Bachelorarbeit

Kien Nghia Tran

Entwicklung eines grafischen Toolkits für
die Erstellung von Mashups

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Kien Nghia Tran

Entwicklung eines grafischen Toolkits für die Erstellung von Mashups

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing. Birgit Wendholdt
Zweitgutachter : Prof. Dr. rer. nat. Gunter Klemke

25.05.2011

Kien Nghia Tran

Thema der Bachelorarbeit

Entwicklung eines grafischen Toolkits für die Erstellung von Mashups

Stichworte

Grafisches Toolkit, Mashup, WSO2 Mashup Server.

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung eines grafischen Toolkits zur Erstellung von Mashups auf Datenebene unter Verwendung des WSO2 Mashup Servers. Anhand eines Evaluierungs-Prototyps wird das Konzept umgesetzt und Erweiterungsmöglichkeiten aufgezeigt.

Kien Nghia Tran

Title of the paper

Development of a graphical toolkit for creating mashups

Keywords

Graphical toolkit, mashups, WSO2 Mashup Server.

Abstract

This thesis describes the development of a graphical toolkit for creating data oriented mashups utilizing the WSO2 Mashup Server. Architecture patterns have been taken into consideration during development. A prototype has been implemented to show the functions.

INHALTSVERZEICHNIS

Abbildungsverzeichnis

1. Einleitung

Für die Gestaltung von (Unternehmens-)Prozessen oder Planung von Produkten sind Informationen verschiedenster Art wie zum Beispiel Steuerungs- und Überwachungsinformationen ein wichtiger Produktionsfaktor (Seidenberg, 1998). Erst kontextbezogene und zielorientiert gewonnene Daten werden zu wertvollen Informationen. So kann das Datum „20255“ eine Postleitzahl sein oder eine Telefonnummer oder die Anzahl der offenen Stellen in Hamburg. Die Gewinnung von Daten und Informationen kann mühselig sein, da die Daten nicht immer bereits in gewünschter oder allgemein zugänglicher Form vorliegen. In der Regel werden in einem Unternehmen Daten von der Informationsabteilung nach Vorgaben der Fachabteilung aufbereitet und bereitgestellt. Die Informationsabteilung übernimmt also die technische Bereitstellung der Daten, während die Fachabteilung die Daten kontextbezogen interpretiert und bewertet.

Eine Aufbereitung, Visualisierung (z.B.: Diagramme, Geo Informationen auf Karten), Transformation, Aggregation (z.B.: Aufsummierung, Zusammenfassung) bzw. Anreicherung (z.B.: zusätzliche Informationen) von Daten kann neue Sichten und Erkenntnisse ermöglichen. In einigen Fällen reicht jedoch eine Bereitstellung der Daten auf einer allgemein zugänglichen Plattform aus (z.B.: Daten einer Datenbank auf einer Webseite publizieren).

Die Aufbereitung und Darstellung von Daten aus verschiedenen Quellen wird als Mashup bezeichnet. Diese Arbeit befasst sich mit der Entwicklung eines grafischen Toolkits zur Erstellung von Mashups. Im Sinne des End User Developments (Lieberman u.a., 2006) soll das grafische Toolkit auch technisch nicht versierten Anwendern die Möglichkeit bieten, Mashups auf einfache Weise mittels einer grafischen Oberfläche zu erstellen. Auf diese Weise wird es direkt dem Anwender, der am besten die Daten verstehen und interpretieren kann, ermöglicht, die Daten aufzubereiten und neue Sichten zu kreieren.

Die Arbeit beginnt mit den Grundlagen von Mashups. Dabei werden die Definition von Mashups erläutert und verschiedene Mashup Kategorien vorgestellt. Im Anschluss erfolgt eine Untersuchung von „Enterprise Mashup Markup Language“ (EMML), einer domänenspezifischen Sprache für Mashups.

In Kapitel 3 werden ausgewählte grafische Werkzeuge zur Mashup Erstellung vorgestellt und evaluiert. Yahoo Pipes, JackBe Presto, IBM Mashup Center und WSO2 Mashup Server werden unter den Gesichtspunkten „Datenquellen“, „Grafische Unterstützung“, „Ausgabeformate und Präsentation“, „Erweiterbarkeit“ und „Kosten“ verglichen und bewertet.

In Kapitel 4 werden die allgemeinen Anforderungen an einer Entwicklungsumgebung und spezielle Anforderungen an einer Entwicklungsumgebung für Mashups auf Datenebene bzw. Präsentationsebene formuliert. Im zweiten Teil des Kapitels werden detailliert die Möglichkeiten und Anforderungen für die Umsetzung einer grafischen Unterstützung bei der Erstellung von Mashups auf Datenebene erörtert.

Kapitel 5 stellt Konzepte vor, auf dessen Basis der Entwurf für die Entwicklung eines grafischen Toolkits für die Erstellung von Mashups auf Datenebene aufbaut. Die Umsetzung der vorgestellten Konzepte stellt eine Wiederverwendbarkeit, Wartbarkeit, Erweiterbarkeit und Flexibilität sicher. Des Weiteren werden Rahmenbedingungen erörtert, die bei der Umsetzung zu berücksichtigen sind. Abschließend wird eine prototypische Implementation vorgestellt, anhand dessen die Tragfähigkeit der Konzepte gezeigt wird.

Das letzte Kapitel fasst das Erreichte zusammen, gibt einen Ausblick auf die nächsten Meilensteine und weist weitere Möglichkeiten auf diese Arbeit zu erweitern.

2. Grundlagen

Dieses Kapitel erklärt den Begriff Mashup und stellt unterschiedliche Mashuptypen vor. Des Weiteren wird die domänenspezifische Sprache EMML[1] vorgestellt, welche sich als Ziel setzt, die Mashup-Erstellung insbesondere im Unternehmensumfeld zu vereinfachen und den inhomogenen Mashup-Markt zu vereinheitlichen.

2.1 Definition Mashup

Der Begriff „Mashup“ entstammt aus der Musikwelt. In der Musik werden Rekombinationen aus typischerweise zwei oder mehr Musikstücken zu einem neuen Musikstück als „Mashup“ bezeichnet. In Anlehnung an diese Bezeichnung werden in der Informationstechnologie Aufbereitungen und Rekombinationen aus typischerweise zwei oder mehreren Ressourcen als „Mashup“ bezeichnet[2]. Das entstandene Mashup bietet dabei kontextbezogen einen größeren Mehrwert als die Nutzung jeder einzelnen Ressource.

2.2 Mashup Kategorien

Im Allgemeinen werden Mashups in folgende Kategorien unterteilt (Hanson, 2009):

- Consumer / Enterprise Mashup
- Client-Side / Server-Side Mashup
- Presentation / Data / Process Oriented Mashup

Diese Kategorien werden im Folgenden einzeln erläutert.

[1] Enterprise Mashup Markup Language
[2] http://www.openmashup.org

2.2.1 Consumer / Enterprise Mashup

Je nach Zielgruppe unterscheidet man zwischen Consumer Mashups und Enterprise Mashups. Consumer Mashups zielen auf den Endanwender. Ein typisches Beispiel für ein Consumer Mashup ist zum Beispiel „Housingmaps“[3]. Dieses Mashup verknüpft Landkarten von Google Maps mit den Ortsinformationen aus Craigslist, einer Internetplattform für Kleinanzeigen. Der Anwender erhält auf diese Weise eine übersichtliche Darstellung von Immobilienstandorten auf einer Landkarte.
Weitere typische Beispiele sind Metasuchmaschinen, Preissuchmaschinen und Personensuchmaschinen. Diese Suchmaschinen aggregieren die Ergebnisse vieler verschiedener Einzelsuchmaschinen und bereiten diese zur Anzeige auf.

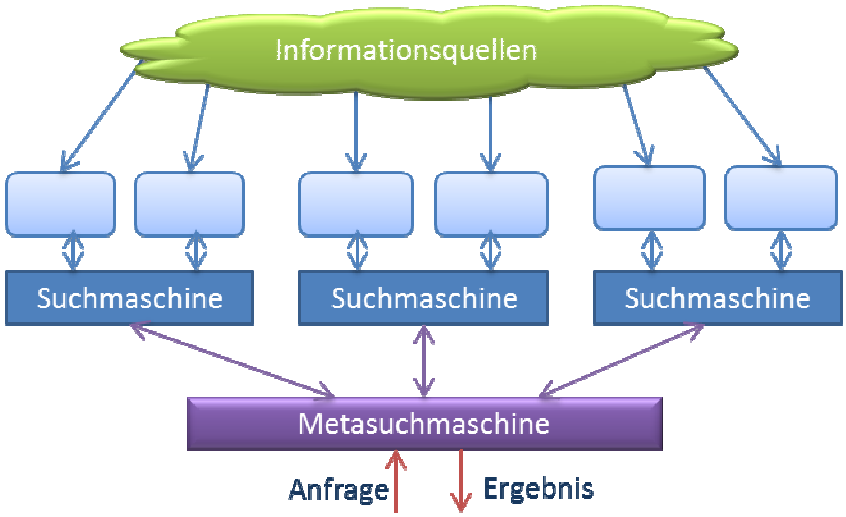


Abb. 1: Architektur Metasuchmaschine[4]

Eine weitere große Gruppe der Consumer Mashups sind vom Anwender anpassbare Social Network und Blog Portale. Diese Seiten erlauben es dem Nutzer, über vorgefertigte Widgets wie zum Beispiel Music Player, Picture Gallery, Themes, Twitter Plug-in, etc. die eigenen Seiten zu bearbeiten, anzureichern und zu individualisieren.

Enterprise Mashups zielen auf Anwender innerhalb eines Unternehmens. Ein Beispiel ist ein Programm für Reisebüros, welches die Informationen verschiedener Fluggesellschaften und

[3] http://www.housingmaps.com
[4] http://upload.wikimedia.org/wikipedia/commons/1/12/Meta-search-de.svg

Reiseveranstalter zusammenfasst und dem Mitarbeiter des Reisebüros eine Übersicht der verfügbaren Flüge bzw. Reisen bietet und vergleichen lässt (Ziegler, 2010).

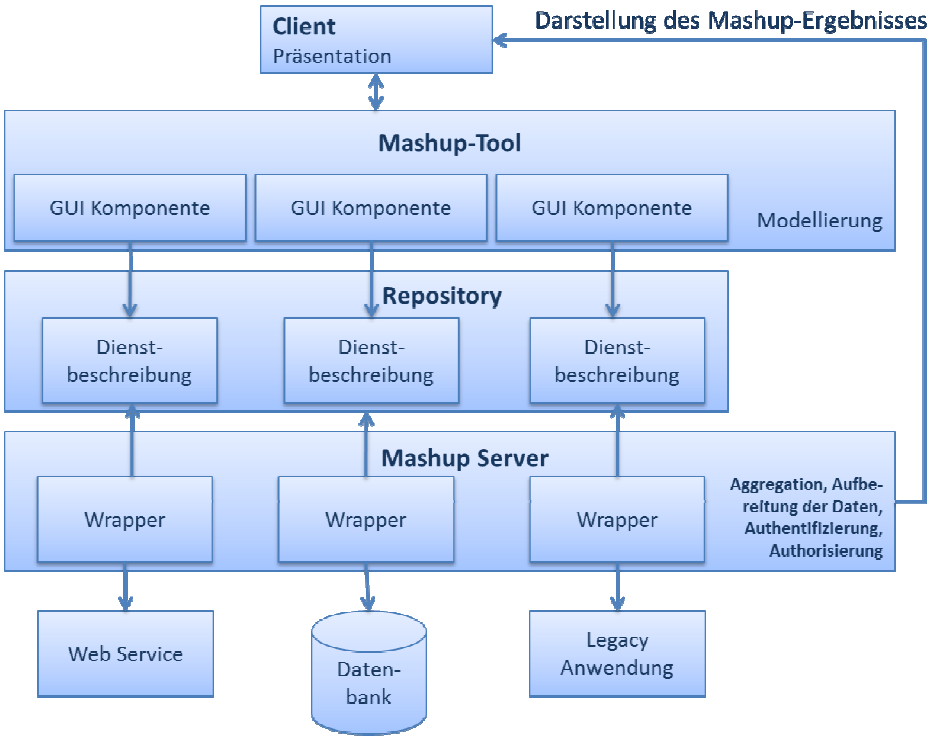


Abb. 2: Aufbau eines Enterprise-Mashup-Infrastruktur (PENTASYS AG)

In der Regel sind Enterprise Mashups lediglich im Intranet einer Firma erreichbar, so dass ein Link zu einem expliziten Beispiel fehlt. Die Abbildung 2 zeigt den schematischen Aufbau einer Enterprise-Mashup-Infrastruktur. Die Enterprise-Mashup-Infrastruktur ist typischerweise mehrschichtig aufgebaut. Dedizierte Mashup Server übernehmen die Aufgaben der Datenaggregation und -aufbereitung sowie die Anbindung verschiedener Datenquellen. Repositories bieten ein Verzeichnis der verfügbaren Dienste und bereitgestellten Mashups an. Über Mashup-Tools kann der Endanwender ohne Programmierkenntnisse die im Repository verfügbaren Dienste rekombinieren und an eigene Bedürfnisse anpassen.

2.2.2 Client-Side / Server-Side Mashup

Anhängig vom Ort der Datenaufbereitung unterscheidet man zwischen Client-Side Mashups und Server-Side Mashups.

Beim Client-Side Mashup werden die Daten im Browser aufbereitet. Als Beispiele können hier die mit Intel Mashmaker erstellten Mashups sowie Skripte der Firefox Erweiterung Greasemonkey genannt werden. Intel Mashmaker erweitert den Browser um die Funktionalität, Daten von verschiedenen Internetseiten auf einer Seite darzustellen. Greasemonkey erlaubt dem Anwender, Internetseiten per DOM Manipulation mit JavaScript anzupassen.

Bei Client-Side Mashups (vgl. Abb. 3) stellt das im Webbrowser implementierte Sicherheitskonzept der Same Origin Policy (SOP) häufig ein Problem dar. SOP verhindert, dass JavaScript einer Quelle auf Daten einer anderen Quelle zugreift. Um dieses Problem zu umgehen, haben sich zwei Techniken, nämlich Proxy Server sowie Callback etabliert: Bei der ersten Technik werden notwendige Daten über einem Stellvertreter, den Proxy Server, bereitgestellt. Beim Callback wird ein JavaScript Code dynamisch über Script Tags eingeschleust, um SOP zu umgehen und Daten von einer anderen Quelle zu erhalten (Hanson, 2009).

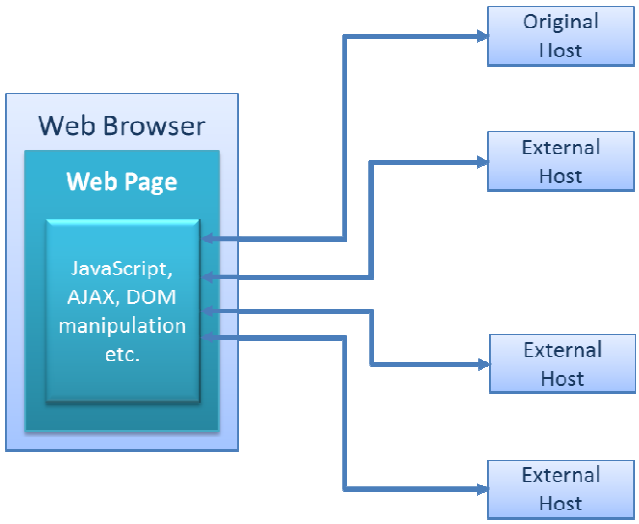


Abb. 3: Architektur Client-Side Mashup

Bei Server-Side Mashups (vgl. Abb. 4) werden die Inhalte der verschiedenen Datenquellen bereits auf dem Server aufbereitet, bevor diese an den Client ausgeliefert werden. Als Beispiel können hier ebenfalls Metasuchmaschinen genannt werden.

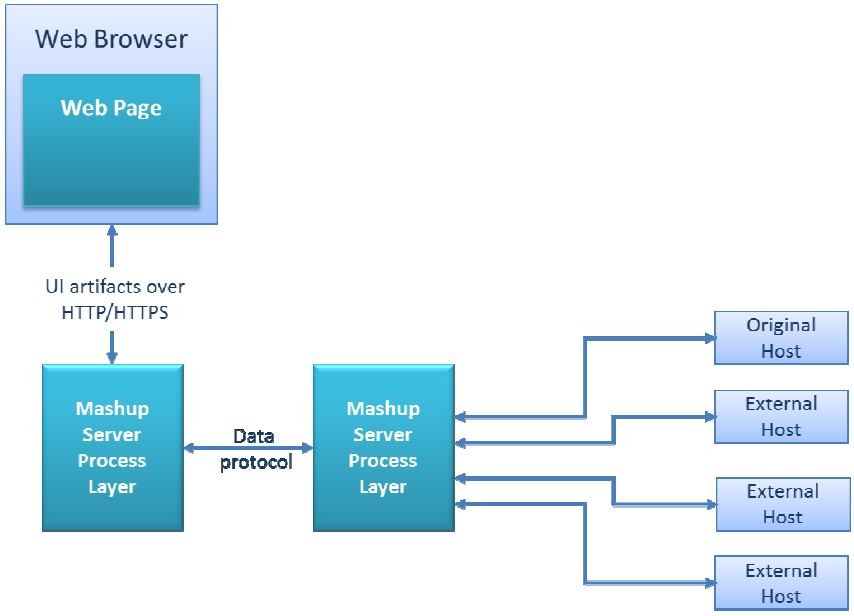


Abb. 4: Architektur Server-Side Mashup

2.2.3 Presentation / Data / Process Oriented Mashup

Eine weitere Möglichkeit ein Mashup zu kategorisieren, ist die Betrachtung der verwendeten Technologie zur Erzeugung des Mashups. Bei dieser Betrachtung werden die Mashups in den drei Hauptkategorien, Presentation-Oriented, Data-Oriented und Process-Oriented, eingeteilt (Hanson, 2009).

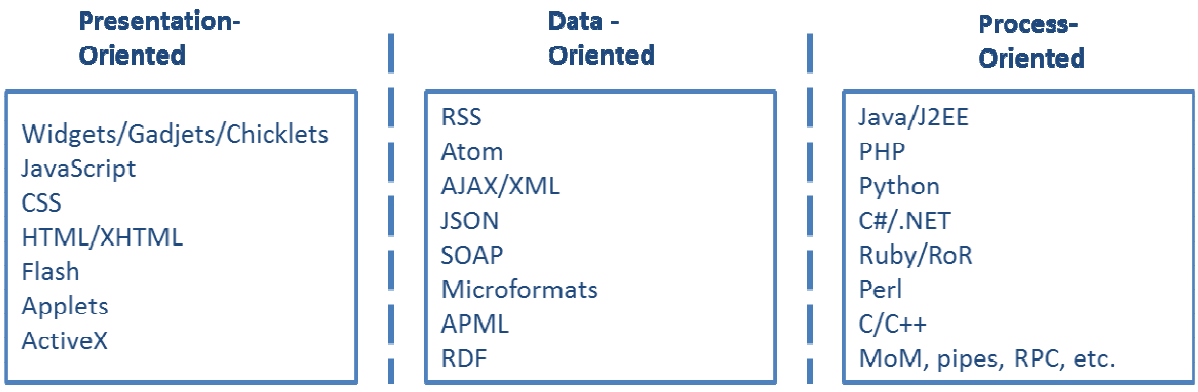


Abb. 5: Kategorisierung nach verwendeter Technologie

Presentation-Oriented

In dieser Kategorie liegt der Fokus in der Präsentation. Dabei werden meist GUI[5] Elemente einem Portal ähnlich zusammengestellt, wobei zwischen den einzelnen GUI Elementen nur wenig oder keine Interaktion herrscht. Als Beispiele sind hier iGoogle Seiten oder Yahoo Portal zu nennen.

Data-Oriented

Bei Data-Oriented Mashup liegt der Fokus in der Datenbereitstellung. Bei diesem Typus liefern Datenquellen Daten in Form von XML, RSS, JSON, Atom, etc. Die anschließende Verarbeitung der Daten geschieht Client-Side per JavaScript oder Server-Side mit Java, Python, Perl, etc.

Process-Oriented

Die Process-Oriented Mashups verarbeiten die Daten auf Programmiersprachenebene. Im Unterschied zu der Server-Side Verarbeitung bei Data-Oriented zielt dieser Typ auf die Bereitstellung von einem Service ab.

In der Praxis werden die oben genannten Technologien nicht isoliert, sondern vermischt eingesetzt (Hanson, 2009).

2.3 EMML - Enterprise Mashup Markup Language

EMML ist eine von Jackbe speziell für die Erstellung von Mashups entwickelte domänenspezifische Sprache[6] (Domain Specific Language - DSL). EMML wurde im September 2009 unter der Creative Common License 3.0 US Attribution-NoDerivs freigegeben. Zeitgleich wurde auch die Open Mashup Alliance (OMA)[7] gegründet, um EMML als offene und einheitliche Entwicklungssprache für Enterprise Mashups weiter zu entwickeln und zu fördern. Durch den offenen Standard soll die Hemmschwelle für die Investition in Mashups verringert werden, damit der Aspekt „Vendor Lock-In“ entfällt und die Portabilität und Interoperabilität

[5] Graphical User Interface - eine Software-Komponente, die dem Benutzer eines Computers die Interaktion mit der Maschine über grafische Symbole erlaubt.
[6] Sprache, die nur für eine bestimmte Aufgabe entworfen und implementiert wurde.
[7] http://www.openmashup.org

von Mashups gewährleistet wird. Die OMA weist in ihrer FAQ[8] darauf hin, dass es sich nicht um ein Standardisierungsgremium handelt. Obwohl der OMA einige namenhafte Firmen wie Adobe und Intel angehören, ist EMML noch nicht weit verbreitet.
IBM, ein weiterer großer Entwickler für Enterprise Mashup Lösungen, ist nicht in diesem Gremium vertreten. IBM führt als Hauptgrund für die Nichtteilnahme das verwendete Lizensierungsmodell an und weist auf für Software geeignetere Lizenzmodelle hin[9].

2.3.1 Funktionsumfang

EMML definiert, wie in Markup Languages üblich, einen Satz von Tags, die Abschnitte und Anweisungen kennzeichnen. Diese werden anschließend von einer serverseitigen Laufzeitumgebung verarbeitet. Eine von der OMA veröffentlichte Referenz-Implementation der Laufzeitumgebung in Java ist unter Downloads von Openmashup[10] erhältlich.

Nachfolgend werden einige Tags in Aufgabenbereiche zusammengefasst, um die Möglichkeiten von EMML zu veranschaulichen.

Kontrollstrukturen

Tag	Kurzbeschreibung
<if>	Prüfung von Bedingungen.
<for>	Zählschleife.
<foreach>	Schleife für jeden Datensatz.
<while>	Schleife gilt solange die Bedingung gilt.
<break>	Abbruch einer Schleife.
<parallel>	Anweisung für parallele Ausführung eines Teils.

[8] Frequently asked questions
[9] http://www.ibm.com/developerworks/forums/thread.jspa?messageID=14447718
[10] http://www.openmashup.org/download/

Datenbanken

Tag	Kurzbeschreibung
<sql>	SQL Select Abfrage oder Verwendung von Stored Procedures einer Datenbank, erwartet Rückgabewert.
<sqlupdate>	SQL Anweisung ohne Rückgabewert, z.B.: insert, update, delete.
<datasource>	Explizite Verbindung an eine Datenquelle per JDBC oder JNDI.
<sqlBeginTransaction>	Markiert den Beginn einer SQL Transaktion.
<sqlCommit>	Beendet eine SQL Transaktion und schreibt Änderungen fest.
<sqlRollback>	Beendet eine SQL Transaktion und verwirft Änderungen.

Datenmanipulationen

Tag	Kurzbeschreibung
<assign>	Zuweisung von Literalen, Variablen.
<filter>	Filterung einer Eingangsvariable nach einem Kriterium.
<group>	Analog zur „Group By“ Funktion in SQL, Berechnung und Aggregation mit „Count“, „Sum“, „Having“ möglich.
<sort>	Sortierung von Werten.
<annotate>	Hinzufügen von Knoten bei Variablen vom Typ „document“.
<script>	Einbinden von eigenem JavaScript oder JRuby Code.
<xslt>	Zur Verwendung von Stylesheets.
<join>	Join Operation ähnlich einem SQL Join.
<merge>	Zusammenführung von Daten mit gleicher Struktur, ähnlich einem SQL Merge.

Variablen und Deklarationen

Tag	Kurzbeschreibung
<emml-meta>	Tag für Meta Informationen, zurzeit ist nur “author” implementiert.
<input>	Aufrufparameter eines EMML Mashups oder Macros, eine beliebige Anzahl ist möglich.
<output>	Ausgabeparameter des EMML Mashups oder Macros.
<user-meta>	Selbstdefinierte Metadaten.

Datenquellen und Wiederverwendung von Services

Tag	Kurzbeschreibung
<include>	Einbinden von Macros.
<macro>	Tag für die Definition von Macros[11].
<invoke>	Aufruf von Mashups, die auf dem eigenen Server publiziert wurden.
<directinvoke>	Aufruf von öffentlich zugänglichen Webservices (andere als vom EMML Mashup Server), die eine REST-, SOAP-, RSS oder Atom-Schnittstelle haben. Im Gegensatz zur JackBe Presto EMML Laufzeitumgebung, fehlt dieser Tag in der OMA Referenzimplementation.

Debugging

Tag	Kurzbeschreibung
<assert>	Prüfung für Debugzwecke.
<display>	Anzeige von Debugging Informationen.

2.3.2 Technische Aspekte

Für die Analyse der technischen Aspekte von EMML wird folgendes Mashup umgesetzt:
Ein RSS Feed wird nach einem einzugebenden Stichwort gefiltert und ausgegeben.

Die Umsetzung des Mashups erfolgt auf zwei Wegen. Die erste Umsetzung verwendet den grafischen Mashup Editor JackBe Presto Wires, welches grafische Elemente verwendet um die Mashup-Erstellung zu erleichtern und den Datenfluss zu visualisieren. Die grafische Repräsentation wird im Hintergrund in EMML Code überführt. Die zweite Umsetzung erfolgt durch einen manuell erstellten EMML Code. Beide Mashups müssen dabei die gleiche Ausgabe erzeugen. Zum Test wird die JackBe EMML Laufzeitumgebung verwendet.

[11] Macros sind EMML Code Teile, die in Mashups wiederverwendet werden können; ein Macro kann kein Macro aufrufen.

EMML Code des Mashups

```
<mashup name=""
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openmashup.org/schemas/v1.0/EMML ../schemas/EMMLPrestoSpec.xsd"
        xmlns="http://www.openmashup.org/schemas/v1.0/EMML"
        xmlns:macro="http://www.openmashup.org/schemas/v1.0/EMMLMacro"
        xmlns:presto="http://www.jackbe.com/v1.0/EMMLPrestoExtensions" >

  <operation name="runMashup" >

    <namespaces></namespaces>

    <variable name="limitRecords" type="number" default="-2" />

    <output output-encoding="UTF-8"
            wires_id="output_0"
            name="output_0"
            type="document"
            service="" />

    <variable name="directinvoke_2_out" type="document" default="" />

    <directinvoke outputvariable="directinvoke_2_out"
                  endpoint="http://www.spiegel.de/schlagzeilen/tops/index.rss"
                  method="GET"
                  _Wires_name="directinvoke_2"
                  _Wires_id="directinvoke_2"
                  _Wires_label="Direct Invoke" />

    <input wires_id="inputparam_4"
           label="Inputparam_4"
           name="inputparam_4"
           type="string"
           default='computer' />

    <filter wires_id="filter_3"
            name="filter_3"
            inputvariable="directinvoke_2_out"
            outputvariable="output_0"
            filterExpr="/*:rss/*:channel/*:item[*:title[matches(.,$inputparam_4,&apos;i&apos;)]]" />

  </operation>
</mashup>
```

Abb. 6: Von JackBe Presto Wires generierter Code

```
<mashup xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openmashup.org/schemas/v1.0/EMML ../schemas/EMMLPrestoSpec.xsd"
        xmlns="http://www.openmashup.org/schemas/v1.0/EMML" >

  <directinvoke outputvariable="directinvoke_2_out"
                endpoint="http://www.spiegel.de/schlagzeilen/tops/index.rss"
                method="GET"/>

  <input name="inputparam_4"
         default='computer'/>

  <filter inputvariable="directinvoke_2_out"
          outputvariable="output_0"
          filterExpr="/*:rss/*:channel/*:item[*:title[matches(.,$inputparam_4,&apos;i&apos;)]]"/>

  <output name="output_0"/>

</mashup>
```

Abb. 7: Manuell geschriebener Code

Erläuterung verwendeter Tags

Verwendeter Tag	Kurzbeschreibung Verarbeitungsschritt
<directinvoke>	RSS Feed wird als Datenquelle eingebunden.
<input>	Eingabe für Filterkriterium.
<filter>	Definition, wonach gefiltert wird; Eingabe wird als Filterkriterium eingebunden.
<output>	Ausgabe

Der Vergleich des generierten Codes mit dem manuell erstellten Code lässt folgende Schlussfolgerungen zu:

- Die Reihenfolge der Tags ist nicht relevant. Daraus folgt, dass der Code zuerst komplett eingelesen und anschließend interpretiert wird.
- Eine Datenflusssteuerung, auch als Wiring bekannt, wird über „inputvariable“ und „outputvariable“ Attribute realisiert.
- Im von JackBe Presto Wires generierten Code werden Datentypen festgelegt. Im Gegensatz dazu werden im manuell erstellten Code keine verwendet. Da beide Mashups ohne Anzeige eines Fehlers das gleiche Ergebnis liefern lässt sich daraus schließen, dass die Festlegung des Datentyps optional ist.
- „wires_id“ Attribute sind nur im generierten Code enthalten. Da sowohl der manuell geschriebene Code als auch der vom Presto Wire erzeugte Code das gleiche Ergebnis liefern, kann daraus geschlossen werden, dass die „wires_id“ lediglich für die Visualisierung als „Wire“, den Verbindungslinien zwischen einzelnen Funktionsblöcken, benötigt wird.

EMML unterstützt nur die Erstellung von Mashups auf Datenebene. Für die Aufbereitung zur Präsentation des Mashup Ergebnisses bietet EMML keine Unterstützung. Diese Funktionalität wird dem Anbieter des Mashup-Entwicklungswerkzeuges überlassen.

2.4 Zusammenfassung

Neben der IT Politik und Philosophie in einem Unternehmen, entscheidet der Grad der technischen und grafischen Unterstützung die Verlagerung der Mashup-Entwicklung von der IT zur Fachabteilung. Nach dem Motto „der Anwender kennt seine Bedürfnisse am besten“, wird die Kreativität der Anwender erschlossen. Die IT übernimmt in diesem Fall Support-Aufgaben, leistet Hilfestellung bei der Entwicklung von Mashups und stellt Datenquellen unter Berücksichtigung des Datenschutzes zur Verfügung. Der IT obliegt die Abnahme in Hinsicht auf Sicherheit, Stabilität, Performance und der ressourcenschonenden Umsetzung der Mashups.

Mashups erleichtern das Erstellen von Protypen. Bei einer weitentwickelten Mashup Plattform kommt man in den Bereich der „Rapid Application Development“ (RAD). Durch die Möglichkeit, schnell Änderungen umzusetzen, kann man nach dem Prinzip der agilen Softwareentwicklung vorgehen. Bei kritischen Anwendungen können Prototypen als „Proof of Concept“ dienen und anschließend als Vorlage einer auf herkömmliche Weise erstellten Anwendung dienen.

Eine weitere Anwendungsmöglichkeit im Unternehmen ist die Erstellung einer einheitlichen und dem Arbeitsprozess optimierten Oberfläche, das sogenannte Streamlining. Man trifft in Unternehmen häufig auf „gewachsene“ IT-Strukturen, in der es viele Insellösungen gibt, deren Ablösung aus verschiedenen Gründen (z.B. Kosten, Legacy Anwendung, die auf alter Plattform bzw. Hardware läuft) nicht möglich ist. Ein Enterprise Mashup könnte eine Möglichkeit darstellen, eine einheitliche und optimierte Benutzer-Oberfläche zu erstellen, in dem es eine browserbasierte Oberfläche über vorhandene Applikationen „stülpt“.

Durch die Kombination verschiedener Datenquellen können neue Sichten auf Daten geschaffen werden, die Zusammenhänge visualisieren und auf diese Weise neue Erkenntnisse bzw. Schlüsse ermöglichen.

Konnektoren bzw. die Funktionalität der Datenanbindung einer Mashup Plattform gewähren dem Mashup Entwickler auf einfache Weise verschiedene Datenquellen anzubinden, ohne dass explizit Schnittstellen programmiert werden müssen.

Analog zu MS Access und MS Excel ermöglichen Mashup Plattformen mit einer grafischen Unterstützung beim Erstellungsprozess auch technisch weniger versierten Anwendern Mashups zu erstellen. Aufgewachsen mit Web 2.0 (s. Consumer Mashups) ist die kommende Generation mit Mashup Techniken vertrauter.

3. Vergleichbare Arbeiten

In diesem Kapitel werden verschiedene Mashup Entwicklungswerkzeuge vorgestellt und verglichen.

3.1 Mashup Entwicklungswerkzeuge

Trotz Bemühungen, wie die der Open Mashup Alliance, die Mashup Entwicklung zu vereinheitlichen, ist der Markt für Mashup Entwicklung inhomogen. Es gibt viele herstellerspezifische Lösungen. Nachfolgend werden vier Mashup Entwicklungswerkzeuge untersucht. Hauptaugenmerk liegt dabei auf folgende Kriterien:

- Datenquellen
- Unterstützung bei der Erstellung von Mashups
- Ausgabeformate und Präsentation
- Erweiterbarkeit
- Kosten

Datenquellen

Die Anzahl, Vielfalt und einfache Anbindung von Datenquellen bestimmen das Einsatzspektrum eines Mashups. Je vielfältiger und flexibler die Datenquellen einzubinden sind, desto einfacher ist es, Mashup Daten bereitzustellen. Im Unternehmensumfeld sind Web Services, Datenbanken, Excel Spreadsheet, CSV Daten, XML und Plain Text Files üblicherweise wichtige Datenquellen. In Sonderfällen kann Webclipping ebenfalls von Bedeutung sein (Ziegler, 2010).

Unterstützung bei der Erstellung von Mashups

Die Mashup Erstellung sollte möglichst einfach, intuitiv und flexibel sein.
Eine grafische Unterstützung sollte diese Punkte soweit wie möglich umsetzen.

Ausgabeformate und Präsentation

Ein wichtiger Aspekt ist die Unterstützung bei der Darstellung der aufbereiteten Daten. Eine ansprechende Darstellung hilft bei der Visualisierung und erhöht die Übersichtlichkeit der Daten. Elemente wie Buttons, Eingabefelder, etc. erhöhen die Vielseitigkeit und ermöglichen eine Interaktion. Für eine weitere Verarbeitung und Interaktion der Daten sind die möglichen Ausgabeformate von Bedeutung.

Erweiterbarkeit

Durch die Möglichkeit, eine Mashup Plattform auf die eigenen Bedürfnisse anzupassen, und um fehlende Funktionen zu erweitern, wird man flexibel und vom Hersteller unabhängig.

Kosten

Zu hohe Kosten können ein „KO"-Kriterium sein. ‚Sowohl die Anschaffungs- und Unterhaltskosten müssen kleiner als der zu erwartende Ertrag und Nutzen sein.

3.2 Yahoo Pipes

Yahoo Pipes ist ein Entwicklungswerkzeug für datenzentrierte Server-Side Consumer-Mashups, die auf Yahoo Servern gehostet und ausgeführt werden. Die Stärke von Yahoo Pipes ist der grafische Editor und dessen intuitive Bedienung. Yahoo Pipes bietet Funktionen zur Daten-Extraktion, -Manipulation und –Aggregation, und wird hauptsächlich zur Feed Aggregation und zum Einbinden von Bildern verwendet[12]. Funktionsblöcke und deren Verkettung (Wiring) werden als grafische Elemente mit Drag and Drop Funktionalität dargestellt. Solch eine Verkettung von Funktionsblöcken bezeichnet Yahoo, in Anlehnung an einer UNIX Pipe, als Pipe. Die grafischen Elemente werden auf Datenebene in JSON abgebildet und serverseitig verarbeitet. Die Reihenfolge der Verkettung spiegelt dabei die Reihenfolge der Verarbeitung wider. Programmiertechnisch ist es mit einer Verschachtelung von Funktionsaufrufen vergleichbar. Voraussetzung für die Nutzung des Dienstes ist eine Anmeldung bei Yahoo.

[12] http://pipes.yahoo.com/pipes/pipes.popular

3.2.1 Datenquellen

Als Datenquellen dienen hauptsächlich verschiedene Feedformate und Webclipping. Des Weiteren sind CSV Daten, Google Base, JSON und XML nutzbar. Eine Besonderheit sind die sogenannten Yahoo Query Language Tabellen (YQL). Diese sind von Yahoo eingebundene XML Dateien, die mit einer SQL ähnlichen Sprache angesprochen werden können. Yahoo stellt für einige ihrer Dienste (Flickr, Digg) Daten in dieser Form bereit.

3.2.2 Unterstützung bei der Erstellung von Mashups

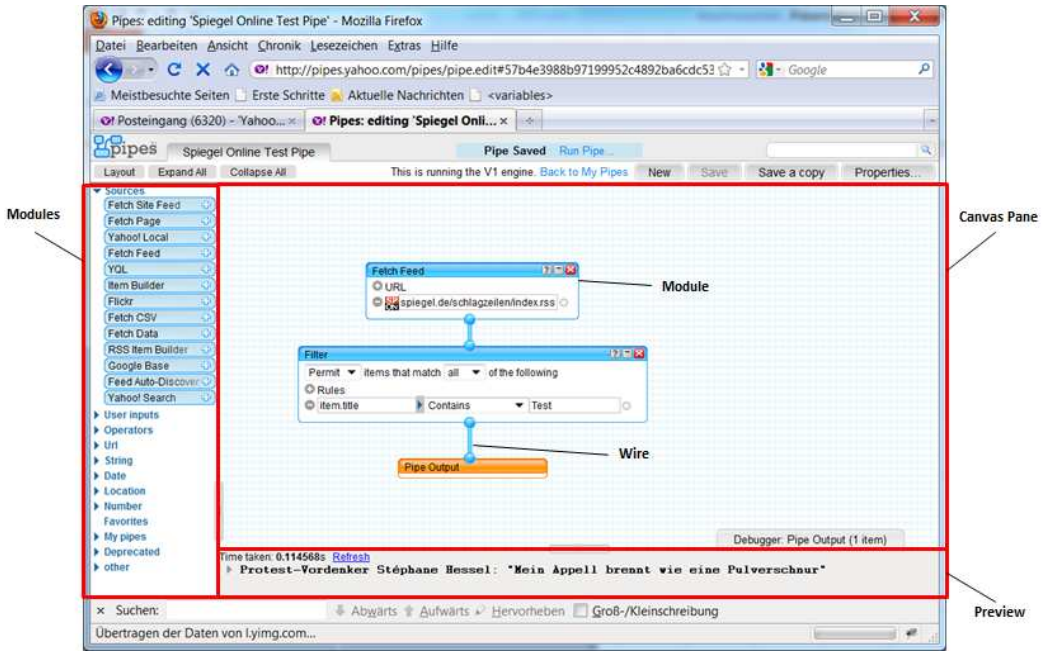


Abb. 8: Grafischer Editor von Yahoo Pipes

Zur Erstellung der Pipes bietet Yahoo ein Webbrowser basierendes IDE, den Yahoo Pipes Editor, an. Der Editor erlaubt es, dem Anwender per Drag and Drop Funktionsblöcke zu kombinieren und aneinander zu reihen. Jeder Funktionsblock repräsentiert einen Verarbeitungsschritt. Über Verbindungen (wires) zwischen den Funktionsblöcken wird der Datenfluss visuell dargestellt und die Verarbeitungsreihenfolge festgelegt. Die Verarbeitung kann mittels Parameterfelder in den Funktionsblöcken beeinflusst werden.

Ein- und Ausgänge an den Funktionsblöcken sind typisiert. Dies bedeutet, dass die Eingänge bestimmte Datentypen erwarten bzw. Ausgänge bestimmte Datentypen ausgeben. Die vorgegebenen Datentypen werden als Tooltip angezeigt. Um eine möglichst hohe Flexibilität zu erhalten, sind bei einigen Eingängen mehrere Datentypen möglich.

Jede Pipe besitzt einen definierten Endpunkt, dem „Output“-Funktionsblock, und mindestens einen Funktionsblock aus der Gruppe „Datenquellen“ (s.u.), der als Startpunkt dient. Des Weiteren können in bestimmten Funktionsblöcken, wie zum Beispiel dem Loop Funktionsblock, eine fertige Pipe eingebunden und somit das Ergebnis der Pipe wiederverwendet werden.

Um einen Eindruck zu vermitteln, was mit Yahoo Pipes möglich ist, erfolgt nachfolgend exemplarisch eine Auflistung der vorhandenen Module mit einer Beschreibung ihrer Funktionen. Eine vollständige Liste der Module mit Beschreibung und Beispielen befindet sich auf der Webseite von Yahoo.

Datenquellen

Zum Einbinden von Feeds als Datenquelle dienen folgende Module:
Fetch Site Feed
Fetch Feed
Feed Auto Discovery

Fetch CSV: Einbinden von CSV Daten
Fetch Data: Einbinden von XML, JSON, KML, iCal Datenquellen

Des Weiteren gibt es spezielle Module für das Anbinden von bestimmten Diensten als Datenquelle:
Flickr: Yahoo Bilder Dienst.
Yahoo Local: Yahoo Geo-Informationsdienst.
YQL: Per Yahoo Query Language ansprechbare Datenquellen.
Yahoo Search: Yahoo Suchergebnis als Datenquelle.
Google Base: Google Online Datenbank.

Anwender Eingaben

Yahoo Pipes bietet für Anwender Eingaben datentypabhängige (Datum, URL, Zahlen, Text, Geo Daten) Eingabe Module an.

Operatoren

Create RSS: Umstrukturierung von Datenstrukturen, um RSS konforme Struktur zu erzeugen.
Regex: Verwendung von Regulären Ausdrücken bei der Verarbeitung.
Tail / Truncate: Liefert die letzten bzw. ersten Elemente.
Web Service: Sendet Pipedaten in JSON an externe WS, erwartet JSON Rückgabewert
Loop: Daten vorhandener Pipes können mit diesem Modul wiederholt bearbeitet werden.

Des Weiteren bietet Yahoo Pipes weitere Module zur Filterung, Zusammenfassung, Sortieren, Aufteilung, Zählen und Umbenennung von Daten. Speziell für die Manipulation hält Yahoo Pipes eine Reihe von Modulen bereit. Beispielsweise gibt es Module für die Ersetzung, Konkatinierung, Kürzung und Vereinzelung von Zeichenketten. Für komplexe Bearbeitung von Zeichenketten können reguläre Ausdrücke verwendet werden.

3.2.3 Ausgabeformate und Präsentation

Um das verarbeitete Ergebnis darzustellen, stellt Yahoo sogenannte Badges bereit. Dies sind für die Darstellung auf einer Webseite vordefinierte Elemente, die an das Layout der Seite angepasst werden können[13].

Es werden momentan drei Typen von Badges bereitgestellt:

- List Badge zur Darstellung von Auflistungen.
- Image Badge zur Darstellung von Bildern mit Slideshow und anderen Funktionen.
- Map Badge zur Darstellung von Geo Koordinaten in einer Karte.

[13] http://pipes.yahoo.com/pipes/badgedocs

Fertige Pipes kann man mit der „publish“ Funktion der Öffentlichkeit zur Verfügung stellen. Des Weiteren lässt sich in veröffentlichten Pipes nach bestimmten Modulen suchen und deren Verwendungsweise einsehen.

3.2.4 Erweiterbarkeit

Yahoo Pipe ist ein von Yahoo gehosteter Dienst. Der Anwender ist von den durch Yahoo angebotenen Funktionen abhängig. Eine Erweiterbarkeit von Yahoo Pipe ist nicht vorgesehen. Diese Einschränkung kann man theoretisch durch Erstellen von Yahoo Pipes konsumierbaren Services auf eigenen bzw. anderen Servern umgehen. Eine Erweiterung der Kernfunktionen ist nicht möglich.

3.2.5 Kosten

Momentan ist Yahoo Pipes noch kostenlos erhältlich. Lediglich die Eröffunung eines Yahoo Accounts ist notwendig.

3.2.6 Architektur

Die Dokumentation der Architektur und Implementierung von Yahoo Pipes ist nicht öffentlich zugänglich. Anfragen bezüglich der Architektur und Implementierung werden in Foren nicht beantwortet[14]. Analysen der Kommunikation zwischen Browser und Yahoo Server mit Firebug haben ergeben, dass die im Pipe Editor erstellten Objekte (modules) und Beziehungen (wires) in JSON definiert sind[15]. Diese Definitionen werden vom Yahoo Server interpretiert und verarbeitet. Einen in Python noch unvollständig implementierter Python Pipes Compiler ist unter https://github.com/ggaughan/pipe2py erhältlich[16].

[14] http://discuss.pipes.yahoo.com/Message_Boards_for_Pipes/threadview?bn=pipgenDiscuss&tid=3012&mid=3012
[15] http://blog.ouseful.info/2010/02/25/starting-to-think-about-a-yahoo-pipes-code-generator/
[16] http://www.wordloosed.com/running-yahoo-pipes-on-google-app-engine

Der Python Pipes Compiler erstellt nach dem Parsen der JSON Module-Definitionen Python Code, der eine Verschachtelung von Funktionsaufrufen nachempfunden ist. Jeder Funktionsaufruf repräsentiert einen Modul und die Verschachtelung der Aufrufe das Wiring. Auf diese Weise wird ein modularer Aufbau und lose Kopplung der Module realisiert.

3.2.7 Yahoo Pipe Beispiel Mashup

Ein RSS Feed wird nach einem Stichwort, welches einem Eingabefeld entnommen wird, gefiltert und ausgegeben.

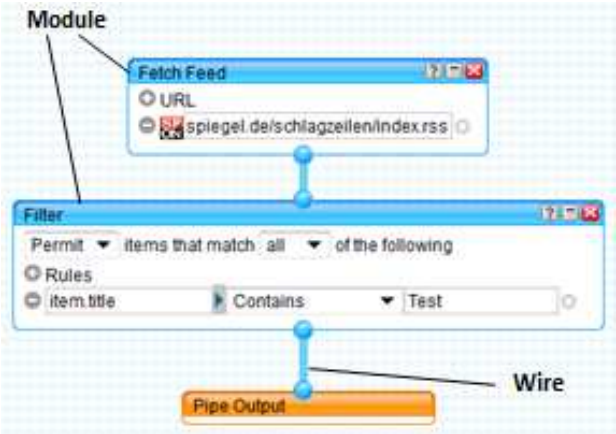


Abb. 9: Pipe Beispiel: Feed nach Stichwort gefiltert

Output(Filter(Fetch(URL), Criteria))

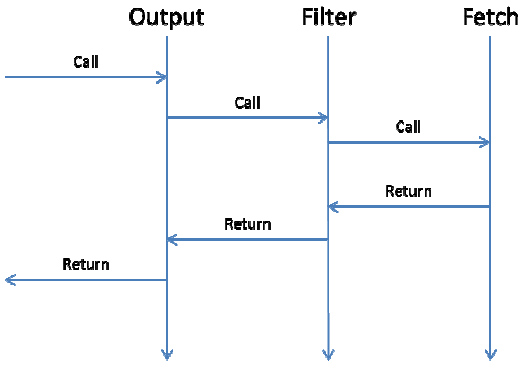


Abb. 10: Ablaufdiagramm: Wiring als Funktionsschachtelung

JSON Code des Beispiels:

Fetch Module

{"type":"fetch","id":"sw-61",
"conf":{"URL":{"value":"http://www.spiegel.de/schlagzeilen/index.rss","type":"url"}}}

Filter Module

{"type":"filter","id":"sw-76",
"conf":{"MODE":{"type":"text","value":"permit"},
"COMBINE":{"type":"text","value":"and"},
"RULE":[{"field":{"value":"title","type":"text"},
"op":{"type":"text","value":"contains"},
"value":{"value":"Test","type":"text"}}]}}

Output Module

{"type":"output","id":"_OUTPUT",
"conf":{}}

Wire Fetch → Filter

{"id":"_w1","src":{"id":"_OUTPUT","moduleid":"sw-61"},
"tgt":{"id":"_INPUT","moduleid":"sw-76"}}

Wire Filter → Output

{"id":"_w3","src":{"id":"_OUTPUT","moduleid":"sw-76"},
"tgt":{"id":"_INPUT","moduleid":"_OUTPUT"}}

Darüber hinaus gibt es noch einen Layout Teil, der die XY-Positionen der einzelnen Module für die Darstellung im Yahoo Pipe Editor beinhaltet.

3.2.8 Zusammenfassung Yahoo Pipes

Yahoo Pipes ist als Server-Side Consumer Mashup Lösung konzipiert und wird hauptsächlich für Feed Aggregation und Einbinden von Bildern verwendet[17]. Die Stärke von Yahoo Pipes ist der grafische Editor und die intuitive Bedienung. Yahoo Pipes senkt durch den grafischen Editor die Einstiegshürde für die Erstellung von Mashups erheblich. Für erste Schritte ist Drag and Drop und das visuell dargestellte Wiring intuitiv und hilfreich, um spielerisch die ersten Schritte zu machen und schnell Erfolge zu erzielen. Viele Beispiele bzw. vorhandene

Pipes anderer Nutzer können als Vorlage für die Entwicklung eigener Pipes dienen. Durch das Hosting durch Yahoo entfällt die Notwendigkeit für eine eigene Server-Infrastruktur. Dies bedeutet allerdings nicht, dass man kein technisches Verständnis braucht. Auch technisch versierte Nutzer brauchen Einarbeitungszeit, um komplexere Pipes zu erstellen[18]. Ein gewisses Maß an technischem Verständnis und Hintergrundwissen darüber, wie Webapplikationen und Protokolle arbeiten, ist für die Extraktion von Daten und Erstellung von komplexeren Pipes notwendig. Für die Einbindung der Pipe in die eigene Webpräsenz sind, trotz Yahoo-seitiger Unterstützung durch Badges, zumindest rudimentäre HTML Kenntnisse notwendig.

Die Einsteigerfreundlichkeit wird aber auf der anderen Seite durch eine Limitierung der Möglichkeiten und eine Abhängigkeit von Yahoo erkauft. Es sind nur Funktionen möglich, die von Yahoo angeboten werden. Eine optionale Erweiterung ist nicht möglich. Weitere Funktionen muss man außerhalb von Yahoo Pipes entwickeln und als Quelle an Yahoo Pipes anbinden oder die Pipe extern per Mashup einbinden.

Ein weiterer Aspekt ist die Abhängigkeit von der Yahoo-Infrastruktur. Man muss sich nach den von Yahoo frei angebotenen Leistungen richten. Eine Skalierung der Serverleistung ist nicht möglich, auch nicht optional wie beim Google App Engine. Des Weiteren ist man auf das Yahoo Hosting angewiesen. Sollte der Dienst eingestellt werden, so sind die erstellten Pipes nicht mehr nutzbar. Die Python Open Source Lösung pipe2py ist momentan noch kein vollwertiger Ersatz. Die Vergangenheit hat gezeigt, dass freie Mashup Lösungen auch großer Software Unternehmen wie von Google, Microsoft und IBM eingestellt wurden (Microsoft Popfly, IBM Lotus Mashup, Google Mashup Editor) oder in ein Bezahlmodell überführt wurden (Google App Engine als PAAS). Vor diesem Hintergrund und im Anbetracht des rückläufigen Marktanteils von Yahoo ist die Zukunft von Yahoo Pipes ungewiss. Die Ankündigung von der Pipes Engine V2 stammt aus Juni 2010[19]. Seit diesem Datum ist keine weitere News zu Yahoo Pipes veröffentlicht worden. Momentan ist über Kundenbindung hinaus kein Geschäftsmodell ersichtlich. Bei Fremdhosting muss man zudem immer den Aspekt Datenschutz beachten. Yahoo ist wie Google ein Unternehmen, dass Daten über Personen und Unternehmen sammelt und zu Werbezwecken vermarktet.

---

[17] http://pipes.yahoo.com/pipes/pipes.popular
[18] http://www.masternewmedia.org/news/2007/02/09/beyond_newsmastering_yahoo_pipes_is.htm
[19] http://blog.pipes.yahoo.net/2010/06/09/yahoo-pipes-v2-engine/

3.3 JackBe Presto

JackBe bietet mit Presto eine umfangreiche Plattform für die Erstellung von Server-Side Enterprise Mashups an[20]. JackBe Presto unterstützt den Prozess von der Datenbereitstellung und -aufbereitung bis zur Präsentation des fertigen Mashups. Ferner besitzt JackBe Presto eine einfache Berechtigungsverwaltung der Mashups, die den Zugang der Mashups regelt. Über einen Mashup App Store können Mashup Entwickler ihre Mashups präsentieren und optional zum Kauf anbieten. Diese Mashups können anschließend direkt verwendet werden oder in eigene Mashups eingebunden werden. Als Entwickler von EMML ist JackBe einer der wenigen Anbieter, die EMML in ihren Mashups implementiert haben.

Nachfolgend wird die Presto 3.0 Cloud Edition (CE) untersucht und beschrieben. Diese Version ist zur Evaluation für 60 Tage kostenlos verwendbar.

3.3.1 Datenquellen:

Bei Presto 3.0 CE lassen sich Wizard-gestützt folgende Quellen anbinden:
Feeds, Web Services, CSV und XML Dateien, Datenbanken, Sharepoint, Gadgets und Excel Spreadsheet/Tabelle.

Über ein Extract Block lassen sich Daten aus Webseiten, die keine andere API anbieten, per Web Clipping extrahieren. Darüber hinaus lassen sich bereits erstellte Mashups einbinden, Mashups von anderen Entwicklern lassen sich suchen und mit entsprechender Berechtigung einbinden. Es steht ebenfalls ein App Store für Mashups bereit, in der nach vorhandenen Presto Mashups gesucht werden kann.

Hervorzuheben ist beim Presto 3.0 CE das Excel Add-In „Excel Presto Toolbar“, mit der Excel Daten direkt in einem Mashup verarbeitet werden können bzw. Mashup Daten direkt in ein Excel Spreadsheet geladen werden können. Diese Funktionalität ähnelt einer SAP BW Anbindung.

[20] Im Weiteren als JackBe Presto genannt.

3.3.2 Unterstützung bei der Erstellung von Mashups

JackBe bietet mit Presto Wires eine grafische Unterstützung zur Erstellung von Mashups an. Presto Wires hat große Ähnlichkeiten mit Yahoo Pipes.

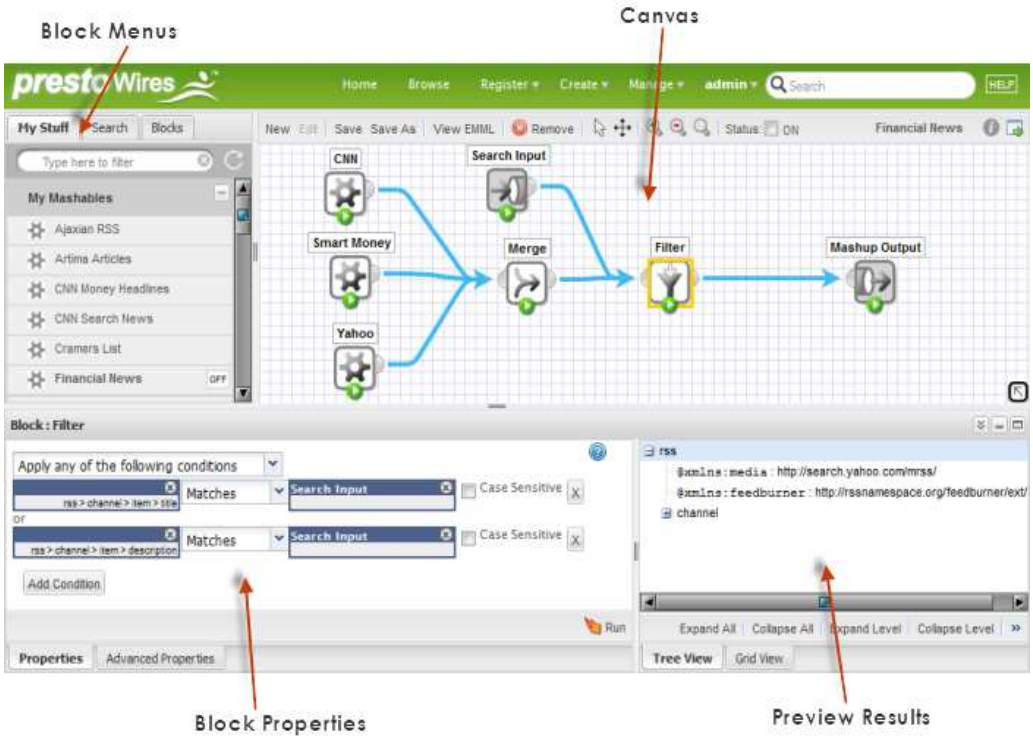


Abb. 11: Bedienoberfläche von JackBe Presto Wires[21]

Wie in Yahoo Pipes repräsentieren JackBe Presto Wires grafische Elemente, bei Presto Wires „Blocks“ genannt, die Datenbereitstellung, Datenverarbeitung und Datenausgabe. Die Blocks können auf einer „Canvas“ genannten Fläche per Drag and Drop angeordnet und mit Pfeilen verbunden (Wiring) werden, die den Datenfluss visualisieren. Die Eigenschaften der grafischen Elemente können im Feld „Block Properties“ festgelegt werden. Auch hierbei erhält der Anwender durch Presto Wires Unterstützung. Sobald der Input eines Blocks verbunden ist und ein Testlauf des vorangehenden Blocks durchgeführt wurde, stehen für die Eigenschaftsfelder mögliche Auswahlkriterien in einer Drop Down Liste zur Verfügung.Auf diese Weise wird das Konfigurieren der Eigenschaften erheblich erleichtert. Zur laufenden Kontrolle der Verarbeitung wird das Ergebnis in einer Vorschau angezeigt. Auf der JackBe Presto Plattform bereits vorhandene eigene Mashups und freigegebene fremde Mashups werden in einer Liste angeboten und können wie gewöhnliche Elemente per Drag and Drop angeordnet und mit Pfeilen verbunden werden. Auf diese Weise wird eine Wiederverwendung erreicht.

View EMML

Presto Wires generiert aus den grafischen Elementen EMML Code, der über die Funktion View EMML eingesehen und kopiert werden kann. Diese Funktion erlaubt es den Entwickler, schnell und komfortabel ein funktionierendes EMML Code-Gerüst zu erstellen, das anschließend im Presto Editor angepasst und erweitert werden kann.

Presto Editor

Neben Presto Wires, dem grafischen Editor, lassen sich Mashups im Presto Editor erstellen. Presto Wires ist eine browserbasierte rudimentäre Entwicklungsumgebung, um Mashups direkt in EMML Code zu schreiben.

Abb. 12: Presto Mashup Editor

Dem Entwickler wird eine Liste mit EMML Anweisungen bereitgestellt, welche nach einem Klick im Texteditor entsprechende EMML Tags erzeugen. Diese Tags müssen anschließend mit den notwendigen Eigenschaften gefüllt werden. Des Weiteren unterstützt der Texteditor EMML Syntax Highlighting. In einem separaten Bereich des Editors kann das Ergebnis in einer Vorschau angezeigt werden.

Die direkte Entwicklung von Mashups in Textform erlaubt es dem Entwickler Funktionen zu verwenden, die in Presto Wires nicht implementiert sind bzw. diese flexibler zu verwenden. Die Liste der verfügbaren EMML Anweisungen zeigt, dass die Presto EMML Implementation gegenüber der OMA Referenz Implementation fortgeschrittener ist und erweitert wurde. Zum Beispiel wird die „Invoke“ Anweisung in der Referenz Implementierung nur für zukünftige Zwecke erwähnt, während JackBe diese im eigenen Produkt bereits implementiert hat.

[21] http://www.jackbe.com/prestodocs/v3.0/wires/wires-designer.html

Presto Mashup Studio

Presto Mashup Studio ist ein von JackBe bereitgestelltes Plug-In für Eclipse. Diese Entwicklungsumgebung wird von JackBe zur Erstellung von Mashups empfohlen.

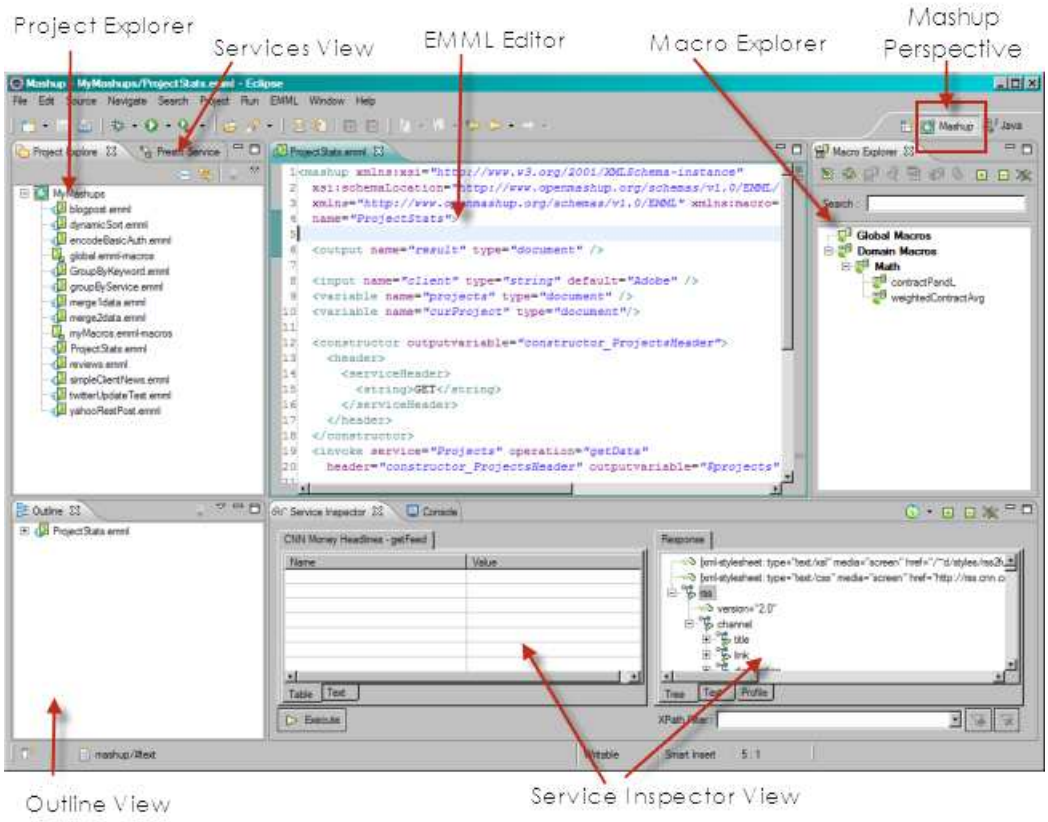


Abb. 13: Presto Mashup Studio

3.3.3 Ausgabeformate und Präsentation

In Presto werden die Präsentations Widgets „Views“ genannt. Die Views sind mit den Badges in Yahoo Pipes vergleichbar. Presto bietet jedoch mit 27 Views eine größere Auswahl an. Die Views sind in folgende fünf Kategorien eingeteilt:

- Tabular: Verschiedene Tabellen
- Maps: Visualisierung von GEO Informationen, z.B: in Google Maps
- Charts: Verschiedene Diagramm Typen
- Sparks: Darstellung von Wertveränderung in Abhängigkeit von der Zeit
- Gauges: Verschiedene grafische Anzeigen für Stärken, z.B.: Thermometer

3.3.4 Erweiterbarkeit

Eine Erweiterung durch den Anwender ist von JackBe nicht vorgesehen. Der Anwender ist von den angebotenen Funktionen abhängig. Diese Einschränkung kann man theoretisch durch Erstellen von Presto konsumierbaren Services auf eigenen bzw. anderen Servern umgehen. Eine Erweiterung der Kernfunktionen ist allerdings nicht möglich.

3.3.5 Kosten

Lizenzkosten für JackBe Presto werden nur bei direkter Anfrage mitgeteilt.
Eine Lizenz für eine CPU wird mit 40.000 US$ pro Jahr angegeben[22].

[22] http://www.infoworld.com/d/developer-world/jackbe-readies-presto-wires-mashup-composer-461

3.3.6 Zusammenfassung JackBe Presto

JackBe Presto ist eine umfangreiche Enterprise Mashup Lösung. Der Anwender wird sowohl bei der Anbindung von verschiedenen Ressourcen unterstützt als auch bei der anschließenden Erstellung des Mashups. Die Einarbeitungszeit für JackBe Presto ist größer als bei Yahoo Pipes. Es bietet allerdings auch einen größeren Funktionsumfang. Yahoo Pipes ist mit der Presto Wires Komponente vergleichbar. JackBe Presto ist nicht so intuitiv bedienbar wie Yahoo Pipes. Es werden zum Beispiel die Presto Komponenten nicht dem Workflow entsprechend von links-nach-rechts angeordnet, sondern in umgekehrter Reihenfolge von rechts-nach-links. Bestandteile wie Berechtigungsverwaltung sind in Untermenüs hinterlegt, so dass ein direkter Zugang erschwert wird. Ein besonderes Merkmal im Hinblick der Nutzung im Unternehmensumfeld ist die Möglichkeit direkt aus Excel heraus, Daten bereitzustellen und in ein Mashup einzubinden.

3.4 IBM Mashup Center

IBM Mashup Center ist eine Enterprise Mashup Plattform, welche sowohl datenorientierte als auch präsentationsorientierte Mashups erstellt. IBM Mashup Center ist eine sehr weit entwickelte, komfortable und in die IBM Produktpalette integrierte Mashup Plattform. Durch die Verflechtung mit anderen IBM Produkten kann IBM Mashup Center auf die Funktionalitäten der anderen Produkte zurückgreifen. Als Beispiele können hierbei „Websphere Information Server“ für Datenanbindungen und „Webphere Portal“ für Portal Integration angeführt werden. Mit „Lotus Widget Factory“ stellt IBM ein mächtiges Werkzeug bereit, um das IBM Mashup Center mit weiteren Komponenten zu erweitern. Weitere Funktionen wie umfangreiche Rechteverwaltung unterstreichen den Enterprise Character.
Die Trial Version von IBM Mashup Center 2.0 ließ sich nicht auf der Trial Version von Microsoft Server 2008 installieren. Mangels Software-Verfügbarkeit erfolgt eine Bewertung des IBM Mashup Centers auf Basis von Tutorial Videos, die IBM in Youtube veröffentlicht hat[23].

[23] http://www.youtube.com/user/ItsMashtastic

3.4.1 Datenquellen

IBM bietet eine große Vielfalt an Datenquellen an. Unter anderem können folgende Datenquellen angebunden werden: Excel Spreadsheets, Access Datenbanken, Feed Daten, CSV- und XML-Dateien, Datenbanken, Web Services, SAP und LDAP Directory.

3.4.2 Unterstützung bei der Erstellung von Mashups

IBM Mashup Center bietet eine Unterstützung für datenorientierte und präsentationsorientierte Mashups. Drag and Drop, Wizards, Tooltips und geeignete Voreinstellungen erleichtern das Erstellungsprozess.

Der Data Mashup Editor ist dem Presto Wires bzw. Yahoo Pipes ähnlich. Es ist ein grafischer Editor für Mashups auf Datenebene. Blöcke symbolisieren die Verarbeitungsschritte, während Verbindungslinien den Datenfluss darstellen. Analog zu Presto Wires und Yahoo Pipes gibt es auch hier einen definierten Endpunkt. Wie in Presto Wired wird das Konfigurieren der Eigenschaftsfelder über Auswahlfelder erleichtert.

Darüber hinaus bietet der IBM Mashup Center auch eine Möglichkeit für die Erstellung von Mashups auf Präsentationsebene. Verschiedene Widgets können auf einer Oberfläche angeordnet und die Widgets per Pub/Sub Wiring miteinander verbunden werden. Sollte keine der vorhandenen Widgets geeignet sein, so steht dem Anwender mit der Lotus Widget Factory ein Werkzeug zur Erstellung von eigenen Widgets zur Verfügung.

3.4.3 Ausgabeformate und Präsentation

Zur Präsentation stehen dem Anwender viele verschiedene Widgets zur Verfügung. Als Beispiel sind Formelemente, Diagramme, Landkarten, Google Gadgets, Diashow, Bildbetrachter und selbsterstellte Widgets zu nennen. Ferner kann der Anwender über Layout Typen, Themes, Styles und Farben Einfluss auf die Gestaltung der Seite nehmen. Fertige Seiten bzw.

Teile können einfach als automatisch generierter HTML Tag in anderen HTML Seiten oder Portalseiten eingebunden werden.

3.4.4 Erweiterbarkeit

Mit der Lotus Widget Factory steht dem Anwender ein mächtiges Werkzeug zur Erstellung eigener Widgets bereit. Hierbei handelt es sich um eine Eclipse-basierte Entwicklungsumgebung für Widgets. Ein Repertoire von über ein hundert vorgefertigten Funktionsblöcken erleichtert dem Anwender die Erstellung von Widgets erheblich.

3.4.5 Kosten

Pro “Processor Value Unit” (PVU) kostet der IBM Mashup Center 467 US$. Der Faktor für einen Single Core Prozessor liegt bei 100. Die Kosten für einen Single Core Server liegen somit bei 46.700 US$[24].

Eine detaillierte Auflistung der PVU ist auf der IBM Webseite erhältlich[25].

3.4.6 Zusammenfassung IBM Mashup Center

IBM zeigt mit IBM Mashup Center eindrucksvoll die Möglichkeiten von Mashup Editoren. Der Anwender wird von vielen Wizards und Voreinstellungen unterstützt und bleibt über selbsterstellbare Widgets flexibel. Lediglich der hohe Preis und hohe Systemvoraussetzung stehen einer weiten Verbreitung im Weg.

[24] Stand Januar 2011, http://www.youtube.com/user/ItsMashtastic
[25] http://www-01.ibm.com/software/lotus/passportadvantage/pvu_licensing_for_customers.html

3.5 WSO2 Mashup Server

Der WSO2 Mashup Server ist eine Enterprise Mashup Lösung zur Erstellung von Server Side Mashups. Es basiert auf verschiedenen Open Source Projekten und ist in der Apache License Version 2.0 freigegeben. Als Entwicklungssprache für die Mashups wird Server Side JavaScript verwendet. Dabei wird der JavaScript Interpreter Mozilla Rhino mit E4X Unterstützung verwendet. Als SOA Plattform ist der WSO2 Mashup Server sehr modular aufgebaut und entsprechend diesem Prinzip werden alle Mashups als Webservice angeboten. Die veröffentlichten Mashup Webservices können in weiteren Mashups konsumiert werden, so dass ein Mashup eine Orchestration von Webservices darstellt. Mit JavaScript Host Objects kann die Funktionalität des Mashup Servers erweitert werden. Dies setzt allerdings weitreichende Kenntnisse in JavaScript und Java Programmierung voraus. Im Auslieferungszustand sind nur einige grundlegende JavaScript Host Objects implementiert.

3.5.1 Datenquellen

Als Datenquellen stehen folgende Host Objects zur Verfügung:

- Scraper: auf Webharvest basierendes Host Object zur Unterstützung bei Webharvesting
- APP: Unterstützung für Atom Publishing Protocol
- File: Lesen, Schreiben und Löschen von Dateien auf dem Server im Verzeichnis des Webservices
- Feed: Verwendung von RSS und Atom Feeds
- WSRequest: Unterstützung bei nicht selbstgehosteten Webservices

Des Weiteren lassen sich weitere Server/Features der anderen WSO2 Server Produkte einbinden. Von besonderem Interesse ist der Data Service Server. Dieser Server bietet weitere Möglichkeiten um andere Datenquellen (z.B.: Datenbanken, Dateien) anzubinden.

3.5.2 Unterstützung bei der Erstellung von Mashups

Der Mashup Server bietet als einzige Unterstützung die automatische Generierung von Stubs an, um das Konsumieren der selber gehosteten Webservices zu erleichtern. Darüber hinaus erhält der Anwender keine weitere Unterstützung zur Mashup Estellung. Ein grafischer Mashup Editor, analog zu den anderen betrachteten Mashup Werkzeugen, fehlt völlig.

3.5.3 Ausgabeformate und Präsentation

In dieser Kategorie gibt es nur wenig Unterstützung seitens des WSO2 Mashup Servers. Der WSO2 Mashup Server bietet lediglich unter einem als „Customized UI“ bezeichneten Punkt einen rudimentären HTML Editor mit Syntax Highlighting an, mit der eine eigene HMTL Seite zur Darstellung der Mashup Daten erstellt und hinterlegt werden kann. Auf diese Weise hat man die Möglichkeit, die Darstellung nach eigener Vorstellung umzusetzen. Diese HTML Seite muss aber bis auf einige Header- und Footer Informationen mit Hinweisen zu WSO2 komplett selbst geschrieben werden. Ein Wizard oder GUI wird nicht geboten. Sollte die Einbindung des Data Service Servers gelingen, so würde als Ausgabemedium auch eine Datenbank zur Verfügung stehen.

3.5.4 Erweiterbarkeit

Der WSO2 Mashup Server ist modular aufgebaut und lässt sich per JavaScript Host Objects erweitern.

3.5.5 Kosten

Für den WSO2 Mashup Server fallen keine Lizenzkosten an. Es sind aber viele Eigenentwicklungen notwendig, welche Entwicklerkapazitäten und somit indirekt Kosten verursacht. Die

Entwicklergemeinde ist klein und es gibt nur wenige Ressourcen bei Problemen. Das Geschäftsmodell beruht auf einem kostenpflichtigen Support.

3.5.6 Zusammenfassung WSO2 Mashup Server

Der WSO2 Mashup Server basiert als einzige Mashup Lösung komplett auf Open Source und es fallen keine Lizenzkosten an. Durch die Verwendung der eigenen Infrastruktur hat man die volle Kontrolle über die Daten. Der Server lässt sich durch die Erweiterbarkeit auf eigene Bedürfnisse anpassen. Allerdings sind viele Funktionen, wie grafische Unterstützung bei der Mashup Erstellung und Ausgabe-Templates, gar nicht implementiert. Benötigte Funktionen müssen selbst entwickelt und getestet werden. Dies setzt Knowhow und Zeit voraus.

3.6 Zusammenfassung

Die Mashup Entwicklungswerkzeuge unterstützen den Anwender unterschiedlich gut bei der Erstellung von Mashups. Je besser ein Programm den Anwender bei der Mashup Entwicklung unterstützt, desto niedriger ist die Hürde für den Anwender und desto höher ist die Akzeptanz das Programm zu verwenden. Bei einer idealen Unterstützung sollten keine besonderen technischen Kenntnisse seitens des Anwenders notwendig sein.

Yahoo Pipes, JackBe Presto und IBM Mashup Center erleichtern durch einen grafischen Editor für Mashups auf Datenebene den Zugang zur Mashup Erstellung für technisch weniger versierte User wie den Fachabteilungen. Insbesondere IBM Mashup Center demonstriert eindrucksvoll, wie ein grafischer Editor und Wizards den Erstellungsprozess vereinfachen. Darüber hinaus hat das IBM Mashup Center als einziger Anbieter einen grafischen Editor für Mashups auf Präsentationsebene.

JackBe Presto und das IBM Mashup Center bieten eine Vielzahl an Konnektoren, um Datenquellen anzubinden und zu verwenden. Im Vergleich fällt die Anzahl der Konnektoren bei Yahoo Pipes und dem WSO2 Mashup Server gering aus. Während der WSO2 Mashup Server

durch Eigenentwicklungen weitere Datenquellen anbinden kann, ist diese Möglichkeit bei Yahoo Pipes nicht gegeben.

Yahoo Pipes, JackBe Presto und IBM Mashup Center bieten vorgefertigte Widgets zur Präsentation der Datenmashups an. Bei Yahoo Pipes ist die Anzahl mit drei vorgefertigten Widgets gering. JackBe Presto bietet mit 27 vorgefertigten Widgets eine größere Auswahl an. Das IBM Mashup Center bietet mit der Lotus Widget Factory, über die vorgefertigten Widgets hinaus, noch ein Werkzeug zur Entwicklung von eigenen Widgets. In diesem Gebiet gibt es keine Unterstützung seitens des WSO2 Mashup Servers.

JackBe Presto und IBM Mashup Center bieten als Enterprise Mashup Plattformen eine umfangreiche Unterstützung bei dem Erstellungsprozess von Mashups, wobei der IBM Mashup Center besonders heraussticht. Von der Anbindung von Datenquellen bis zur Präsentation der Mashups wird der Nutzer vom IBM Mashup Center unterstützt.

JackBe Presto und IBM Mashup Center lassen sich den hohen Entwicklungskomfort gut bezahlen. Die Kosten im Bereich von über 40 TEUR bedeuten für kleinere Unternehmen eine hohe Investition. Yahoo Pipes, als kostenloses Mashup Werkzeug mit grafischem Editor, lässt sich nicht auf eigene Bedürfnisse erweitern.

Der WSO2 Mashup Server bietet keinen grafischen Editor an. Als quelloffene Software fallen für den WSO2 Mashup Server aber keine jährlichen Lizenzkosten an. Das Unternehmen ist dadurch vor stetig steigenden Lizenzkosten geschützt. Mit der Erweiterbarkeit durch eigene JavaScript Host Objects kann der WSO2 Mashup Server um neue Funktionen erweitert werden. Die Möglichkeit zur Anbindung an andere ebenfalls kostenlose WSO2 Produkte wie Data Service Server ist ein weiterer Vorteil. Die Schwächen vom WSO2 Mashup Server liegen in der schlechten Unterstützung beim Erstellungsprozess. Kenntnisse in JavaScript ist die Voraussetzung, um Mashups zu erstellen. Mit der Erweiterbarkeit des WSO2 Mashup Servers besteht die Möglichkeit, den Server um zusätzliche Funktionen, wie einem grafischen Editor, zu erweitern, die den Anwender bei der Erstellung von Mashups unterstützen. Auf diese Weise wird der WSO2 Mashup Server anwenderfreundlicher und somit zugänglicher für Anwender ohne technische Kenntnisse.

Vergleichstabelle der Mashup Werkzeuge

Nachfolgende Tabelle fasst die hier diskutierten Ergebnisse zusammen.

Kriterien	Yahoo Pipes	JackBe Presto	IBM Mashup Center	WSO2 Mashup Server
Datenquellen	O	+	++	O
Mashup Unterstützung	O	+	++	-
Präsentation	O	+	++	--
Erweiterbarkeit	--	--	+	+
Kosten	++	--	--	++

Abb. 14: Vergleich der Mashup Entwicklungswerkzeuge: Bewertungstabelle

++ sehr gut, + gut, O neutral, - schlecht, -- sehr schlecht

4. Analyse

Dieses Kapitel betrachtet die Möglichkeiten den WSO2 Mashup Server so zu erweitern, dass auch technisch nicht versierte Anwender in die Lage versetzt werden, Mashups zu erstellen.

Das Kapitel ist in zwei Teilen aufgeteilt. Der erste Teil betrachtetet, unabhängig von den zur Verfügung stehenden Ressourcen, die Merkmale, die in einer Mashup Entwicklungsumgebung realisiert werden sollten. Diese werden in drei Kategorien eingeteilt:

- Allgemeine Merkmale einer Entwicklungsumgebung,
- Merkmale für Mashups auf Datenebene,
- Merkmale für Mashups auf Präsentationsebene.

Der Erstellungsprozess soll durch Wizards und Vorgaben von validen Eingabemöglichkeiten maximal unterstützt werden und minimale Eingriffe des Anwenders erfordern. Als Vorbild dienen die technisch ausgereiften kommerziellen Produkte, vor allem das IBM Mashup Center. Der zweite Teil erörtert detailliert die Umsetzung eines dieser Merkmale, nämlich die grafische Unterstützung bei der Erstellung von Mashups auf Datenebene.

Im Rahmen dieser Arbeit wird der WSO2 Mashup Server um die Möglichkeit, Mashups auf Datenebene grafisch zu erstellen, erweitert (vgl. Kapitel 5). Als „Proof of concept“ wird ein Evaluierungs-Prototyp erstellt mit dem ein Mashup, analog zu den Beispiel-Mashups aus den vorherigen Kapiteln, erstellt werden kann. Dieses Mashup wird folgende Funktionalitäten beinhalten:

- Feed einlesen
- Titel nach einem bestimmten Stichwort filtern

Beispielhaft werden grafische Funktionsblöcke für eine Feed Datenquelle, einem Filter und einer Ausgabe implementiert. Des Weiteren soll ein grafisches Wiring, zur Visualisierung des Datenflusses, analog zu den kommerziellen Produkten, wie Yahoo Pipes, Presto Wires und dem IBM Data Mashup Editor, möglich sein.

4.1 Allgemeine Merkmale einer Entwicklungsumgebung

Unabhängig vom Mashup Typ, sollten Mashup Entwicklungsumgebungen allgemeine Funktionsmerkmale besitzen, die den Entwickler bei der Erstellung von Mashups helfen.

- Vorgaben von validen Eingaben und Plausibilitätsprüfung
- Tooltipps und Hilfeseiten
- Wiederverwendbarkeit der Mashups innerhalb der Mashup Plattform
- Suche, Mashup Katalog, Cloud Tagging
- Bewertung, Feedbackmöglichkeit

Vorgaben von validen Eingaben und Plausibilitätsprüfung

Vorgaben von validen Eingaben und Plausibilitätsprüfung erleichtern dem Entwickler die Erstellung von Mashups. Fehlerhafte Eingaben können von vornherein verhindert bzw. vermindert werden. Zum Beispiel sollten Eingabefelder, die eine Zahl erwarten, nur Zahlen akzeptieren. Eingabefelder mit begrenzten Eingabemöglichkeiten sollten die eingegrenzten Möglichkeiten anzeigen und diese zum Beispiel über ein Pulldown-Menu automatisch anbieten.

Tooltipps und Hilfeseiten

Tooltipps erläutern durch kurze Texte auf bequeme Weise Verwendungs- und Eingabemöglichkeiten. Im Gegensatz zu expliziten Hilfeseiten sind Tooltipps besser im Erstellungsprozess integriert, da lediglich ein Mouseover notwendig ist. Explizite Hilfeseiten können detailliertere Hilfestellung geben, da mehr Platz für Erläuterungen zur Verfügung steht. Diese Möglichkeit sollte dem Entwickler auch für sein Mashup zur Verfügung stehen.

Wiederverwendbarkeit der Mashups innerhalb der Mashup Plattform

Eine Wiederverwendbarkeit der Mashups innerhalb der Mashup Plattform erspart dem Entwickler, alle Funktionen / Mashups von Grund auf neu zu entwickeln. Der Entwickler kann ähnliche bestehende Mashups auf seine eigenen Bedürfnisse anpassen, um den Erstellungsprozess zu beschleunigen. Des Weiteren kann der Entwickler bestehende komplexe Mashups von erfahrenen Entwicklern als Implementierungsbeispiel verwenden, um neue Techniken zu erlernen und nachzuvollziehen.

Suche, Mashup Katalog, Cloud Tagging

Suchfunktionen, Mashup Kataloge und Cloud Tags helfen dem Entwickler und dem Anwender Mashups zu finden. Entwickler können solche Funktionen verwenden, um gezielt nach Funktionen / Mashups zur Wiederverwendung zu suchen bzw. um diese als Vorlage für eigene Entwicklungen zu verwenden. Möglicherweise lassen sich vorhandene Mashups bereits durch kleine Anpassungen an die eigenen Bedürfnisse anpassen.

Bei Yahoo Pipes lassen sich zum Beispiel gezielt nach in Pipes verwendeten Funktionsblöcken suchen, anschließend den Aufbau der gefundenen Pipes einsehen und gegebenenfalls für eigene Zwecke kopieren und anpassen.

Bewertung, Feedback

Ähnlich wie bei Amazon sollen Bewertungsfunktionen beim Identifizieren von besonders interessanten Mashups helfen. Eine Feedbackfunktion gibt dem Entwickler die Gelegenheit, Benutzerkommentare zu erhalten und Mashups auf Benutzerbedürfnisse anzupassen und zu verbessern.

4.2 Mashups auf Datenebene

Yahoo Pipes, Presto Wires oder der Data Mashup Editor des IBM Mashup Centers zeigen wie einfach Mashups auf Datenebene mit einem grafischen Editor erstellt werden kann, so dass keine Programmierkenntnisse notwendig sind. Die grafischen Editoren haben folgende Eigenschaften, die den Erstellungsprozess unterstützen:

- Vordefinierte Funktionsblöcke
- Kombinierbarkeit der Funktionsblöcke
- Anpassung über Eigenschaftsfelder und Anwender Eingabefelder
- Datenflussvisualisierung und -steuerung
- Drag & Drop, Freie Anordnung der Funktionsblöcke
- Ergebnisvorschau

Vordefinierte Funktionsblöcke

Vordefinierte grafische Funktionsblöcke ermöglichen es dem Anwender, ohne Programmierkenntnisse komplexe Funktionen zu verwenden. Damit die Verwendung der Funktionsblöcke anwenderfreundlich ist, sollten die Funktionsblöcke möglichst universell einsetzbar und eindeutig benannt sein. Zum Beispiel sollte ein „Datenquelle Feed“ Funktionsblock möglichst verschiedene Feed-Formate kennen und akzeptieren.

Datenquellen

Es sollten Funktionsblöcke für möglichst viele Datenquellen zur Verfügung stehen. Über übliche Datenquellen wie Feeds, Dateien, Datenbanken, Web Services, Web Clipping, usw. hinaus, wäre eine Implementierung von Laufzeitumgebungen für fremde Mashup Formate wie EMML oder Yahoo Pipes denkbar.

Datenmanipulationen

Mit Funktionsblöcken zur Datenmanipulation stehen dem Entwickler Werkzeuge zur Verfügung, um die Quelldaten zu bearbeiten und auf eigene Bedürfnisse anzupassen.

Kombinierbarkeit der Funktionsblöcke

Mit der Kombinationsmöglichkeit von Funktionsblöcken können komplexere Funktionen abgebildet und somit auch komplexere Mashups realisiert werden.

Anpassung über Eigenschaftsfelder und Anwender Eingabefelder

Analog zu Funktionsaufrufen mit Parameterübergabe lassen sich Funktionsblöcke über Eigenschaftsfelder anpassen und universell einsetzen. Anwender Eingabefelder ermöglichen das Mashup zur Laufzeit zu beeinflussen.

Datenflussvisualisierung und -steuerung

Die in Kapitel 3 betrachteten grafischen Editoren ermöglichen dem Entwickler, durch einfaches Ziehen von Verbindungslinien vom Ausgang eines Funktionsblocks zum Eingang eines Funktionsblocks auf intuitive Weise den Datenfluss zu steuern und zu visualisieren.

Drag & Drop, Freie Anordnung der Funktionsblöcke

Die freie Positionierung der grafischen Funktionsblöcke per Drag & Drop ist intuitiv und erhöht die Übersichtlichkeit für den Entwickler. Der Entwickler kann je nach Geschmack für sich individuell entscheiden, ob er eine horizontale oder vertikale Anordnung bevorzugt. Horizontale Anordnung steht im Einklang mit der Leserichtung, während eine vertikale Anordnung der Tatsache Rechnung trägt, dass vertikales Scrolling durch das Mausrad bereits etabliert und einfacher ist.

Ergebnisvorschau

Eine Ergebnisvorschau gibt dem Entwickler schnell und unkompliziert Feedback. Dem Entwickler ist sofort ersichtlich welche Auswirkung eine Maßnahme oder Änderung auf das Ergebnis hat, und kann sich mit dieser Funktion an das gewünschte Ergebnis Schritt für Schritt herantasten.

4.3 Mashups auf Präsentationsebene

Neben der Erstellung von Mashups auf Datenebene sollte der Entwickler Mashups auf Präsentationsebene erstellen können. Diese Vorgehensweise ist für den Entwickler noch intuitiver, da nach dem WYSIWYG Prinzip per Drag and Drop frei anordbare Widgets zusammengestellt werden können, die der Oberfläche für den Endanwender entspricht.

- Vordefinierte Widgets
- Globale Themes, Templates und CSS
- Mehrsprachenunterstützung
- Wiring – Widget zu Widget Datentransfer

Vordefinierte Widgets

Analog zu IBM Mashup Center kann der Entwickler einfach aus einem Repertoire von vordefinierten Widgets (Karten, sortierbare Tabellen, Diagramme, etc.) wählen. Diese Widgets bieten die Möglichkeit, die Mashups auf Datenebene ansprechend zu präsentieren. (Vgl. Yahoo Badges oder Presto Views)

Globale Themes, Templates und CSS

Eine einfache Möglichkeit, die Präsentation anzupassen sind globale, anpassbare Themes, Templates und CSS-Dateien. Für Corporate Identity könnten zum Beispiel Schrift und Farben global angepasst werden, so dass sich das Mashup nahtlos in die Intranet bzw. Internet Präsenz einfügt. Des Weiteren vereinheitlichen solche Vorgabemöglichkeiten das Look & Feel. Dies ist besonders wichtig, wenn es das Ziel ist, das Mashup Entwicklungswerkzeug einer breiteren Anwenderschicht bereitzustellen. Mit dem Anwenderkreis steigen ansonsten die Variationen.

Mehrsprachenunterstützung

Besonders in global agierenden Unternehmen ist eine Mehrsprachenunterstützung hilfreich. Bei einer mehrsprachenfähigen Software müssen nicht alle Seiten neu erstellt werden. Es sind lediglich Übersetzungen der Labels zu hinterlegen, so dass bei einem Sprachwechsel die Labels der gewählten Sprache angezeigt werden.

Wiring - Widget zu Widget Datentransfer

Ein Widget zu Widget Datentransfer ermöglicht dem Entwickler einen Datenfluss im Mashup zu realisieren und eine Interaktion der Widgets zu erreichen. Auf diese Weise können interaktive Anwendungen erstellt werden, welche Merkmale von Rich Internet Applications (RIA) besitzen.

4.4 Grafische Unterstützung für Mashups auf Datenebene

Im Rahmen dieser Arbeit ist eine Implementierung aller wünschenswerten Funktionen nicht möglich. Diese Arbeit beschränkt sich auf die Teilfunktion der grafischen Unterstützung für Mashups auf Datenebene. Nachfolgend wird diese Teilfunktion detailliert erörtert. Zur besseren Vergleichbarkeit wird als Beispiel ein Mashup erstellt, welches ähnlich den Mashups ist, die in Kapitel 3 mit den kommerziellen Produkten erstellten wurden.

Ein Feed wird nach einem bestimmten Stichwort gefiltert.

Für dieses Beispiel sind folgende Funktionen für eine grafische Unterstützung notwendig:

- Grafisches Element, welches einfach mit einer Funktion gemappt werden kann und eine Drag & Drop Funktionalität aufweist. Es bildet die grafische Benutzerschnittstelle.
- Feed Datenquelle
- Grafische Repräsentation einer Feed Datenquelle mit Eingabemöglichkeit für die Feed-Datenquelle
- Filterfunktion
- Grafische Repräsentation eines Filters, mit Eingabemöglichkeit für das Filterkriterium; optional kann eine bestimmte Filterfunktion gewählt werden.
- Endpunkt
- Grafische Repräsentation eines Endpunktes
- Wiring - Visualisierung des Datenflusses durch Verbindungslinien. Erstellung einer Vorgänger-Nachfolger Beziehung zwischen benachbarten Elementen.

4.4.1 Anwendungsfälle

Nachfolgend werden die Punkte erläutert und ihre Anforderungen beschrieben.

Grafisches Element

Das grafische Element bildet die Schnittstelle der Anwendung mit den Benutzer. Es repräsentiert grafisch die Datenquellen, Filter und Datensenke. Über Interaktionskomponenten wie Eingabefelder kann der Anwender auf die jeweilige Komponente Einfluss nehmen. Das grafische Element sollte einfach zu erstellen und leicht anpassbar sein. Des Weiteren sollte eine Funktion einfach mit dem grafischen Element verbunden werden können. Diese Anforderungen ermöglichen ein universelles Modell für weitere Funktionsblöcke.

Je nach Bedarf symbolisiert ein grafisches Element einen Eingang bzw. einen Ausgang. Eine freie Positionierung der grafischen Elemente per Drag & Drop sollte möglich sein. Es ist daher erforderlich, dass die grafischen Elemente ihre Positionen kennen und Drag & Drop anbieten.

Eingang	• 0 bis 1 Eingang, im Spezialfall von Vereinigungsfunktionen können entsprechend mehrere Eingänge vorhanden sein. • Parameter für Funktion • Gibt an, welche Datentypen akzeptiert werden.
Eigenschaftsfelder	Eigenschaftsfelder bzw. Eingabefelder, um Mashup zu beeinflussen.
Ausgang	• In der Regel 1 Ausgang, Spezialfall: Endpunkt → kein Ausgang • Ist Parameter für nächste Funktion • Gibt an, welcher Datentyp vorliegt.
Assoziierte Funktion	Gibt an, welche Funktion mit dem grafischen Element assoziiert ist.
Interne Daten	Hält interne Daten vor, die nicht für den Anwender sichtbar sind. ID: eigene Referenz Position: Koordinaten des Elementes Vorgänger: wenn vorhanden, Referenz des Vorgängers

Feed Datenquelle

Eine Funktion, die unter Angabe einer Feed URL ein Feed bereitstellt.

Grafische Repräsentation einer Feed Datenquelle

Eingang	Kein Eingang
Eigenschaftsfelder	Eingabefeld für Feed URL
Ausgang	Feed Daten
Assoziierte Funktion	Feed Datenquelle
Interne Daten	ID; Position

Filterfunktion

Filterfunktion ist eine Funktion, die eine Zeichenkette nach einem bestimmten Stichwort abgleicht. Nur Übereinstimmungen werden angezeigt. Optional können weitere Filterfunktionen implementiert werden: „beginnt mit“, „nicht übereinstimmend“, usw. Bei Implementierung von verschiedenen Filterfunktionen muss eine Wahlmöglichkeit der Filterfunktion bereitgestellt werden.

Grafische Repräsentation eines Filters

Eingang	Zeichenkette
Eigenschaftsfelder	Eingabefeld für Filterkriterium Optional: Wahlmöglichkeit für Filterfunktion
Ausgang	Zeichenkette, die auf das Kriterium zutrifft
Assoziierte Funktion	Filter
Interne Daten	ID; Position; Vorgänger

Endpunkt

Ein Endpunkt dient als definierten Einstiegspunkt für das Mashup. Durch Verweise auf den jeweiligen Vorgänger kommt man, wie in einer verketteten Liste, von einem Element zum nächsten bzw. von einer Funktion zur nächsten Funktion. Es ist für den korrekten Aufbau der Funktionsschachtelung zuständig.

Grafische Repräsentation eines Endpunktes

Eingang	Keine Daten
Eigenschaftsfelder	Eingabefeld für Filterkriterium Optional: Wahlmöglichkeit für Filterfunktion
Ausgang	Kein Ausgang
Assoziierte Funktion	Endpunkt
Interne Daten	Position; Vorgänger

Wiring - Visualisierung des Datenflusses durch Verbindungslinien

Verbindungslinien sollen den Datenfluss visualisieren. Die Verbindungslinien werden durch Ziehen eines grafischen Elementes auf ein anderes grafisches Element erzeugt. Dabei wird die ID des Vorgängers in das Vorgängerfeld des Nachfolgers eingetragen, so dass eine Vorgänger-Nachfolger Beziehung zwischen den beiden Elementen entsteht.

4.5 Zusammenfassung

Inspiriert von den kommerziellen Mashup Entwicklungswerkzeugen lassen sich sehr viele Funktionen identifizieren, um die sich der WSO2 Mashup Server erweitern lässt, damit dieser anwenderfreundlicher und somit auch für Anwender ohne Programmierkenntnisse zugänglich wird.

Eine Realisierung aller Funktionen sprengt jedoch den Rahmen dieser Arbeit. Es wurde daher die Funktion „grafische Unterstützung für Mashups auf Datenebene“ ausgesucht und ein bekanntes Beispiel Mashup im Detail beschrieben. Im nächsten Kapitel werden die grundlegende Konzepte erörtert und anhand eines Evaluierungs Prototyps die Tragfähigkeit der Konzepte gezeigt.

5. Konzeption, Design und Implementierung

In diesem Kapitel werden die grundlegenden Konzepte für die Entwicklung eines flexiblen Gerüstes für die Erstellung von datenorientierten Mashups mit dem WSO2 Mashup Server mittels einer grafischen Benutzeroberfläche vorgestellt. Es werden zunächst relevante Architektur Muster vorgestellt und erläutert. Anschließend werden Randbedingungen erörtert, die ebenfalls Einfluss auf die weiteren Design Entscheidungen haben. Unter Beachtung der Randbedingungen und der vorgestellten Architektur Muster wird ein flexibles und erweiterbareres Design vorgestellt. Abschließend wird ein Prototyp, welcher im Rahmen dieser Arbeit erstellt wurde, als „Proof of Concept“ vorgestellt.

5.1 Architektur Muster

Architektur Muster bieten, im Unterschied zu Entwurfsmuster, generelle Richtlinien, um ein bestimmtes Konzept zu unterstützen. Im nachfolgenden werden die Architektur Muster „Pipes and Filters“, „Client-Server“ und „Model View Controler“ (MVC) vorgestellt. Durch Verwendung dieser Architektur Muster in der Konzeption wird Wiederverwendbarkeit, Erweiterbarkeit und Flexibilität sichergestellt.

5.1.1 „Pipes and Filters“

Das Architektur Muster „Pipes and Filters“ bietet eine Aufbaustruktur für Systeme, die Datenströme verarbeitet (Buschmann u.a., 1996). Das Muster sieht eine Verkettung von Arbeitsschritten zu einer sogenannten „Pipeline“ vor (vgl. Abb. 15), die aus folgenden drei Basis-Komponenten bestehen:

- Datenquelle: stellt Daten für die Pipeline bereit
- (Transformations)-Filter: nimmt Daten entgegen, verarbeitet Daten und stellt verarbeitete Daten bereit
- Datensenke: konsumiert Daten

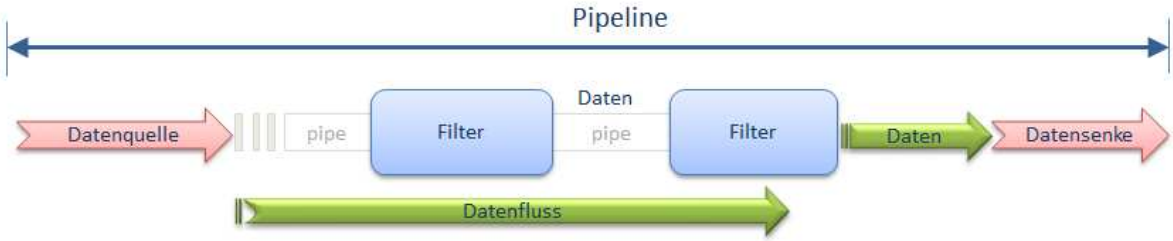


Abb. 15: Schematischer Aufbau einer Pipe and Filter Musters zu einer Pipeline[26]

Die Daten einer Datenquelle gelangen über eine Pipe an einen Filter. Im Filter erfolgt eine Verarbeitung (z.B.: Anreicherung, Transformation, Aggregation,…) der Daten. Dabei stellt jeder einzelne Filter einen Verarbeitungsschritt dar. Durch Kombination einzelner Filter hintereinander, können Daten weiteren Anpassungen unterworfen werden. Die Datensenke stellt das Ende der Pipeline dar. Hier steht das gewünschte Ergebnis für die weitere Verarbeitung bereit (Speicherung, Darstellung,…).

Die drei Basis-Komponenten werden über eine sogenannte „Pipe“ verbunden, die abhängig von der implementierten Variante, implizit durch direkte Aufrufe der benachbarten Filter untereinander (passive Pipe) oder explizit als eine weitere Komponente dargestellt wird (aktive Pipe). Die aktive Pipe wird verwendet, um parallel laufende Filter zu synchronisieren und Daten zu puffern.

Eine passive Pipe in der „Pull“-Variante wird in Abbildung 16 dargestellt. Die Abbildung zeigt, dass von der Datensenke ausgehend geschachtelte Funktionsaufrufe erfolgen bis die Datenquelle erreicht wird. Anschließend erfolgt das Durchlaufen der Verarbeitungsschritte. Neben dieser genannten Variante gibt es ebenfalls eine „Push“ Variante, welche eine Betrachtung von der Datenquelle aus darstellt.

[26] http://de.wikipedia.org/wiki/Pipes_und_Filter

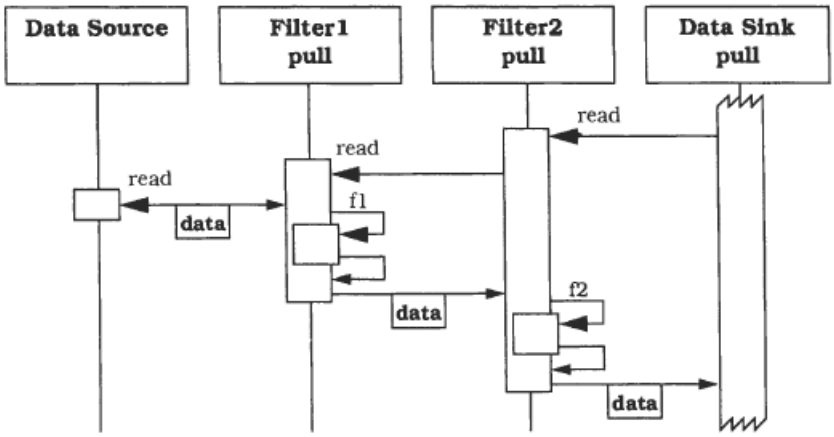


Abb. 16: Sequenzdiagramm: Pipes and Filters – Pull Variante (Buschmann u.a., 1996)

In der Basis-Variante des Architektur Musters „Pipes and Filters“ erfolgt eine lineare Verarbeitung des Datenstromes. Jede Komponente bildet eine unabhängige Einheit und hat maximal nur einen direkten Vorgänger und einen direkten Nachfolger.

In der weiteren Variante „Tee-and-Join-Pipeline” können Filter mehrere Ein- und Ausgänge besitzen. Diese Variante erlaubt zum Beispiel die Zusammenführung von unabhängigen Datenströmen, wodurch zum Beispiel eine „Union“-Filterkomponente abgebildet werden kann. Eine weitere Verwendungsmöglichkeit wäre zum Beispiel die Umsetzung einer Verzweigung oder einer Schleife. Bei komplexeren Pipelines stellt die „Tee-and-Join-Pipeline“ Variante eine besondere Herausforderung im Design dar. Es muss zum Beispiel vorher gründlich geprüft werden, ob die Pipeline terminiert (Buschmann u.a., 1996).

In der Modularität bzw. im Baukastenprinzip liegt der größte Vorteil des Architektur Musters „Pipes and Filters“. Im Gegensatz zu einem monolithischen Aufbau lassen sich in einem modularen Aufbau Teile (hier: Filter) schnell anpassen und austauschen. Übertragen auf das Muster „Pipes and Filter“ bedeutet dies:

- Flexibilität:
  Eine Pipeline kann schnell an neue Anforderungen angepasst werden, in dem einzelne Filter ausgetauscht oder angepasst werden.
- Wiederverwendbarkeit:
  Eine Filterkomponente kann von verschiedenen Pipelines verwendet werden.

Dieses Architektur Muster weist viele Parallelen zu den grafischen Entwicklungswerkzeugen auf (s. Kapitel 3). Die Filter entsprechen den Funktionsblöcken, während die Pipes den Wires in den betrachteten Werkzeugen entsprechen. Eine Umsetzung, die stark an den betrachteten Werkzeugen angelehnt ist, setzt gleichzeitig dieses Architektur Muster mit all ihren Vorteilen um.

Durch die Kapselung der Funktionsblöcke in Filterkomponenten erhält man die Flexibilität und Austauschbarkeit. Man kann auf einfache Weise weitere Filterkomponenten implementieren und um weitere Funktionen erweitern. Kombinationen der Filterkomponenten miteinander erhöhen weiter die Anwendungsmöglichkeiten[27].

5.1.2 „Model View Control“

Das Architektur Muster „Model View Control“ (MVC) ist ein Muster für interaktive Systeme. Dabei steht „Model“ für das zugrundeliegende Datenmodell. Es enthält die Daten und bietet die Kernfunktionalität an. „View“ steht für die Präsentationsebene. Die View zeigt dem Anwender Informationen an. „Control“ nimmt Anwenderaktionen entgegen und verarbeitet bzw. delegiert diese entsprechend. „View“ und „Control“ umfassen die Benutzerschnittstelle. Im klassischen MVC Muster werden diese drei Komponenten separiert, so dass eine Entkopplung der Komponenten möglich wird. Auf ein Datenmodell können somit verschiedene Sichten ermöglicht werden. Zum Beispiel können Daten in einer tabellarischen Form (Tabelle) präsentiert werden, als auch in einer grafischen Form (Diagramm). Ebenso können verschiedene Controller implementiert werden, die das gleiche Datenmodell verwenden (Reenskaug 2007). Die Abbildung 17 zeigt eine schematische Darstellung des MVC Architektur Musters. Von dieser klassischen Form abweichend gibt es Varianten, in der zum Beispiel „View“ und „Control“ zusammengefasst werden oder „View“ vom Model entkoppelt wird[28] .

[27] http://eprints.cs.univie.ac.at/2698/1/ArchPatterns.pdf
[28] MVP; http://de.wikipedia.org/wiki/Model_View_Presenter

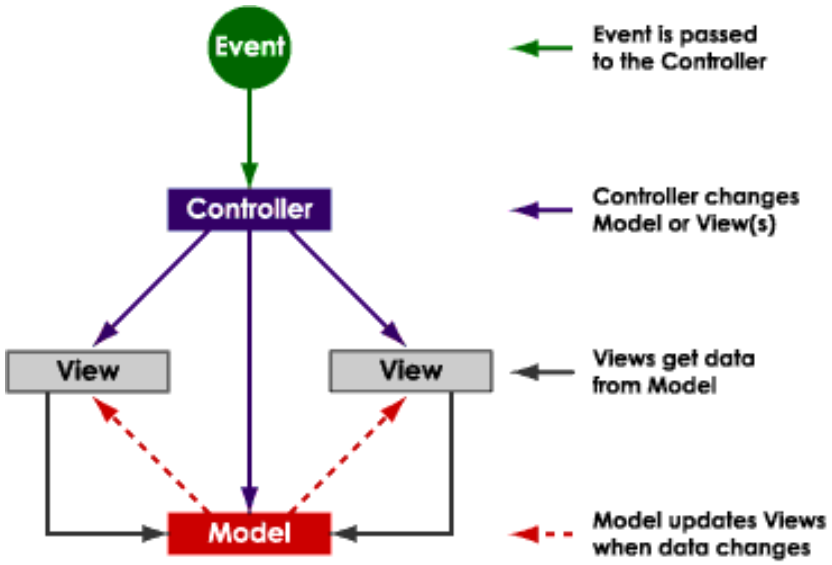


Abb. 17: Schematische Darstellung des Architektur Musters MVC[29]

View

Übertragen auf das grafische Entwicklungswerkzeug bildet die grafische Repräsentation der Funktionsblöcke bzw. der Filterkomponenten die „View“. Jeder Funktionsblock bzw. jede Filterkomponente erhält eine View, mit den für diesen Funktionsblock / diese Filterkomponente erforderlichen Eingabefeldern. Des Weiteren sollte die View Drag and Drop unterstützen.

Model

Das Model repräsentiert die Daten einer Applikation, die durch den „Controller“ geändert werden. Dabei beschränkt sich der Begriff „Daten“ nicht nur auf die Daten, auf die sich eine Applikation konzentriert und die beim Durchlaufen einer Applikation Änderungen erfahren. Sondern es gehören dazu auch Daten, wie zum Beispiel die eingegebenen Parameter oder Daten über den Zustand.

Control

Der „Controller“ nimmt Anwenderaktionen entgegen und steuert bzw. ändert entsprechend die View und das Model. Übertragen auf das grafische Entwicklungswerkzeug wäre es zum Beispiel das Eintragen eines Parameters in ein Eingabefeld. Der Controller übernimmt die veränderten Eingabedaten entgegen und aktualisiert das zugrundeliegende Model.

[29] http://www.enode.com/x/markup/tutorial/mvc.html

Editor

In der ursprünglichen Variante des „Model View Control“ von 1979 war noch eine Erweiterung des Controllers, dem Editor, vorgesehen (Reenskaug 2007). Der Editor wird durch eine View nach explizitem Aufruf erzeugt. Die Lebensdauer des Editors ist auf eine Änderungsaktion beschränkt.

5.1.3 „Client-Server“

Das Client-Server Modell beschreibt die Verteilung der Aufgaben zwischen einem Dienst-Anbieter, dem Server, und einem Konsumenten, dem Client. Die Abbildung 18 zeigt in einer Übersicht die verschiedenen Kombinationsmöglichkeiten und Unterscheidungsmerkmale. Es kann sowohl nach Art des Clients (s. 5.1.3.1 und 5.1.3.2) als auch nach Anzahl der Schichten in der Architektur (s. 5.1.3.3).

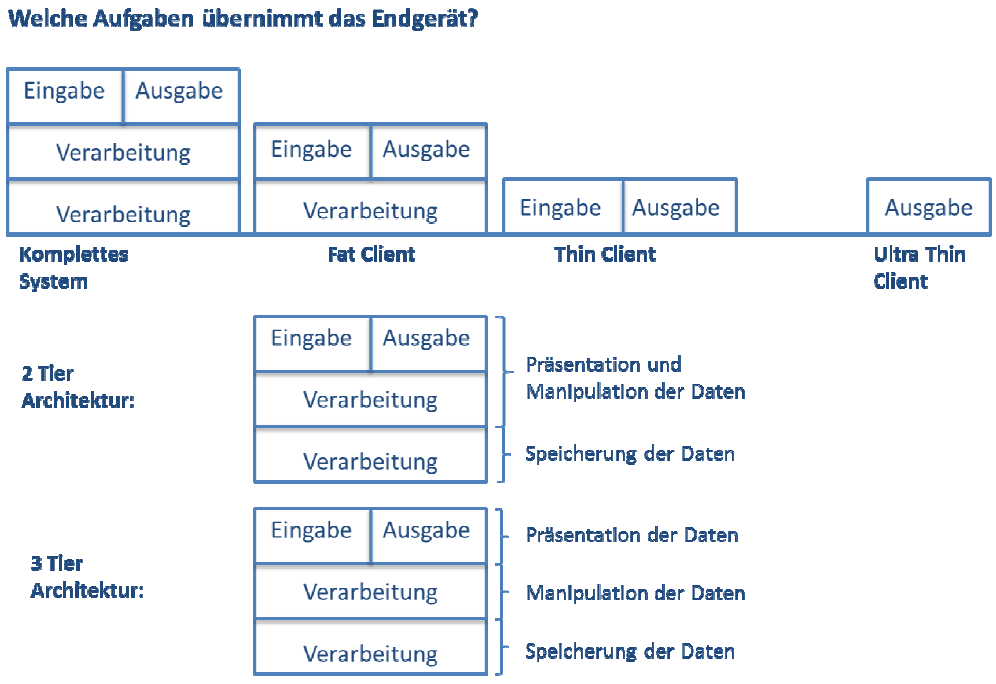


Abb. 18: Client Kategorien und Schichten Architektur[30]

[30] http://de.wikipedia.org/wiki/Thin_Client

5.1.3.1 „Fat Client“ / „Thin Client“

Je nachdem, welche Aufgaben (Eingabe, Ausgabe, Verarbeitung, Verwaltung) ein Client wahrnimmt, werden die Clients unterschiedlich bezeichnet. In der klassischen Definition der Clients werden zwischen einem „Fat Client“ und einem „Thin Client“ unterschieden. Im „Fat Client“ erfolgen sowohl die Ein- und Ausgabe als auch die Verarbeitung. Im „Thin Client“ erfolgt lediglich die Ein- und Ausgabe. In der Regel stellt ein „Fat Client“ im Vergleich zum „Thin Client“ höhere Anforderungen an die verwendete Hardware, auf die der Client läuft, da für die Verarbeitung zusätzliche Rechenleistung erforderlich ist. Hingegen stellt der „Thin Client“ höhere Anforderungen an die Infrastruktur und den Server. Da die Verarbeitung serverseitig erfolgt, muss der Server Ressourcen für die Verarbeitung bereitstellen. In der Regel mehr Datenpakete zwischen dem Server und dem Client ausgetauscht, so dass eine höhere Netzwerklast erzeugt wird. Um einen reibungslosen Betrieb, insbesondere wenn es eine hohe Anzahl von „Thin Clients“ gibt, zu gewährleisten, ist eine ausreichend dimensionierte Infrastruktur erforderlich. Die Vorzüge eines „Fat Clients“ liegen im hohen Komfort für den Anwender. Da die Verarbeitung lokal auf dem Client Rechner erfolgt, erhält der Anwender entsprechend gute Antwortzeiten. Ferner lassen sich Komfortfunktionen wie Plausibilitätsprüfung oder die Anbindung an lokal installierte Programme realisieren. Der Thin Client hat zum Vorteil, dass in der Regel keine Installation notwendig ist. Der Footprint des Programmes ist relativ klein. Ein Extrem stellt der „Ultra Thin Client“ dar. Der „Ultra Thin Client“ ist lediglich für die Ausgabe verantwortlich.

5.1.3.2 „Smart Client“

Eine weitere Variante ist der „Smart Client“ (Javamagazin, 2006). Der „Smart Client“ versucht die Vorzüge der beiden klassischen Varianten, „Fat Client“ und „Thin Client“, zu vereinen. In Javamagazin wurden folgende Merkmale genannt, die einen „Smart Client“ ausmachen:

„Benutzungskomfort“ und „Integration lokaler Standardsoftware“

Der „Smart Client“ bietet das Antwortverhalten und die Benutzerführung eines „Fat Client“. Es findet eine Plausibilisierung und Validierung der Eingaben statt. Ausserdem ist der „Smart

Client“ an die lokalen Programme angebunden (Office, Mail, PDF Viewer). Datenfluss zwischen Client und Server findet nur statt, wenn gemeinsam verwendete Daten bearbeitet werden.

„Verteilte Komponenten in einer gemeinsamen Architektur“

Client und Server verwenden die gleiche technologische Basis und die gleiche Entwicklungsumgebung. Funktionalitäten sollen dadurch nach Bedarf zwischen Server und Client verschoben werden können. Kommunikation mit dem Server erfolgt über auf den Smart Client zugeschnittene WebServices. Der Fokus der WebServices liegt dabei auf der optimalen Abstimmung mit dem „Smart Client“ und nicht auf der Wiederverwendbarkeit des WebServices.

„Keine lokale Datenhaltung“

Es erfolgt keine lokale Datenhaltung von veränderlichen Daten. Lediglich statische Daten können zwecks Performance Optimierung auf dem Client vorgehalten werden.

„Integration fremder Software“

Informationen von zentral laufender Software werden mittels WebServices in den „Smart Client“ integriert. Daten und Funktionen verbleiben dabei in der Fremdsoftware.

„Software-Update“

Idealerweise erfolgt eine Versionsprüfung und Softwareaktualisierung bei jedem Start statt, so dass der Client immer in der aktuellsten Version vorhanden ist. Auf diese Weise sollen Probleme, die durch verschiedene Softwareversionen verursacht werden, vermieden werden. Ähnlich wie beim „Windows Service Pack Webinstaller“ erfolgt ein inkrementelles Software-Update, so dass nur benötigte Komponenten geladen werden. Auf diese Weise werden die Netzwerkressourcen geschont.

„Copy & Run Install“

Der Smart-Client soll nach dem Kopieren sofort startbereit sein. Bei der Installation erfolgt keine Systemveränderungen, wie zum Beispiel Registry Einträge.

5.1.3.3 Architektur Schichten

Neben der Kategorisierung über den Client, erfolgt eine Kategorisierung über die Anzahl der Schichten in der Architektur. Es wird dabei zwischen einer „2 Tier“-Architektur und einer „3 Tier“-Architektur unterschieden. In der „2 Tier“-Architektur erfolgt lediglich die Speicherung der Daten auf den Server. Der Rest (Ein-, Ausgabe und Verarbeitung) erfolgt clientseitig. Bei einer „3 Tier“-Architektur (vgl. Abb. 19) erfolgt die Ein- und Ausgabe auf dem Client. Eine Verarbeitung erfolgt auf einem Applikationsserver, während die Datenspeicherung auf einem Backend-Server stattfindet.

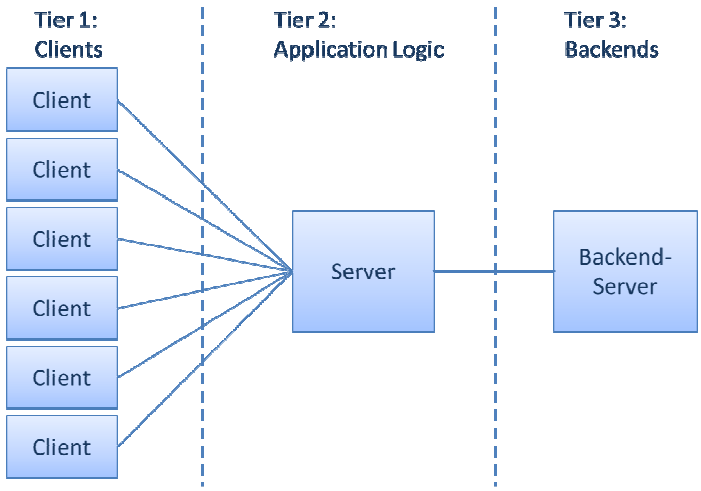


Abb. 19: Beispiel einer „3-Tier“-Client-Server-Architektur[31]

5.2 Technische Rahmenbedingungen

Serverseitig ist Javascript als Entwicklungssprache mit der Wahl von des WSO2 Servers vorgegeben. Mit der Wahl des Browsers als Client Plattform, stehen clientseitig Javascript und Adobe Flash Actionscript als Programmiersprache zur Auswahl. Beide Programmiersprachen basieren auf der ECMA Spezifikationen 262, wobei alle etablierten Browser (Mozilla Firefox, Microsoft Internet Explorer, Google Chrome, Opera und Apple Safari) Javascript bzw. einen Javascript Derivat nativ unterstützen. Adobe Flash Actionscript ist proprietär und es ist ein

[31] Architectural Patterns Revisited – A Pattern Language; http://eprints.cs.univie.ac.at/2698/1/ArchPatterns.pdf, S. 31

Plug-In notwendig, damit die Browser Actionscript verstehen. Aus diesem Grund wird clientseitig ebenfalls Javascript als Programmiersprache verwendet.

5.2.1 Javascript

Javascript basiert auf der ECMA Spezifikation 262 (European Computer Manufacturers Association) und hat seit seiner Entstehung (1995) mehrere Versionen durchlebt. Aktuell ist die Javascript Version 1.8. Mit den Versionen wurde Javascript stetig erweitert.
Mit dem Zugang zum Internet und der wachsenden Bedeutung des Internets wächst auch die Bedeutung des Browsers als Client Plattform. Heutzutage ist nahezu auf jedem Computer mindestens ein Browser installiert. Mit dem wachsenden Anspruch an die Webseiten, ein interaktives Surferlebnis zu bieten, ist Javascript bzw. Flash momentan noch unersetzlich. Dies trägt zur einer weiter wachsenden Bedeutung von Javascript bei. In der momentanen Entwicklung befindet sich die Javascript Version 1.8.5. In dieser Version werden einige Bestandteile von ECMAScript 5 implementiert (c't 2011).

Nachfolgend werden einige wichtige Merkmale von Javascript genannt:

- Javascript ist eine objektorientierte Programmiersprache.
- Javascript ist klassenlos. Objekte werden über Konstruktor Funktionen erstellt.
- Vererbung wird in Javascript über *Prototyping* realisiert. Es wird dabei die Prototype Eigenschaft eines Objektes die Prototype Eigenschaft eines anderen Objektes zugewiesen (vgl. Abb. 20). Auf diese Weise entsteht eine Prototype Hierarchie. Bei Abruf einer nicht vorhandenen Eigenschaft oder Funktion erfolgt ein Lookup über die in der Prototype abgelegten Referenz zum Prototype des übergeordneten Objektes.
- In Javascript sind Variablen dynamisch typisiert, d.h. erst zur Laufzeit wird der Typ zugewiesen.

Die Browser verwenden unterschiedliche Javascript Compiler (Mozilla Firfox → Rhino, Microsoft Internet Explorer → JScript, Google Chrome → V8, Apple Safari → KJS), welche die ECMA Spezifikation 262 nicht immer einheitlich umsetzen. Aus diesem Grund können browserabhängige Inkompatibilitäten des Javascript Codes auftreten. Um diesem Problem zu

begegnen, ist es üblich, den Browser und ihre Version zu identifizieren und für bekannte Inkompatibilitäten eine entsprechend angepasste Implementierung bereitzustellen.

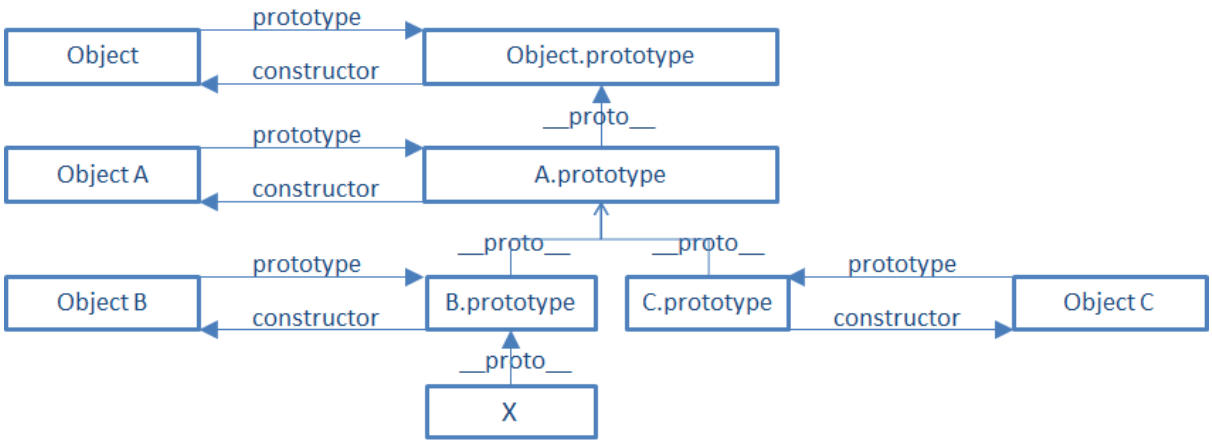


Abb. 20: Schaubild Vererbung per Prototyping

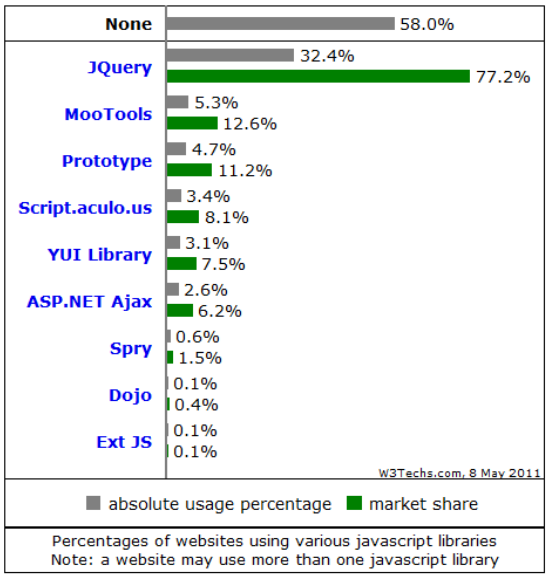
5.2.2 jQuery und jQueryUI

Mit der Browser- und Versionsvielfalt sind auch die zu berücksichtigten Ausnahmen angewachsen. Aus diesem Grund hat es sich für mittlere bis größere Projekte etabliert, Javascript Bibliotheken zu verwenden, die den verwendeten Browser detektieren, eventuelle Inkompatibilitäten berücksichtigen und automatisch einen geeigneten Code verwenden. Des Weiteren bieten Javascript Bibliotheken für häufig verwendete Operationen, wie DOM Manipulation, optimierte Funktionen an. Inperformanter Programm Code trägt zu einer schlechten Software Ergonomie und Akzeptanz bei.

Aus den oben genannten Gründen wird für die Client Komponente ebenfalls eine Javascript Bibliothek verwendet. Als Javascript Bibliothek wird, die mit wachsender Tendenz (vgl. Abb. 22) verbreitetste (vgl. Abb. 21) Javascript Bibliothek, jQuery und dessen Erweiterung für Benutzeroberflächen jQueryUI verwendet[32]

[32] http://w3techs.com/technologies/overview/javascript_library/all

How to read the diagram:
58.0% of the websites use none of the javascript libraries that we monitor.
JQuery is used by 32.4% of all the websites, that is a javascript library market share of 77.2%.



The following javascript libraries have a market share of less than 0.1%

- DHTMLX
- MochiKit
- DOMAssistant
- UIZE
- Glow
- Midori
- SproutCore
- Google Closure Library
- JavaScriptMVC

Abb. 21: Marktanteile verschiedener JavaScript Bibliotheken[33]

Historical trend

This diagram shows the historical trend in the percentage of websites using JQuery.
Our dedicated trend survey shows more javascript libraries usage and market share trends.

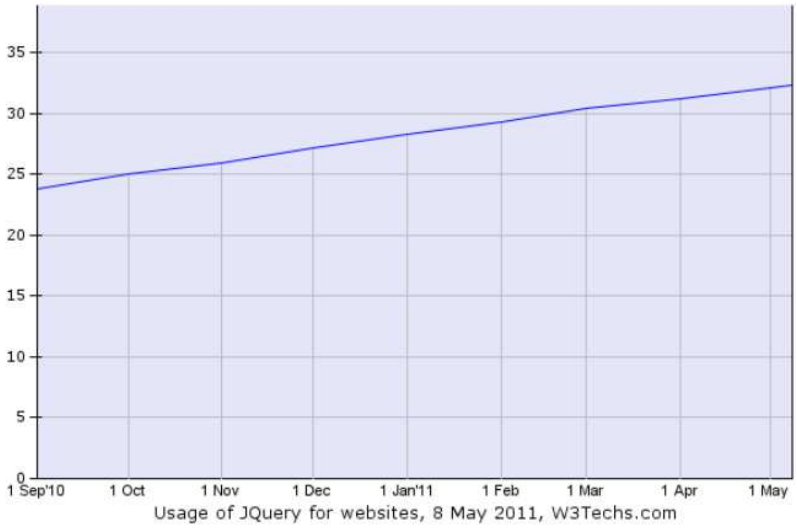


Abb. 22: Trend der Verwendung von jQuery[34]

[33] http://w3techs.com/technologies/overview/javascript_library/all
[34] http://w3techs.com/technologies/details/js-jQuery/all/all

jQuery ist eine auf kleinen Footprint und hohe Geschwindigkeit optimierte Javascript Bibliothek mit einer vergleichsweise niedrigen Lernkurve. Die niedrige Lernkurve ermöglicht es Interessierten, eine Weiterentwicklung des im Rahmen dieser Arbeit erstellten Prototyps bereits nach relativ kurzer Einarbeitungszeit vorzunehmen. Durch die hohe Verbreitung von jQuery sind viele Informationsquellen zu jQuery vorhanden. Zudem gibt es viele kostenlose Erweiterungen für jQuery.

jQuery bietet Grundfunktionen wie DOM Manipulation und Eventhandling. Das auf jQuery basierende jQueryUI bietet vordefinierte Widgets (Accordion, Autocomplete, Button, Datepicker, Dialog, Progressbar, Slider, Tabs) und Funktionen für Interaktion wie zum Beispiel GUI spezifische Eventhandler (Draggable, Droppable, Resizeable, Selectable, Sortable). Außerdem werden Effekte über CSS Manipulation unterstützt. Über die genannten Punkte hinaus achtet jQuery auf eine korrekte Implementation eines Lifecycle Managements für die über jQuery erstellten Objekte. Ein Indiz für die Performanz von jQuery ist, dass hoch frequentierte Webseiten wie Amazon.com, Twitter.com und Microsoft.com die jQuery Bibliothek verwenden (vgl. Abb. 23).

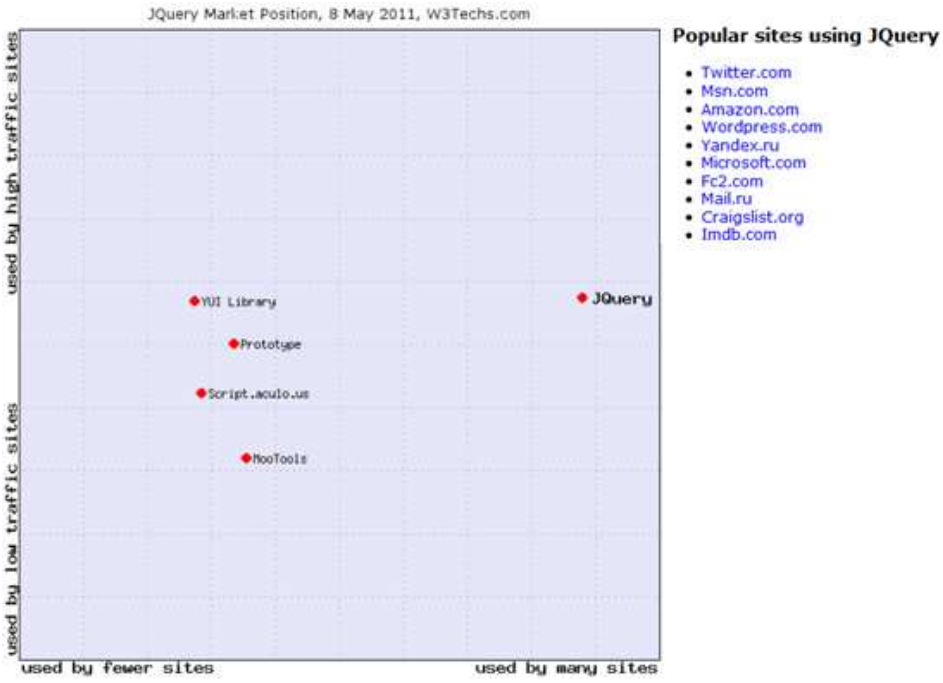


Abb. 23: Verbreitung in Relation zur Frequentierung[35]

[35] http://w3techs.com/technologies/details/js-jQuery/all/all

5.3 Umsetzung der Architektur Muster

Das Schaubild (Abb.24) zeigt den Aufbau, die Verteilung und das Zusammenspiel der in Kapitel 5.1 erläuterten Architektur Muster.

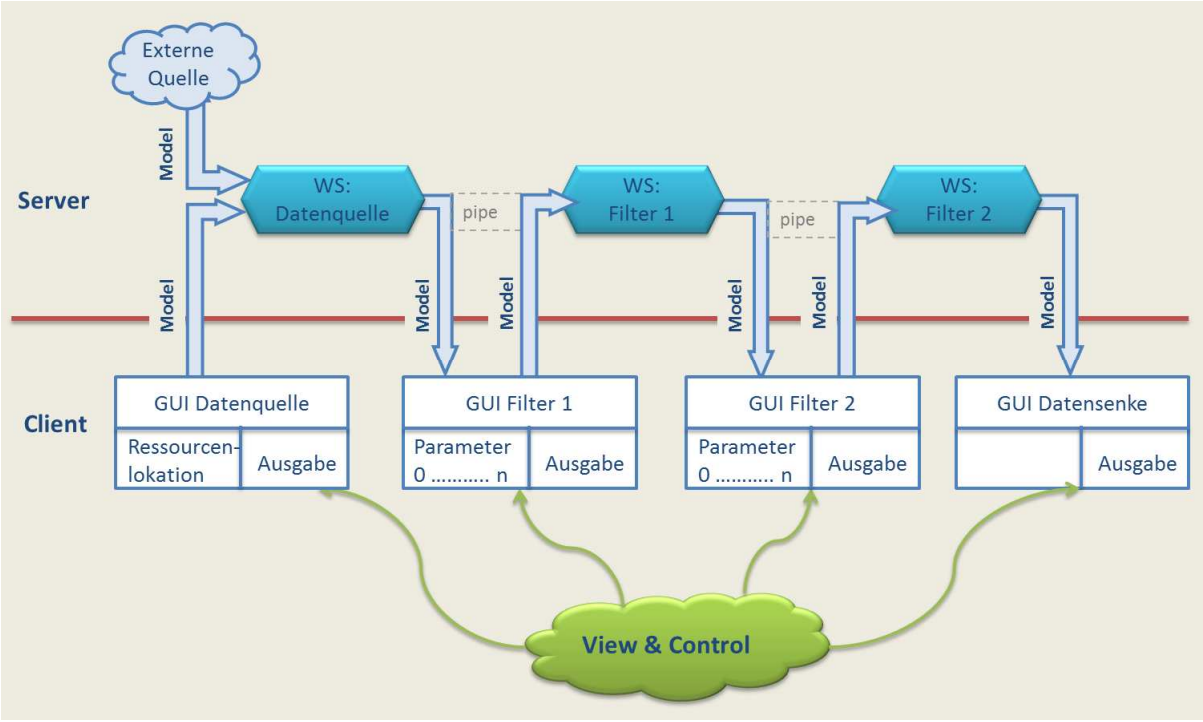


Abb. 24: Umsetzung der Architekturmuster[36]

5.3.1 „Pipes and Filters“

Filter

Filter nach dem „Pipes and Filter“ Architektur Muster werden serverseitig über zustandslose Webservice Funktionen auf dem WSO2 Mashup Server abgebildet. Die Webservice Funktion erwartet im Allgemeinen eine Dateneingabe und gegebenenfalls Parameter, verarbeitet die eingegebenen Daten und gibt das Ergebnis wieder zurück an den Client. Dadurch, dass immer nach dem gleichen Muster vorgegangen wird (Dateneingabe, Verarbeitung, Datenrückgabe), wird eine lose Koppelung realisiert und es ist auf einfache Weise eine Kaskadierung von Filtern möglich. Eine Verkettung von Webservice-Aufrufen erlaubt eine flexible Steuerung der

[36] Eigene Darstellung

Verarbeitung. Die Möglichkeit der Parametereingabe erhöht die Flexibilität und Wiederverwendbarkeit der Filter.

Datenquelle

Datenquellen-Filter erwarten die Angabe der Datenquelle und geben Daten von der Datenquelle zurück. In der Implementation des Prototyps wurde das Feed Host Object des WSO2 Mashup Servers verwendet, um ein Feed Datenquellen Filter zu erstellen. Der Feed Datenquellen Filter erwartet als Eingabe die URL des Feeds und liefert als Ergebnis das entsprechende Feed zurück. Analog zu dieser Datenquelle lassen sich weitere Datenquellen realisieren. Mit der Auswahl an Datenquellen, steigen die Möglichkeiten des WSO2 Mashup Servers.

Transformation

Verarbeitungs- bzw. Transformations-Filter verändern die am Eingang vorliegenden Daten und stellen die veränderten Daten am Ausgang bereit. Bezogen auf die WSO2 Mashup Server Webservices erwartet der Webservice Eingangsdaten und gibt die veränderten Daten wieder an den Client zurück. Für den Prototyp wurde ein Selektions-Filter für in XML vorliegende Daten implementiert. Der Selektionsfilter erlaubt es dem Anwender, XML-Knoten nach Vorkommen von bestimmten Zeichenketten, ohne Beachtung von Groß- und Kleinschreibung, zu durchsuchen und liefert als Ergebnis den ganzen Elternknoten für die weitere Verarbeitung zurück. Die zu suchende Zeichenkette wird beim Aufruf des Webservice mit den zu verarbeitenden Daten als Parameter mitgegeben.

Datensenke

In der Datensenke erfahren die Eingangsdaten keine weitere Verarbeitung. Die Datensenke dient lediglich dazu, einem nachgelagerten Prozessschritt wie zum Beispiel einer Daten-Präsentation einen fest definierten Punkt bereitzustellen. Für den Prototyp wurde keine explizite Datensenke erstellt. Im Prototyp wurde eine Anzeigemöglichkeit in jedem Filter implementiert, so dass analog zu JackBe Presto eine Überprüfung der Daten in jedem Filterschritt möglich ist. Hierzu wird der vom Webservice zurückgelieferte Inhalt bei Bedarf in einer modalen Dialogbox angezeigt. Als nächster Schritt ist eine Implementierung der Anzeige in einer nicht modalen Konsole möglich.

Pipe

Die Pipe kann als passive oder als aktive Pipe konzipiert werden (vgl. Kapitel 5.1.1). Eine passive Pipe kann einfach durch Schachtelung von Funktionsaufrufen modelliert werden.Soll eine aktive Pipe implementiert werden, so muss man folgende Punkte beachten: Eine aktive Pipe setzt nebenläufige Filter voraus. Ferner müssen die Pipes eine Pufferung der Daten bieten, damit unterschiedlich schnelle Filter parallel arbeiten können. Dies hat zur Folge, dass aktive Pipes modelliert werden müssen. Die aktive Pipe braucht über einen Puffer hinaus noch eine Art Semaphoren Konzept, damit eine gesteuerte Vergabe der vorhanden Ressourcen (Pufferplatz) möglich wird.

Die Pipe wird im ersten Schritt und für den „Proof of Concept“ Prototyp als eine passive Pipe in der „Pull“-Variante implementiert. Dieses Vorgehen hat den Vorteil, dass durch den Wegfall einer Komponente im ersten Schritt Komplexität reduziert wird. Zudem bieten parallel laufende Filter für Mashups auf Datenebene keinen Vorteil gegenüber sequentiellen Filtern. Vielmehr handelt es sich bei den Mashups auf Datenebene um eine Aneinanderreihung von Transformationsfiltern, die auf das Ergebnis des vorherigen Filters aufbauen.

Für die Implementierung bedeutet es, dass benachbarte Filter sich kennen und aufrufen können. Die Ergebnisse eines Filters bilden die Eingangsdaten des nachfolgenden Filters. Auf diese Weise lassen sich nach jedem Verarbeitungsschritt die Ergebnisse überprüfen. Das nachfolgende Schaubild (Abb.25) ist eine schematische Darstellung einer Pipeline im Zusammenspiel mit den Grundfunktionen bzw. Webservices des WSO2 Mashup Servers. Die passive Pipe befindet sich vollständig auf dem Client.

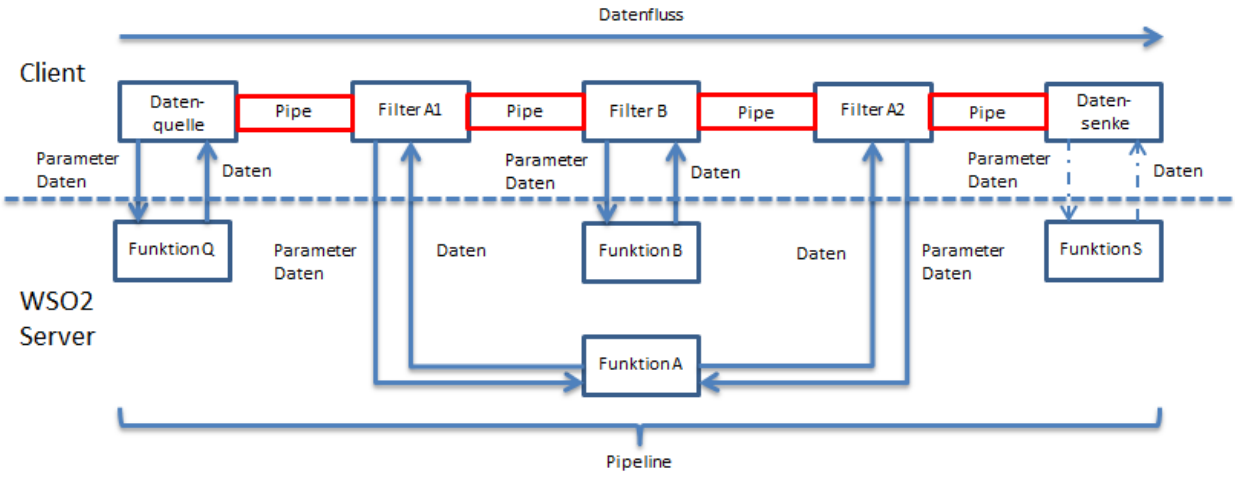


Abb. 25: Zusammenspiel Filter und Grundfunktionen / Webservices

5.3.2 „Model View Control“

View / Control

Für die grafische Darstellung der Filter wird ein Dialog Widget des jQueryUI als Basis Widget verwendet. Je nach abzubildender Filterkomponente werden nach Bedarf dem Basis Widget GUI Komponenten wie Eingabefelder und Tasten per Prototyping hinzugefügt. Neben globaler Steuerung des Aussehens (Theme) durch globale CSS Vorgabemöglichkeiten von jQueryUI, hat dies den weiteren Vorteil, dass in dem Basis Dialog Widget bereits ein Lifecycle Management implementiert ist. Das Fehlen bzw. die inkorrekte Implementation eines Lifecycle Managements kann dazu führen, dass nicht mehr benutzte Speicherbereiche nicht wieder freigegeben werden (Memory Leak).

Die Kombination von HTML und Javascript führt zu einer engen Kopplung von der Darstellung (View) und dem Eventhandling (Control). Diese enge Kopplung macht sich auch in jQuery bzw. jQueryUI bemerkbar.

In der konkreten Implementation wird das Basis Dialog Widget verwendet und per Prototyping erweitert. Die Konstruktor Funktion des Basis Dialog Widget wird der Prototype Eigenschaft eines Objektes zugewiesen. Auf diese Weise erbt das Objekt alle Eigenschaften und Methoden des Basis Dialog Widgets. Das Objekt wird anschließend um spezifische GUI Komponenten erweitert und mit entsprechenden Actionlistenern dekoriert. Anschließend wird die Konstruktor Funktion des Objektes für die Erzeugung der Widget Objekte verwendet. In Anlehnung an Schaubild 20 wäre das jQueryUI Basis Dialog Widget das Pendant zu Objekt A, während die auf die eigenen Bedürfnisse angepassten Widgets Datenquelle und Selektionsfilter Objekt B bzw. Objekt C entsprechen.

Model

Das Model besitzt in diesem Konzept zwei Teile. Der Hauptteil stellt die Daten dar, die durch die WSO2 WebServices verarbeitet und transformiert werden. Der zweite Teil stellt die Daten dar, die der Client vorhält. Hierzu gehören zum Beispiel die Parameter oder die Referenzen zum benachbarten Widget.

5.3.3 „Client – Server“

Server

Mit der Wahl des WSO2 Mashup Servers ist bereits eine Entscheidung für eine Client Server Architektur gefallen.

Client

Durch die Verwendung von zustandslosen Webservices zur Abbildung der Filter Komponenten und der teilweisen Verlagerung des Models auf den Client, ist eine Implementation eines Smart Clients vorgegeben. Für zustandsbehaftete Webservices sind weitere Maßnahmen wie zum Beispiel ein Sessionhandling auf dem Server notwendig.

5.4 Implementierung

Im Rahmen dieser Arbeit wird ein funktionsfähiger Evaluierungs-Prototyp eines grafischen Werkzeugs zur Erstellung von Mashups auf Datenebene implementiert. Der Prototyp soll als „Proof of Concept“ dienen und zeigen, dass die zuvor betrachteten Konzepte tragfähig sind.

Datenquelle

Für den Evaluierungs-Prototyp wurde ein auf den Feed Host Object des WSO2 Mashup Servers basierender Feed Webservice erstellt. Der Feed Webservice fungiert im Rahmen des Evaluierungs-Prototyps als eine Datenquelle nach dem „Pipes and Filter“ Architektur Musters. Der Webservice erwartet als Eingabe eine Feed URL und gibt den entsprechenden Feed zurück. Feeds liegen meist als XML Format wie RSS und ATOM vor. Durch die Verwendung des Javascript Compilers Mozilla Rhino, welches E4X (ECMAScript for XML) unterstützt, kann der WSO2 Mashup Server mit XML und XMLList (Liste von XML Objekten) als Datentyp umgehen und bietet für die XML Bearbeitung entsprechende Funktionen wie appendChild, parent (vgl. ECMA-357) an. XML wird im IBM Mashup Center ebenfalls für die interne Kommunikation verwendet. Mit XML als vorherrschendes (nicht notwendigerweise exklusives) Datenformat ergeben sich einige Vorteile. Einige andere Datenformate wie zum Bei-

spiel JSON[37], CSV, Datenbank Tabellen und Excel Spreadsheets lassen sich auf relativ einfache Weise in eine XML Struktur überführen. Für die anschließende weitere Aufbereitung für eine Präsentation der Daten ist XML ebenfalls gut geeignet.

Selektionsfilter

Analog zu den erstellten Beispiel Mashups (vgl. Kapitel 2 und 3) der kommerziellen Produkte erfolgt als nächster Verarbeitungsschritt die Filterung bzw. Selektion von Knoten nach einem einzugebenden Stichwort. Hierzu wurde eine Filter- bzw. Selektionsfunktion als Webservice implementiert. Der Webservice erwartet Daten im XML Format und gibt, wenn eine Übereinstimmung mit dem Stichwort vorliegt, anschließend den gesamten Elternknoten des durchsuchten Knotens zurück. Die zurück gegebenen Daten liegen ebenfalls im XML Format vor.

Datenquellen Widget

Als Basis Widget für die Datenquelle und den Selektionsfilter wird ein Dialog Widget des jQueryUI verwendet. Dieses Basis Widget wird um benötigte Eingabefelder und einen Button erweitert. Für die Datenquelle ist mindestens ein Eingabefeld für die Lokationsangabe der Datenquelle notwendig. Es wurden darüber hinaus weitere Felder für Test- und Debugging Zwecke implementiert. Zum Beispiel wurden Felder für die Speicherung und Anzeige der eigenen ID und die ID des Nachfolgers erstellt. Ausserdem wurde ein Feld für die Anzeige der Funktion, die am Eingang anliegt, erstellt.

Selektionsfilter Widget

Für den Selektionsfilter wurde über die genannten Felder hinaus noch zusätzlich ein Feld für die Vorgänger ID erstellt. Eine Vorgänger ID ist für Datenquellen nicht notwendig, da die Datenquelle das erste Glied in der Kette bildet.

Actionlistener / Eventhandler

Alle relevanten Komponenten des Widgets werden mit einem Actionlistener dekoriert. Eingabefelder achten auf Änderungen des Inhalts eines Eingabefeldes. Bei einer Änderung werden entsprechende Maßnahmen (Eventhandler) durchgeführt. Eine Änderung eines Parameters muss sich im Resultat bemerkbar machen. Eine Änderung des Selektionskriteriums muss zu

[37] http://www.xml.com/pub/a/2006/05/31/converting-between-xml-and-json.html?page=1

einer Änderung des Ergebnisses führen. Entsprechend müssen ebenfalls die Nachfolger des Widgets benachrichtigt werden.

Drag and Drop, Resize, Droppable

Das Basis Dialog Widget besitzt bereits Actionlistener / Eventhandler für Drag and Drop und Resize. Als weiterer Actionlistener wird Droppable von jQuerUI dem Basis Dialog Widget hinzugefügt. Dieser Actionlistener detektiert, wenn ein anderes Widget über das Basis Dialog Widget gezogen wurde.

Wiring

Mit Hilfe des Droppable Actionlisteners wird ein Wiring implementiert. Der Eventhandler überträgt die IDs und die Funktion mit der ein Widget assoziiert ist. Im ersten Schritt erfolgt die Visualisierung eines erfolgreichen Wirings über eine Farbänderung des Widgets. Im nächsten Schritt wird es angestrebt, analog zu den kommerziellen Produkten, eine Verbindungslinie zwischen den Widgets zu zeichnen und dadurch eine auch visuell nachvollziehbare Verkettung zu erreichen.

Ausgabe

Jedes Widget besitzt einen „Debug“-Button. Ein Klick auf den „Debug“-Button gibt die Daten von der Funktion zurück, mit der das jeweilige Widget assoziiert ist. Als Eingabe für die Funktion wird die Ausgabe der vorrangegangenen Funktion verwendet. Die Daten werden in einer Javascript Alert Box ausgegeben.

Erweiterbarkeit

Um einen weiteren Filter zu implementieren sind folgende Schritte notwendig. Als erstes muss ein für den Filter zugrundeliegender Webservice auf dem WSO2 Mashup Server implementiert werden. Anschließend ist ein um entsprechende Eventhandler und GUI Komponenten erweitertes Basis Dialog Widget zu erstellen.

Anwendungsbeispiel

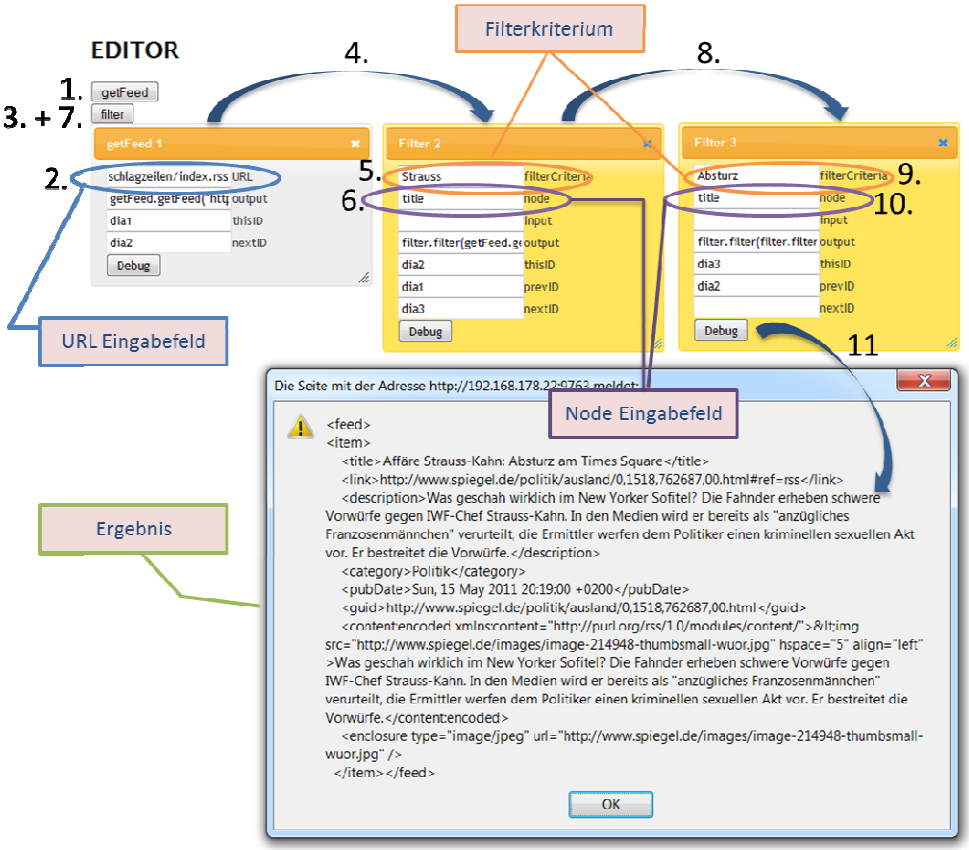


Abb. 26: Verwendungsbeispiel

Die Abbildung 26 fasst die nachfolgenden Einzelschritte zur beispielhaften Verwendung des Evaluierungs-Prototyps zusammen

1. „getFeed“-Button klicken → es erscheint ein „getFeed 1“-Widget.
2. Feed URL im URL-Eingabefeld des „getFeed 1“-Widgets eintragen.
3. „filter“-Button klicken → es erscheint ein „Filter 2“-Widget.
4. „getFeed 1“-Widget über das „Filter 2“-Widget ziehen bis sich die Farbe des „Filter 2“-Widget ändert → Wiring ist erfolgreich.
5. In „filterCriteria“-Eingabefeld das zu suchende Stichwort eintragen.
6. In „node“-Eingabefeld den Knoten eintragen, in dem gesucht werden soll.
7. „filter“-Button klicken → es erscheint ein „Filter 3“-Widget.
8. „Filter 2“-Widget über das „Filter 3“-Widget ziehen bis sich die Farbe des „Filter 3“-Widget ändert → Wiring ist erfolgreich.
9. In „filterCriteria“-Eingabefeld das zu suchende Stichwort eintragen.
10. In „node“-Eingabefeld den Knoten eintragen, in dem gesucht werden soll.
11. Klick auf „Debug“-Button ruft das Alert Fenster mit dem Ergebnis auf.

5.5 Zusammenfassung

In Anlehnung an die theoretischen Grundlagen wurde in diesem Kapitel ein grafisches Toolkit für die Erstellung von Mashups konzipiert und entwickelt.

Für die Sicherstellung der Wiederverwendbarkeit, Erweiterbarkeit und Flexibilität des Toolkits wurden zunächst die Architektur Muster „Pipes and Filters“, „Client-Server“ und „Model View Controler“ untersucht und bei der Entwicklung eingesetzt. Sowohl clientseitig als auch serverseitig wurde Javascript verwendet. Um eventuelle Browser-Inkompatibilitäten zu berücksichtigen und automatisch einen geeigneten Code zu verwenden, wurden die Javascript Bibliotheken jQuery und jQueryUI eingesetzt. Diese häufig verwendeten Bibliotheken bieten browserspezifische Funktionen für die Aspekte DOM Manipulation, Eventhandling und GUI an.

Im Rahmen dieser Bachelorarbeit wurde der Fokus auf die Basisfunktionen eines Mashups gelegt, um zu zeigen, dass die Konzepte grundsätzlich umsetzbar und tragfähig sind und weitere Möglichkeiten für zukünftige Erweiterungen aufzuzeigen. Es wurde ein Widget für eine Feed Datenquelle, ein Widget für einen Selektionsfilter sowie die notwendigen Webservices auf dem WSO2 Mashup Server implementiert. Des Weiteren konnte ein Wiring der Widgets per Drag and Drop demonstriert werden. Ein visuelles Wiring ist im ersten Schritt in Form eines Farbwechsels des Widgets implementiert. Ein sequentielles Wiring von mehreren Selektionsfiltern ist ebenfalls umgesetzt. Der Evaluierungs-Prototyp ist in der Lage für jeden verwendeten Selektionsfilter ein unterschiedliches Filterkriterium zu verwenden. Es können sowohl verschiedene Stichworte als auch verschiedene Knoten angegeben werden. Während der Entwicklungsphase konnten bereits weitere Funktionalitäten und Arbeitsmodi des Selektionsfilters erfolgreich getestet werden. Des Weiteren befindet sich eine Datei-Datenquelle bereits in der Konzeptphase. Erste Tests konnten erfolgreich Dateien einlesen.

Der Evaluierungs-Prototyp bestätigt die einfache Erweiterbarkeit um weitere Funktionalitäten. Um einen neuen Filter zu implementieren, müssen lediglich ein entsprechender Webservice und ein entsprechendes Widget erstellt werden. Der Webservice stellt die Funktionalität bereit, während das Widget die grafische Repräsentation der neuen Funktionalität für den Anwender darstellt. Der Wiring Mechanismus bleibt gleich, da das neue Widget ebenfalls nur die benachbarten Widgets kennen muss.

6. Fazit und Ausblick

Ohne das Wissen Daten bzw. Informationen im erforderlichen Kontext zu sehen und zu bewerten, bleiben Daten lediglich eine Ansammlung von Buchstaben, Zahlen und Zeichen. Erst die Synthese aus den Daten mit dem erforderlichen Wissen über den Kontext macht die Daten verwertbar und wertvoll. Das erforderliche Wissen, um die Daten richtig zu interpretieren, befindet sich in Unternehmen üblicherweise in den Fachabteilungen. Allerdings fehlt häufig im Gegenzug die Möglichkeit oder die Fertigkeit um die Daten zu aggregieren. In diesem Fall könnte ein End User Development Tool wie ein anwenderfreundliches grafisches Mashup Werkzeug verwendet werden. Das Mashup Werkzeug könnte dem Mitarbeiter aus der Fachabteilung, der üblicherweise keine bis wenige Programmierkenntnisse hat, die Möglichkeit bieten sich seine Daten „zusammen zu klicken“. Auf diese Weise liegen sowohl Fähigkeit Daten zu beschaffen als auch die Daten anschließend zu interpretieren in einer Hand.

Im Rahmen dieser Arbeit wurde ein Konzept für die Erstellung eines Toolkits für die Erstellung von Mashups auf Datenebene entwickelt und anhand eines Evaluierungs-Prototyps realisiert. Dazu wurden zunächst die verschiedenen Mashup Arten erörtert und die DSL EMML untersucht. Anschließend wurde eine Auswahl an grafischen Mashup Werkzeugen analysiert und bewertet. Der WSO2 Mashup Server konnte in Hinsicht auf Erweiterbarkeit und Kosten überzeugen und ist nicht zuletzt durch die Verwendung von Open Source Software die erste Wahl. Von kommerziellen Produkten konnten für die Anforderungs-Analyse viele Anregungen für die Erweiterungsmöglichkeiten des WSO2 Mashup Servers übernommen werden.

Bei der Konzeptionierung wurden die Architektur Muster „Pipes and Filters“, „Model View Control“ und „Client Server“ näher betrachtet und mit deren Berücksichtigung bei der Umsetzung die Wiederverwendbarkeit und Flexibilität sichergestellt werden.
Nach dem „Pipes and Filters“ Architektur Muster ist ein wesentlicher Bestandteil des Toolkits aufgebaut. Die Filter und Datenquellen werden durch zustandslose Webservices des WSO2 Mashup Servers abgebildet. Das Konzept der passiven Pipe wird auf kaskadierende Aufrufe von Webservices abgebildet. Der Client ist als als Smart Client konzipiert, der die Steuerung der Pipe übernimmt. Das Gesamtsystem ist nach dem MVC Muster entworfen, die Realisierung erfolgte auf Basis von jQuery/jQueryUI. Dabei entsprechen Filter und Datenquellen auf der Präsentationsebene jeweils einer GUI Komponente. Das Model verteilt sich über Client

und Server. Die Zustandsbehafteten Daten werden auf dem Client verwaltet. Die zustandslosen Transformationen erfolgen auf dem Server. Das Wiring – die Datenflusssteuerung – wird clientseitig ebenfalls grafisch unterstützt. Auf den Evaluierungs-Prototypen aufbauend können weitere Filter, Funktionen und Datenquellen implementiert werden.

Neben einfacheren kleineren Erweiterungsmöglichkeiten sind einige Punkte vorhanden, die konzeptionell eine Herausforderung darstellen.

Eine Erweiterungsmöglichkeit, die eine Herausforderung darstellt, ist die Implementierung einer aktiven Pipe. Erste Ansatzpunkte zur Implementation wurden in Kapitel 5.3.1 betrachtet. Eine aktive Pipe setzt Filter voraus, die selbstständig ihre Vorgänger-Pipe nach Arbeitspaketen bzw. Eingangsdaten abprüft. Wenn Eingangsdaten vorliegen, so soll die Verarbeitung der Eingangsdaten erfolgen. Die Eingangsdaten werden durch den Filter von der Pipe abgeholt, verarbeitet und in eine Nachfolger-Pipe geschrieben. Sollten keine Daten vorliegen, so soll der Filter in Ruhestellung gehen. Dieser Mechanismus könnte über einen Timer realisiert werden. Die aktive Pipe wird periodisch von dem Filter durch Polling abgeprüft ob Daten vorhanden sind. Die Pipe hat einen endlichen Speicherplatz. Für die Koordination und Steuerung dieser Ressource ist ein Semaphoren Konstrukt notwendig. Die Semaphoren müssen die Filter steuern und synchronisieren. Ein Filter kann nur so viel in eine Pipe schreiben, wie die Pipe Speicherplatz besitzt. Sind die Verarbeitungsgeschwindigkeiten der Filter unterschiedlich, so kann es dazu führen, dass ein Filter auf freien Platz in der Nachfolger-Pipe warten muss bevor eine weitere Verarbeitung von Eingangsdaten erfolgen kann.

Ein weiterer Punkt ist die Erstellung einer generischen Transformationskomponente, die ein Bindeglied zwischen Datenquellen und Filtern unterschiedlicher Datentypen darstellt. Die generische Transformationskomponente müsste Methoden zur Prüfung der Transformierbarkeit und der Transformation bereitstellen. Nicht kompatible Datenquellen und Filter dürfen nicht verbunden werden. Stattdessen muss ein entsprechender Hinweis auf die Inkompatibilität gegeben werden. Hierfür müssten die Filter mitteilen können welche Datentypen akzeptiert werden bzw. ausgegeben werden. Die generische Transformationskomponente müsste diese Informationen auswerten und eine geeignete Transformationsmethode aufrufen bzw. eine Fehlermeldung geben, wenn keine Transformierbarkeit vorliegt.

Ein weiterer Punkt ist die Möglichkeit der Speicherung der erstellten Mashups. Eine Möglichkeit ist die Implementierung von der Yahoo Pipes realisierten Methode. Yahoo Pipes verwendet das JSON Datenformat um die Mashups auf dem Server zu speichern (vgl. Kapitel 3.2.7). Zur Speicherung auf dem WSO2 Mashup Server könnte das File Host Object des WSO2 Mashup Servers verwendet werden. Mit diesem Host Objekt könnte eine Datei mit den JSON Daten des Mashups geschrieben werden. Die Datei könnte beim Aufruf des Mashups ausgelesen und ausgeführt werden. Eine Erweiterung dieser Funktionalität ist die Speicherung der Koordinaten der Widgets. Mit diesen Daten könnte auch die Position der Widgets wiederhergestellt werden.

Des Weiteren könnte der Evaluierungs-Prototyp, analog zu den Yahoo Badges oder JackBe Presto Views, um vordefinierte Präsentationen für die nur bisher auf Datenebene vorhandenen Mashups erweitert werden. Hierzu müsste ein Konzept von Grund auf durchdacht werden. Es müssen weitere Plug-Ins, Bibliotheken, Frameworks evaluiert werden, um geeignete Mittel zur Umsetzung zu finden. Für die Erstellung von Diagrammen könnte zum Beispiel „flot“[38], eine von Google entwickelte Erweiterung für jQuery speziell zum Zeichnen von Diagrammen, verwendet werden. Zur flexiblen Darstellung von Tabellen könnte „DataTables“[39], eine weitere Erweiterung für die jQuery Bibliothek. Mit dieser Erweiterung lassen sich interaktive Tabellen erstellen, die zum Beispiel eine Sortierung der Spalten ermöglichen. Ein reichhaltiger Fundus an Erweiterungen für die jQuery Bibliothek ist auf der jQuery Webseite zu finden[40]. Für die Implementierung einer umfangreichen Widget zu Widget Kommunikation können PubSub[41] Bibliotheken verwendet werden. Als Beispiel für eine PubSub Bibliothek ist die „OpenAjax Hub“[42] zu nennen. Allerdings wird diese Bibliothek von der jQuery Gemeinde wegen ihrer restriktiven Mitgliederpolitik und stellenweise inperformanten Implementation kritisiert[43].In Kombination mit dem Punkt „Speicherung des erstellten Mashups“ könnten die Präsentationen einfach die JSON Daten der Mashup Datei auslesen und somit auf bestehende Mashups aufbauen.

Eine weitere Möglichkeit zur Erweiterung liegt in der Erhöhung des Nutzungskomforts. In Eingabefeldern bzw. Auswahlfeldern soll dynamisch die möglichen Eingabemöglichkeiten in

[38] http://code.google.com/p/flot/
[39] http://www.datatables.net/
[40] http://plugins.jquery.com/
[41] Der PubSub Mechanismus ist vergleichbar mit dem Observer Muster [Gamma].
[42] http://www.openajax.org/index.php

Abhängigkeit von den Eingangsdaten bzw. Vorgänger-Widget angezeigt werden. Für diese Erweiterung müsste der Filter die anliegenden Daten prüfen und sinnvolle Vorgaben liefern. Ein Selektionsfilter müsste zum Beispiel die Eingangsdaten beim Wiring analysieren und die selektierbaren Knoten zur Filterauswahl in einem Auswahlfeld zur Selektion anbieten. Dies könnte durch parsen der XML Struktur geschehen und per DOM Manipulation in den Optionen eines Auswahlfeldes schreiben.

Durch sukzessive Erweiterung der in dieser Arbeit gelegten Basis um die drei letztgenannten Punkte lässt sich auf diese Weise das Toolkit für einen Open Source Server auf Funktionalitäten erweitern, wie sie zum Beispiel aus der JackBe Presto Umgebung bekannt sind.

[43] http://ejohn.org/blog/thoughts-on-openajax/

7. Glossar

Atom

Ein standardisiertes Syndication-Dateiformat, vergleichbar RSS-Feed, das von Online-Redaktionen benutzt wird, um ihre Artikel und News ins Web zu stellen und mit anderen Webseiten auszutauschen. Atom wird aber ebenso von Bloggern für ihre Weblogs benutzt. Wie bei RSS werden die Atom-Dateien als Feeds oder Kanäle bezeichnet. Atom-Feeds sind etwas komplexer als RSS-Feeds, beinhalten allerdings auch komplexere Informationen. Es bietet eine hohe Konsistenz hinsichtlich des gesamten Inhalts, dessen Speicher- und Editierbarkeit.

Cloud-Computing

Ein IT-Konzept, das auf verteilten Ressourcen Cloud-Dienste von einem oder von mehreren Service-Providern anbietet. Cloud-Dienste werden hierarchisch in drei Ebenen gegliedert, wobei die unterste Ebene die IT-Leistungen der Infrastruktur umfassen. Auf dieser Ebene geht es um Hardware, um Rechenkapazitäten, Speicherplatz und die Kommunikationsverbindungen mit der darunter liegenden Netzinfrastruktur angeboten. Als Cloud-Services stehen Infrastructure-as-a-Service (IaaS), Storage-as-a-Service und Communication-as-a-Service zur Verfügung.

Cloud Tag

Eine Methode zur Informationsvisualisierung, bei der eine Liste aus Schlagwörtern, oft alphabetisch sortiert, flächig angezeigt wird, wobei einzelne unterschiedlich gewichtete Wörter größer oder auf andere Weise hervorgehoben dargestellt werden. Es ist somit möglich, zwei Ordnungsdimensionen (die alphabetische Sortierung und die Gewichtung) gleichzeitig darzustellen und auf einen Blick erfassbar zu machen. Wortwolken werden zunehmend beim gemeinschaftlichen Indexieren und in Weblogs eingesetzt.

ECMAScript für XML (E4X)

E4X bringt native XML-Unterstützung für ECMAScript-konforme Programmiersprachen (z. B. JavaScript). Das Ziel ist es, eine alternative, einfachere Syntax für das Bearbeiten von XML-Dokumenten zu bieten als die bekannte DOM-Schnittstelle.

Extensible Stylesheet Language Transformation (XSLT)

Eine vom World Wide Web Consortium (W3C) standardisierte XML-basierte Beschreibungssprache für die Konvertierung zwischen verschiedenen XML-Dialekten. Mit XSLT können XML-Daten für die Darstellung und Ausgabe in ein gewünschtes Ausgabeformat konvertiert werden. Dabei werden die XML-Dateien nach den XSLT-Regeln von XSLT-Prozessoren in das entsprechende Ausgabeformat für die Darstellung und Ausgabe transformiert. Als Transformation von XSL (Extensible Stylesheet Language) kann XSLT Dokumente nach bestimmten Regeln in andere Dokumente, wie HTML-Dokumente, transformieren und nutzt bei der Weiterverarbeitung von Dokumenten XPath. Mit XSLT wird eine Trennung zwischen Daten und deren Darstellung erreicht.

Mashup

Der Begriff Mashup bedeutet so viel wie etwas vermischen. Es ist einer der vielen neuen Begriffe aus dem Umfeld von Web 2.0. In diesem Kontext versteht man darunter die Kombination verschiedener Webservices über eine offene Programmierschnittstelle (API).

Mashables

Informationen oder Datenquellen, die für den Einsatz in Mashups / oder Apps registriert wurden. Mashable-Quellen können interne Anwendungen in Organisationen sein, die einen Web-Feed oder Web Service-Schnittstelle haben. Sie basieren meistens auf gemeinsame Standards wie SOAP, REST, RSS-oder Atom-Protokollen. Mashable-Quellen können auch externe Web-Feeds oder Web Services sein, oder sie können aus Datenbanken, Excel-Arbeitsblättern, CSV-, XML-Dateien, SharePoint-Listen, durch Web-Clipping-Daten von Web-Sites erstellt werden.

Middleware

Eine zusätzliche Schicht in einer komplexeren Software-Struktur, deren Aufgabe es ist, die Zugriffsmechanismen auf unterhalb angeordnete Schichten zu vereinfachen und die Details deren Infrastruktur nach außen hin zu verbergen. Dazu stellt die Middleware Funktionen zur Verteilung sowie Dienste zur Unterstützung der Anwendung bereit. Dahingehend ist das Ergebnis einer Middleware die Entlastung der Anwendungs-Software und ebenfalls es ist Ziel, durch eine höhere Produktivität den Entwicklungsprozess zu optimieren. Man unterscheidet zwei verschiedene Kategorien von Middleware - kommunikationsorientierte sowie anwendungsorientierte Middleware. Die kommunikationsorientierte Middleware stellt zunächst eine logische Infrastruktur zur Kommunikation bereit. Die anwendungsorientierte Middleware bietet über den Rahmen der Kommunikation hinaus eine Erweiterung der Laufzeitumgebung, Dienstkomponenten sowie ein Komponentenmodell. Durch den Aufwand einer zusätzlichen Schicht trägt die Einführung einer Middleware nicht positiv zu einer höheren Performance des Gesamt-Systems bei. Bekannte Middleware-Produkte sind die MQSerie von IBM, der Websphere Application Server oder die SAP Exchange Infrastructure von SAP.

Platform as a Service (PaaS)

Ein Cloud-Dienst, bei dem der Cloud-Provider dem Anwender eine Entwicklungsumgebung in Form eines Frameworks bereitstellt. Mit diesem Cloud-Dienst und den darin zur Verrfügung gestellten Programmierschnittstellen können Entwicklern über das Cloud-Computing ganze Entwicklungsumgebungen mit Datenbanken, Middleware und Anwendungssoftware angeboten werden. Dies dürfte für Webapplikationen besonders interessant sein, weil damit dem Anwender eine Infrastruktur für die Entwicklung hochperformanter PHP-Anwendungen zur Verfügung steht.

Representational State Transfer (REST)

Ein Architekturstil, mit dem Webservices realisiert werden können. Das REST-Architekturmodell beschreibt die Funktionsweise des WWW und dient als Referenz für zukünftige Erweiterungen und als Anleitung wie Web-Standards Web-gerecht eingesetzt werden können. Es abstrahiert die Architektur-Elemente in einem verteilten Hypermedia-System, konzentriert sich aber nicht auf Details, wie einzelne Komponenten implementiert oder in

welcher Syntax die Protokolle verfasst wurden, sondern es beschäftigt sich mit der Rolle und der Funktion der Komponenten. Ihre Beziehungen und Interaktionen untereinander werden durch ihre Uniform Resource Locator (URL) bestimmt. REST zeigt den fundamentalen Zusammenhang der Komponenten und Daten, die die Basis der Web-Architektur bilden und ihr Verhalten in netzwerkbasierten Anwendungen.

RSS Feed (Really Simple Syndication Feed)

Eine textbasierte Datei, vergleichbar XML, die aus einer Liste von Einträgen besteht. Ein RSS-Feed ist zuerst durch seine Version gekennzeichnet. Des Weiteren besteht ein RSS-Feed aus dem Titel, einer Zusammenfassung und einem Link zu der URL der Webseite, auf der die komplette Nachricht steht. Die Einträge können weitere Angaben umfassen wie den Namen des Autors, das Datum usw. In RSS-Feeds werden die einzelnen Einträge in einer strukturierten Form dargestellt und können dadurch von einem RSS-Reader gelesen werden. Die aus Headlines und der Zusammenfassung bestehenden RSS-Feeds haben über die URL einen direkten Link zum Artikel. RSS-Feeds haben keine einheitliche Extension, sondern benutzen u.a. *.xml, *.rss oder *.rdf. Verschiedene Feeds werden auch für Podcasts benutzt. Dazu muss ein <item>-Tag geändert werden und die Angaben über die URL, die Länge und den Typ enthalten.

Software-as-a-Service (SaaS)

SaaS basiert darauf, dass der Kunde seine Software nach Bedarf aus dem Internet als Software on Demand herunterlädt. Das SaaS-Konzept mit seinem On-Demand-Computing entstand durch die stärker werdende Einbindung des Internet in private und geschäftliche Anwendungen und damit der Entwicklung des Cloud-Computing. Da beim SaaS-Ansatz die Kosten für den Betrieb der informationstechnischen Systeme in den Verantwortungsbereich des Anbieters fallen, ist die Nähe zu Application-Service-Providern unverkennbar.

Screen Scraping

Alle Verfahren zum Auslesen von Texten aus Computerbildschirmen, speziell die Technologien, die der Gewinnung von Informationen durch gezieltes Extrahieren der benötigten Daten dienen.

Simple Object Access Protocol (SOAP)

Ein von Microsoft entwickeltes Kommunikationsprotokoll zum Zugang zu einzelnen Projekten im Internet. Es ist aus einem Remote Procedure Call (RPC) mit XML-Syntax entstanden, um Textbefehle über das Internet auf Basis von HTTP zu senden. SOAP ist ein schlankes Protokoll, mit dem proprietäre Module verpackt und mit allgemein verständlichen Schnittstellen versehen werden können. SOAP definiert einen Message-Austausch zwischen dem Programmobjekt, das nach SOAP-Diensten fragt, und dem Programmobjekt, das Dienste anbietet. Das Protokoll ist herstellerneutral und völlig unabhängig von der verwendeten Programmiersprache, dem Objektmodell und der jeweiligen Betriebssystemplattform. SOAP wird als Anwender-Schnittstelle bei Webservices eingesetzt. Es ist vergleichbar mit dem Internet Inter-ORB Protocol (IIOP). Ein SOAP-Dienst wird über eine entsprechende Anfrage angefordert. Die Anfrage führt über den Webserver, der als Transportprotokoll SMTP, HTTP oder das FTP-Protokoll implementiert hat. Der Webserver übergibt die Anfrage an einen SOAP-

Application-Server. Nach Überprüfung und Validierung des Anfragenden, kann die Anwendung die Nachricht interpretieren und den gewünschten SOAP-Dienst aufrufen.

Same Origin Policy (SOP)

Ein wesentliches Sicherheitselement in allen modernen Browsern und Webanwendungen zum Schutz vor Angriffen. Aus Sicherheitsgründen dürfen JavaScript und ActionScript nur dann auf Objekte einer Webseite zugreifen, wenn sie aus derselben Domain stammen, beispielsweise auf Elemente, Cookies, XML-Dateien und HTML-Dateien. Die SOP soll verhindern, dass in Formulare eingegebene Daten oder Cookies in die falschen Hände geraten.

Web 2.0

Die Entwicklung des Internets zeigt sich nicht zuletzt an den wesentlich höheren Datenraten, die zu neuen Webservices geführt haben. Parallel zu dieser Entwicklung haben sich interaktive Communities gebildet, auf deren Kommunikationsplattformen Ideen und Vorstellungen, Fotos, Videos, Daten und Software ausgetauscht werden. Diese neuen Webservices, mit denen technische, soziale, wissensbasierte und freundschaftliche Beziehungen zwischen Benutzern aufgebaut werden, haben zu der Bezeichnung Web 2.0 geführt, die 2004 auf einer Conference von Tim O'Reilly kreiert wurde.

Web Clipping

Eine Technik zur Reduzierung von größeren Webseiten, damit diese auf kleineren Displays dargestellt werden können. Diese von Palm entwickelte Technik hat den Vorteil, dass wichtige Informationen aus einer Standard-Webseite extrahiert und auf dem PDA gespeichert werden und jederzeit darstellbar sind, ohne dass der kleine Bildschirm mit Grafiken und Texten überladen wird. Durch die direkte Speicherung auf dem PDA werden außerdem die Zugriffszeiten reduziert, da die Webseiten direkt geladen werden können und nicht erst eine Kommunikation über Funknetze mit niedriger Datenrate aufgebaut werden muss. Die Datensicherheit ist insofern gewährleistet, als dass beim Abruf einer Clipping-Seite diese verschlüsselt mit dem SSL-Protokoll übertragen wird und somit eine vollständige verschlüsselte Verbindung zwischen PDA und Webserver besteht.

Webclipping-Application (WCA)

WCA ermöglichen dem Benutzer statische Informationen eines Webservers zu extrahieren und sie auf einen Web-fähigen PDA wie beispielsweise den Palmtop zu laden. Palm hat die WCA-Technik entwickelt mit der, der vom Web extrahierte Content hinsichtlich des Seitenlayouts und der Darstellung der Grafiken dem kleineren PDA-Display angepasst wird. Die WCA-Technik extrahiert dabei statische Informationen des Webservers und speichert die Daten wegen des einfacheren Zugriffs auf dem PDA.

Web Harvesting

Web-Harvesting-Software extrahiert automatisch Informationen aus dem Web, wo Suchmaschinen hierfür nicht in der Lage sind. Extraction Tools automatisieren Lesen, Kopieren und Einfügen, die als Informationssammlung zur Analyse notwendig sind. Sie haben sich für die Bündelung Informationen über Wettbewerber, Preise und finanziellen Daten aller Art als nützlich erwiesen.

Webservices

Selbstbeschreibende und eigenständig agierende Software-Komponenten, die sich auch untereinander aufrufen können. Möglich gemacht wird dies über Universal Description, Discovery and Integration (UDDI) - einem übergeordneten Verzeichnisdienst zur Veröffentlichung von Webservices. Plattformen zur Realisierung von Webservices sind: .NET, PHP, Java mit JAX-RPC. Es sind aber auch Werkzeuge für die Realisierung diesbezüglicher Entwicklungen für die Programmiersprachen C und C++ verfügbar. Zur Realisierung der integrierten Plattform für Dienste-orientierte Geschäftsmodelle, Service Oriented Architecture (SOA) bilden Webservices neben anderen Verfahren eine der Basismethoden. Webservices können über die ihnen zugeordneten Uniform Resource Identifier (URI) erreicht werden. Bekannte Webservices werden beispielsweise durch Google oder Amazon betrieben, wo auf das Internet bezogene Dienstleistungen angeboten werden.

Wizard

Programmtechnisch realisierte Assistenten/Hilfe, die durch einen Prozess anleiten.

8. Literaturverzeichnis

Buschmann, Frank; Meunier, Regine; Rohnert, Hans; Sommerlad, Peter: *Pattern-Oriented Software Architecture Volume 1: A System of Patterns.* John Wiley & Sons Ltd., England, 1996.

de Vrieze, Paul; Xu, Lai; Bouguettayay, Athman; Yangz, Jian; Chen, Jinjun: *Process-oriented Enterprise Mashups*, St. Gallen, Switzerland, 2009.

Hanson, J. Jeffrey: *Mashups – Strategies for the Modern Enterprise*. Pearson Education, Boston, 2009.

Gamma, Erich; Helm, Richard;. Johnson, Ralph E: *Design Patterns. Elements of Reusable Object-Oriented Software,* Addison-Wesley Longman, Amsterdam, 1994

Java Magazin, Software & Support Verlag, September 2006.

Lieberman, Henry; Paternó, Fabio; Wulf, Volker*: End User Development*, Springer, Dordrecht 2006.

Ogrinz, Michael: *Mashup Patterns – Designs and Examples for the Modern Enterprise*. Pearson Education, Boston, 2009.

Seidenberg, Ulrich: *„Ist Informatioin als eigenständiger Produktionsfaktor aufzufassen?“*, Siegen 1998.

Yee, Raymond: *Pro Web 2.0 Mashups - Remixing Data and Web Services*. Springer-Verlag, New York, 2008.

Ziegler, Peter-Michael: *Kratzbürstig - Landgericht entscheidet im Screen-Scraping-Rechtsstreit.* c’t 23, 2010.

http://download.oracle.com/docs/cd/E19509-01/820-5699/ref_mdm-primer-sdm_c/index.html – Zugriffsdatum: 30.03.2011

http://nestor.sub.uni-goettingen.de/handbuch/artikel/nestor_handbuch_artikel_146.pdf – Zugriffsdatum: 13.01.2011

http://pipes.yahoo.com/pipes/ – Zugriffsdatum: 30.01.2011

http://upload.wikimedia.org/wikipedia/commons/1/12/Meta-search-de.svg – Zugriffsdatum: 13.01.2011

http://wso2.com/products/mashup-server/ – Zugriffsdatum: 22.01.2011

http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-357.pdf

http://www.ecma-international.org/publications/standards/Ecma-357.htm – Zugriffsdatum: 30.01.2011

http://www.enzyklo.de – Zugriffsdatum: 02.02.2011

http://www.enzyklo.de – Zugriffsdatum: 10.01.2011

http://www.gartner.com/it/products/podcasting/asset_184657_2575.jsp – Zugriffsdatum: 12.04.2011

http://www.heise.de – Zugriffsdatum: 20.02.2011

http://www.housingmaps.com – Zugriffsdatum: 10.01.2011

http://www.it-academy.cc – Zugriffsdatum: 10.01.2011

http://www.itwissen.info – Zugriffsdatum: 10.01.2011

http://www.jackbe.com – Zugriffsdatum: 05.02.2011

http://www.jackbe.com/resources/jackopedia.php – Zugriffsdatum: 07.02.2011

http://www.openajax.org/index.php

http://www.openmashup.org – Zugriffsdatum: 10.02.2011

http://www.perl.com/pub/2003/01/22/mechanize.html – Zugriffsdatum: 18.01.2011

http://www.xml.com/pub/a/2007/11/28/introducing-e4x.html – Zugriffsdatum: 18.01.2011

Krishnamurthy, Raj; Alur, Deepak; “EMML in 15 Minutes”;
http://www.jackbe.com/presto/emml15/ - Zugriffsdatum: 14.03.2011

PENTASYS AG: Mashup-Frameworks;
http://wiki.computerwoche.de/doku.php/web_2.0/enterprise-mashups – Zugriffsdatum: 02.02.2011

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §22(4) bzw. §24(4) bzw. §25(4) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 25. Mai 2011 _____

Ort, Datum Unterschrift