2. Erfolgreich geladene Dashboards werden nicht automatisch angezeigt.
12. Der Nutzer kann definieren, welches **Dashboard zu beim Programmstart angezeigt** werden soll (Start-Dashboard).
    1. Diese muss in Odysseus zentral eingestellt werden können.
    2. Wird der angegebene Dashboard nicht gefunden bzw. kann nicht geladen werden, wird die Standardoberfläche angezeigt.
       1. Die Fehlermeldung wird protokolliert.
       2. Die Meldung muss aussagekräftig genug sein, damit der Nutzer entsprechende Korrekturen vornehmen kann.
    3. Wird der Start-Dashboard erfolgreich geladen, wird dieser sofort angezeigt.

# Technische Anforderungen

1. Dashboards sind unabhängig von der Datenstrombeobachtung.
2. Odysseus ist unabhängig von Dashboards.
   1. Odysseus kann ohne Dashboards gestartet werden.
3. Entwickler können **neue Dashboard Elementtypen** definieren.
   1. Entwickler können beschreiben, wie die Dashboard Elementtypen aussehen.
   2. Entwickler können beschreiben, wie sich die Elementtypen verhalten.
   3. Entwickler können Einstellungen definieren, die für jede Instanz des Typs festgelegt werden müssen.
      1. Es kann ein Standardwert definiert werden.
      2. Die Einstellung kann selbst prüfen, ob sie einen korrekten Wert enthält.
      3. Es kann definiert werden, dass eine Einstellung optional ist.
      4. Es kann definiert werden, dass eine Einstellung nur bei der Erstellung der .Visualisierung gesetzt werden darf. Ist dies der Fall, kann bei der Bearbeitung der Visualisierung die Einstellung nicht geändert werden.
4. Die zugrunde liegende Anfrage muss aus **einer „qry“-Datei ausgelesen** werden können.
   1. Optional kann bei der Erstellung eines Dashboard Elements eine Anfrage direkt eingegeben werden, welche dann neben der Erstellung des Elements zu einer Erstellung der Anfrage-Datei führt.
5. Die Speicherung der Elementinformationen muss in **XML** geschehen, um eine manuelle Bearbeitung zu ermöglichen.
   1. Das Format muss so definiert werden, dass zu jedem Dashboard Element beliebige Einstellungen eingetragen werden können.
   2. Die Dateien müssen fehlertolerant gegenüber benutzerdefinierten Einstellungen zu spezifischen Visualisierungsformen sein.
   3. Die Elementtypen sind selbst verantwortlich, die Einstellungen in den Instanzen korrekt umzusetzen.