UNIVERSITÄT LEIPZIG
Fakultät für Mathematik und Informatik
Institut für Informatik

Vergleich workflow-basierter Mashup-Werkzeuge

am Beispiel von *„Apatar“*, *„Microsoft Popfly“*, *„IBM Damia“* und *„Yahoo! Pipes“*

Problemseminar „Integration von Web-Daten“

(Betreuer: Dipl.-Inf. Andreas Thor)

Leipzig, 30. Januar 2008

vorgelegt von: Martin Meinhold
geb. am: 17.09.1980
Studiengang: Informatik

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einführung

1.1 Was sind Mashups ?

Laut [WH06] existieren heute Unmengen von Informationen im Internet, die jedoch nicht immer in der vom Endbenutzer gewünschten Form zur Verfügung stehen. Häufig sind alle benötigten Informationen zugänglich, so dass eine Kombination der Daten möglich ist.

Am Anfang der Entwicklung von Datendarstellungen im Internet stehen die heute als *„Web 1.0“* bezeichneten Varianten. Dabei wurden statische HTML-Seiten verwendet und es gab keinerlei Trennung von Daten und Layout. In der folgenden Zeit wurden Informationen zunehmend dynamisch dargestellt und Webservices(1) entwickelten sich.

Nach [Mer06] sind Mashups eine neue Art von Anwendungen, die sich auf Daten stützen, die von externen Datenquellen extrahiert werden, um gänzlich neue Dienste zur Verfügung zu stellen. Sie sind ein Markenzeichen der zweiten Generation von Internet-Anwendungen, auch bekannt als *„Web 2.0“*. Die Internetseite `www.programmableweb.com` verzeichnet derzeit über 2500 derartiger Mashups. Die Entwicklung dieser Anwendungen erfolgt durch eine Entwicklergemeinschaft, deren Mitglieder nicht zwingend in einem wirtschaflichen Verhältnis zueinander stehen.

(1) Webservices sind ein platformübergreifendes Kommunikationsprotokoll [WsW].

1.2 Mashup-Arten

Mashup-Internetseiten können nach den Arten der Datenquellen unterschieden werden. Im Folgenden wird ein Überblick über häufig zu findende Mashups nach [Mer06] gegeben.

Mapping-Mashups

Diese Arten von Mashups integrieren Daten, die Ortsinformationen enthalten, in online verfügbare Karten. Seit der Veröffentlichung der *„Google Maps API*[(2)]*“* ist es einer Vielzahl von Personen möglich derartige Daten zu integrieren. *„Microsoft“* und *„Yahoo!“* veröffentlichten ebenfalls kurz darauf APIs für die Dienste *„Virtual Earth“* und *„Yahoo! Maps“*.

Foto- und Video-Mashups

Das Aufkommen von Foto-Hosting-Seiten wie *„`flickr.com`“* hat durch die zugehörigen APIs zu einer Reihe interessanter Mashups geführt. Da häufig Metadaten zu den Bildern bzw. Videos gespeichert werden, ist es möglich, Mashups zu erstellen, die externe Daten mit Hilfe dieser Metadaten integrieren. So könnten zum Beispiel aktuelle Nachrichten mit zugehörigen Bildern oder Videos integriert werden.

Such- und Shopping-Mashups

Diese Art von Mashups existierte lange vor der Begriffsbildung des Mashup. Anbieter wie *„Google Froogle“* oder *„PriceGrabber“* benutzten Business-to-Business-Technologien oder Screen-Scraping[(3)], um Vergleichsinformationen von verschiedenen Anbietern zu erhalten. Heute stellen Anbieter, wie *„Amazon“* oder *„eBay“*, APIs zur Verfügung, die die Erstellung derartiger Mashups erleichtern.

Nachrichten-Mashups

Nachrichtenagenturen, wie *„Reuters“* oder *„BBC“*, verwenden bereits seit 2002, die im folgenden Abschnitt näher erläuterten Technologien, RSS oder ATOM, um Nachrichten zu verbreiten. Damit ist es möglich, personalisierte Nachrichtendienste zu erstellen. So kombiniert zum Beispiel die Seite *„`digdot.us`“* Nachrichten der Seiten *„`digg.com`“*, *„`slashdot.org`“* und *„`del.icio.us`“*

(2) API ist ein Akronym für Application Programming Interface
(3) Technologie zur Extraktion von Informationen aus bestehenden Datenquellen [HC03]

1.3 Mashup-Technologien

Im Folgenden wird eine Zusammenstellung der mit Mashups in Verbindung stehenden Technologien nach [Mer06] angegeben.

1.3.1 Architektur

Eine Mashup-Anwendung besteht in der Regel aus drei Parteien: dem Daten-Provider, dem Mashup-Provider und dem Klienten.

Die Daten-Provider bieten ihre Informationen häufig über, die im nächsten Abschnitt genauer erläuterten Protokolle, an. Der Mashup-Provider stellt die benötigte Logik zur Verfügung. Das kann zum Beispiel, analog zu traditionellen Web-Anwendungen, in Form von Servlets, CGIs oder PHP-Scripten geschehen.

1.3.2 Datenformate

SOAP[4] und REST[5]

Beides sind plattformunabhängige Kommunikationsprotokolle, die von Klienten benutzt werden können, ohne Kenntnisse der Plattform zu besitzen, die den verwendeten Dienst zur Verfügung stellt. Beide Protokolle verwenden XML zum Austausch von Daten. Neben HTTP ist es ebenfalls möglich, SOAP-Nachrichten über andere Transportmechanismen, wie zum Beispiel E-Mail oder, JMS[6], zu übertragen.

RSS[7] und ATOM[8]

Beides sind, auf XML basierende, Kooperationsprotokolle mit deren Hilfe es möglich ist, Daten zur gemeinsamen Verwendung freizugeben. Diese Technologien werden häufig in so genannten *„Newsfeeds[9]“* verwendet und eignen sich hervorragend zur Erstellung von Mashups.

(4) Ursprünglich für Simple Object Access Protocol [SOA07]
(5) Representational State Transfer bezeichnet einen Softwarearchitekturstil für verteilte Informationssysteme [Fie00].
(6) Abkürzung für Java Message Service [HB+02]
(7) Abkürzung für Really Simple Syndication [RSS]
(8) Atom Syndication Format [ATO05]
(9) Oder auch nur Feed: Oberbegriff über XML-Dokumente in den Formaten RSS oder ATOM

Semantic Web[10] und RDF[11]

Die Ineffektivität von Screen-Scraping hängt damit zusammen, dass für Menschen aufbereitete Informationen nur schlecht oder gar nicht von Maschinen interpretierbar sind. Im Rahmen des Semantic Web wird versucht, diese Informationen mit Metadaten so weit anzureichern, dass eine maschinelle Verarbeitung möglich wird. In der Form von RDF stehen Methoden zur Verfügung, um derartige syntaktische Strukturen zu erstellen. Diese Metainformationen können mit Hilfe von Semantic-Web-Suchmaschinen, wie *„Swoogle“*, analysiert und durchsucht werden.

1.3.3 Softwaretechnologien

Web-Anwendungen

Web-Anwendungen zeichnen sich durch eine Darstellung in einem Browser aus. Sie haben den Vorteil, dass auf dem Rechner des Klienten keine weitere Software installiert werden muss. Die Verarbeitung der Daten erfolgt zentral auf einem Server.

Rich-Client-Anwendungen

Es ist jedoch auch möglich, die Integration der Daten direkt auf der Klient-Seite auszuführen. Diese so genannten *„Rich Client Applications“* sind ebenfalls ein Merkmal der Internet-Anwendungen der zweiten Generation.

AJAX[12]

Dieses Web-Applikationsmodell besteht aus Technologien zur asynchronen Übertragung und Präsentation von Informationen. Damit können kleine Datenmengen zwischen Server und Browser ausgetauscht werden, ohne die gesamte Seite neu zu übertragen und zu interpretieren. Derartige Applikationen verhalten sich nahezu wie lokal auf dem Klient-Rechner installierte Anwendungen.

---

[10] Framework zur Wiederverwendung von Daten über Anwendungs- und Unternehmensgrenzen hinaus [SwW]
[11] XML-basierte Sprache zur Anreicherung von Daten mit Metadaten [RDF04]
[12] Akronym für Asynchronous JavaScript and XML [BB06]

Screen-Scraping

Screen-Scraping [HC03] ist der Prozess, mit Hilfe von Software-Werkzeugen vorhandene Darstellungen von Daten zu analysieren und daraus Informationen zu extrahieren, die in die Erstellung von Mashups einfließen. Diese Methode der Informationsgewinnung wird oft als wenig „elegant“ angesehen, da sie unter anderem von der Darstellung der Informationen auf der Seite des Daten-Providers abhängt.

1.4 Abgrenzung workflow-basierter Mashup-Werkzeuge

Nach [Hoh07] lassen sich Werkzeuge zur Erstellung von Mashups in drei große Gruppen einteilen. Die erste Gruppe beinhaltet Werkzeuge zur Extraktion von Informationen aus bestehenden Datenquellen, wie zum Beispiel *„dapper“* [Dap] oder *„openkapow“* [kap].
Anwendungen zur workflow-basierten Manipulation von Daten, wie *„Apatar“* [Apa], *„IBM DAMIA“* [IBMa] oder *„Yahoo! Pipes“* *„Pipes“*, stellen die zweite Werkzeuggruppe dar.
Die dritte und letzte Gruppe beinhaltet Anwendungen, wie *„QED-Wiki“* [IBMb] oder den *„Google Mashup Editor“* [GME], die das Zusammensetzen vorhandener Mashup-Komponenten unterstützen.
Des Weiteren existieren Werkzeuge, wie *„SnapLogic“*, die an der Grenze zwischen den ersten beiden Gruppen stehen, oder auch Anwendungen, wie *„Microsoft Popfly“* [MSPa], die an der Grenze zwischen Gruppe zwei und drei stehen.

Werkzeuge der ersten Gruppe zeichnen sich durch ausgereifte Methoden zur Extraktion von Daten aus bestehenden Datenquellen, zum Beispiel HTML-Seiten, aus. Sie bedienen sich der im letzten Abschitt beschriebenen Techniken, wie Screen-Scraping, um XML-Daten, idealerweise RSS- oder ATOM-Feeds, zur Verfügung zu stellen.

Anwendungen der dritten Werkzeuggruppe verfolgen in der Regel eine Art Portalansatz, um die zu itegrierenden Informationen darzustellen. Sie werden auch als Front-End-Werkzeuge bezeichnet und sind häufig als AJAX-Anwendung implementiert.

Die im folgenden genauer betrachtete Gruppe der workflow-basierten Werkzeuge zeichnet sich durch die Modellierung eines Datenflusses aus, der durch verschiedene Operatoren transformiert werden kann.

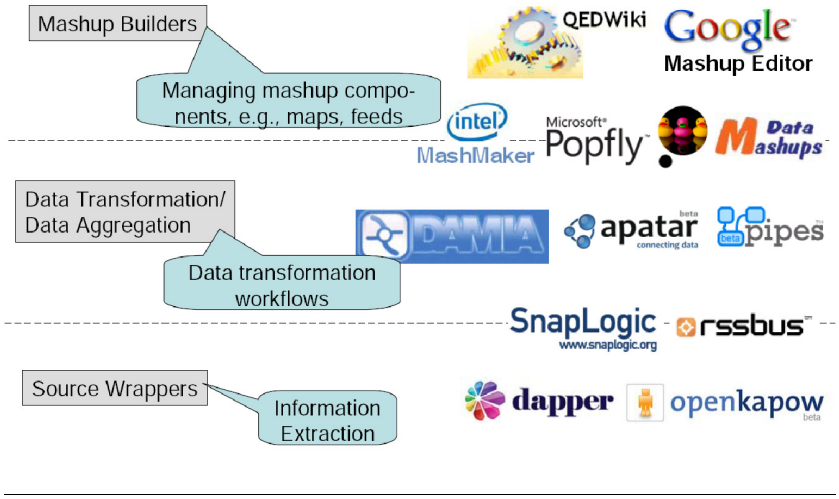

Abbildung 1.1: Übersicht über Mashup-Werkzeuge [RTA05, S. 23]

In Abbildung 1.1 sind alle drei Gruppen von Werkzeugen dargestellt. Dabei ist die erste Gruppe, die Anwendungen zu Datenextraktion, in der untersten Schicht dargestellt. Die mittlere Schicht stellt die workflow-basierten Werkzeuge dar und die oberste Schicht die Front-End-Werkzeuge.

Kapitel 2

Vorstellung der Werkzeuge

In den folgenden Abschnitten werden vier Werkzeuge zur workflow-basieren Integration von Daten verglichen. Dabei handelt es sich um Produkte führender Softwarehersteller und um eine OpenSource-Anwendung. Die zu vergleichenden Anwendungen sind *„Apatar“*, *„Microsoft Popfly“*, *„IBM DAMIA“* und *„Yahoo! Pipes“*. Darüber hinaus existieren, ohne Anspruch auf Vollständigkeit, noch weitere Anwendungen, wie *„Intel MashMaker“*, *„Teqlo“* oder *„MashupMania“*.

Alle vier im folgenden betrachteten Werkzeuge erlauben dem Nutzer die einfache Modellierung eines Datenflusses und danach die Anzeige, beziehungsweise Veröffentlichung der integrierten Daten. Zunächst werden die Werkzeuge allgemein vorgestellt und individuelle Besonderheiten aufgezeigt. In Kapitel 3 werden die Ein- und Ausgabemöglichkeiten sowie die Prozessoren der einzelnen Anwendungen näher erläutert und verglichen. In Kapitel 4 wird ein Beispielszenario vorgestellt, dass mit allen Werkzeugen umgesetzt werden soll.

2.1 Apatar

„Apatar“ [Apa] ist eine in Java geschriebene Anwendung, die als Kommandozeilenprogramm, als Desktopanwendung oder als integrierte Anwendung in einem J2EE-Container zur Ausführung gebracht werden kann. Damit hebt sie sich von den anderen drei Anwendungen ab und erfüllt nur bedingt die in [NV07] beschriebenen Eigenschaften für Mashup-Anwendungen. Sie ist sowohl zur Erstellung von Mashups, als auch zur Umsetzung eines ETL-Prozesses während der Erstellung, beziehungsweise Aktualisierung von

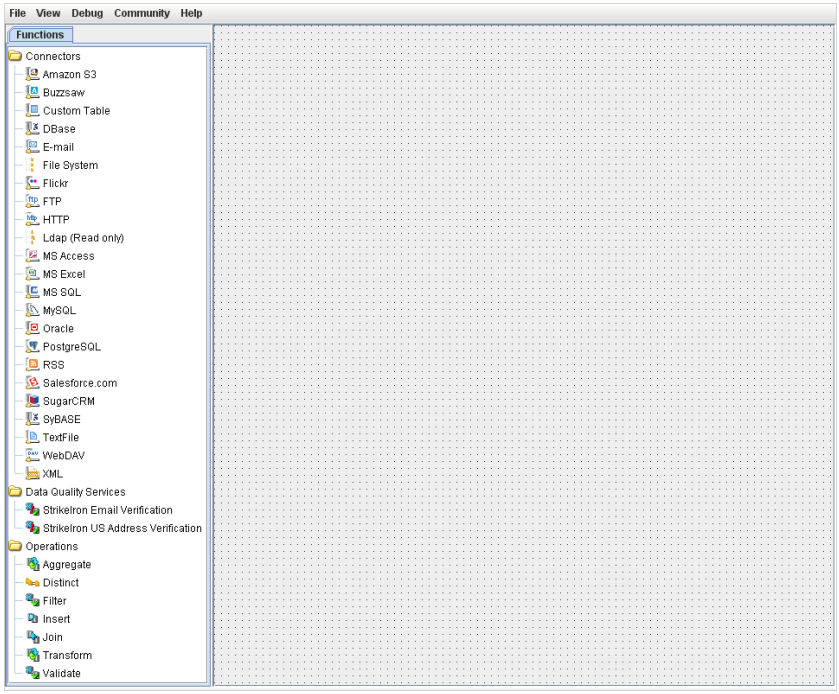

Abbildung 2.1: Anwendungsbereiche in „Apatar“

Datawarehouses(1) geeignet. Daher besitzt *„Apatar“* zahlreiche Konnektoren für relationale Datenbanken oder CRM-Anwendungen(2).

Die Anwendung gliedert sich, wie in Abbildung 2.1 dargestellt, in zwei Bereiche: rechts der Zeichenbereich und links der Bereich der Konnektoren und Operatoren.
Objekte des linken Anwendungsbereiches können mit der Maus per „Drag and Drop“ im Zeichenbereich angeordnet und untereinander verbunden werden. Dabei existieren keine Einschränkungen, so dass beliebige Operatoren untereinander verbunden werden können. Außerdem müssen in *„Apatar“* die Abbildungen der Eingangs- auf die Ausgangsschemata, analog zu Abbildung 2.2, explizit angegeben werden. Im modellierten Datenfluss können an jeder Stelle persistente Zustände von Daten, zum Beispiel in Form von Tabellen in relationalen Datenbanken, enthalten sein und als Ausgangsdaten für weitere Verarbeitungen dienen.

(1) „Ein Data Warehouse ist eine physische Datenbank, die eine integrierte Sicht auf beliebige Daten zu Analysezwecken ermöglicht.“ [BG04]

(2) „Kundenbeziehungsmanagement oder Kundenpflege (engl. Customer Relationship Management, CRM) bezeichnet die Dokumentation und Verwaltung von Kundenbeziehungen . . . “ [ME+03]

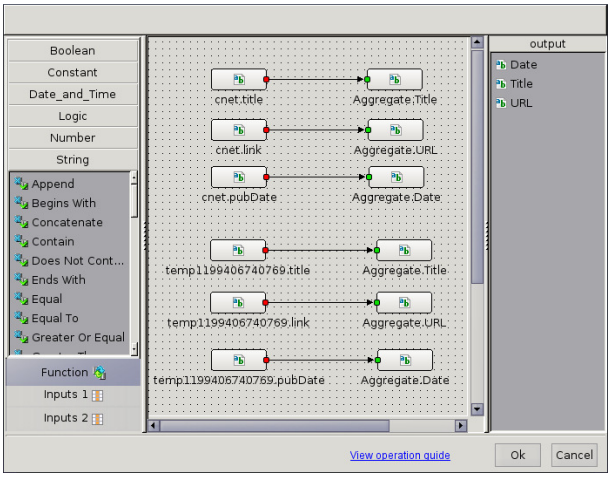

Abbildung 2.2: Schemaabbildung

2.2 Microsoft Popfly

„Popfly“ [MSPa] ist der Mashup-Editor und Mashup-Maker von Microsoft. Als Programmierumgebung wird ein Browser mit einer, auf *„Silverlight*[(3)]*“* basierenden, grafischen Oberfläche verwendet. Programmierkenntnisse sind zur Verwendung der Umgebung weitgehend nicht erforderlich.

Datenquellen, Operatoren und Datensenken werden in so genannten Blöcken erfasst, die sehr einfach mit der Maus per „Drag and Drop“ in der Oberfläche angeordnet werden können. Als Datenquellen können bestehende Web-Inhalte genutzt werden, für die der jeweilige Block mehrere Funktionen bereitstellt. Pro Block kann allerdings nur eine Funktion für die Datenquelle bestimmt werden. Die Datensenke richtet sich nach der gewählten Funktion. Während ein Block die empfangenen Web-Daten transformiert, kann ein anderer Block die Daten visuell als Newsreader, Video- und Musikplayer oder Photoalbum darstellen. Die Art der Funktionalität wird in *„Popfly“* nicht unterschieden.

Sollen Blöcke kombiniert werden, müssen die Eingabedaten des Folgeblocks mit den Ausgabedaten des Vorgängers kompatibel sein, was sich aber durch die Verschiedenheit der Datenquellen und implementierten Funktionen als schwierig erweist. Da die Anzahl an Blöcken unüberschaubar wirkt, bietet *„Popfly“* eine Entscheidungshilfe an. Datenquelle, Operator und Datensenke sind fest in einem Block eingebunden. Das heißt, alle Funktionen sind speziell für eine konkrete Datenquelle ausgelegt.

(3) Silverlight bezeichnet eine Web-Präsentationstechnik von Microsoft [MSS].

Abbildung 2.3: Anwendungsbereiche in „Microsoft Popfly“

Die Funktionen eines Blocks stehen fest und können nicht durch weitere Operatoren verändert werden. Microsoft stellt bereits verschiedene vorgefertigte Blocks bereit. Zur Erstellung nutzerdefinierter Blöcke sind Kenntnisse in JavaScript und XML nötig. Als Programmierumgebung stehen der *„PopflyExplorer*[4]*“* und das *„Popfly BlockSDK*[5]*“* zur Verfügung. Die Datensenke dient der visuellen Darstellung in einer Internetseite, dem *„MySpace-Blog*[6]*“* oder als *„Sitebar Gadget*[7]*“* in *„Windows Vista“*. Die erneute Veröffentlichung der Ergebnisse ist nicht möglich.

2.3 IBM DAMIA[8]

„DAMIA“ [IBMa] verwendet eine AJAX-basierte grafische Oberfläche für die der Browser Mozilla Firefox ab der Version 1.5 [Moz] erforderlich ist. Eine installierbare Version ist ebenfalls erhältlich, dazu wird jedoch ein Apache HTTP-Server benötigt. Diese Version ist auf der Internetseite verfügbar und enthält den Mashupmaker *„QEDWiki“* [IBMb] zur grafischen Darstellung der erzeugten Mashups.

(4) Plug-In für *„Visual Studio Web Developer“* [MSPb]
(5) Download unter `http://go.microsoft.com/fwlink/?LinkId=102098`
(6) MySpace ist eine werbefinanzierte Internetseite, die es den Nutzern erlaubt Benutzerprofile mit Fotos, Videos und Blogs einzurichten. [MyS]
(7) Als Sidebar Gadget wird ein kleines Programm bezeichnet, dass in der Sidebar von Windows Vista auf dem Desktop angezeigt werden kann. [Gad]
(8) Akronym für „Data Mashup for Intranet Applications“

Die zu verarbeitenden Quelldaten sind eigene Dateien, die auf den Server geladen werden, oder Daten die im Internet verfügbar sind. Mit Hilfe der zur Verfügung stehenden Prozessoren werden sie zusammengefasst, sortiert und schließlich veröffentlicht. Die Modellierung eines Datenstroms erfolgt durch die geordnete Anwendung der Prozessoren auf den Eingangsdatenstrom, beziehungsweise die Eingangsdatenströme.

Intern werden XML-Datenstrukturen zur Verarbeitung der Daten verwendet, das heißt ein Datenstrom besteht aus Elementen mit ihren Attributen die wiederum Werte haben. Die Navigation auf den Elementen erfolgt mit XPath[9], wobei Kenntnisse darüber nicht nötig sind, da hierfür ein Assistent zur Verfügung steht.

Die Ausgabedaten werden von *„DAMIA“* nicht grafisch bereitgestellt. Zur Darstellung sollten andere Anwendungen, wie *„QEDWiki“* oder einfache RSS-Reader benutzt werden. Die Anzeige innerhalb der Oberfläche erfolgt auf einfachste Weise im Preview-Bereich der Anwendung. Außerdem werden die Daten ebenfalls unter der angegebenen Web-Adresse veröffentlicht[10]. Das Mashup kann anderen Nutzern von *„DAMIA“* zur weiteren Bearbeitung oder Verwendung bereitgestellt werden.

Die Anordnung der Elemente in der Oberfläche erfolgt ebenso wie in den bereits vorgestellten Anwendungen per „Drag and Drop“ mit der Maus.

2.4 Yahoo! Pipes

„Yahoo! Pipes“ [Pipa] ist eine reine, auf AJAX basierende, Web-Anwendung, die die Erstellung von Mashups in einem grafischen Editor ermöglicht. Dieser wird derzeit von den Browsern *„Firefox“*, *„Internet Explorer 7“* und *„Safari“* unterstützt. Wie der Name *„Yahoo! Pipes“* schon sagt, ist die Integration der Quelldaten an die Verwendung von Pipes[11] unter UNIX angelehnt. Dabei werden einfache Operatoren benutzt, um aus gegebenen Quelldaten ein gewünschtes Ergebnis zu erzeugen.

Wie in Abbildung 2.5 dargestellt, besteht die Anwendung aus drei Bereichen: der Operatorbibliothek, dem Debugger und dem Zeichenbereich.

[9] „Die Sprache XPath dient zur Adressierung von Teilen eines XML-Dokuments. Sie wurde für die Verwendung sowohl in XSLT, als auch in XPointer entworfen.“ [XPa07]

[10] Es ist in *„DAMIA“* möglich, Mashups für andere Nutzer zu veröffentlichen oder nur für den angemeldeten Benutzer zugänglich zu machen

[11] Anlehnung an das, unter UNIX bekannte, Konzept zur Kommunikation zwischen Prozessen nach dem Produzent-Konsument-Prinzip. [Tan02, S. 55,741f.]

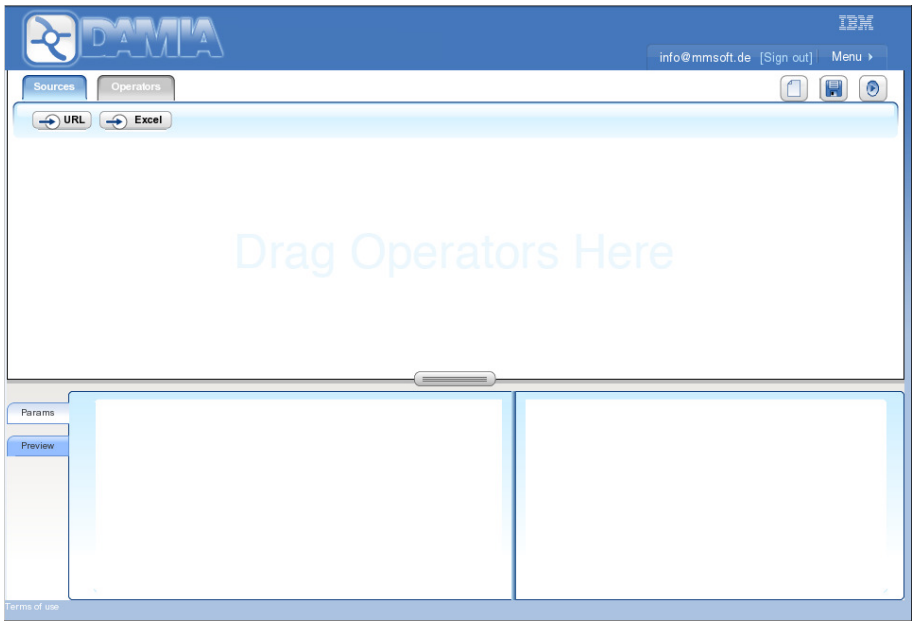

Abbildung 2.4: Anwendungsbereiche in „IBM DAMIA“

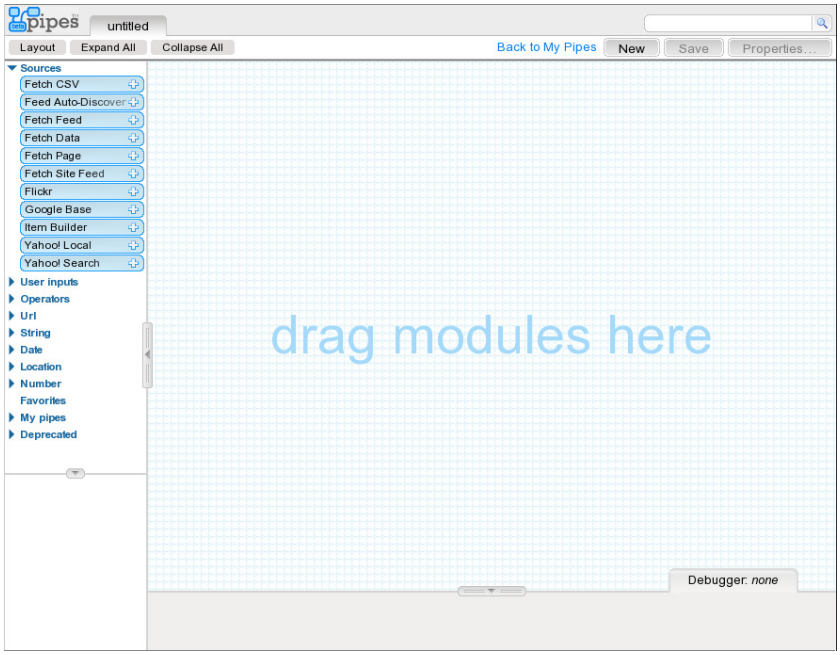

Abbildung 2.5: Anwendungsbereiche in „Yahoo! Pipes“

Objekte der Bibliothek können per *„Drag and Drop“* mit der Maus auf den Zeichenbereich gezogen und auf die gleiche Art untereinander verbunden werden. Wichtig ist hierbei die, später noch genauer diskutierte, Unterscheidung in primitive oder Feed-Prozessoren. Beide Gruppen können jeweils, bis auf wenige Ausnahmen, nur untereinander verbunden werden. Feed-Prozessoren werden durch blau dargestellte Pipes[12] verbunden, primitive Prozessoren durch grau gefärbte. Die Ausgaben von Prozessoren können, sowohl als Eingabedaten für weitere Prozessoren verwendet werden, sowie, im Fall primitiver Prozessoren, auch als Parameter der folgenden Prozessoren.
Der Debugger am unteren Rand des Fensters zeigt immer, zum jeweils gerade bearbeiteten Prozessor, die aktuelle Ausgabe an.

2.5 Übersicht über die Anwendungen

In Tabelle 2.1 ist eine Übersicht über die vier Anwendungen und die jeweils verwendeten, grundlegenden Technologien dargestellt. *„IBM DAMIA“* und *„Yahoo! Pipes“* verwenden beide eine AJAX-basierte Web-Oberfläche. *„Popfly“* ist ebenfalls eine Web-Anwendung, verwendet jedoch *„Silverlight“* als Laufzeitumgebung. *„Apatar“* unterscheidet sich, als Desktop-Anwendung, deutlich von den anderen Werkzeugen. Zur Ausführung wird eine Java-Laufzeitumgebung benötigt. Die Integration der Daten kann mit *„Apatar“*, im Gegensatz zu den anderen Anwendungen, auch ohne Verbindung zum Internet erfolgen wenn die Daten lokal verfügbar sind.

Anwendung	Anwendungsart	Technologie	Hersteller	Datenformat
„Apatar“	Desktop	Java	Apatar	Tabelle
„Microsoft Popfly“	Web	Silverlight	Microsoft	unbekannt
„IBM DAMIA“	Web	AJAX	IBM	XML
„Yahoo! Pipes“	Web	AJAX	Yahoo!	RSS

Tabelle 2.1: Übersicht über die zu vergleichenden Anwendungen

[12] Vgl. Fußnote 11 auf Seite 11

Kapitel 3

Operatoren

3.1 Einordnung von Operatoren

Wie auch in den Veröffentlichungen von [WH07] und [AB+07] sollen alle Operatoren, die die vier Beispielanwendungen bieten, zur besseren Vergleichbarkeit in die Gruppen *„Quellen“*, *„Prozessoren“* und *„Senken“* eingeordnet werden. Des Weiteren kann die Gruppe der *„Prozessoren“* weiter in *„primitive Prozessoren“*, *„Feed-Prozessoren“* und *„höherwertige Prozessoren“* unterteilt werden.

3.2 Ein- und Ausgabe

3.2.1 Daten-Quellen

Dateien

Sowohl *„Apatar“*, als auch *„Yahoo! Pipes“* unterstützen Datenquellen aus Textdateien mit Trennzeichen. *„Apatar“* kann außerdem Daten in diesem Dateiformat speichern. Weiterhin ist es mit *„Apatar“* oder *„IBM DAMIA“* möglich, reine XML-Dateien und Excel-Sheets zu verarbeiten. Im Moment existiert für *„DAMIA“* noch keine Schnittstelle, um Textdateien mit Trennzeichen zu verarbeiten; sie befindet sich aber in der Entwicklung. *„Microsoft Popfly“* ist nicht in der Lage, mit Daten aus Dateien umzugehen.

Feeds

Alle Werkzeuge unterstützen verschiedene Arten von Feeds, wie RSS, ATOM und XML. In *„Apatar“* ist es außerdem möglich, mit Hilfe des Konnektors für XML-Daten, einzelne Felder eines Feeds zu extrahieren und zu verarbeiten. Eine ähnliche Funktion steht ebenfalls für *„IBM DAMIA“* zur Verfügung, indem die Datenquelle mit dem Prozessor *„Transform“* kombiniert wird. *„Popfly“* kann Datenströme durch den Einsatz unterschiedlicher Blöcke aufteilen.

Anwendungs-APIs

„Apatar“ besitzt eine größere Vielfalt von Schnittstellen im Vergleich zu anderen Anwendungen. Zunächst werden Konnektoren für HTTP, FTP, LDAP und WebDAV angeboten, um Dateien zu lesen, beziehungsweise zu schreiben. Es werden ebenfalls Schnittstellen zu zahlreichen relationalen Datenbanken, wie zum Beispiel *„Oracle“*, *„SyBASE“* oder *„PostgreSQL“* angeboten. Außerdem sind Konnektoren zu *„Amazon“*, *„Autodesk Buzzsaw“*, *„Flickr“*, *„Salesforce“* und *„SugarCRM“* vorhanden. Die Konnektoren bieten zahlreiche Parameter und damit im Vergleich zu den anderen Anwendungen eine deutlich höhere Flexibilität. Mit der kostenpflichtigen Version von *„Apatar“* werden noch weitere Konnektoren für *„SAP“*, *„Siebel“* oder *„StrikeIron“* zur Verfügung gestellt.

„Popfly“ besitzt einen Satz an vorgefertigten Schnittstellen zu diversen Onlinediensten, wie *„flickr“*, *„Picture2Life“*, *„Facebook“*, *„Virtual Earth“*, *„MSN Shopping“* oder *„Live Spaces“*. Direkte Schnittstellen zu Datenbanken sind nicht verfügbar. Dem Nutzer wird jedoch die Möglichkeit geboten, eigene Schnittstellen zu implementieren, solange diese online verfügbar sind. Auch Dienste, die eine Authentifizierung benötigen, können genutzt werden.

„IBM DAMIA“ besitzt im Moment keine derartigen Schnittstellen. Eine Anbindung an externe Datenbanken ist jedoch momentan in der Entwicklung. Die Nutzung verschiedener Onlinedienste ist zur Zeit nur möglich, wenn der Provider einen RSS-Service anbietet.

„Yahoo! Pipes“ besitzt zahlreiche Schnittstellen zu anderen Anwendungen, wie zum Beispiel *„flickr“*, *„Google Base“* oder *„Yahoo! Search“*. Diese Schnittstellen sind, verglichen mit *„Apatar“*, sehr einfach zu bedienen, bieten allerdings auch wenig Flexibilität.

Nutzerdefinierte Datenquellen

Sowohl *„Apatar“*, als auch *„Yahoo! Pipes“* unterstützen Datenquellen, die der Nutzer in Form von Attribut-Wert-Paaren frei definieren kann. In *„Pipes“* steht diese Funktion über das Modul *„Item Builder“* zu Verfügung, in *„Apatar“* über den Konnektor *„Custom Table“*. In *„IBM DAMIA“* ist ein vergleichbares Konzept nicht verfügbar. *„Popfly“* bietet nur die im letzten Abschnitt erwähnten nutzerdefinierten Schnittstellen an.

Zusammenfassung

In Tabelle 3.1 ist gegenübergestellt, welche Anwendungen welche Datenquellen unterstützen. Als kleinster gemeinsamer Nenner sind dabei die Feed-Formate zu erkennen, die von allen Anwendungen unterstützt werden. *„Apatar“* bietet die größte Auswahl an Datenquellen, wohingegen *„DAMIA“* die wenigsten Eingabemöglichkeiten für Daten unterstützt. Gerade für *„DAMIA“* wären Konnektoren für Datenbanken, CRM-Systeme oder SAP sinnvoll, da dieses Werkzeug auf die Integration von Unternehmensdaten abzielt. *„Yahoo! Pipes“* unterstützt als einzige Anwendung die Abfrage von Suchmaschinen, wie *„Yahoo! Search“* oder *„Google Base“*. *„Microsoft Popfly“* besitzt ebenfalls einige Konnektoren zu verschiedenen, im Internet verfügbaren, Diensten, unterstützt aber keine Verarbeitung von Dateien der Anwender.

3.2.2 Datensenken

In *„Apatar“* können nahezu alle Konnektoren als Datenquelle und Datensenke benutzt werden. Eine Ausnahme ist hier der LDAP-Konnektor, der nur als Quelle verwendbar ist.

Die Ausgabe der Daten ist in *„Popfly“* nur visuell möglich. Dazu existieren zahlreiche Darstellungsblöcke für Texte, Videos, Landkarten, Musik oder Fotos. Diese Ausgaben können allerdings nur in dafür erstellten Internetseite, *„MySpace-Blogs*[(1)]*“* oder *„Vista Sidebar Gadgets*[(2)]*“* eingebunden werden. Eine weitere Verwendung der aufbereiteten Daten ist nicht möglich.

(1) Vgl. Fußnote 6 auf Seite 10
(2) Vgl. Fußnote 7 auf Seite 10

Datenquellen	*„Apatar“*	*„Microsoft Popfly“*	*„IBM DAMIA“*	*„Yahoo! Pipes“*
RSS, ATOM, XML	√	√	√	√
XLS oder CSV	√	–	√	√
HTTP, FTP, WebDAV	√	–	–	–
E-Mail	√	–	–	–
LDAP	√	–	–	–
„flickr.com“	√	√	(√)	√
„Picture2Life“	–	√	–	–
„Live Spaces“	–	√	–	–
„Virtual Earth“	–	√	–	–
„Facebook“	–	√	–	–
Rel. Datenbanken	√	–	–	–
„SugarCRM“	√	–	–	–
„salesforce.com“	√	–	–	–
„Autodesk Buzzsaw“	√	–	–	–
„Amazon S3“	√	–	–	–
„SAP“	√	–	–	–
„Siebel“	√	–	–	–
„StrikeIron“	√	–	–	–
„Google Base“	–	–	–	√
„MSN Shopping“	–	√	–	–
„Yahoo!“ Suchmaschinen	–	–	–	√
Nutzereingaben	√	–	–	√
Nutzerdef. Tabellen	√	–	–	√

Tabelle 3.1: Vergleich der Datenquellen

Die Datenausgabe in *„IBM DAMIA“* erfolgt durch den Prozessor *„Publish“*, in dessen Parametern auch das genaue Format festgelegt wird. Der resultierende Feed wird auf einem IBM-Server abgelegt, um ihn anderen Nutzern zur Verfügung zu stellen zu können. Eine grafische Darstellung oder eine Anzeige als HTML-Seite wird nicht unterstützt.

„Yahoo! Pipes“ bietet nur eine mögliche Datensenke: den *„Pipe Output“*. Die Ausgabe kann als HTML-Seite oder auch als RSS-Feed erfolgen. Ist der Feed ein GeoRSS-Feed, können die Daten ebenfalls in *„Yahoo! Maps“* annotiert werden.

Zusammenfassung

Tabelle 3.2 zeigt, welche Anwendungen welche Datensenken unterstützen. *„Apatar“* ist, wie bereits bei der Betrachtung der Datenquellen, die flexibelste Anwendung. Sie erlaubt nahezu alle Konnektoren sowohl als Quelle und Senke für den Datenfluss zu verwenden. *„Popfly“* bietet keine Möglichkeit die Daten als Feed oder Datei auszugeben. Es sind nur visuelle Darstellungen möglich. Damit können die Daten nach der Vararbeitung in keiner weiteren Anwendung verwendet werden. Alle anderen Anwendung unterstützen die Ausgabe von Feeds und können somit gemeinsam genutzt werden, um eine Aufgabe zu lösen, indem die Ausgabe eines Programms als Eingabe für ein weiteres dient.

Anwendung	RSS, ATOM, XML	XLS oder CSV	HTTP, FTP, WebDAV	E-Mail	*„flickr.com“*	Webseite	*„MySpace-Blog“*	*„Vista Sidebar“*	Rel. Datenbanken	CRM-Systeme	*„Buzzsaw“*	*„Amazon“*
„Apatar“	√	√	√	√	√	–	–	–	√	√	√	√
„Microsoft Popfly“	–	–	–	–	–	√	√	√	–	–	–	–
„IBM DAMIA“	√	–	–	–	–	–	–	–	–	–	–	–
„Yahoo! Pipes“	√	–	–	–	–	√	–	–	–	–	–	–

Tabelle 3.2: Vergleich der Datensenken

3.3 Prozessoren

3.3.1 Primitive Prozessoren

Als primitive Prozessoren werden im Rahmen dieser Arbeit alle die Prozessoren betrachtet, die sich auf einzelne Datensätze beziehen.

„Apatar“ besitzt fünf Gruppen primitiver Prozessoren. Die Gruppe *„Boolean“* beinhaltet die üblichen Funktionen, wie *„AND“*, *„OR“* und *„NOT“*, aber auch *„Is Null“* und *„Is Not Null“* sowie die zwei Prozessoren *„To Boolean“*, um Strings in Wahrheitswerte umzuwandeln, und *„To String“* für die Gegenrichtung.

Die Gruppe der Konstanten enthält keine eigentlichen Prozessoren, sondern dient dazu, dem Ersteller des Mashups die Möglichkeit zu geben, Konstanten verschiedenen Typs zu definieren, die dann in späteren Berechnungen verwendet werden können. Die möglichen Konstantentypen sind *„Boolean“*, *„Date“*, *„Decimal“*, *„File“*, *„Numeric“*, *„Text“* und *„Time“*.

Die dritte Gruppe bilden die Datums- und Zeitprozessoren. Damit ist es möglich zu überprüfen ob zwei Zeitpunkte identisch sind, oder ein Zeitpunkt größer, beziehungsweise kleiner als ein anderer ist. Außerdem existieren Prozessoren, um die Datentypen zwischen String, Date und Time zu konvertieren.

Die Gruppe der nummerischen Prozessoren beinhaltet, analog zur Gruppe der Datums- und Zeitprozessoren, Prozessoren zur Erkennung, ob zwei Zahlen identisch sind oder eine größer, beziehungsweise kleiner ist. Es existieren ebenfalls Prozessoren zur Konvertierung der Datentypen zwischen *„String“*, *„Integer“*, *„Single“* und *„Double“*.

Die letzte Gruppe beinhaltet die String-Prozessoren. Es existieren ebenfalls wieder die Prozessoren für Gleichheit, *„größer als“*, *„kleiner als“* und die Konvertierung der Datentypen. Weiterhin können Zeichenketten in Groß- oder Kleinbuchstaben (*„Upper Case“* und *„Lower Case“*) umgewandelt werden und es gibt die Möglichkeit in einer Zeichenkette alle Satzanfänge oder alle Wortanfänge in Großbuchstaben zu konvertieren. Außerdem sind Prozessoren zum Einfügen, Suchen, Extrahieren, Ersetzen und Löschen von Teilstrings sowie zum Eliminieren von initialen oder finalen Leerzeichen verfügbar. Es besteht ebenfalls die Möglichkeit, Zeichenketten zusammenzufügen oder zu überprüfen, ob eine Zeichenkette leer ist. Weiterhin existieren noch fortgeschrittenere String-Prozessoren zur Überprüfung, ob die Zeichenkette eine gültige URL, E-Mail-Adresse, IP-Adresse oder Kreditkartennummer enthält.

„Popfly“ besitzt nur wenige primitive Prozessoren. Der Block *„Calculator“* stellt einfache mathematische Funktionen, wie Addition, Subtraktion, Multiplikation, Division, Modulo-Division und Potenzierung zur Verfügung.

Der Block *„RegExp“* unterstützt den Anwender bei der Untersuchung von Zeichenketten auf die Erfüllung regulärer Ausdrücke. Dabei kann eine Liste von Ausdrücken übergeben werden, die auf Übereinstimmung geprüft werden. Die Rückgabe dieser Funktion sind alle Übereinstimmungen aller Ausdrücke. Außerdem stellt dieser Block noch zwei weitere Funktionen zu Verfügung. Es kann ein Zeichenkette darauf hin überprüft werden, ob sie einen gegebenen Ausdruck erfüllt oder ob es sich um eine Zahl handelt. Beide Möglichkeiten liefern einen Wahrheitswert zurück.

Mit dem Block *„User Input“* wird ein Textfeld zur Eingabe bereitgestellt, das zur Laufzeit befüllt werden kann. Mit der Verwendung dieser Eingaben wird dem Endbenutzer die Möglichkeit gegeben, das Mashup zu parametrisieren.

„IBM DAMIA“ unterstützt keine primitiven Prozessoren, da die Datenquellen ausschliesslich Feeds beinhalten und sich alle Prozessoren auf den gesamten Feed beziehen.

„Yahoo! Pipes“ stellt eine ganze Reihe verschiedener primitiver Prozessoren zur Verfügung, deren Dokumentation unter [Pipb] abrufbar ist. Darunter befinden sich Eingabefelder, URL-Generatoren, verschiedene Zeichenkatten- und Datumsfunktionen, einfache mathematische Berechnungen und der so genannte *„Location Builder“*.

Die Eingabefelder dienen dazu, dem Benutzer des Mashups die Möglichkeit zu geben, Einfluss auf die Bearbeitung der Daten zu nehmen. Zum Beispiel könnte mit einer solchen Eingabe die Anzahl der im Ausgabe-Feed enthaltenen Datensätze vom Benutzer selbst festgelegt werden.

Mit Hilfe des URL-Generators ist es möglich, URLs aus einzelnen Bestandteilen zusammenzusetzen. Die einzelnen Bestandteile können entweder per Hand eingegeben werden oder das Ergebnis anderer primitiver Operatoren sein.

Die Prozessoren zur Bearbeitung von Zeichenketten ermöglichen dem Ersteller des Mashups, Zeichenketten, mit Perl-ähnlichen[(3)] regulären Ausdrücken, zu verändern, oder Teil-Strings zu extrahieren, beziehungsweise zu ersetzen. Außerdem bietet *„Pipes“* die Möglichkeit,

[(3)] Pipes unterstützt die von der Bibliothek PCRE [Haz] unterstützten regulären Ausdrücke.

Zeichenketten in einzelne Token aufzubrechen, Schlagwörter zu extrahieren, oder in andere Sprachen zu übersetzen. Eine weitere sehr interessante Funktion stellen die privaten Zeichenketten dar. Sie werden ebenfalls vom Ersteller des Mashups angegeben, jedoch beim Kopieren des Mashups gelöscht und sind damit für alle anderen Benutzer nicht ersichtlich. Damit eignen Sie sich zur Angabe von Passwörtern für benutzte Dienste.

Die beiden Prozessoren zur Bearbeitung von Datumsinformationen ermöglichen, sowohl das Erstellen eines Datums aus der Ausgabe eines anderen Operators, als auch das Formatieren eines entsprechenden Datums. Die Parameter zur Angabe des Formats entsprechen den Parametern der PHP-Funktion „`strftime`[(4)]“.

Der Prozessor zur Berechnung einfacher mathematischer Ausdrücke enspricht dem Block *„Calculator“* in *„Microsoft Popfly“* und stellt die gleichen Funktionen zur Verfügung. Als Eingabe können entweder die Ausgaben zweier anderer primitiver Prozessoren dienen oder einer der beiden Parameter kann vom Ersteller des Mashups fest vorgegeben werden.

Zusammenfassung

Wie in Tabelle 3.3 zu sehen ist, bietet *„Yahoo! Pipes“* neben *„Apatar“* das vollständigste Portfolio primitiver Operatoren an. *„DAMIA“* hingegen unterstützt keinen einzigen Prozessor in dieser Kategorie.

Anwendung	Bool. Prozessoren	Konstanten	Datumsprozessoren	Zeitprozessoren	Nummerische Proz.	String-Prozessoren	Reguläre Ausdrücke	URL-Generator
„Apatar“	√	√	√	√	√	√	√	–
„Microsoft Popfly“	–	√	–	–	√	–	√	–
„IBM DAMIA“	–	–	–	–	–	–	–	–
„Yahoo! Pipes“	–	√	√	√	√	√	√	√

Tabelle 3.3: Vergleich primitiver Prozessoren

(4) Beschreibung der Funktion „`strftime`“ unter `http://de.php.net/strftime`

3.3.2 Feed-Prozessoren

Im Gegensatz zu primitiven Prozessoren bezieht sich diese Art von Prozessoren immer auf eine Menge von Datensätzen. Die Ein- oder Ausgabe eines solchen Prozessors ist eine Liste von Werten.

„Apatar“ besitzt keine eigentlichen Feed-Prozessoren. Die, unter dem Begriff *„Operations“* gruppierten, Prozessoren in *„Apatar“* entsprechen am ehesten den Feed-Prozessoren der anderen Anwendungen.

Mit Hilfe des Prozessors *„Distinct“* kann die Eliminierung von Duplikaten erfolgen. Dabei kann angegeben werden, welche Attribute, beziehungsweise welche Gruppe von Attributen zur Erkennung von Duplikaten verwendet werden.
Außerdem steht der Prozessor *„Insert“* zur Verfügung, der eine nutzerdefinierte Tabelle zur Verfügung stellt, die in die weiteren Berechnungen mit einbezogen werden kann.

Der Prozessor *„Aggregate“* dient der Vereinigung der Daten zweier Datenquellen. Im Gegensatz zu den anderen Anwendungen muss dazu die Abbildung der Quellschemata auf das Zielschema mit Hilfe primitiver Prozessoren explizit angegeben werden. Analog dazu arbeitet der Prozessor *„Join“*, der die, aus dem SQL-Standart bekannte, Funktionalität bietet.

Weiterhin existieren die Prozessoren *„Filter“* und *„Validate“*. Der erste ermöglicht die Überprüfung der Datensätze auf bestimmte Bedingungen. Das Ergebnis enthält nur Datensätze, die die angegebenen Bedingungen erfüllen. Der zweite Prozessor ermöglicht ebenfalls die Prüfung bestimmter Bedingungen, allerdings beinhaltet das Ergebnis in einer Tabelle die Datensätze, die die Bedingungen erfüllen und in einer zweiten Tabelle die Datensätze, die die Bedingungen nicht erfüllen.

Der Prozessor *„Transform“* ermöglicht die Angabe einer Abbildung zwischen dem Schema der Eingabedaten und dem Schema der Ausgabedaten.

„Microsoft Popfly“ bietet nur eine sehr begrenzte Auswahl an Feed-Prozessoren unter den vordefinierten Blöcken. Es handelt sich hierbei um *„Filter“*, *„Sort“* und *„Combine“*. Der erste Prozessor arbeitet analog zum bereits vorgestellten Prozessor bei *„Apatar“*. Der zweite Block erlaubt die Sotierung der Elemente der Eingabedaten. Der Prozessor *„Combine“* erlaubt das Zusammenführen von maximal drei Listen zu einer neuen Liste. Die Kombination von

Objekten, zum Beispiel Elementen eines RSS-Feeds, ist nicht möglich. Ausschließlich das Zusammenführen einzelner Parameter der Objekte zu einer neuen Liste ist möglich. Somit handelt es sich um das Kombinieren von Listen, nicht von Objekten.

„IBM DAMIA“ unterstützt generell nur Prozessoren zur Bearbeitung ganzer Feeds. Der Prozessor *„Augment“* kombiniert zwei Datenströme so, dass den Elementen des einen Datenstroms ein Attribut des anderen Datenstroms hinzugefügt wird. Ein Feed kann also mit Daten eines weiteren Feeds angereichert werden.

Die Prozessoren *„Merge“* und *„Union“* kombinieren jeweils zwei Datenströme. Dabei bildet *„Union“* die Vereinigung und *„Merge“* führt einen, mit dem Prozessor in Apatar vergleichbaren, Join mit einem Join-Attribut und einer Join-Operation aus.

Mit Hilfe des Prozessors *„Group“* können Elemente eines Feeds, die in einem anzugebenden Attribut übereinstimmen, als Sub-Elemente zu einem Element zusammengefasst werden.

Weiterhin wird der Prozessor *„Sort“* angeboten, mit dem die Elemente eines Eingabedatenstroms aufsteigend, beziehungsweise absteigent sortiert werden können.

Außerdem bietet *„DAMIA“* den Prozessor *„Filter“* an, der die Elemente des Eingabedatenstroms, mit Hilfe verschiedener Vergleichsoperationen, mit einem Ausdruck vergleicht. Wird die Bedingung für ein Element als wahr ausgewertet, wird dieses in den Ausgabedatenstrom übernommen.

Der Prozessor *„Unique“* ermöglicht, ähnlich wie *„Distinct“* in *„Apatar“*, die Erkennung und Eliminierung von Duplikaten anhand eines anzugebenden Attributs. Die Erkennung dieser Duplikate erfolgt ebenfalls nur bei exakter Übereinstimmung des Attributwerts bei zwei oder mehr Elementen im Feed.

Mit Hilfe des Prozessors *„Transform“* können einem Element des Datenstroms neue Subelemente, beziehungsweise Attribute hinzugefügt werden, oder deren Werte manipuliert werden.

Der Prozessor *„Publish“* stellt den Datenstrom als RSS-, Atom- oder XML-Feed öffentlich zur Verfügung.

„Yahoo! Pipes“ bietet zahlreiche Prozessoren in dieser Kategorie [Pipb]. Der Prozessor *„Count“* bestimmt die Anzahl der Datensätze im Feed. Diese kann als Eingabe für einen primitiven Prozessor verwendet werden, jedoch nicht für andere Feed-Prozessoren.

Weiterhin wird der Prozessor *„Filter“* angeboten, mit dem Datensätze innerhalb eines Feeds ausgewählt werden können, die für die Weiterverarbeitung zur Verfügung stehen sollen. Es ist möglich, „positive“ oder „negative“ Regeln zu definieren. Positive Regeln erlauben Datensätze, die der Regel entsprechen, negative Regeln blockieren Datensätze, die der Regel entsprechen. Die Anzahl der Regeln ist nicht beschränkt.

Mit Hilfe des Prozessors *„Loop“* können primitive Prozessoren auf alle Elemente eines Feeds angewendet werden. Dieser Prozessor kann als For-Schleife, wie in herkömmlichen Programmiersprachen, aufgefasst werden. Die Iteration erfolgt immer über alle Elemente des Feeds. Es ist mit diesem Prozessor nicht möglich, zum Beispiel nur über die ersten fünf Elemente zu iterieren. Dazu müsste der Feed zunächst auf fünf Elemente gekürzt werden. Im Schleifenkörper des Prozessors dürfen alle primitiven Prozessoren außer den Eingabefeldern enthalten sein. Außerdem kann noch ausgewählt werden, ob die Ergebnisse der primitiven Prozessoren aus dem Schleifenkörper als neue Elemente zum Eingabe-Feed hinzugefügt werden sollen, oder ob sie in einem Attribut eines Feed-Datensatzes gespeichert werden sollen.

Der Prozessor *„Regex“* erlaubt die Anwendung von Perl-ähnlichen[(5)] regulären Ausdrücken auf Attribute der Feed-Elemente. Innerhalb dieses Prozessors können beliebig viele Regeln definiert werden, die sich jeweils auch auf unterschiedliche Attribute beziehen können.

Außerdem werden die Prozessoren *„Rename“*,*„Reverse“* und *„Sort“* angeboten. Damit ist es möglich, Attribute in allen Feed-Elementen umzubenennen, beziehungsweise zu kopieren, oder Feed-Elemente umzusortieren. *„Reverse“* kehrt die Sortierung um, wohingegen *„Sort“* beliebige Sortierungen nach beliebigen Feed-Attributen erlaubt.

Mit Hilfe des Prozessors *„Split“* kann ein kompletter Feed kopiert werden. Die Ausgabe sind zwei identische Kopien des Eingabe-Feed.

Der Prozessor *„Sub-Element“* extrahiert ein Attribut aus den Datensätzen eines Feed und erstellt einen neuen Feed, dessen Elemente nur dieses eine Attribut besitzen.

Weiterhin werden die Prozessoren *„Tail“* und *„Truncate“* angeboten. *„Tail“* gibt die letzten n Elemente eines Feeds aus, der komplementäre Prozessor *„Truncate“* die ersten n Elemente.

(5) Vgl. Fußnote 3 auf Seite 20

Mit Hilfe des Prozessors *„Union“* können mehrere Datenquellen verbunden werden. Es ist damit möglich, aus mehreren Datenströmen einen gemeinsamen Feed zu erstellen.

Der Prozessor *„Unique“* erlaubt die Erkennung und Eliminierung von Duplikaten anhand eines anzugebenden Attributes. Die Erkennung von Duplikaten erfolgt allerdings nur bei exakter Übereinstimmung des Attributwerts bei zwei oder mehr Elementen im Feed.

Außerdem wird der Prozessor *„Web Service“* angeboten mit dem es möglich ist, komplette Feeds im *„JSON-Format“* [Cro06] an einen Webservice zu übergeben. Dafür können entweder bereits existierende Dienste [Str] in Anspruch genommen werden oder Eigene nach den genauen Bedürfnissen erstellt werden. Dazu sind jedoch Programmierkenntnisse erforderlich.

Zusammenfassung

In Tabelle 3.4 ist dargestellt, welche Anwendungen welche Prozessoren unterstützen. *„Apatar“* stellt die Prozessoren, deren Haken in Klammern stehen nicht selbst zur Verfügung, sondern erfordert die Verwendung einer relationalen Datenbank um die gewünschten Funktionen abbilden zu können. Das hat den Nachteil, dass vor der Erstellung des Mashups ein passendes Datenbankschema in der zu verwendenden Datenbank angelegt werden muss. Der Prozessor Webservice kann durch ein eigenes Plugin erstellt werden.

Der Prozessor *„Vereinigung“* wird von *„Popfly“* nur sehr rudimentär implementiert, da ausschließlich das Zusammenführen von Listen möglich ist, deren Sortierung identisch sein muss. Außerdem fehlen wichtige Prozessoren wie Duplikaterkennung, Transformation oder Gruppierung.

„IBM DAMIA“ bietet nahezu alle wichtigen Grundfunktionen, allerdings fehlt die Erkennung von Duplikaten und eine Schnittstelle für nutzerdefinierte Webservices.

Den vollständigsten Satz an Prozessoren bietet *„Yahoo! Pipes“*, es fehlen jedoch auch hier die Prozessoren *„Join“* und *„Gruppieren“*. Die Transformation von Elementen eines Feeds erfolgt in *„Pipes“* nicht mit einem expliziten Prozessor, wie bei *„DAMIA“* oder *„Apatar“*, sondern implizit über die Prozessoren, die auf den Feed angewendet werden.

Anwendung	Vereinigung	Join	Duplikate	Filter	Sortieren	Gruppieren	Transformieren	Webservice	Schleifen	Reg. Ausdruck	Abschneiden	Kopieren	Zählen
„Apatar“	√	√	√	√	(√)	(√)	√	(√)	(√)	–	√	(√)	(√)
„Microsoft Popfly“	(√)	–	–	√	√	–	–	–	–	–	–	–	–
„IBM DAMIA“	√	√	–	√	√	√	√	–	–	–	–	–	–
„Yahoo! Pipes“	√	–	√	√	√	–	(√)	√	√	√	√	√	√

Tabelle 3.4: Vergleich von Feed-Prozessoren

3.3.3 Höherwertige Prozessoren

Prozessoren dieser Kategorie lassen sich ebenfalls einer der beiden anderen Gruppen zuordnen. Hinter ihnen stehen jedoch sehr komplexe Funktionen, weshalb sie hier gesondert betrachtet werden sollen. Darunter fallen zum Beispiel *„Dynamic Entity Resolution“*, die Behandlung *„unscharfer“* Daten oder das Suchen nach Mashups [AB+07, S. 3f].

„Apatar“ implementiert einige Prozessoren in dieser Kategorie an. So existiert die Möglichkeit Kreditkartennummern, E-Mail-Adressen, IP-Adressen und URLs unter Angabe eines Strings auf Gültigkeit zu überprüfen. Außerdem können die Dienste der Webservices *„StrikeIron Adressverifikation“* und *„StrikeIron Email Verification“* auf einen gesamten Datenstrom angewendet werden. Weiterhin ist es möglich, eigene Prozessoren zu erstellen, da der Quelltext von *„Apatar“* öffentlich zugänglich und damit auch erweiterbar ist. Allerdings werden für eine solche Erweiterung Kenntnisse in der Programmiersprache Java benötigt.

„Microsoft Popfly“ implementiert keine Prozessoren dieser Art. Zur Durchführung dieser Operationen muss auf externe Dienste zurückgegriffen oder es müssen eigene Blöcke mit der gewünschten Funktionalität implementiert werden.

„IBM DAMIA“ bietet derzeit ebenfalls keine der in [AB+07, S. 3f] beschriebenen Prozessoren. Es ist aber zu erwarten, dass entsprechende Prozessoren in Zukunft in *„DAMIA“* enthalten sein werden.

Yahoo! Pipes bietet vier Prozessoren in dieser Kategorie [Pipb]. Der *„Location Builder“* ist ein Prozessor, der es ermöglicht, aus einem String, der eine Adresse oder Ortsangabe enthält, geografische Informationen abzuleiten. Er erstellt ein so genanntes Location-Objekt, in dem Informationen wie Land, Stadt, Längen- oder Breitengrad enthalten sind. Der Prozessor erkennt Adressen, Postleitzahlen, Ländernamen und Codes von Flughäfen. Eng damit verbunden ist der *„Location Extractor“*, der versucht, entsprechende Informationen aus einem Datensatz eines Feeds zu extrahieren und dem Datensatz danach ein weiteres Unterelement hinzufügt, in dem die entsprechenden Informationen enthalten sind.

Der dritte Prozessor, der so genannte *„Term Extractor“*, versucht aus einem String wichtige Schlagwörter zu extrahieren. Diese können dann als Liste oder wiederum als String zurückgegeben und weiterverarbeitet werden.

Mit Hilfe des vierten Prozessors, *„Yahoo! Shortcuts“*, können Entitäten aus Texten extrahiert werden. Leider ist in *„Pipes“* der Prozessor *„Join“* nicht implementiert, so dass zum jetzigen Zeitpunkt keine *„Entitiy Resolution“* gemäß [AB+07, S. 3f] möglich ist.

Zusammenfassung

Wie in Tabelle 3.4 zu sehen ist, unterstützen nur *„Apatar“* und *„Yahoo! Pipes“* Prozessoren in dieser Kategorie. In *„Microsoft Popfly“* besteht jedoch die Möglichkeit, eigene Blöcke zu definieren und damit zu Webservices zu verwenden, die zum Beispiel Entitäten auflösen können. *„Apatar“* bietet ebenfalls die Möglichkeit, eigene Konnektoren oder Prozessoren zu erstellen und damit den Funktionsumfang zu erweitern. Die Erweiterung von *„IBM DAMIA“* um eigene Operatoren ist zwar prinzipiell möglich, da die Programmiersprache PHP verwendet wird, jedoch aus lizenzrechlichen Gründen untersagt.

Die Verwendung der Prozessoren in dieser Kategorie ist zwischen *„Apater“* und *„Yahoo! Pipes“* vergleichbar. Beide Anwendung verwenden zur Realisierung dieser Dienste Webservices, die automatisch im Hintergrund aufgerufen werden. Die Erkennung von Entitäten wird derzeit leider von keiner Anwendung unterstützt. Mit *„Pipes“* ist es im Moment nur möglich Entitäten zu extrahieren.

Anwendung	Adressverifikation	E-Mail-Verifikation	IP-Verifikation	URL-Verifikation	Kreditkartenverif.	Schlagwortsuche	Orts-Extraktion	Entity-Extraktion	Entity-Resolution
„Apatar“	√	√	√	√	√	–	–	–	–
„Microsoft Popfly“	–	–	–	–	–	–	–	–	–
„IBM DAMIA“	–	–	–	–	–	–	–	–	–
„Yahoo! Pipes“	√	–	–	–	–	√	√	√	–

Tabelle 3.5: Vergleich höherwertiger Prozessoren

Kapitel 4

Beispielszenario

Im folgenden Kapitel soll mit allen vier Werkzeugen ein einfaches Nachrichten-Mashup erstellt werden. Dazu werden RSS-Feeds von `www.news.com`, `slashdot.com` und von `del.icio.us` kombiniert. Außerdem sollen im resultierenden RSS-Feed doppelte Einträge eliminiert werden, wenn möglich ein zum Thema passendes Bild von *„flickr.com“* hinzugefügt werden und die Elemente sollen nach Datum sortiert vorliegen.

4.1 Apatar

Die Feeds können mit dem Operator *„Aggregate“* kombiniert werden, jedoch ist darauf zu achten, dass die verwendeten Felder der beiden Quell-Feeds identische Namen haben. Ist das nicht der Fall, muss ein Feed zunächst in eine benutzerdefinierte Tabelle überführt werden, bevor die Aggregation durchgeführt werden kann. Das Aggregationsergebnis muss wiederum in einer benutzerdefinierten Tabelle gespeichert werden.

Danach kann mit Hilfe des Operators *„Distinct“* die Eliminierung von Duplikaten vorgenommen werden. Es ist möglich alle Felder in die Duplikateliminierung einzubeziehen, allerdings können Duplikate, ebenso wie bei allen anderen Anwendungen, nur bei exakter Übereinstimmung erkannt werden.

Eine Sortierung der Einträge ist mit *„Apatar“* nur unter Verwendung einer relationalen Datenbank möglich. Dazu werden alle Elemente in einer Tabelle abgelegt und anschließend geordnet wieder ausgelesen. Die Zuordnung eines Bildes zu den Einträgen des Feeds ist nicht möglich gewesen, da der Konnektor zu *„flickr.com“* offensichtlich fehlerhaft ist.

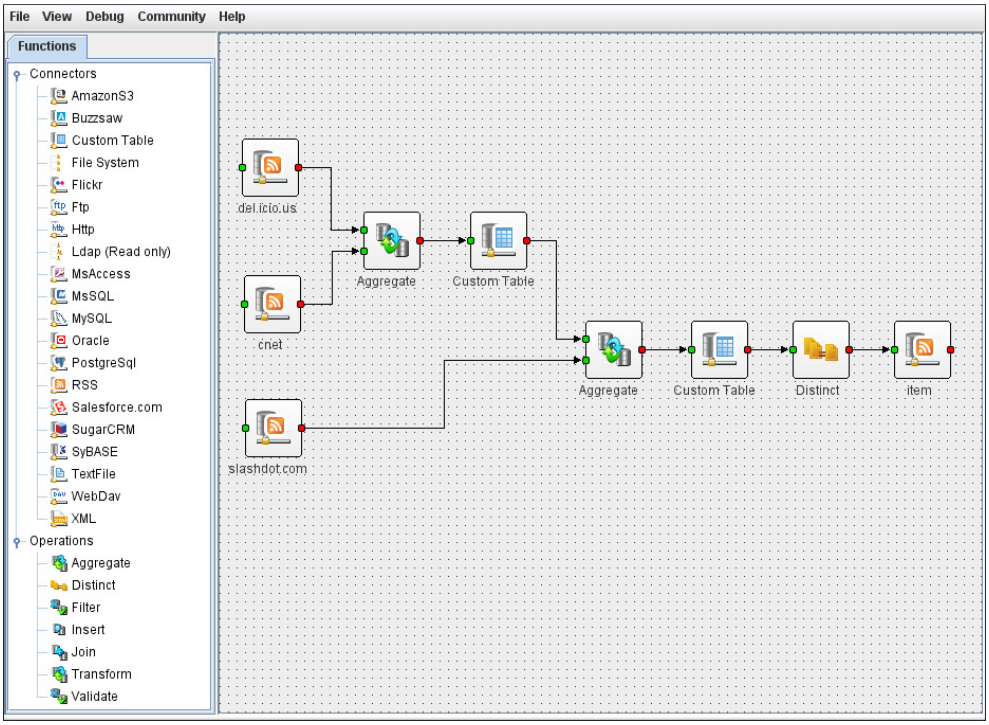

Abbildung 4.1: Beispielszenario in „Apatar“

Die Umsetzung des Szenario erforderte aufgrund des Beta-Stadiums der Software und der manuellen Angabe der Schematransformationen etwa 30 Minuten. In dieser Zeitangabe ist das Anlegen des, für die Sortierung benötigten, Datenbankschemas nicht enthalten. Somit war es nicht möglich das geforderte Szenario komplett umzusetzen.

4.2 Microsoft Popfly

Zwar stehen alle Datenquellen wie RSS-Reader und eine Schnittstelle zur Flickr-Suche bereit, doch war es nicht möglich diese mit den vorhandenen Mitteln zu vereinen. Wie bereits kurz beschrieben, gibt es keine Blöcke, die gleichartige Objekte vereinigen können und somit die drei RSS-Feeds zu einem verbinden könnten.

Ein „Workaround“ wurde damit geschaffen, dass die vier Elemente Titel, Datum, Beschreibung und Link einzeln durch je einen Block des Typs *„Combine“* mit den gleichartigen Elementen der anderen Feeds vereinigt wurden. Damit entstanden vier Listen die zwar die Attribute eines RSS-Feeds hatten, aber keiner waren. Es war allerdings trotzdem möglich, diese vier Datenströme an einen RSS-News-Reader-Element zu knüpfen, da alle Datenströme die selbe Reihenfolge hatten.

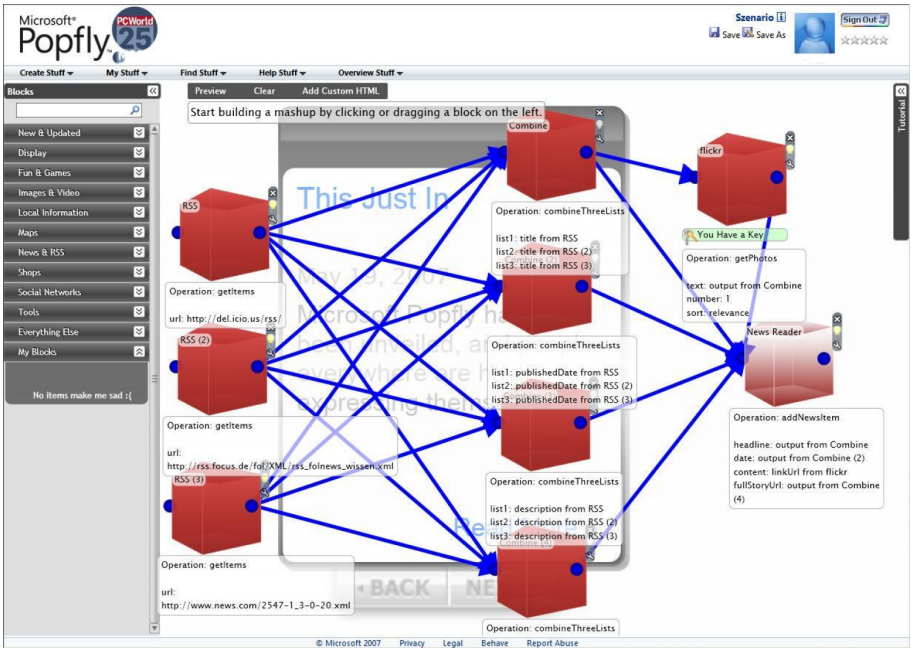

Abbildung 4.2: Beispielszenario in „Microsoft Popfly“

Die prinzipielle Anbindung des Foto-Dienstleisters *„flickr.com“* war problemlos. Der entsprechende Block nutzt die Überschirft der RSS-Feeds als Suchstring und liefert die Beschreibung, die URL des Fotos und die Geodaten zurück. Allerdings waren diese Daten nicht nutzbar, da alle Ausgaben Bildschirmelemente sind. Weil der Block *„Newsreader“* bereits den gesamten Bildschirmplatz eingenommen hatte, konnte kein Block zur Darstellung von Bildern hinzugefügt werden. Als Alternative wurde der Bildlink innerhalb des *„Newsreaders“* angezeigt.

Das Szenario konnte mit *„Popfly“* nur über den Umweg der vier getrennten Listen realisiert werden. Zur Umsetzung wurden etwa 10 Minuten benötigt.

4.3 IBM DAMIA

Die Nachrichten-Feeds können problemlos über die URLs eingebunden werden. Da sie sich aber in den Elementen unterscheiden, werden sie mit dem Prozessor *„Transform“* auf ein einheitliches Schema gebracht, dass die Elemente *„title“*, *„link“*, *„date“* und *„description“* beinhaltet.

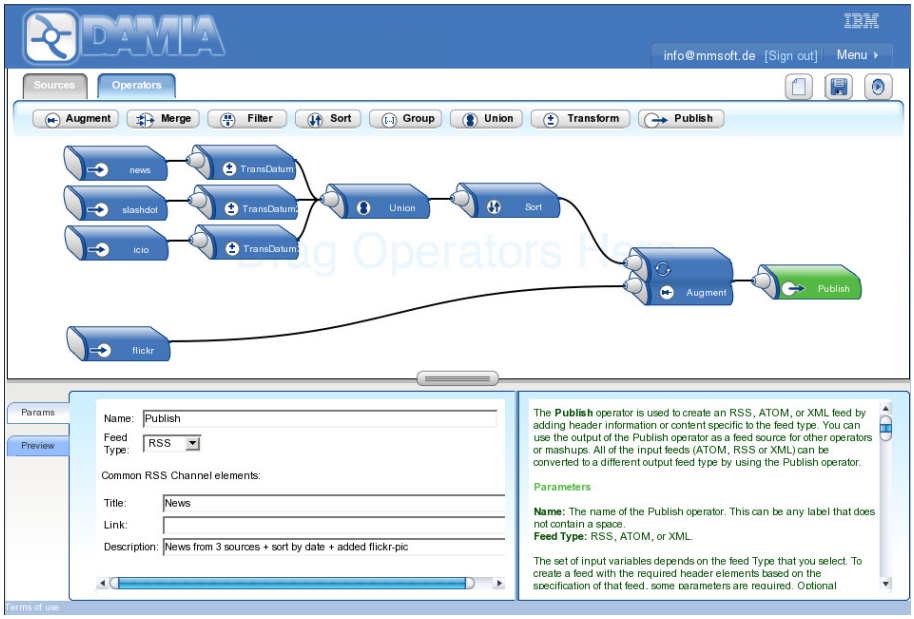

Abbildung 4.3: Beispielszenario in „IBM DAMIA“

Ein Problem stellt das unterschiedliche Datumsformat der Feeds dar. Mit sehr viel Aufwand wäre es wahrscheinlich möglich gewesen das Element *„date“* in ein einheitliches Format zu konvertieren. Aus diesem Grund ist die Sortierung zwar möglich, aber nur bedingt richtig.

Die Datenquelle des Bildes [fli] kann über *„Augment“* mit dem Wert des Attributs *„title“* angesteuert werden. Allerdings wird die gesamte Zeichenkette zur Suche in *„flickr.com“* verwendet, was nur zu wenigen Suchergebnissen führt. Die Ausgabe erfolgt als Feed, in einem gewünschten Format, über den Prozessor *„Publish“*.

Die Umsetzung des Szenarios war mit *„DAMIA“* nur teilweise erfolgreich, da die Suche nach Bildern nur sehr selten zu Ergebnissen führt. Außerdem ist es nicht möglich doppelte Einträge zu erkennen und zu entfernen. Zur Erstellung des Mashups wurden etwa 10 Minuten benötigt.

4.4 Yahoo! Pipes

Die Feeds sind gemeinsam mit dem Quell-Operator *„Fetch Feed“* abrufbar und mit dem Operator *„Sort“* können die Einträge nach einem beliebigen Feld sortiert werden.

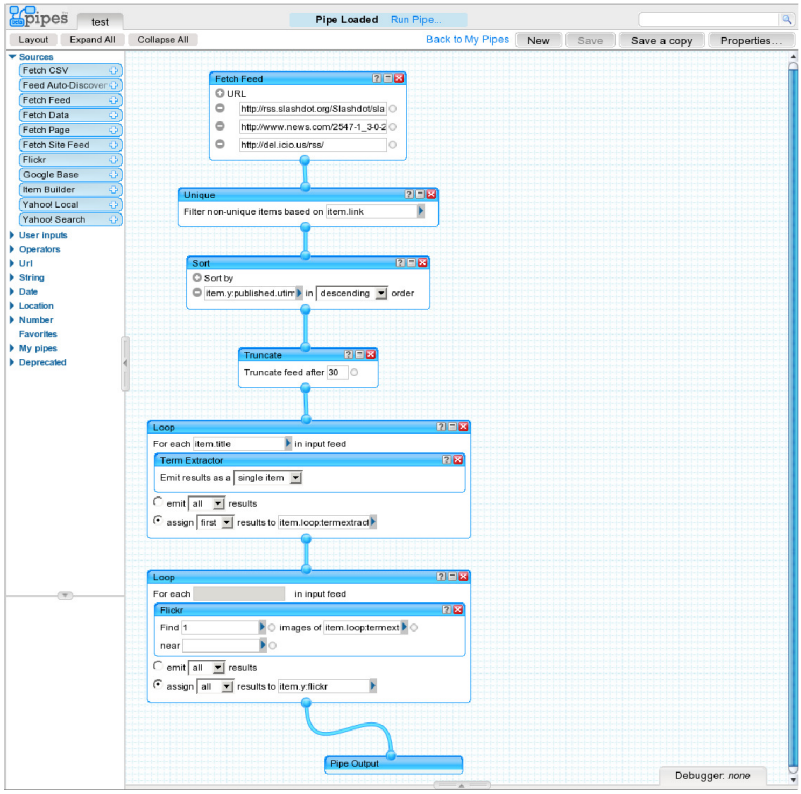

Abbildung 4.4: Beispielszenario in „Yahoo! Pipes“

„Pipes“ bietet keine Möglichkeit, einzelne Einträge auf Ähnlichkeit zu überprüfen und somit doppelte Einträge zu eliminieren. Es ist, ebenso wie beiden anderen Anwendungen, nur möglich, Duplikate anhand eines exakten Wertes in einem Feld zu erkennen.

Um den Elementen ein Bild zuzuordnen, wurde zunächst der Prozessor *„Term Extractor“* auf jedes Element des Feeds angewendet, um Schlagworte aus dem Titel zu extrahieren. In der folgenden Schleife wurden anhand dieser Schlagworte Bilder bei *„flickr.com“* gesucht und das jeweils erste zum jeweiligen Feed-Element hinzugefügt.

Das Umsetzen des Szenarios erfordert mit Pipes etwa 10 Minuten. Bis auf die ungenügende Duplikaterkennung konnten alle Anforderungen des Szenarios umgesetzt werden.

Anwendung	Vereinigung	Duplikatentfernung	Sortierung	Bildzuordnung	Umsetzung erfolgreich	Benötigte Zeit
„Apatar“	√	√	(√)	–	–	30 Min.
„Microsoft Popfly“	(√)	–	√	√	–	10 Min.
„IBM DAMIA“	√	–	√	(√)	–	10 Min.
„Yahoo! Pipes“	√	√	√	√	√	10 Min.

Tabelle 4.1: Auswertung des Beispielszenario

4.5 Auswertung

Wie in Tabelle 4.1 dargestellt ist, war es nur mit *„Yahoo! Pipes“* möglich das Szenario vollständig umzusetzen. *„Apatar“* bietet ebenfalls alle, für das Szenario benötigten, Prozessoren an, jedoch ist deren Implementierung derart mangelhaft, dass die Umsetzung des Szenarios nicht vollständig erfolgen konnte. Die Anwendung *„Microsoft Popfly“* ist für dieses Szenario am wenigsten geeignet, da es nur auf eine sehr unbefriedigende Art möglich war, die Vereinigung der einzelnen RSS-Feeds zu realisieren. Mit *„IBM DAMIA“* konnte das Szenario ebenfalls nicht komplett abgebildet werden, da, ebenso wie bei *„Popfly“*, kein Prozessor zur Erkennung von Duplikaten zur Verfügung steht. Auffallend ist weiterhin, dass die Umsetzung des Szenarios mit *„Apatar“* deutlich mehr Zeit in Anspruch nahm, als mit den anderen drei Anwendungen.

Kapitel 5

Zusammenfassung

Obwohl die Anwendung *„Apatar“* auf den ersten Blick für die Erstellung von Mashups sehr gut geeignet erscheint, stellt sich während der Benutzung heraus, dass einige Funktionen nur schlecht oder gar nicht implementiert sind. So ist zum Beispiel die Extraktion von Daten aus RSS-Feeds sehr unzuverlässig und die Konvertierung von Strings in andere Datentypen funktioniert gar nicht. Ein weiterer Nachteil sind die fehlenden Operatoren, wie Zählen aller Datensätze oder Sortieren. Die einzige Möglichkeit diese Funktionalität mit *„Apatar“* zu erreichen, ist, das Speichern der Daten in einer relationalen Datenbank und nachfolgendes Abfragen mit „`Select Count(*) from Table`“, beziehungsweise „`Select * from Table Order by Field1`“. Das ist sehr umständlich, da *„Apatar“* davon ausgeht, dass die benötigten Tabellen bereits in der Datenbank existieren. Es gibt keine Möglichkeit, die Schemata der Datenbank aus der Anwendung heraus zu verändern.

Sowohl *„Apatar“*, als auch *„Microsoft Popfly“* und *„IBM DAMIA“* sind noch immer im Beta-Stadium der Software-Entwicklung. Das macht sich an vielen Stellen bemerkbar. So sind in jedem der Programme wichtige Funktionen nur mangelhaft implementiert. Das hat zur Folge, dass zum jetzigen Zeitpunkt keine der Anwendungen wirklich produktiv genutzt werden können, ausser eventuell für sehr ausgesuchte Szenarien. *„Apatar“* befindet sich offensichtlich noch im frühesten Entwicklungsstadium, bezogen auf die Erstellung von Mashups. Für die Integration von Daten aus relationalen Datenbanken ist die Anwendung jedoch deutlich besser geeignet, da viele der noch fehlenden Funktionen durch SQL-Befehle ersetzt werden können.

„Yahoo! Pipes“ hingegen ist eine bereits sehr ausgereifte Anwendung, die es Benutzern mit sehr geringen bis gar keinen Programmierkenntnissen erlaubt, Mashups zu erstellen. Sie bietet zwar nicht die komplette Flexibilität, die von *„Apatar“* angestrebt wird, ist allerdings für die schnelle Integration von Daten sehr gut geeignet. Außerdem ist *„Pipes“* unter den betrachteten Anwendungen die einzige, die die Extraktion von Entitäten unterstützt. Damit ist zu erwarten, dass in zukünftigen Versionen ein Prozessor zur Entitiy-Resolution enthalten sein könnte.

„IBM DAMIA“ besitzt sicherlich das beste Konzept, bezüglich der Erstellung von Mashups. Es benutzt intern XML-Daten und XPath zur Navigation und Suche. *„Yahoo! Pipes“* hingegen verwendet immer RSS-Feeds, was die Flexibilität deutlich einschränkt. *„Apatar“* arbeitet, als ETL-Werkzeug, intern mit Tabellen zur Verarbeitung der Informationen.

Da keine der Anwendungen alle Wünsche zur Integration von Daten abdeckt, liegt die Verwendung mehrerer Werkzeuge nahe. Da bis auf *„Microsoft Popfly“* alle Werkzeuge RSS-, ATOM- und XML-Feeds lesen und erstellen können, ist es möglich, mit Hilfe dieser Datenformate Informationen zwischen den einzelnen Anwendungen auszutauschen. Damit ist die gemeinsame Verwendung der Werkzeuge zur Integration von Daten möglich, indem die Ausgabe einer Anwendung in einer zweiten als Eingabe benutzt wird. Auf diese Weise können nahezu alle Anwendungen alle möglichen Prozessoren in einem anwendungsübergreifenden Datenfluss nutzen.
So wäre es zum Beispiel möglich, mit *„Apatar“* Adressdaten aus zwei unterschiedlichen Datenbanken abzufragen und daraus jeweils einen Feed zu erstellen. *„Yahoo! Pipes“* könnte im Anschluß benutzt werden, um die, hinter den Adressdaten stehenden, Entitäten zu extrahieren. Die resultierenden Feeds könnten wiederum in Apatar verwendet werden, um die extrahierten Entitäten zu vergleichen. Damit wäre mit der Benutzung der beiden Anwendungen die Funktion der Entity-Resolution implementierbar.

Literaturverzeichnis

[AB+07] A , Mehmet ; B , Paul ; C , Susan ; K , Rajesh ; L , Eric ; M , Volker ; M , Louis ; N , Yip-Hing ; S , David E. ; S , Ashutosh: DAMIA - A Data Mashup Fabric for Intranet Applications. In: *VLDB*, 2007, S. 1370–1373

[Apa] A I . (Hrsg.): *Apatar – Open Source Data Integration & ETL*. `http://www.apatar.com`, Abruf: 20.01.2008

[ATO05] T I E T F (Hrsg.): *The Atom Syndication Format*. 46000 Center Oak Plaza, Sterling, VA 20166: The Internet Engineering Task Force, Dezember 2005. `http://www.ietf.org/rfc/rfc4287.txt`, Abruf: 20.01.2008. – Request for Comments 4287

[BB06] B , Olaf ; B , Carsten: *Ajax, Frische Ansätze für das Webdesign*. Teia Lehrbuch Verlag, 2006. – ISBN 978–3935539265

[BG04] B , Andreas (Hrsg.) ; G¨ , Holger (Hrsg.): *Data Warehouse Systeme*. 2., überarbeitete und aktualisierte Auflage. dpunkt.varlag GmbH, 2004. – ISBN 3–89864–251–8

[Cro06] C , Douglas ; T I E T F (Hrsg.): *The application/json Media Type for JavaScript Object Notation (JSON)*. 46000 Center Oak Plaza, Sterling, VA 20166: The Internet Engineering Task Force, Juli 2006. `http://www.ietf.org/rfc/rfc4627.txt`, Abruf: 20.01.2008. – Request for Comments 4627

[Dap] D I . (Hrsg.): *Dapper: The Data Manager*. http://www.dapper.net/, Abruf: 20.01.2008

[Fie00] F , Roy T.: *Architectural Styles and the Design of Network-based Software Architectures*, University of California, Irvine, Diss., 2000

[fli] Y ! I . (Hrsg.): *flickr*. http://www.flickr.com, Abruf: 20.01.2008

[Gad] M D G H (Hrsg.): *Skripting für Windows Vista: Erstellen von Gadgets*. http://www.microsoft.com/germany/technet/scriptcenter/topics/vista/gadgets-pt1.mspx#EZB, Abruf: 20.01.2008

[GME] G I . (Hrsg.): *Google Mashup Editor*. http://code.google.com/gme/, Abruf: 20.01.2008

[Haz] H , Philip: *PCRE - Perl Compatible Regular Expressions*. http://www.pcre.org/, Abruf: 20.01.2008

[HB+02] H , Mark ; B , Rich ; S , Rahul ; F , Joseph ; S , Kate ; S M , I . (Hrsg.): *Java Message Service*. Version 1.1 FCS. : Sun Microsystems, Inc., April 2002. http://java.sun.com/products/jms/docs.html, Abruf: 23.01.2007

[HC03] H , Kevin ; C , Tara: *Spidering Hacks*. O'Reilly Verlag, 2003. – ISBN 0–596–00577–6

[Hoh07] H , Gregor: Mashups Tools Market. Version: August 2007. http://www.eaipatterns.com/ramblings/60_mashupmarket.html, Abruf: 20.01.2008. Gregor Hohpe, August 2007. – Forschungsbericht

[IBMa] IBM C (Hrsg.): *IBM DAMIA*. http://damia.alphaworks.ibm.com, Abruf: 20.01.2008

[IBMb] IBM C (Hrsg.): *IBM QEDWiki*. http://services.alphaworks.ibm.com/qedwiki, Abruf: 20.01.2008

[kap] K T (Hrsg.): *openkapow*. http://openkapow.com, Abruf: 20.01.2008

[ME+03] M , Peter ; E ; A ; Y ; M ; K - ; A : *Kundenbeziehungsmanagement*. http://de.wikipedia.org/wiki/Kundenbeziehungsmanagement. Version: Juni 2003, Abruf: 20.01.2008

[Mer06] M , Duane: Mashups: The new breed of Web app. Version: August 2006. http://www-128.ibm.com/developerworks/library/x-mashups.html, Abruf: 20.01.2008. International Business Machines Corporation, August 2006. – Forschungsbericht

[Moz] M C (Hrsg.): *Firefox web browser*. http://www.mozilla.com/firefox/, Abruf: 20.01.2008

[MSPa] M C (Hrsg.): *Microsoft Popfly*. http://www.popfly.ms, Abruf: 20.01.2008

[MSPb] M C (Hrsg.): *Popfly Explorer Visual Studio add-in*. http://www.popfly.ms/Overview/Explorer.aspx, Abruf: 20.01.2008

[MSS] M C (Hrsg.): *Microsoft Silverlight*. http://silverlight.net, Abruf: 20.01.2008

[MyS] M S I . (Hrsg.): *MySpace*. http://www.myspace.com, Abruf: 20.01.2008

[NV07] N , Jasminko ; V , Benjamin J. J.: Mashups: Strukturelle Eigenschaften und Herausforderungen von End-User Development im Web 2.0. In: *i-com* 6 (2007), Mai, Nr. 1, S. 19–24. – ISSN 1618–162X

[Pipa] Y ! I . (Hrsg.): *Yahoo! Pipes*. http://pipes.yahoo.com, Abruf: 20.01.2008

[Pipb] Y ! I . (Hrsg.): *Yahoo! Pipes Dokumentation : How do I use Pipes?* http://pipes.yahoo.com/pipes/docs, Abruf: 20.01.2008

[RDF04] W W W C (Hrsg.): *Resource Description Framework (RDF)*. ttp://www.w3.org/TR/rdf-primer/. Version: Februar 2004, Abruf: 20.01.2008

[RSS] RSS A B (Hrsg.): *RSS 2.0 Specification*. http://www.rssboard.org/rss-specification, Abruf: 20.01.2008

[RTA05] R , Erhard ; T , Andreas ; A ¨ , David: *Dynamic Fusion of Web Data: Beyond Mashups*, September 2005. http://dbs.uni-leipzig.de/file/xsym07-rahm_0.pdf, Abruf: 20.01.2008

[SOA07] W W W C (Hrsg.): *SOAP Specifications*. http://www.w3.org/TR/soap/. Version: April 2007, Abruf: 20.01.2008

[Str] S I I . (Hrsg.): *Strikeiron Marketplace*. http://www.strikeiron.com, Abruf: 20.01.2008

[SwW] W W W C (Hrsg.): *W3C Semantic Web Activity*. http://www.w3.org/2001/sw/, Abruf: 20.01.2008

[Tan02] T , Andrew S.: *Moderne Betriebssysteme*. 2., überarbeitete Auflage. Pearson Education Deutschland GmbH, Prentice Hall Inc., 2002. – ISBN 3–8273–7019–1

[WH06] W , Jeffrey ; H , Jason: Marmite: end-user programming for the web. In: *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*. New York, NY, USA : ACM, 2006. – ISBN 1–59593–298–4, S. 1541–1546

[WH07] W , Jeffrey ; H , Jason I.: Making mashups with marmite: towards end-user programming for the web. In: *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM, 2007. – ISBN 978–1–59593–593–9, S. 1435–1444

[WsW] W W W C (Hrsg.): *Web Services @ W3C*. `http://www.w3.org/2002/ws`, Abruf: 20.01.2008

[XPa07] W W W C (Hrsg.): *XML Path Language (XPath) 2.0*. `http://www.w3.org/TR/xpath20`. Version: Januar 2007, Abruf: 20.01.2008