



Freitextsuche Weikatec Magellan Explorer

White Paper

Autoren: Dirk Weitemeyer, Weikatec Software GmbH, dw

Historie:

01.04.2006 dw Erstellung, Version 1.0

20.02.2009 dw Überarbeitung für Version 1.1

19.03.2009 mg Überarbeitung für Version 1.2

Die in diesem Dokument enthaltenen Informationen können jederzeit ohne Benachrichtigung geändert werden. Weikatec übernimmt daher keine Garantie für die Richtigkeit der dargestellten Informationen nach dem Datum der Veröffentlichung. Diese White Paper dient ausschließlich Informationszwecken. Weikatec sichert mit diesem Dokument keine Produkteigenschaften zu.

Diese Dokument ist einschließlich aller seiner Teile urheberrechtlich geschützt. Alle Rechte sind ausdrücklich vorbehalten, einschließlich der Rechte auf Vervielfältigung, Reproduktion, Übersetzung, Mikroverfilmung, Speicherung auf elektronischen Medien und Verarbeitung in elektronischer Form.

Alle Firmen-, Produkt- und Markennamen sind Warenzeichen oder eingetragene Warenzeichen der entsprechenden Inhaber.

Copyright © 2006-2009 Weikatec Software GmbH. Alle Rechte vorbehalten.

Sollten Sie Fragen zu Weikatec Magellan haben, so wenden Sie sich bitte an:

Weikatec Software GmbH
Borstels Ende 23a
22391 Hamburg
Tel: 040-2700337
weitemeyer@weikatec.com
<http://www.weikatec.com>
<http://www.magellan-explorer.de>

Inhaltsverzeichnis

1	Allgemeines	4
2	Zielgruppe	4
3	Fehlertoleranz	6
4	Wortstammerkennung	6
5	Wortteilung	6
6	Synonyme	6
7	Ersetzung	6
8	Taxonomien.....	6
9	Suchreihenfolge	7
10	Sortieren und Filtern	7
10.1	Sortierung.....	7
10.2	Filterung	7
11	Statische Suche.....	7
12	Mehrsprachigkeit	7
13	Mandantenfähigkeit	7
14	Top Suchbegriffe	8
15	Ähnliche Suchbegriffe.....	8
16	Systemarchitektur	10
16.1	Betriebssystem.....	10
16.2	Web-Server	10
16.3	File-Server.....	10
16.4	Application-Server.....	10
16.5	Datenbank-Server	10
17	Externer oder integrativer Ansatz	11
18	Caching.....	11
19	Skalierbarkeit	11
20	TCO - total cost of ownership	11
20.1	Hardware.....	11
20.2	Hosting	11
20.3	Lizenzkosten	11
20.4	Template-Entwicklung (nur in externer Variante)	12
20.5	Modulentwicklung und –anpassung	12
21	Performance-Beispiel	12

1 Allgemeines

Weikatec Magellan ist eine sehr schnelle und fehlertolerante Freitextsuche, die individuell konfigurierbar ist und damit auf nahezu beliebige Daten angewendet werden kann.

Sie kann ohne große Probleme in jeden beliebigen Online-Shop integriert werden und läuft nach der Inbetriebnahme nahezu pflegefrei.

2 Zielgruppe

Weikatec Magellan Explorer wendet sich an Kunden,

- denen die Standardsuche Ihres Online-Shops nicht ausreicht
- oder eine sonstige Applikation betreiben, die eine Suche benötigt.
- die eine fehlertolerante Freitextsuche benötigen
- die die beschriebenen Leistungen mit möglichst geringen Lizenz- und Hardwarekosten nutzen wollen.

Freitextsuche Weikatec Magellan Explorer

White Paper

Funktionaler Überblick

3 Fehlertoleranz

Fehler, d.h. Abweichungen von der korrekten Schreibweise, können sowohl bei der Eingabe des Suchbegriffs als auch in den Daten vorkommen. Daher ist es wichtig, dass auch ähnliche Begriffe zuverlässig gefunden werden.

Es werden 2 Typen von Fehlern unterschieden:

- Rechtschreibfehler: Hier werden Suchbegriffe herausgesucht, die phonetisch ähnlich sind. Z.B. Micro und Mikro, oder Photo und Foto.
- Tippfehler: Hier werden mit einem erweiterten Levenshtein-Algorithmus Suchbegriffe ermittelt, die dem ursprünglich eingegebenen ähnlich sind. Dazu zählen einzelne weggelassene oder hinzugefügte Buchstaben oder Buchstabendreher, z.B. Adidsan => Adidas

4 Wortstammerkennung

Sowohl bei den eingegebenen Suchbegriffen als auch in den zu durchsuchenden Daten wird eine Wortstammerkennung durchgeführt. Dadurch werden Plurali auf den Singular zurückgeführt (Häuser => Haus) als auch unterschiedliche Verbformen auf den gleichen Stamm zusammen geführt (löschen, lösche, gelöscht => lösche)

5 Wortteilung

Ein spezifisch deutsches Problem sind die zusammengesetzten Wörter. Wird z.B. „Lederstiefel“ eingegeben, soll möglichst auch der „Stiefel aus Leder“ gefunden werden. Dazu werden sowohl die Suchbegriffe als auch die Produktdaten einer Wortteilung unterzogen. In dem Beispiel wird dann neben „Lederstiefel“ auch nach „leder“ und „stiefel“ gesucht. Werden diese Begriffe nah beieinander gefunden, ist auch dies ein gültiges Suchergebnis. Es wird aber immer niedriger bewertet als eine Fundstelle des zusammengesetzten Wortes, denn es wird ja z.B. auch „Stiefel mit Ledersohle“ gefunden.

6 Synonyme

Obwohl die Suche möglichst pflegefrei konzipiert ist, gibt es doch einige Fälle, wo man um den Einsatz von Synonymen nicht herumkommt.

Wenn es 2 oder mehr Worte für die gleiche Sache gibt, soll bei der Suche nach einem der Begriffe gleichzeitig auch nach den anderen gesucht werden, z.B. Teleskop und Fernrohr. Desweiteren gibt es die Möglichkeit bei nicht hundertprozentig passenden Synonymen einen Malus zu definieren, so dass ein Suchtreffer beim Synonym schlechter gerankt wird als ein Suchtreffer beim Originalsuchwort.

Die Liste der Synonyme kann in Form einer Textdatei gepflegt werden.

7 Ersetzung

Falls es für einen Begriff mehrere Wörter gibt und bereits bekannt ist, dass in den Daten nur einer davon vorkommt, gibt es eine noch wirksamere Methode als Synonyme. Wenn z.B. bei Fernsehern das Wort „Fernseher“ im Text nie vorkommt, da es als „TV“ benannt wird, dann kann man die Ersetzung so definieren, dass bei Eingabe von „Fernseher“ der Suchbegriff automatisch in „TV“ umgewandelt wird.

Die Liste der Ersetzungen kann in Form einer Textdatei gepflegt werden.

8 Taxonomien

Synonyme sind sinnvoll, wenn beide Begriffe das Gleiche beschreiben. Anders verhält es sich, wenn ein Begriff der Oberbegriff für einen anderen ist. So ist z.B. ein Mountain Bike ein Fahrrad, aber nicht jedes Fahrrad ist ein Mountain Bike. Für diese Fälle kann eine Baumstruktur definiert werden, so dass bei einer Suche nach einem allgemeinen Begriff (z.B. Fahrrad) gleichzeitig auch nach den spezielleren Begriffen (z.B. Rennrad, MTB) gesucht wird. Die Baumstruktur kann beliebig viele Ebenen haben.

9 Suchreihenfolge

Die Reihenfolge bestimmt sich aus der Güte des Suchergebnisses. Dabei geht es darum, wie genau das Suchwort getroffen wurde und ob es an wichtigen oder eher unwichtigen Stellen steht. Daneben gibt es eine Vielzahl von weiteren Einflusskriterien.

Es ist zudem möglich, dass externe Daten die Reihenfolge beeinflussen. Dies kann z.B. der Umsatz des Artikels sein, um Topseller zu bevorzugen. Die Stärke des Einflusses dieser externen Faktoren ist einstellbar.

10 Sortieren und Filtern

10.1 Sortierung

Das Suchergebnis kann nach verschiedenen Parametern sortiert werden. Die Standardsortierung ist nach der Güte des Suchergebnisses. Weitere mögliche Parameter sind theoretisch alle Datenfelder, sinnvoll sind davon nur einige, wie z.B. Preis oder Hersteller.

10.2 Filterung

Das Suchergebnis kann nach verschiedenen Parametern gefiltert werden. Theoretisch sind wieder alle Datenfelder möglich, sinnvoll einzusetzen sind z.B. Hersteller, Kategorie, Farbe, Größe, Preis, Zielgruppe und Warengruppe. Es kann nach mehreren Felder gleichzeitig gefiltert werden.

Die Daten können dabei in verschiedenen Formen zur Verfügung gestellt werden

- Einfache Liste (z.B. bei Farben)
- Baumstruktur (z.B. bei Kategorien)
- Bereiche, bzw. freie Eingabe eines Bereichs (z.B. beim Preis)

11 Statische Suche

Hiermit ist die Suche in Hilfe- bzw. Nicht-Angebots-Seiten des Shops gemeint. Auch hier war das Ziel, eine möglichst pflegearme Lösung zu finden. Es kam daher nicht in Frage, für jeden erdenklichen Suchbegriff, die Suchergebnisse einzeln pflegen zu müssen. Eine vollautomatische Suche, die z.B. in den Enfinity-Templates direkt sucht, ist auch nicht praktikabel, weil hier intensiv mit Includes gearbeitet wird und wenn dann ein Suchbegriff in einem Include gefunden wird, ist immer noch unklar, in welchem Template dieses Include eingebunden sein könnte.

Die eingesetzte Lösung sieht folgendermaßen aus: Der Administrator pflegt eine Liste von URLs, die die zu durchsuchenden statischen Seiten aufrufen. Innerhalb der Templates kann mit bestimmten Tags der Teil, der tatsächlich durchsucht werden soll, eingeschränkt werden, so macht es zum Beispiel keinen Sinn, die Hauptnavigation durchsuchen zu lassen, sonst erscheinen bei einem Suchbegriff aus diesem Teil sämtliche Seiten als Suchergebnis.

Die URLs werden vom Suchserver in einem zu definierenden Intervall aufgerufen, die erhaltenen html-Seiten werden gespeichert und dann für die Freitextsuche aufbereitet.

12 Mehrsprachigkeit

Die Mehrsprachigkeit ist gewährleistet:

- Die sprachabhängigen Teile wie Umlauterkennung und Wortstammerkennung werden durch sprachspezifische Varianten ersetzt.
- Der Wortteilungsalgorithmus entfällt bei allen Sprachen außer Deutsch.
- Templateseitig bietet die Velocity-Engine entsprechende Funktionalitäten.

13 Mandantenfähigkeit

Es können auf einem Server theoretisch beliebig viele Mandanten nebeneinander betrieben werden. Alle Einstellungen, Synonymlisten sowie Datenimporte und Templates können pro Mandant einzeln gepflegt werden.

14 Top Suchbegriffe

Weikatec Magellan Explorer kann die Top-Suchbegriffe eines konfigurierbaren Intervalls (z.B. letzter Tag) bei Berücksichtigung einer pflegbaren Blacklist von unerwünschten Begriffen ermitteln und diese dem Suchergebnis anhängen, so dass auf dem Suchergebnistemplate eine Liste der Top-Suchbegriffe oder eine Tag-Cloud ausgegeben werden kann.

15 Ähnliche Suchbegriffe

Weikatec Magellan Explorer kann aus den eingegebenen Suchbegriffen von Kunden eine Statistik erzeugen, die erlaubt, zu allgemein gehaltenen Suchworten, z.B. Hose, eine Liste von häufig benutzten ähnlichen Suchbegriffen zu liefern z.B. Sport-Hose, Hose Adidas, Anzug-Hose. Dies hilft dem Benutzer, seine Suche sinnvoll einzuschränken.

Freitextsuche Weikatec Magellan Explorer

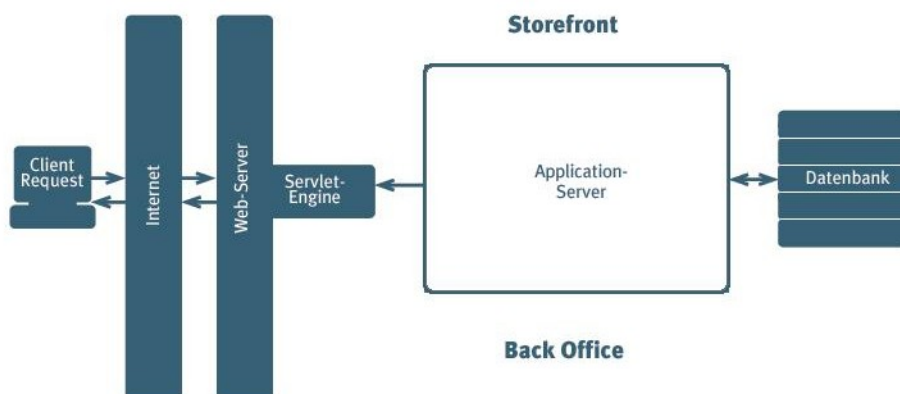
White Paper

Technischer Überblick

16 Systemarchitektur

Weikatec Magellan besteht aus einer modernen Mehrschichtarchitektur mit den folgenden Komponenten

- Web-Server
- Servlet-Engine
- Application-Server
- Datenbank



16.1 Betriebssystem

Weikatec Magellan wird auf folgenden Betriebssystemen angeboten:

- Linux (Debian, Red Hat u.a.)
- Microsoft Windows (2000, XP, 2000/2003 Server)

Generell wird Linux empfohlen. Windows wird nur für Test- und Entwicklungsmaschinen empfohlen.

16.2 Web-Server

Unter Linux wird mit Apache 2 gearbeitet. Der Apache Webserver ist Open Source und auf vielen Servern schon vorinstalliert.

Unter Windows kann MS-IIS oder der Tomcat-eigenen Webserver (nur QA- oder Entwicklungsmaschinen) eingesetzt werden

16.3 File-Server

Der Web-Server dient auch immer als File-Server für Bilder, Stylesheets, Javascript-Bibliotheken und andere Multimediadateien. Dies ist nur in der externen Variante nötig.

16.4 Application-Server

Der Magellan-Application-Server ist eine Eigenentwicklung von Weikatec, die in Java programmiert wurde. Für den Betrieb ist die Java Version 1.5 notwendig.

Im Application-Server tritt die Hauptlast auf. Daher ist hier eine Skalierung möglich, indem mehrere Application-Server parallel geschaltet werden. Des Weiteren wird durch intelligentes Caching die Last gesenkt(s. 18 Cacheing).

16.5 Datenbank-Server

Weikatec Magellan arbeitet mit MySQL in der Version 5.0 oder höher. MySQL ist eine robuste, leistungsfähige, sehr schnelle und transaktionsfähige Datenbank. MySQL ist Open Source, so dass keine Lizenzkosten anfallen.

Des Weiteren ist MySQL auf den meisten Mietservern bereits vorinstalliert.

17 Externer oder integrativer Ansatz

Die Suche kann komplett extern aufgesetzt werden. Sie bekommt dann die Daten in Form eines csv-Files zur Verfügung gestellt und übernimmt das komplette Templating, wobei bestimmte Teile, wie z.B. die zentrale Navigation, per Iframe aus dem Shop eingebunden werden können. Beim Klick auf ein Suchergebnis wird wieder in den Shop zurückgesprungen.

Alternativ ist auch ein integrativer Ansatz für Intershop Multisite und Enfinity Suite6 möglich. Dazu wird eine Cartridge im Shop integriert. Über ein Pipelet wird die Suchanfrage an den Suchserver geschickt. Als Ergebnis kommt dann eine Liste der Style-Namen plus weitere Listen für die After-Search-Navigation im XML-Format zurück. Der Aufbau des Suchtemplates wird dann vom Shop übernommen.

18 Caching

Zur Performance-Optimierung sind in die Suche zwei jeweils zwei-stufige Caches eingebaut. Die erste Stufe wird im Speicher gehalten und ist in der Größe beschränkt. In der zweiten Stufe werden die gecachelten Daten in der Datenbank gespeichert.

Der erste Cache speichert Suchergebnisse. Der zweite Cache speichert ganze Template-Seiten, in die nur noch die aktuelle SessionId integriert werden muss. Dieser Cache ist nur bei der externen Variante relevant.

19 Skalierbarkeit

Sollte der unwahrscheinliche Fall (s. 21 Performance-Beispiel) eintreten, dass eine 1-Server-Lösung die Anfragen nicht abarbeiten kann, so gibt es verschiedene Möglichkeiten der Skalierung:

- Aufteilen der 3-Schicht-Architektur (Webserver, Suchapplikation, Datenbank) auf verschiedene Rechner.
- Aufbau mehrerer identischer Parallelsysteme, die über einen Lastverteiler gleichmäßig mit Anfragen versorgt werden.
- Mischformen der beiden Methoden

20 TCO - total cost of ownership

Neben der Leistungsfähigkeit der Suche lag das Hauptaugenmerk darauf, ein System zu schaffen, das im Verhältnis zur Leistungsfähigkeit geringe TCO aufzuweisen hat.

20.1 Hardware

Ziel muss es hier sein, die Anforderungen an die Hardware so gering wie möglich zu halten. Maßgeblich dafür ist die Leistungsfähigkeit des Application-Servers. Grundvoraussetzung ist die Verwendung einer leistungsfähigen und schnellen Programmiersprache wie Java im Vergleich mit reinen Interpretersprachen wie Perl oder PHP.

Des Weiteren verzichtet Weikatec Magellan auf aufwändige Persistence-Layer wie EJB oder Servlet Frameworks wie Struts, sondern nutzt schlanke, selbst entwickelte, auf den Anwendungsbereich optimierte Klassen. Das dies führt zu einer deutlich schlankeren Architektur, die entsprechend geringere Hardware-Anforderungen nach sich zieht.

20.2 Hosting

Neben den Hardwarekosten spielt hier der Einrichtungsaufwand eine zentrale Rolle. Dieser ist bei Weikatec Magellan besonders gering, weil ein sogenannter Standard-LAMP-Server mit Linux, Apache und MySQL schon fast alle benötigten Komponenten enthält. Es muß nur noch Java installiert werden, der Rest wie Velocity und Tomcat ist Teil der eigentlichen Applikation.

20.3 Lizenzkosten

Neben den Lizenzkosten für Weikatec Magellan fallen keine weiteren Lizenzkosten an, da alle verwendeten Fremdkomponenten

- Linux (OS)
- Apache (WebServer)
- Tomcat (Servlet-Engine)
- MySQL (Datenbank)
- Velocity (Template-Engine)

Open Source sind.

20.4 Template-Entwicklung (nur in externer Variante)

Durch den Einsatz der Template-Engine Velocity ist die Template-Entwicklung von der eigentlichen Programmierung getrennt und kann kostengünstig durch den Kunden selbst durchgeführt werden. Velocity ist eine der verbreitetsten Template-Engines und sehr simpel aufgebaut, so dass auch bei bisher fehlendem Know-How dieses schnell aufgebaut werden kann.

20.5 Modulentwicklung und -anpassung

Durch den Verzicht auf externe Frameworks, wie im Kapitel 20.1 Hardware beschrieben, wird die Komplexität des Gesamtsystems deutlich gesenkt. Außerdem ist damit der komplette Sourcecode des Systems von Weikatec und damit im Zugriff für Änderungen. Das erleichtert das Debugging und verhindert die Programmierung aufwändiger Workarounds. Zusammen führt dies dazu, dass individuelle Anpassungen schnell und preisgünstig ausgeführt werden können.

21 Performance-Beispiel

Die Baur-Gruppe (www.baur.de, www.universal.at, www.imwalking.de), einer der größten Versandhändler Deutschlands, nutzt die Weikatec Magellan Suche mit 3 Mandanten in der externen Variante, d.h. der Suchserver übernimmt auch das komplette Templating. Als Hardware kommen 2 redundante Standard-Server der 1.000 EUR Klasse zur Anwendung. Dieser Server hat keinerlei Probleme die Suchanfragen zu bewältigen:

- Die Serverlast liegt sehr selten über 10%.
- Der tägliche Import der Quelldaten dauert ca. 15 Min.
- Die durchschnittliche Antwortzeit beträgt unter 100ms
- Pro Tag werden ca. 100.000 Anfragen beantwortet.