PROOF[1]: Produktmonitoring im Web

Christian Wartner, Sven Kitschke

WDI-Lab, Institut für Informatik, Universität Leipzig
{wartner | kitschke}@informatik.uni-leipzig.de

Abstract: Für Verbraucher, Händler und Hersteller ist die Beobachtung der Entwicklung von Angebotspreisen interessant, seien dies Preise von Produkten, Flügen oder Dienstleistungen. Wir präsentieren mit PROOF einen neuartigen Ansatz Produktmonitoring durchzuführen. PROOF ist ein erweiterbares System zum Integrieren und Analysieren von Webdaten. Die definierbaren Workflows erlauben unter anderem Anfrageoptimierungs- und Objekt-Matching-Operationen, um eine hohe Datenqualität bei guter Performance zu erreichen.

1 Motivation

Der Onlinehandel mit Produkten ist in den vergangenen Jahren stetig gewachsen. Neben Online-Shops sind über 1000 Preisvergleichsportale entstanden [KH07], welche die Angebote der Händler aggregieren. Um einen möglichst vollständigen Überblick über Produktangebote zu erhalten, ist man gezwungen, viele Shops zu besuchen oder sich mehrerer Preisvergleichsportale zu bedienen. Wenn Angebote eines Händlers in unterschiedlichen Portalen gefunden werden, sind die gewonnenen Ergebnisse um Duplikate zu bereinigen. Will man die Preisentwicklung über einen längeren Zeitraum beobachten, muss diese Aufgabe zudem häufig wiederholt werden.

Wir demonstrieren mit unserer Applikation, dass das Auffinden und die Identifikation relevanter Datensätze zu Produktangeboten, das Zusammenführen dieser oft heterogenen Datensätze und ihre Analyse in vielen Teilen automatisiert und damit effizienter gestaltet werden kann. Produktmonitoring mit PROOF besitzt folgende Vorteile:

- Skriptbasierte Steuerung der Workflows für Erweiterbarkeit, Adaption an konkrete Analyseaufgaben und Einsatz in mehreren Domänen (z.B. auch für Flüge und Immobilien),
- Querygeneratoren [ETR09] zur Anpassung an unterschiedliche Anfrageschnittstellen und zur Optimierung hinsichtlich Qualität und Effizienz,
- flexible Objekt-Matching-Verfahren zur Duplikatbereinigung,
- Operatoren zur effizienten Verarbeitung heterogener Daten.

[1] Product Offer Fusor

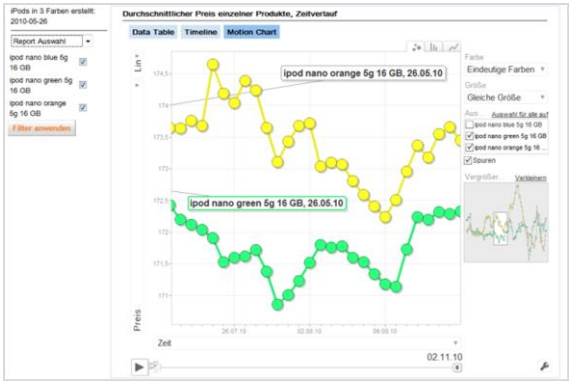

Abbildung 1: Screenshot – Darstellung der Preisentwicklung zweier Produkte

Die periodisch ausgeführte Aggregation der Angebote befreit den Nutzer von der wiederkehrenden und zeitraubenden Aufgabe, verschiedene Webshops und Preisvergleichsportale zu besuchen und deren Angebote und Produktpreise manuell zu extrahieren. Langfristig von Vorteil ist, dass der Workflow für andere Produkte wiederverwendet werden kann und lediglich parametrisiert werden muss.

2 Unser Ansatz

Architektur: Bei der vorgestellten Anwendung handelt es sich um eine AJAX[2]-basierte Webapplikation. Der im Webbrowser ausgeführte clientseitige Teil der Anwendung dient zum Erstellen und Verwalten von *Monitoring-Jobs* sowie zur Visualisierung und Auswertung der gesammelten Daten. Aufgabe des Servers ist die periodische Ausführung von *Extraktionsworkflows* und das Bereitstellen von aus den gesammelten Daten erstellten Reports. Die dazu nötigen Operationen werden mit Hilfe von *weFuice*, einer Weiterentwicklung von *iFuice*[3] [Ra05], realisiert. *weFuice* stellt eine Skriptsprache zum Definieren von Datenintegrationsworkflows bereit. In ihr werden Operatoren für die Generierung von Anfragen, zum Zugriff auf verschiedenste Datenquellen, zum Objekt-Matching sowie für weitere Funktionen wie z.B. Mengenoperationen und statistische Funktionen genutzt.

Ansatz: Ausgangspunkt für das Produktmonitoring ist das Erstellen von *Monitoring-Jobs*, dem eine vom Benutzer zu erstellende Liste von Produkten zu Grunde liegt. Für jeden *Monitoring-Job* werden ein Extraktionsintervall und ein *Extraktionsskript* festgelegt. Das Sammeln von Daten zu Produktangeboten erfolgt durch die periodische Ausführung des *Extraktionsskriptes* (siehe Abb. 2 links), welches Anfragen aus der Liste der zu überwachenden Objekte an *Monitoring-Datenquellen*, wie Preisvergleichsseiten oder Metasuchmaschinen, generiert, ausführt und deren Ergebnisse verarbeitet und

[2] Asynchronous JavaScript and XML
[3] Information Fusion utilizing Instance Correspondences and Peer Mapping

archiviert. Zur Auswertung der gesammelten Daten werden *Analyseskripte* (siehe Abb. 2 rechts) definiert, deren Ergebnisse als XML-Daten verfügbar sind oder im Browser dargestellt werden können.

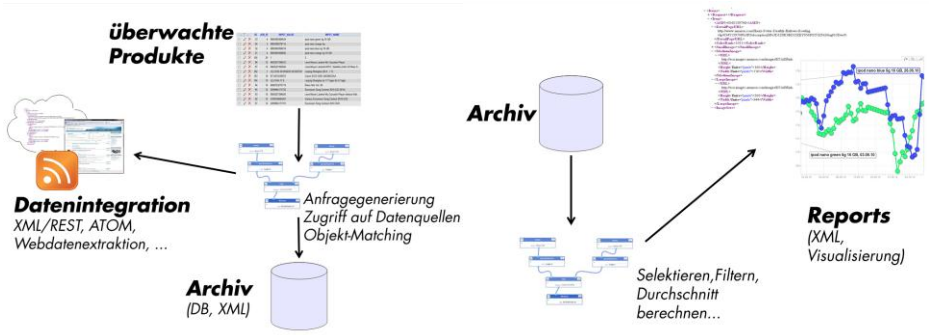

Abbildung 2: Generischer Workflow

Erstellen von Monitoring-Jobs: Je nach Typ des Extraktionsskripts werden unterschiedliche Anfragen an die *Monitoring-Datenquellen* erzeugt. Zum Erzielen guter Treffer in den *Monitoring-Datenquellen* sollten für jedes überwachte Produkt eindeutige Merkmale vorhanden sein. Diese können zum Beispiel die EAN[4] oder ein eindeutiger Titel sein, der das Produkt von ähnlichen Produkten und Zubehörartikeln ausreichend abgrenzt. Dabei wird vom Benutzer nicht erwartet, dass er diese Produkteigenschaften genau kennt. Vielmehr wählt er nach Eingabe von Schlüsselwörtern aus Produkttiteln einen oder mehrere daraufhin generierte Vorschläge aus, die, falls verfügbar, Attribute wie EAN, Hersteller und Produktnamen enthalten.

Ausführung des Extraktionsskripts: In den *Extraktionsskripten* wird festgelegt, welche *Monitoring-Datenquellen* benutzt und wie Anfragen an diese Datenquellen erzeugt werden. Datenquellen können z.B. RSS-Feeds, Webservices oder per Screen-Scraping-Verfahren aufbereitete Webseiten sein. Je nach Anfragemöglichkeiten einer *Monitoring-Datenquelle* und vorhandenen Attributen der Eingabeobjekte ist die Strategie maßgebend, mit der Anfragen erzeugt werden, um alle relevanten Einträge in einer Datenquelle zu finden. Beim Verarbeiten großer Mengen von Eingabedaten kann die Reduktion der Anzahl gesendeter Anfragen entscheidend sein. Konfigurierbare Querygeneratoren dienen zum Erzeugen von Anfragen aus einer Liste von Eingabeobjekten. Bei Produkten können dafür im Normalfall Produkttitel, Metainformationen wie Hersteller oder EAN-Code verwendet werden. Die Ergebnismenge bei Anfragen, die aus dem Titel erzeugt werden, ist meist größer, beinhaltet aber oft Angebote für ähnliche Produkte oder Produktzubehör. Werden Anfragen mit Hilfe des EAN-Codes generiert, dann sind die Ergebnismengen in der Regel exakter aber von geringerer Kardinalität. Querygeneratoren ermöglichen es, bestimmte Attribute zu bevorzugen und auf andere Attribute auszuweichen, falls Attribute fehlen oder nicht durch Anfrageschnittstellen unterstützt werden. Datenquellen, welche die Anzahl erlaubter Anfragen limitieren oder

[4] European Article Number - eindeutige Produktkennzeichnung für Handelsartikel

lange Antwortzeiten haben, benötigen Querygeneratoren, die die Anzahl der Anfragen reduzieren. Dazu können häufig vorkommende Schlüsselwörter oder OR-Verknüpfungen von Attributwerten unterschiedlicher Eingabeobjekte genutzt werden. Zur Identifikation der relevanten Treffer im Anfrageergebnis werden Objekt-Matching-Abläufe definiert, die eine String-Ähnlichkeit bestimmen. Zudem dienen diese Verfahren zum Entfernen von Duplikaten aus der Menge der gefundenen Angebote.

Auswertung: *Analyseskripte* greifen auf die archivierten Daten zu und sollen aggregierte Informationen bereitstellen, beispielsweise den Zeitverlauf des durchschnittlichen Preises oder die Händler mit den günstigsten Angeboten der vergangenen Tage. Das System bietet die Möglichkeit weitere Analyseskripte hinzuzufügen.

3 Einsatzszenarien

Für *Verbraucher* kann es relevant sein, schnell einen Überblick über die aktuellen Angebote zu erhalten, um daraus das günstigste auszuwählen. Das Monitoring von Produktpreisen kann ihn bei der Auswahl eines günstigen Kaufzeitpunkts unterstützen. Für *Online-Händler* ist es interessant, den aktuellen Marktpreis zu bestimmen, um z.B. Angebotspreise initial festzulegen. Eine Beobachtung über einen Zeitraum gibt Händlern die Möglichkeit, schnell auf Preisänderungen zu reagieren, den Angebotspreis nach oben oder nach unten zu korrigieren. Für *Hersteller* bietet sich eine Möglichkeit den aktuellen Marktpreis ihrer eigenen Produkte zu beobachten und sie in Relation zu Konkurrenzprodukten zu stellen. Zudem können Hersteller ermitteln, in welchen Shops und zu welchem Preis ihre Produkte angeboten werden.

4 Demonstration

Wir haben unsere Anwendung in den Domänen Produkt- und Flugpreisbeobachtung getestet. Teilnehmer der Vor-Ort-Demonstration können sich Ergebnisse bisher erfolgter Preisbeobachtungen ansehen. Dabei werden die verschiedenen Auswertungs- und Visualisierungsmöglichkeiten vorgeführt. Außerdem wird die Vorgehensweise bei der Erstellung von Monitoring-Jobs sowie die Abfrage und Aggregation aktueller Preisinformationen demonstriert.

Literaturverzeichnis

[ETR09] Endrullis, S.; Thor, A. und Rahm, E.: Evaluation of Query Generators for Entity Search Engines. Proc. of Intl. Workshop on Using Search Engine Technology for Information Management (USETIM), 2009.

[KH07] Klaus, P.; von Hören, Th.: Branchenführer Preisvergleichsportale. mpEXPERT, Grasberg, 2007.

[Ra05] Rahm, E.; Thor, A.; Aumueller, D.; Do, H.H.; Golovin, N.; Kirsten, T.: Information Fusion utilizing Instance Correspondences and Peer Mapping. 8th International Workshop on the Web and Databases (WebDB), 2005.