DAVID AUMÜLLER · ANDREAS THOR

Mashup-Werkzeuge zur Ad-hoc-Datenintegration im Web

Mashup-Werkzeuge ermöglichen eine einfache Erstellung von Mashups, d.h. von Webapplikationen, die Informationen aus verschiedenen Quellen kombinieren und in integrierter Form wieder selbst als Datenquelle oder Dienst anbieten. Dieser Artikel gibt einen Überblick über den aktuellen Stand der Technik von Mashups und einschlägigen Werkzeugen und geht dabei speziell auf die Möglichkeiten zur Datenintegration ein. Am Anfang werden Mashups und typische Anwendungsszenarien vorgestellt, um dann die wesentlichen funktionalen Komponenten einer Mashup-Anwendung zu charakterisieren. Dabei wird der Einsatz von Mashups zur Datenintegration betrachtet und mit bestehenden klassischen Datenintegrationsansätzen verglichen. Anschließend wird eine Reihe von Mashup-Werkzeugen vorgestellt und kategorisiert, wobei Tools zur Modellierung von Datenflüssen u.a. bzgl. der verfügbaren Operatoren untersucht und gegenübergestellt werden.

1 Einführung

Mashups bezeichnen Webanwendungen, die Daten bzw. Dienste verschiedener Quellen miteinander kombinieren und dadurch einen Mehrwert für den Nutzer schaffen. Zusammenhänge zwischen Informationen aus unterschiedlichen Quellen (z.B. Preise gleicher Produkte bei verschiedenen Anbietern) lassen sich so leichter erschließen. Die zunehmende Verbreitung solcher Applikationen (ProgrammableWeb.com listete im Frühjahr 2008 über 3000 Mashups mit einer Steigerungsrate von ca. drei Mashups pro Tag) geht zum einen auf den Anstieg frei verfügbarer Webservices zurück. So lässt sich durch entsprechende Schnittstellen (APIs), z.B. von Google, Amazon und eBay, einfach auf deren Datenbestände oder Dienste (Suche, Kartenvisualisierung) zugreifen, um sie zu neuen Anwendungen zu kombinieren. Ein weiterer Grund ist die breite Akzeptanz von Ajax, einem aktuellen Webapplikationsmodell mit Technologien zur asynchronen Übertragung und Präsentation von Informationen. Damit werden meist geringe Datenmengen zwischen Server und Browser dynamisch im Hintergrund ausgetauscht, sodass sich derartige Web 2.0-Applikationen nahezu wie lokale Clientanwendungen verhalten.

Die Definition von Mashups erlaubt eine Vielzahl möglicher Realisierungsformen, weshalb Mashups nach verschiedenen Kriterien kategorisiert werden können. Novak und Voigt [Novak & Voigt 2007] identifizieren als Kriterien u.a. den Applikationstyp, die verwendete Technologie, die soziale Infrastruktur und die Offenheit der Webanwendung. Im Folgenden werden die häufigsten Applikationstypen kurz vorgestellt (vgl. [Novak & Voigt 2007; Merill 2006]):

- Mapping-Mashups integrieren Daten, die beliebige Ortsinformationen enthalten, in online verfügbare Karten (maps). Seit der Veröffentlichung entsprechender Mapping-APIs von Google, Yahoo und Microsoft ist es relativ einfach, derartige Daten zu kombinieren und auf interaktiven Landkarten mit den zugehörigen ortsbezogenen Informationen zu visualisieren. Dabei wird meist ein geografischer Bezug zu bereits online verfügbaren Informationen (z.B. Restaurantadressen) geschaffen.
- Foto- und Video-Mashups erhalten durch das Aufkommen von Foto-Hosting-Seiten (z.B. Flickr) und Videoportalen (z.B. YouTube) mit ihren zugehörigen Webservices einen großen Auftrieb. Da häufig beschreibende Metadaten zu den Bildern bzw. Videos gespeichert werden, ist es möglich, Mashups zu erstellen, die externe Daten mithilfe dieser Metadaten integrieren. So können zum Beispiel aktuelle Nachrichten mit zugehörigen Bildern oder Videos kombiniert werden. Durch die Verwendung geografischer Koordinaten der Fotos (z.B. mittels eines integrierten GPS-Empfängers in der Kamera oder Zeitstempelabgleichs) können diese auch mit einem örtlichen Bezug dargestellt werden (vgl. Mapping-Mashups).
- Such- und Shopping-Mashups existierten lange vor der Begriffsbildung des Mashups und Web 2.0.-Anbieter wie Google Froogle oder PriceGrabber benutzen Business-to-Business-Technologien oder Screen-Scraping, um Vergleichsinformationen zu Produkten von verschiedenen Anbietern zu erhalten. Heute stellen große E-Commerce-Anbieter, z.B. Amazon oder eBay, Webschnittstellen zur Verfügung, die den Zugriff auf die Produktdaten und damit die Erstellung derartiger Mashups erleichtern.
- Nachrichten-Mashups kombinieren die von Nachrichtenagenturen erstellten Meldungen und/oder Beiträge in Weblogs, Foren u.Ä. Neben der Möglichkeit, gleichzeitig in mehreren Nachrichtenquellen zu suchen (z.B. Google News), können Nutzer Beiträge bewerten und dadurch die Gewichtung bzw. Reihenfolge der Nachrichten beeinflussen (z.B. Digg.com). Auch hier ist oft ein geografischer Bezug der Nachrichtenmeldung möglich (vgl. Mapping-Mashups).

Viele im Web verfügbaren Mashups haben spielerischen Charakter oder liefern nur einer eingeschränkten Community einen Nutzen (z.B. die Visualisierung privater Gebrauchtwagenangebote auf einer Landkarte). Dennoch können Mashups auch im Geschäftsumfeld Erfolg versprechend eingesetzt werden, z.B. durch die Visualisierung von Hotelstandorten inklusive (unternehmensabhängiger) Zimmerpreise in der Nähe eines Ortes für einen Geschäftstermin. Die Kartendarstellung erlaubt dabei eine schnelle Entscheidung, da Entfernung, Verkehrsanbindung und Kosten integriert dargestellt werden. Zusätzlich entsteht im Geschäftsumfeld das Verlangen nach einer schnellen Entwicklung sogenannter Situational Applications [Jhingran 2006], d.h. Anwendungen, die für einen bestimmten Zweck ad hoc realisiert

werden müssen. Daher kommt der effizienten Mashup-Erstellung, die durch entsprechende Tools unterstützt wird, zukünftig eine wichtige Bedeutung zu. So könnten z.B. im Firmenintranet unternehmensinterne Daten auch aus (für Datenintegration unüblichen) Quellen wie E-Mails, Spreadsheets, Präsentationen, Web usw. zusammengebracht und dargestellt werden. Der Erfolg derartiger bottom-up erstellter Nischenprodukte kann zudem als Indikator für weiteren Bedarf interpretiert werden.

2 Mashups und Datenintegration

Mashups als Webanwendungen kombinieren die Daten und Dienste verschiedener Daten- und Serviceprovider und stellen das (integrierte) Ergebnis in einem standardisierten Format zur Verfügung, sodass Nutzer es mittels entsprechender Clients betrachten und weiterverarbeiten können. Somit interagiert (siehe Abb. 1) ein Mashup mit einem oder mehreren Daten- oder Serviceprovidern sowie dem (Nutzer-)Client [Merrill 2006]. Dabei lassen sich drei funktionale Komponenten innerhalb eines Mashups identifizieren (vgl. [Jhingran 2006]).

Datenextraktion: Die für Mashups verfügbaren Daten können über verschiedene Schnittstellen angefordert werden und in verschiedenen Ergebnisformaten bereitstehen. Anfragen an die Provider werden mithilfe standardisierter Protokolle realisiert, wobei HTTP, SOAP und REST am weitesten verbreitet sind. Für die von den Quellen zurückgelieferten Daten gibt es eine Vielzahl möglicher Formate. Webseiten werden im (X)HTML-Format repräsentiert; die Ergebnisse von Webservice-Aufrufen liegen üblicherweise im XML-Format vor. Zunehmend gewinnen auch leichtgewichtigere Austauschformate (z.B. JSON) an Bedeutung, die eine kompakte und generische Repräsentation von Datenobjekten ermöglichen. Speziell für News-Feeds haben sich die (XML-basierten) Standards RSS und Atom etabliert, die Listen von Datenobjekten mit definierten Attributen ermöglichen. Weitere mögliche Formate zum Datenaustausch sind das (aus Semantic-Web-Anwendungen bekannte) Resource Description Framework (RDF) sowie zunehmend sogenannte Microformats, d.h. explizite (semantische) Auszeichnungen von potenziell interessanten Dateneinheiten auf Webseiten, wie z.B. Datums- und Ortsangaben zu Veranstaltungen. Dabei erfolgt die Auszeichnung innerhalb HTML anhand eines standardisierten Vokabulars, ohne eine Unterstützung aufseiten des Webbrowsers zu verlangen. Um eine konsistente Weiterverarbeitung zu gewährleisten, müssen die Daten in einheitliche Formate überführt werden. Dabei können auch Techniken des Screen-Scrapings zur Anwendung kommen.

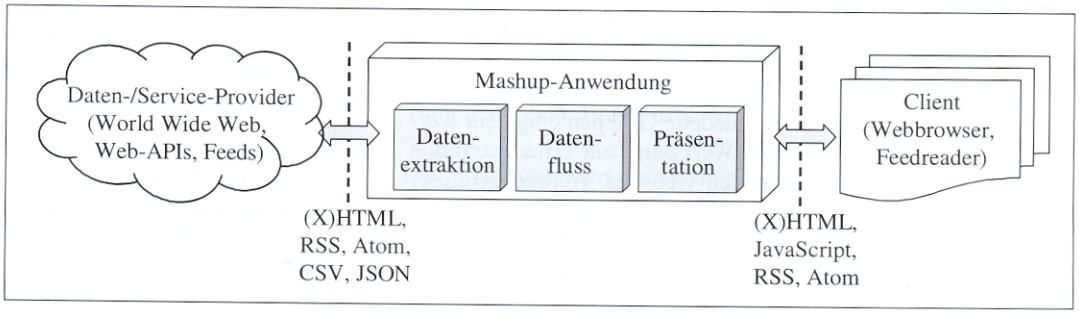
Datenfluss: Innerhalb der Mashup-Anwendung werden die extrahierten Daten transformiert und miteinander kombiniert. Die Mashup-Anwendung stellt dazu die benötigte Logik zur Verfügung, die z.B. (analog zu traditionellen Webanwendungen) in Form von Servlets, PHP-Skripten o.Ä. realisiert sein kann. Die Verarbeitung, d.h. Integration der Daten und Dienste, stellt damit aus Sicht der Datenintegration das zentrale Element dar und ist daher Schwerpunkt innerhalb dieses Beitrags.

Präsentation: Die Mashup-Anwendung wird mithilfe eines Clients aufgerufen. Typischerweise ist dies ein Webbrowser, der das Mashup-Ergebnis visualisiert und Möglichkeiten zur Interaktion bietet. Dazu generiert die Mashup-Anwendung entsprechenden (X)HTML-Code, der um CSS und/oder JavaScript angereichert wird. Kommt als Client ein Newsreader zum Einsatz, müssen die Daten in einem entsprechenden Feed-Format (RSS, Atom) ausgegeben werden.

Aufgrund der identifizierten funktionalen Komponenten können Mashups als eine besondere Art von Anwendungen zur Datenintegration angesehen werden, die wir im Folgenden von »klassischen« Datenintegrationsansätzen (DI-Ansätze), wie z.B. Data Warehouses oder Query-Mediatoren, abgrenzen. Dazu werden Gemeinsamkeiten und Unterschiede bzgl. der Anwendungsentwicklung und Verwendung sowie der Integrationsart aufgezeigt.

Entwicklung: Ein wesentlicher Unterschied zwischen Mashups und klassischen DI-Ansätzen ist, dass der potenzielle Kreis der Mashup-Entwickler größer ist als für DI-Ansätze, da Kenntnisse zur Webprogrammierung weiter verbreitet sind als z.B. die Verwendung von ETL-Tools für Data Warehouses oder Mechanismen zur Anfrageverarbeitung bei Query-Mediatoren. Bei entsprechender Toolunterstützung ist evtl. eine Entwicklung sogar ohne Programmierkenntnisse möglich. Damit besteht im Geschäftsumfeld die Möglichkeit, Mashup-Anwendungen ohne zeitaufwendige Involvierung der IT-Abteilung bereits durch Mitarbeiter der Fachabteilungen realisieren zu lassen. Gleichzeitig sind bei Mashups durch die enge Verzahnung von Datenquellen, Datenfluss und Präsentation erste Ergebnisse bereits binnen weniger Stunden oder Tage möglich, wodurch sich Mashup-Anwendungen frühzeitig evaluieren und ggf. anpassen lassen. Die Erstellung von Mashups kann daher als prototypische Entwicklung von Datenintegrationsanwendungen angesehen werden, wobei erfolgreiche bzw. vielversprechende Anwendungen im Geschäftsumfeld anschließend von der IT-Abteilung weiterentwickelt und betreut werden können. Dem agilen und iterativen Entwicklungs-

Abb. 1: Mashup-Gesamtarchitektur und typische Komponenten



modell von Mashups steht bei klassischen DI-Ansätzen ein Prozess gegenüber, der erst nach einer gewissen Vorlaufzeit (u.a. Data Cleaning, Schemaintegration) für den Endnutzer Ergebnisse produziert.

Integrationsart: Der Zugriff auf die Datenquellen erfolgt ähnlich zu klassischen DI-Ansätzen mittels Wrappern. Dabei kann die Wrapper-Funktionalität sowohl innerhalb des Mashups implementiert sein oder durch den Aufruf entsprechender Dienste realisiert werden. Der wesentliche Unterschied zwischen Mashups und klassischen DI-Ansätzen besteht in der Definition des Integrationsprozesses. Mashups realisieren eine »Low-Level-Integration«, d.h., es erfolgt keine explizite semantische Beschreibung der Quellen und ihrer Verbindung zueinander (z.B. über ein Schema Mapping). Vielmehr definiert der Mashup-Entwickler einen fest codierten Datenfluss, was den späteren Verwendungseinsatz einschränkt (siehe nächsten Punkt). Gleichzeitig führen Mashups eine virtuelle Integration durch, d.h., die Extraktion und Kombination der Daten (innerhalb des Datenflusses) geschieht zur Laufzeit und wird im Allgemeinen nicht vorberechnet. Nutzer erwarten von Mashups als interaktiven Webanwendungen ein ansprechendes Laufzeitverhalten. Da die Daten oft erst zur Laufzeit integriert werden, sind Mashups daher auf wenige (gut ausgewählte) Quellen begrenzt und verarbeiten nur relativ kleine Datenvolumina.

Verwendung: Mashups verknüpfen relativ starr die Daten ausgewählter Quellen mit einer zugehörigen Nutzeroberfläche. Sie lassen sich daher als aufgabenspezifische Anwendungen charakterisieren. Demgegenüber stehen die klassischen DI-Ansätze als datenorientierte Anwendungen. So lässt z.B. ein Data Warehouse beliebige Analysen mit den integrierten Daten zu und ein Query-Mediator realisiert die Ausführung beliebiger Anfragen (sofern diese von der Anfragesprache unterstützt werden). Daraus resultiert ein unterschiedlicher Nutzerkreis bzw. Nutzungsgrad, d.h., klassische DI-Ansätze finden in vielfältiger Art und Weise Verwendung, während Mashups stark zweckgebunden sind. Gleichzeitig kann vermutet werden, dass die Lebensdauer von Mashups im Allgemeinen kürzer ist als von DI-Ansätzen, u.a. weil der Aufwand zur Erstellung bei Mashups deutlich niedriger gesetzt wird und z.B. auf Ausfallsicherheit nur geringer Wert gelegt wird.

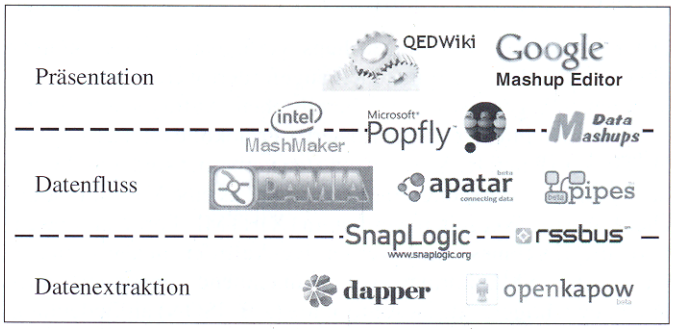
Der Vergleich von Mashups zur Datenintegration mit klassischen DI-Ansätzen zeigt, dass beide Richtungen ihre Berechtigung haben. Damit die Vorteile der Mashup-Entwicklung, insbesondere die schnelle Erstellung durch einen breiten Entwicklerkreis ohne größere Programmierkenntnisse, zur Geltung kommen, haben sich in der letzten Zeit Werkzeuge herausgebildet, die die Mashup-Erstellung entsprechend unterstützen.

3 Werkzeuge zur Mashup-Erstellung

Analog zu den geschilderten Mashup-Funktionen können derzeitige Mashup-Tools in drei verschiedene Gruppen eingeteilt werden. Die erste Gruppe beinhaltet Werkzeuge zur Datenextraktion von Informationen aus bestehenden Websites. Werkzeuge dieser Gruppe zeichnen sich insbesondere durch ausgereifte Methoden zur Extraktion von Daten aus HTML-Seiten aus. Innerhalb der Anwendungslogik konzentrieren wir uns auf Mashup-Tools zur Modellierung und Ausführung von Datenflüssen. Entsprechende Tools bieten Komponenten zur Datenverarbeitung (z.B. Transformation und Aggregation von Datenwerten und -objekten) an, die miteinander kombiniert werden können. Die dritte Gruppe beinhaltet Anwendungen zur Unterstützung der Präsentation, d.h. Werkzeuge, die eine integrierte Darstellung vorhandener Mashup-Komponenten innerhalb eines Frontends ermöglichen sowie Interaktion mit dem Endnutzer erlauben.

Eine Auswahl von Tools inklusive ihrer Einordnung in die genannten drei Gruppen zeigt Abbildung 2. Dabei ist die Einteilung in eine der drei Gruppen nicht scharf, d.h., es existieren Werkzeuge, die Funktionalitäten mehrerer Gruppen bieten. So unterstützt z.B. Microsoft Popfly sowohl die Erstellung von Datenflüssen als auch die Präsentation der Ergebnisse. Im Folgenden werden ausgewählte Tools zur Datenextraktion und Präsentation vorgestellt. Ein ausführlicher Vergleich mehrerer Tools zur Modellierung des Datenflusses erfolgt im darauffolgenden Kapitel 4.

Abb. 2: Kategorisierung von Mashup-Tools (Auswahl)



3.1 Tools zur Datenextraktion

Strukturierte Austauschformate stellen im Allgemeinen kein großes Hindernis zur Weiterverarbeitung der Daten dar, solange Struktur und Semantik bekannt sind. Schwieriger fällt die Verarbeitung von HTML-Daten, die primär zur Darstellung im Browser für den Nutzer gedacht sind und neben den eigentlichen Informationen auch Layoutangaben enthalten. Screen-Scraping bezeichnet den Prozess, mithilfe von Softwarewerkzeugen vorhandene Darstellungen von Daten zu analysieren und daraus Informationen zu extrahieren, die u.a. in die Erstellung von Mashups einfließen können. Diese Methode der Informationsgewinnung wird oft als weniger elegant angesehen, da sie unter anderem von der Darstellung der Informationen auf der Seite des Datenproviders abhängt und diese sich unerwartet ändern kann. Zur Extraktion bieten gängige Programmiersprachen (z.B. PHP, Perl, Java) entsprechende Funktionen und Bibliotheken an. Dieses Vorgehen setzt jedoch Programmierkenntnisse voraus, weshalb sich Tools entwickelt haben, mit deren Hilfe Anwender einfache Screen-Scraping-Aufgaben bewältigen können. Wir stellen im Folgenden mit Dapper und RoboMaker exemplarisch zwei Tools vor.

Dapper [Dapper] ist eine Webanwendung zur nutzerfreundlichen Erstellung und Ausführung von Wrappern (sog. Dapps) zur Datenextraktion von Websites. Der Anwender spezifiziert idealerweise mehrere Beispielwebseiten einer Site, die von Dapper auf strukturelle Ähnlichkeiten hin verglichen werden. Anschließend markiert der Benutzer visuell die ihn interessierenden Berei-

che zur Extraktion und benennt diese Elemente. Intern selektiert Dapper entsprechende Teilbäume der DOM-Repräsentation der Webseite. Die erstellten Dapps sind über eine eigene URL verfügbar, d.h., die Extraktion findet auf den Servern von Dapper statt. Dapps können im Online-Repository für andere Nutzer hinterlegt werden, die diese wiederum erweitern können. Als Ausgabe steht neben HTML eine Reihe von Formaten (XML, RSS, Atom) zur Weiterverarbeitung zur Verfügung. Weiterhin ist es möglich, mehrere Dapps zu einem komplexeren zu kombinieren, indem die Ausgaben einer oder mehrerer Anwendungen als Eingabe für einen anderen Wrapper benutzt werden.

OpenKapow RoboMaker [RoboMaker] ist eine frei verfügbare Desktop-Anwendung zur Erstellung von Extraktionsregeln (Robot), die anschließend als Webservice oder Webfeed aufgerufen werden können. Die Erstellung eines Robots erfolgt visuell durch Auswählen der gewünschten Bereiche einer Webseite. RoboMaker enthält eine Reihe nützlicher Extraktionsbausteine, z.B. Schleifen zur Extraktion sich wiederholender Bereiche innerhalb einer Webseite (z.B. Suchmaschinentreffer) sowie zur Weiterverfolgung von Next-Links. Der Nutzer wird durch ein Vorschaufenster, das die aktuelle Website enthält, sowie eine Ansicht der zugrunde liegenden DOM-Struktur und des HTML-Codes unterstützt, sodass im Vergleich zu Dapper eine genauere Kontrolle über die Extraktionsregeln vorherrscht. Sowohl Robots als auch Dapps können durch nutzerdefinierte Eingabewerte weiter parametrisiert werden.

Als Nachteil dieser Anwendungen ist einerseits die Abhängigkeit (Datensicherheit, Verfügbarkeit) zum Serviceprovider zu nennen, die sich aber durch entsprechende Verträge mit dem Anbieter zu definierten Konditionen als unproblematisch erweisen sollte. Andererseits bieten die Tools nicht die volle Kontrolle über den Extraktionsprozess wie der Einsatz einer Programmiersprache.

3.2 Tools zur Datenpräsentation

Mashup-Tools für die Präsentationsschicht unterstützen den Mashup-Entwickler bei der schnellen Erstellung (optisch) ansprechender Ergebnisdarstellungen. Dazu zählen auch die Möglichkeiten der Nutzerinteraktion, wobei Nutzer die Darstellung und/oder die Daten manipulieren können. Aus Sicht der Datenintegration sind diese Tools allerdings nur von sekundärem Interesse, weshalb im Folgenden nur zwei Tools kurz vorgestellt werden.

Google Mashup Editor [GME] ist eine webbasierte Programmierumgebung, die das Erstellen von Mashups über eine spezielle Erweiterung von HTML unterstützt. Datenobjekte wie z.B. Feeds können mittels Tags und Attributen direkt in eine Webseite auf Googles Servern integriert werden. Als Besonderheit sei hier die Unterstützung persistenter Nutzereingaben erwähnt.

IBM QED Wiki [QEDWiki] verfolgt einen Portal- bzw. Content-Management-Ansatz zur Erstellung von Mashups. Einzelne funktionale Programmbausteine (sog. Widgets) lassen sich gemeinsam auf einer Seite platzieren. Durch die Kommunikation zwischen Widgets wird eine interaktive Anwendung ermöglicht. So kann z.B. nach der Auswahl einer Person in einem Adressbuch-Widget der zugehörige Ort in einem Karten-Widget visualisiert werden.

4 Vergleich von Mashup-Werkzeugen zur Datenflussmodellierung

Aufbauend auf einer Ende 2007 durchgeführten Evaluation von Mashup-Werkzeugen im Rahmen eines Problemseminars zur Integration von Webdaten [Meinhold 2008] werden im Folgenden vier Tools vorgestellt und miteinander verglichen:
Apatar [Apatar], Microsoft Popfly [Popfly], IBM Damia [Damia] und Yahoo! Pipes [Pipes].

Die Funktionsweise der betrachteten Tools ist recht ähnlich. Als Eingabe fungieren eine oder mehrere Datenquellen, auf die durch die jeweiligen Tools zugegriffen werden kann. Alle Daten werden in ein einheitliches Datenformat konvertiert, das als eine Liste von Elementen (Datenobjekten) interpretiert werden kann. Je nach Tool wird ein unterschiedliches internes Format verwendet. Die einzelnen Elemente enthalten Attribute mit entsprechenden Werten und können, z.B. bei Damia, auch Unterelemente beinhalten.

Alle Tools stellen Operatoren für Datenquellen und Datensenken (die auch als Konnektoren bezeichnet werden) sowie zur Datentransformation zur Verfügung. Nutzer können den Datenfluss per Drag & Drop modellieren, d.h., sie ziehen Operatoren auf eine virtuelle Arbeitsfläche und verbinden diese miteinander, sodass die Ausgabe eines Operators als Eingabe für andere Operatoren fungiert. Zusätzlich können dem Datenfluss Parameter hinzugefügt werden. Die Ausführung eines Datenflusses beginnt mit dem Laden/Abfragen der Daten aus den definierten Datenquellen. Anschließend werden die Elementlisten durch die Transformationsoperatoren gemäß dem Datenfluss verarbeitet, ehe die resultierende Liste mittels einer Datensenke in ein definiertes Ausgabeformat überführt wird.

Die erstellten Datenflüsse werden auf den Webservern der Toolanbieter zusammen mit Metadaten (Namen, Schlagworte) hinterlegt, sodass auch hier (ähnlich der vorgestellten Datenextraktionswerkzeuge) eine Abhängigkeit vom Anbieter bzgl. Verfügbarkeit etc. vorliegt. Die Ausführung erfolgt jeweils auf den Servern, wobei auch freigegebene Datenflüsse anderer Nutzer ausgeführt werden können.

4.1 Übersicht der Tools

Es werden im Folgenden die Tools Apatar, Microsoft Popfly, IBM Damia und Yahoo! Pipes kurz vorgestellt. Tabelle 1 fasst deren Hauptmerkmale zusammen.

Tab. 1: Kurzübersicht der Mashup-Tools zur Datenflussmodellierung

	Apatar	Popfly	Damia	Pipes
Technologie	Desktop (Java)	Web (Silverlight)	Web (Ajax)	Web (Ajax)
Hersteller	Apatar	Microsoft	IBM	Yahoo
Datenformat	Tabelle	Unbekannt	XML	RSS

Apatar ist eine Java-Anwendung, die sowohl zur Erstellung von Mashups als auch zur Umsetzung eines Data-Warehouse-ETL-Prozesses geeignet ist. Daher besitzt Apatar zahlreiche Konnektoren für relationale Datenbanken oder CRM-Anwendungen. Apatar verwendet intern ein relationales Datenformat, sodass für die Kopplung von Operatoren jeweils die Abbildung des Eingangs- auf das Ausgangsschema explizit angegeben werden muss.

Microsoft Popfly nutzt das hauseigene Präsentations-Plug-in Silverlight zur interaktiven Darstellung in Webbrowsern. Popfly kapselt Datenquellen, Operatoren und Datensenken in sogenannten Blöcken, sodass z.B. für eine Datenquelle nur bestimmte Operatoren zur Verfügung stehen. Das Blockkonzept vereinfacht die Konfiguration einzelner Teile des Datenflusses für den Nutzer, verringert jedoch gleichzeitig die Flexibilität insbesondere bzgl. der Kombinierbarkeit von Operatoren bzw. Blöcken. Nutzer können eigene Blöcke definieren, was jedoch Programmierkenntnisse (XML, JavaScript) voraussetzt. Im Gegensatz zu den anderen Tools fokussiert Popfly bei Datensenken auf die visuelle Darstellung, sodass Ergebnisse in entsprechende Darstellungsblöcke für Texte, Videos, Landkarten, Musik oder Fotos eingebunden werden können. Diese Ausgaben können allerdings nur in ausgewählten Webseiten wie z.B. Facebook, MySpace-Blogs oder als Vista Sidebar Gadget verwendet werden, stehen also nicht weiteren Programmen als Feed zur Verfügung.

IBM Damia verwendet eine Ajax-basierte grafische Weboberfläche. Zur internen Repräsentation der Daten wird XML verwendet, sodass Elemente neben Attributen selbst wieder Unterelemente enthalten können. Die Navigation auf den Elementen erfolgt mittels XPath, wobei tiefere Kenntnisse darüber nicht nötig sind, da hierfür ein Assistent zur Verfügung steht.

Yahoo! Pipes ist ebenfalls eine Ajax-basierte Webanwendung, deren Name und auch mancher der Operatoren an die Verwendung von Pipelines unter Unix angelehnt ist. Pipes verwenden RSS und Atom als interne Datenformate und erlauben somit für nahezu alle Operatoren nicht nur eine Voranzeige im Debugger, sondern auch eine Ausgabe als Feed.

4.2 Datenquellen und Datensenken

Tabelle 2 zeigt einen Vergleich der unterstützten Formate für Datenquellen und -senken. Dabei werden nicht alle Datenquellen und -senken einzeln aufgeführt, sondern nur deren allgemeinere Typen betrachtet.

Alle Werkzeuge unterstützen die gängigen Feed-Formate RSS und Atom. Sehr heterogen ist die Unterstützung von Web-APIs, d.h., die Tools bieten jeweils unterschiedliche Spektren einschlägiger Dienste. Generell bietet Apatar die meisten Konnektoren.

4.3 Transformationsoperatoren

Für eine vergleichende Analyse der durch die Tools bereitgestellten Datentransformationsoperatoren wurden diese in zwei Klassen eingeteilt. Basisoperatoren erstellen aus einem oder mehreren Werten eines Elements einen neuen Wert. Die Anwendung eines Basisoperators auf eine Liste bewirkt somit eine Ausführung für jedes einzelne Listenelement. Demgegenüber transformieren Listenoperatoren komplette Listen von Elementen, z.B. durch Sortierung oder Filterung. Die Ein- und Ausgabe von Transformationsoperatoren (d.h. sowohl Basis- als auch Listenoperatoren) erfolgt jeweils in Listen, sodass die Operatoren beliebig kombiniert werden können.

Tabelle 3 zeigt einen Vergleich der Basisoperatoren, die von allen Tools mit Ausnahme von Damia bereitgestellt werden. Damia verwendet zur Manipulation von Attributen den Listenoperator Transform, d.h., auch wenn nur einzelne Werte verändert werden sollen, muss die komplette Liste transformiert werden, wobei alle anderen Werte unverändert übernommen werden. Die Bandbreite der von den Werkzeugen angebotenen Funktionalitäten reicht von einfachen Wertdefinitionen durch den Anwender bis hin zu speziellen höherwertigen Operatoren. String-Verarbeitung und String-

Tab. 2: Vergleich der Datenquellen und -senken

Konnektor		Apatar	Popfly	Damia	Pipes
Datenquellen	Dateien	Ja (CSV, XML, XLS)	Nein	Ja (XML, XLS)	Ja (CSV)
	Feeds	Ja	Ja	Ja	Ja
	APIs	Ja (HTTP, FTP, Web-APIs, DB)	Ja (Web-APIs)	Nein	Ja (Web-APIs)
	Nutzerdefiniert	Ja	Ja	Nein	Ja
Datensenken		Alle Quellen	HTML, MySpace, Vista Sidebar (keine Feeds)	RSS, Atom, XML	RSS, HTML

Tab. 3: Vergleich von Basisoperatoren

	Apatar	Popfly	Damia	Pipes
Nutzereingabe	Nein	Ja	(Ja)	Ja
String-Verarbeitung	Ja	Nein	Nein	Ja
String-Matching	Ja	Ja	(Ja)	Ja
Datumsverarbeitung	Ja	Nein	Nein	Ja
URL-Generator	Nein	Nein	Nein	Ja
Numerik	Ja	Ja	Nein	Ja
Höherwertige Funktionen bzw. Operatoren (Auswahl)	Plug-ins (u.a. Validierung von Adressen, Kreditkarte)	Nein	Nein	Location Builder/ Extractor, Term Extractor, Yahoo Shortcuts

Matching mittels regulärer Ausdrücke spielen eine wichtige Rolle und werden dementsprechend meist unterstützt, da insbesondere im Newsfeed-Bereich viele Elemente Textinhalt aufweisen.

Apatar und Pipes bieten neben einfachen Basisoperatoren auch höherwertige Operatoren, von denen einige in Tabelle 3 aufgeführt sind. Apatar unterstützt z.B. mit der Validierung von Adress- und Kreditkartendaten typische Anforderungen in E-Commerce-Anwendungen. Der Location Builder von Pipes ermittelt aus einem Adress-String geografische Informationen und liefert u.a. Längen- und Breitengrad, die insbesondere in Mapping-Mashups Anwendung finden können.

Tabelle 4 zeigt einen Vergleich der Listenoperatoren. Das Sortieren von Listen ist in allen betrachteten Werkzeugen möglich. Eine Listenreduzierung durch Filterung wird ebenfalls von allen Tools unterstützt, wobei jeweils ein entsprechendes Filterkriterium angegeben wird. Die Erkennung von Duplikaten innerhalb einer Liste ist nur über die Verwendung eines eindeutigen Attributwerts möglich. Dazu wird eine Liste mittels Distinct (Apatar) bzw. Unique (Pipes) um diejenigen Objekte reduziert, die in einem Attributwert (oder einer Gruppe von Attributwerten) mit einem anderen Objekt übereinstimmen.

Listentransformationen modifizieren die Struktur der Listen. Apatar, das Listen intern als relationale Datenbanktabelle repräsentiert, ermöglicht bei der Transformation eine Änderung des Schemas, d.h. z.B. eine Änderung der Attributnamen. Der Transform-Operator bei Damia ermöglicht die Erstellung neuer Subelemente bzw. Erstellung und Umbenennung von Attributen. Apatar bietet dazu analog die Operatoren Rename und Sub-Element an. Damias Transform-Operator dient zusätzlich der Erstellung und Veränderung von Attributwerten, was die fehlenden Basisoperatoren ersetzt.

Bei der Bildung der Vereinigung zweier Listen zeigen sich unterschiedliche Semantiken. Die Vereinigung als Konkatenation der Listen liefern Aggregate von Apatar sowie Union von Damia und Pipes. Eine andere Semantik der Vereinigung ist die Anreicherung eines Feeds mit den Elementen eines anderen. Dazu wird jeweils das n-te Objekt des ersten Feeds um die Elemente des n-ten Objekts des zweiten Feeds ergänzt. (Es handelt sich sozusagen um einen Verbund/Join unter Verwendung der Listenposition der Objekte.) Dieser Semantik folgt der Operator Augment in Damia. Demgegenüber kombiniert Combine von Popfly einzelne Attribute einer Liste zu einer neuen Liste. Einen Verbund unter Verwendung von Vergleichsattributen liefern Join und Merge von Apatar bzw. Damia.

Damias Group-Operator ermöglicht die Umgruppierung einer Liste, indem alle Objekte, die in einem Attributwert übereinstimmen, als Subobjekte eines Objekts zusammengefasst werden. Pipes bietet mit Count die Möglichkeit, die Anzahl der Listenobjekte zu ermitteln, wobei das Ergebnis jedoch nur innerhalb eines Basisoperators verwendet werden kann. In Apatar lassen sich durch die Verwendung von Datenbankfunktionen ebenfalls Listenoperationen realisieren, z.B. das Zählen oder Gruppieren von Objekten.

4.4 Mashup-Szenario

Für einen praktischen Vergleich wurde mit den vorgestellten Werkzeugen jeweils ein einfaches Nachrichten-Mashup erstellt. Dazu werden mehrere RSS-Feeds (news.com, slashdot.com und del.icio.us) kombiniert, wobei im resultierenden RSS-Feed doppelte Einträge eliminiert werden sollen. Zusätzlich soll ein zum Thema passendes Bild von Flickr.com hinzugefügt werden. Abschließend sollen die Elemente nach Datum sortiert angezeigt werden.

Mit einigen Abstrichen konnte das gewählte Szenario mit allen Werkzeugen innerhalb einer Stunde umgesetzt werden. Die Duplikaterkennung war mit allen Tools nur eingeschränkt möglich, da nur bei exakt gleichen Attributwerten (z.B. Titel) eine Eliminierung erfolgen konnte. Im Folgenden wird kurz auf die aufgetretenen Schwierigkeiten bei den einzelnen Tools eingegangen.

Apatar: Zur Speicherung und Verarbeitung (z.B. Sortieren) der Daten forderte Apatar das Anlegen benutzerdefinierter Tabellen, was den größten zeitlichen Aufwand ausmachte. Leider funktionierte in unserem Test der Konnektor zu Flickr nicht korrekt, sodass keine Bilder zugeordnet werden konnten.

Popfly: Hier stellte sich die Kombination der drei Feeds als größtes Hindernis heraus, da Popfly keinen Operator für die Konkatenation von Listen bereithält (siehe Vereinigung bei den Listenoperatoren). Hier musste ein Umweg über die Bildung von vier Listen (jeweils eine für Titel, Datum, Text und Link der News-Einträge) realisiert werden.

Damia: Zur Vereinigung der Feeds mit Damia war es nötig, die Feeds zuvor mittels Transform-Operator in die gleiche Struktur zu bringen. Dabei mussten die heterogenen Datumsformate vereinheitlicht werden, was durch fehlende Datumsfunktionen nicht vollständig gelang, und daher konnte die Sortierung nicht korrekt erfolgen.

Pipes: Mit Pipes konnte das Szenario am leichtesten umgesetzt werden. Hervorzuheben ist der Term Extractor als ein höherwer-

Tab. 4: Vergleich von Listenoperatoren

	Apatar	Popfly	Damia	Pipes
Sortieren	(DBS-Funktion)	Sort	Sort	Sort, Reverse
Filter	Filter	Filter	Filter	Filter, Tail, Truncate
Duplikate entfernen	Distinct	-	-	Unique
Transformation	Transform	-	Transform	Rename, Sub-Element
Vereinigung	Aggregate	Combine	Augment, Union	Union
Join	Join	-	Merge	-
Sonstige (Auswahl)	(DBS-Funktionen)	-	Group	Count

tiger Basisoperator, mit dessen Hilfe relevante Begriffe aus dem News-Titel extrahiert und für die Suchanfrage bei Flickr verwendet werden konnten.

5 Ausblick und Fazit

Die hier vorgestellten Tools sind relativ neu und z.T. wie Yahoo! Pipes im Web 2.0-typischen Beta-Stadium, was evtl. die Nutzer vor überzogenen Erwartungen bzgl. Erreichbarkeit und Funktionalität abhalten soll. Dennoch möchten wir, wie bereits in [Rahm et al. 2007] identifiziert, das Fehlen mächtigerer Operatoren zur Datenintegration erwähnen. Eine Unterstützung sogenannter Anfragestrategien wäre wünschenswert, was z.B. mittels Erstellung von erweiterten Query-Strings zur Anfrage an Suchmaschinen durch relevante Textbausteine verwirklicht werden könnte und auch schon mit Yahoo! Pipes' Term Extractor ansatzweise möglich ist. Nicht zuletzt um auch die Rückgabewerte solcher heterogener Anfragen sinnvoll zu verarbeiten, ist ein Operator zur unscharfen Duplikaterkennung nötig, der die Erkennung, Angleichung oder Eliminierung von Dubletten auch bei nicht identischen Werten (z.B. über Ähnlichkeitsfunktionen) unterstützt.

Mashups stellen aktuell ein interessantes und hochdynamisches Feld dar und sind dabei, erwachsen zu werden. Wurden sie lange vorrangig als Spielereien abgetan, so steckt in der Unterstützung zur einfachen Erstellung auch ein Potenzial zur Ad-hoc-Datenintegration. Mit den sogenannten Situational Applications entstehen im Geschäftsumfeld Anwendungen, die auf einen stark eingeschränkten Nutzerkreis spezialisiert sind, aber dort evtl. täglich anfallende Arbeiten erheblich erleichtern und je nach Bedarf die Grundlage zur Entwicklung einer vollwertigen Anwendung darstellen.

Danksagung

Wir danken Herrn Meinhold für die praktische Durchführung der Toolevaluation.

Tools

[Apatar] Apatar, *www.apatar.com.*
[Damia] IBM Damia, *http://services.alphaworks.ibm.com/damia.*
[Dapper] Dapper, *www.dapper.net.*
[GME] Google Mashup Editor, *http://code.google.com/gme/index.html.*
[Pipes] Yahoo! Pipes, *http://pipes.yahoo.com.*
[Popfly] Microsoft Popfly, *www.popfly.com.*
[QEDWiki] QEDWiki, *http://services.alphaworks.ibm.com/qedwiki.*
[RoboMaker] RoboMaker, *http://openkapow.com.*

Literatur

[Jhingran 2006] *Jhingran, A.:* Enterprise information mashups: integrating information, simply. Proc. of VLDB, 2006.
[Meinhold 2008] *Meinhold, M.:* Vergleich workflow-basierter Mashup-Werkzeuge. Problemseminar »Integration von Web-Daten«, Universität Leipzig, 2008, *http://dbs.uni-leipzig.de/stud/ws0708/webdatenintegration.*
[Merill 2006] *Merrill, D.:* Mashups: The new breed of Web app. 2006, *www.ibm.com/developerworks/library/x-mashups.html.*
[Novak & Voigt 2007] *Novak, J.; Voigt, B. J. J.:* Mashups: Strukturelle Eigenschaften und Herausforderungen von End-User Development im Web 2.0. In: i-com Zeitschrift für interaktive und kooperative Medien, (6) 2007.
[Rahm et al. 2007] *Rahm, E.; Thor, A.; Aumüller, D.:* Dynamic Fusion of Web Data: Beyond Mashups, Proc. of XSym07, 2007.

David Aumüller
hat seinen Master in Communications Studies an der University of Leeds erlangt und ist seit 2004 wissenschaftlicher Mitarbeiter an der Abteilung Datenbanken der Universität Leipzig.

Andreas Thor
studierte Informatik und ist seit 2003 wissenschaftlicher Mitarbeiter an der Abteilung Datenbanken der Universität Leipzig. Im Jahr 2008 promovierte er im Bereich Webdatenmanagement.

David Aumüller MSc
Dr. Andreas Thor
Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik
Johannisgasse 26
04103 Leipzig
{david, thor}@informatik.uni-leipzig.de
www.informatik.uni-leipzig.de