

Portal-U InGrid 1.0

DV-technisches Feinkonzept
iPlug

Version 1.0

Stand 23.06.2005

Erstellt durch das Konsortium

 GIS-tec

GIStec GmbH

Rundeturmstraße 12
64283 Darmstadt
Tel (+49)6151 / 155-250
Fax (+49)6151 / 155-259
Mail info@gistec-online.de
<http://www.gistec-online.de>



Fraunhofer Institut für Graphische Datenverarbeitung

Fraunhoferstraße 5
64283 Darmstadt
Tel (+49)6151 / 155-0
Fax (+49)6151 / 155-199
Mail info@igd.fhg.de
<http://www.igd.fhg.de>



media style GmbH

Mansfelder Straße 13
06108 Halle
Tel (+49)345 / 5222702
Fax (+49)345 / 5222719
Mail info@media-style.com
<http://www.media-style.com>

 wemove

wemove digital solutions GmbH

Eschersheimer Landstraße 5-7
60322 Frankfurt
Tel (+49)69 / 759003-0
Fax (+49)69 / 759003-22
Mail info@wemove.com
<http://www.wemove.com>

Im Auftrag der Koordinierungsstelle UDK/GEIN



**Niedersächsisches Umweltministerium –
Koordinierungsstelle UDK/GEIN**

Archivstrasse 2
30167 Hannover
Tel (+49)511 / 120-3480
Fax (+49)511 / 120-3697
Mail kug@numis.niedersachsen.de
<http://www.udk-gein.de>

Dokumentenhistorie

Version	Datum	Bearbeiter	Anlass	Status
1.0	20.06.2005	GIS-tec / media style	Phase 1 Portal-U / InGrid	FINAL

Inhaltsverzeichnis

1	Einleitung	5
1.1	Java Management Extension	6
1.2	JuXTApose	6
1.3	Apache Lucene	7
1.3.1	Boolean Operators	8
1.3.2	Phrase Search	8
1.3.3	Grouping	8
1.3.4	Fields	8
1.3.5	Term Modifiers	8
1.3.6	Fuzzy Searches	8
1.3.7	Proximity Searches	9
1.3.8	Range Searches	9
1.3.9	Term Boosting	9
1.3.10	Escaping Special Characters	9
1.4	Apache Nutch	9
1.4.1	Plugin System	10
1.4.2	Crawler	10
1.4.3	Parser	10
1.4.4	Web-Datenbank	10
1.4.5	Indizierung	10
2	Zielsetzung	11
2.1	Allgemeine Anforderung an die Suche	11
2.1.1	Suche im zeitlichen Bezug	12
2.1.2	Suche im räumlichen Bezug	12
2.1.3	Ähnliche Begriffe	12
2.1.4	Suche in Attributen	13
3	iPlugs	14
3.1	iPlug-Architektur	14
3.1.1	iPlugInterface	15
3.1.2	Pre-/PostProcessingPipeline	16
3.1.3	Transformationsschicht	16
3.1.4	Index	16
3.1.5	QueryBuilder	16
3.1.6	ServiceAdapter	17
3.1.7	QuellenAdapter	17
3.1.8	Zeitgesteuerte Jobs	18
3.2	Datasource Client-iPlug	18
3.2.1	Administration	20
3.3	SearchEngine-iPlug	21
3.3.1	Administration	21
3.3.2	Dateiformate	23
3.4	UdkDB-iPlug	24
3.4.1	Administration	25

3.5	CswDB-iPlug	25
3.5.1	Administration	26
3.6	UDK-iPlug	27
3.6.1	Administration	27
3.7	CSW-iPlug	28
3.7.1	Administration	29
3.8	g2k-iPlug	29
3.8.1	Administration	30
3.9	SNS-iPlug	30
3.9.1	Verwendung des SNS in Portal-U und InGrid 1.0	31
3.9.2	Administration	33
3.10	Forschungsportal.net-iPlug	33
3.10.1	Administration	34
3.11	Sonderfall Forschungsdatenbanken	35
3.11.1	Suchkonzepte	35
3.11.2	Untersuchung einiger Datenbanken	36
4	Anhang	38
4.1	Abbildungsverzeichnis	38
4.2	Literatur	38

1 Einleitung

Derzeit ist das Umweltinformationsnetz Deutschland gein® (German Environmental Information Network) „das Portal für Umweltfragen“ in Deutschland, welches im Internet verteilte Informationen zahlreicher öffentlicher Einrichtungen erschließt. Es wurde vom Umweltbundesamt (UBA) in den Jahren 1998 bis 2000 aufgebaut und bis Ende 2002 weiterentwickelt. Beim Umweltdatenkatalog (UDK) handelt es sich um das Informationssystem zum Nachweis von Umweltinformationen der öffentlichen Verwaltung in Deutschland. Seine Grundkonzeption entstand in den Jahren 1991 bis 1995 im Niedersächsischen Umweltministerium.

Im Jahre 2003 wurden gein® und UDK organisatorisch zusammengeführt. Beide Systeme sollen nun konzeptionell und technologisch in der Anwendung InGrid 1.0 (Information Grid, ehemaliger Arbeitstitel: gein® 2.0) integriert werden. Mit Hilfe von InGrid 1.0 soll das zukünftige „Portal für Umweltfragen“ Portal-U aufgebaut werden.

Die neue Anwendung soll verschiedene Schnittstellen bedienen können. Hierbei wird zwischen den Anfrage- und Abfrage-Schnittstellen unterschieden. Zu den Anfrage-Schnittstellen gehören die UDK-, CSW- und die Portal-Schnittstelle. Die Abfrage-Schnittstellen werden als iPlugs realisiert. Ein iPlug ist eine kleines in sich geschlossenes Modul, dessen DV-technische Feinkonzeption Thema dieses Dokumentes ist. Im Folgenden sind die iPlugs, die für InGrid 1.0 umgesetzt werden aufgeführt:

- Datasource Client-iPlug (DSC-iPlug)
- SearchEngine-iPlug (SE-iPlug)
- UdkDB-iPlug
- CswDB-iPlug
- UDK-iPlug
- CSW-iPlug
- g2k-iPlug
- SNS-iPlug
- Forschungsportal.net-iPlug (FPN-iPlug)

In dem hier vorliegenden Feinkonzept werden zunächst die verwendeten Open-Source-Frameworks beschrieben; anschließend wird in Kapitel 2 auf die Zielsetzung des iPlug-Konzeptes eingegangen. Im Hauptteil des Konzeptes werden zunächst die einzelnen Komponenten beschrieben, die ein iPlug besitzen kann, um dann in den folgenden Unterkapiteln auf die iPlugs einzugehen. Hierbei wird beschrieben, aus welchen

Komponenten der entsprechende iPlug besteht und wie die technische Realisierung aussieht. Darüber hinaus wird die Administration des einzelnen iPlugs und die Installation erläutert.

1.1 Java Management Extension

Die Java Management Extension (JMX)¹ ist eine Art Mini-Container, in dem kleine Komponenten, sogenannte „managable beans“ (MBeans), betrieben werden können.

JMX ist eine Spezifikation (Java Specification Request) – also ein definierter Standard, welcher durch verschiedene Anbieter und Open-Source-Projekte implementiert wurde und so verschiedenen Produkten mit gleichem Funktionsumfang zur Verfügung steht. Die Java Management Extension bietet eine Reihe an nützlichen Funktionalitäten für dauerhaft (24/7) laufende Softwaresysteme.

So können MBeans während der Laufzeit einzeln ausgetauscht werden, ohne die gesamte Plattform neu starten zu müssen. Dies ist insbesondere für Wartungsarbeiten wichtig.

Darüber hinaus bietet JMX die Möglichkeit, Status und Metadaten von MBeans zu beobachten oder Einstellungen vorzunehmen. Für dieses Remotemanagement bietet JMX eine offene Adapter-Schnittstelle, die durch beliebige Protokolle erreichbar gemacht werden kann. So gibt es neben HTML- oder SOAP-Adapttern auch RMI- oder Simple-Network-Monitor-Protokoll-Adapter (SNMP-Adapter).

Für rund um die Uhr laufende Systeme in verteilten Installationsszenarien sind Monitoring- und Remote-Management-Funktionalitäten unerlässlich.

In der Realisierung des InGrid-Such-Moduls wird eine JMX Implementierung mit dem Namen MX4J² verwendet. Die MX4J-Implementierung zeichnet sich durch eine Open-Source-Lizenz aus und ist mit Version 3.0.1 eine ausgereifte und vielfach erprobte Implementierung der JMX-Spezifikation.

1.2 JuXTApose

JuXTApose (JXTA)³ ist die Java-Peer-to-Peer-Plattform. JXTA ist im Wesentlichen die Implementierung einer durch Sun erstellten Spezifikation für den Austausch von XML-basierten Nachrichten zwischen zwei Rechnern.

JXTA spezifiziert eine Nachrichtensyntax, legt aber die Kommunikation zweier Rechner nicht auf ein spezielles Kommunikationsprotokoll fest, wie es zum Beispiel bei einer Technologie wie JINI der Fall ist.

¹ <http://java.sun.com/products/JavaManagement/>

² <http://www.mx4j.org>

³ <http://www.jxta.org>

Daraus ergibt sich der Vorteil, dass zwei Rechner mittels eines beliebigen Protokolls kommunizieren können, und dennoch durch die definierte Syntax in der Lage sind, die Nachrichten zu verarbeiten.

So kann das http-Protokoll, im Falle dass beide Rechner durch eine Firewall getrennt sind, verwendet werden. Es kann aber auch ein performanteres Protokoll, zum Beispiel in binärer Form via TCP-IP-Sockets, zum Einsatz kommen.

Darüber hinaus bietet JXTA eine Anzahl an Funktionalitäten, die es erlauben, Rechner in einem Netzwerk zu finden, in Gruppen zu organisieren und zwischen den Rechnern zu kommunizieren. Die Rechner verwenden dabei, wie bereits erwähnt, jeweils das beste Kommunikationsprotokoll, das zwischen den beiden Kommunikationspartnern möglich ist.

Gerade in verteilten Systemen, bei denen die Netzwerkstruktur zwischen den Subsystemen nicht klar definiert ist, kann die Kommunikation eine große Hürde sein. So können zum Beispiel einfache Kommunikationswege durch Firewalls blockiert werden oder aber Firewall freundliche Protokolle wie SOAP ein erhebliches Performanceproblem darstellen.

Daher ist es in großen Systemen nötig, die Nachrichten zwischen Kommunikationspartnern intelligent zu transferieren. Dabei werden Nachrichten zwischen zwei Kommunikationspartnern eventuell über Drittsysteme weitergeleitet.

JXTA bietet ein solches ‚intelligentes‘ Routing-System. Dabei ist es möglich, dass die Nachricht zwischen zwei Systemen über ein externes System weitergeleitet wird, welches als Protokollübersetzer fungiert.

JXTA ist unter einer der Apache Lizenz Version 2.0 ähnelnden Open-Source-Lizenz publiziert und wird von Sun Microsystems weiterentwickelt.

Im InGrid-Suchmodul wird zudem ein Open-Source-Tool mit dem Namen weta-G zum Einsatz kommen, das die JXTA Technologien abstrahiert und dem Entwickler ein benutzerfreundliches API zur Verfügung stellt.

1.3 Apache Lucene

Apache Lucene⁴ ist eine java-basierte Bibliothek zum Erstellen von Volltextindizes, die unter der Apache Open-Source-Lizenz publiziert wurde. Lucene hat eine fast 15-jährige Geschichte und liegt aktuell als Version 1.4.3 vor. Lucene hat sich tausendfach bewährt und gehört zu den populärsten Open-Source-Tools überhaupt. Es existieren neben der Java-Implementierung auch eine Anzahl an Portierungen in andere Programmiersprachen.

Die Lucene-Bibliothek wandelt eine Anzahl an Textdokumenten in einen sogenannten ‚inverted Index‘ um. Dabei wird der Text eines Dokumentes in ein Term-Dokument-Wertepaar zerlegt und dieses in einer sehr performanten Datenstruktur abgelegt. Somit ist es möglich, innerhalb sehr kurzer Zugriffszeiten zu ermitteln, in welchen Dokumenten ein bestimmter Term vorkommt.

⁴ <http://lucene.apache.org>

Darüber hinaus bietet Lucene eine große Anzahl weiterer Funktionalitäten. Im Wesentlichen ermöglichen diese einen hoch performanten Zugriff auf einen Index mit Hilfe von Suchanfragen. Lucene ist keine Suchmaschine, sondern bietet ausschließlich Funktionen zur Indizierung von Dokumenten und Auffinden von Dokumenten im Index.

Durch Lucene werden folgende Suchmethodiken unterstützt:

1.3.1 Boolean Operators

Boolesche Operatoren erlauben die Verwendung von logischen Operatoren in Suchanfragen. Somit ist es möglich, Wörter oder Wortgruppen mit UND bzw. ODER zu verknüpfen. Auch ist es möglich Anfragen zu formulieren, die Treffer mit bestimmten Worten ausschließen.

1.3.2 Phrase Search

Neben einzelnen Wörtern kann auch nach dem Vorkommen ganzer Wortgruppen in exakter Übereinstimmung gesucht werden. Dies erfolgt durch Umschließung der Phrase mit Anführungszeichen.

1.3.3 Grouping

Die Grouping-Funktionalität erlaubt es, im Zusammenhang mit booleschen Operatoren komplexe Suchanfragen zu formulieren, die sich durch Grouping hierarchisch gliedern lassen. So werden die durch runde Klammern definierten Gruppen einzeln als zusammenhängende Anfrage interpretiert und ausgeführt. Im Wesentlichen ist dies das gleiche Prinzip, welches beim Verwenden von Klammern in mathematischen Ausdrücken zum Einsatz kommt.

1.3.4 Fields

Ein Dokument – im Kontext von Lucene – kann als Datensatz mit einer Anzahl an Feldern verstanden werden. Um die Gliederung eines Dokumentes in Felder auch in einer Suchanfrage zu unterstützen, wird durch Lucene eine Schlüssel-Wert-Syntax unterstützt. Diese erlaubt es, den Feldnamen eines Dokumentes gefolgt durch einen Doppelpunkt und den gesuchten Wert in einer Suchanfrage anzugeben. Die Suche wird dann auf das angegebene Feld eingeschränkt.

1.3.5 Term Modifiers

Lucene unterstützt sogenannte Wildcard-Zeichen zur Term-Modifikation. Diese Zeichen stehen in Vertretung für Buchstaben oder Buchstabengruppen. Dies kann nützlich sein um nach Wortstämmen oder Worten in verschiedenen Schreibweisen zu suchen. Als Wildcards stehen '*' und '?' zur Verfügung.

1.3.6 Fuzzy Searches

Wenn nach einem Term mit Fuzzy Search gesucht wird, so werden auch alle Dokumente mit ähnlichen Termen (Ausdrücken) gefunden. Dabei lässt sich eine mathematische

Distanz in der Ähnlichkeit angeben. So kann z.B. nach dem Wort *Korn* gesucht werden, es würde aber auch das Wort *Horn* gefunden werden.

1.3.7 Proximity Searches

Lucene erlaubt das Suchen von Dokumenten, in denen Wörter in einer definierten Distanz zueinander stehen. So kann man beispielsweise nach Dokumenten suchen, in denen die Wörter Deponie und Grundwasser nicht weiter als 50 Wörter von einander entfernt sind.

1.3.8 Range Searches

Range Searches bieten die Möglichkeit der Suche nach einem Ausschnitt einer Menge, die durch eine lexikographisch Sortierung definiert ist. Es werden sowohl Text- als auch Zahlenwerte bei Range-Anfragen unterstützt. So kann man von Abfall bis Zweckverband oder von 01012002 bis 10102005 suchen, um beispielsweise die Treffer zeitlich einzuschränken. Letzteres setzt natürlich voraus, dass ein Zeitwert im Index abgelegt wurde.

1.3.9 Term Boosting

In Suchanfragen können einzelne Terme höher gewichtet werden als andere. Damit lässt sich das Ranking beeinflussen. Es werden Dokumente, in denen der höher gewichtete Term eine wichtigere Rolle spielt, weiter oben in die Trefferliste sortiert.

1.3.10 Escaping Special Characters

Verschiedene Zeichen wie das Plus- oder Minus-Symbol werden in einer Suchanfrage als Syntaxzeichen interpretiert. Darüber hinaus ist es jedoch mit sogenannten Escaping Characters möglich, ihre syntaktische Funktion auszuschalten und ebenfalls nach diesen Symbolen zu suchen.

Die Formulierung von Boolean Operators, Phrase Search, Grouping und Fields wird durch den Suchassistenten in der Erweiterten Suche unterstützt.

Term Modifiers, Fuzzy Searches, Proximity Searches, Range Searches, Term Boosting und Escaping Special Characters stehen dem Anwender textbasiert für das Sucheingabefeld zur Verfügung. Eine Erklärung der Syntax kann der Nutzer der Hilfe entnehmen.

1.4 Apache Nutch

Apache Nutch⁵ ist eine Software, die den vollen Funktionsumfang einer Suchmaschine abdeckt. Nutch ist ebenfalls unter einer Apache Open-Source-Lizenz verfügbar. Dabei wird Apache Lucene zur Erstellung und Abfrage des Textindex verwendet.

⁵ <http://incubator.apache.org/nutch/>

In der aktuellen Implementierung ist Nutch eine Anzahl an Command Line Tools, die zusammenarbeiten und Inhalte in definierten Dateiformaten austauschen. So werden die Daten der einzelnen Schritte in Dateien abgelegt, um dann vom nächsten Tool weiter verwendet zu werden.

Nutch lässt sich im Wesentlichen in folgende Module untergliedern:

1.4.1 Plugin System

Das Nutch Pluginsystem erlaubt es, Nutch an verschiedensten Stellen durch Plugins zu erweitern, ohne dass das Kernsystem manipuliert werden muss. Ein Plugin ist ein kleines, in sich geschlossenes Modul, das mit Hilfe einer XML Beschreibungsdatei an Nutch installiert wird.

1.4.2 Crawler

Nutch verfügt über einen Crawler, der in der Lage ist, verschiedene Protokolle zu nutzen. Zum einen kann HTTP verwendet werden, es gibt aber auch ein Plugin für das FTP Protokoll. Der Crawler selbst arbeitet eine Liste an URLs ab, setzt das nötige Protokoll ein, um den Content der URL zu laden und speichert diesen auf der Festplatte ab.

1.4.3 Parser

Auch bei den Parsern, die zur Extraktion von Texten aus Mediaformaten wie HTML oder PDF verwendet werden, kommt das Pluginsystem zum Einsatz. Neben HTML und PDF stehen auch Parser-Plugins für Microsoft-Office-Formate zur Verfügung. Eine genaue Auflistung der im Rahmen des Projektes verfügbaren und zu realisierenden Parser wird im Abschnitt Nutch-Dateiformate gegeben.

1.4.4 Web-Datenbank

Die Webdatenbank ist keine relationale Datenbank. Sie basiert auf einem eigenen proprietären Format, das ausschließlich darauf ausgelegt ist, URLs und deren Beziehungen zueinander abzulegen. Darüber hinaus wird in der Web-Datenbank das Datum des letzten Crawls einer Webseite abgelegt, welches nötig ist, um Seiten regelmäßig neu zu crawlen und zu indizieren.

1.4.5 Indizierung

Wie erwähnt kommt bei Nutch Apache Lucene als Indizierungswerkzeug zum Einsatz. Nachdem die Textinhalte aus den geladenen Inhalten extrahiert wurden, werden diese indiziert. Während des Indiziervorgangs wird bereits eine Art Dokument-Wichtung als Teil des Rankings berechnet. Die Wichtung eines Dokuments wird bei Nutch durch den so genannten PageRank⁶-Algorithmus errechnet.

⁶ <http://citeseer.ist.psu.edu/page98pagerank.html>

2 Zielsetzung

Das iPlug-Feinkonzept beschreibt die einzelnen iPlugs von InGrid und geht auf deren Architektur ein. Die einzelnen iPlugs decken die Abfrage-Schnittstellen sowie das Modul Suchmaschine der Leistungsbeschreibung ab. Die Umsetzung der iPlugs beruht auf einem Konzept, das es ermöglicht, durch Ersetzen einzelner Bausteine aus einem iPlug ein anderes mit anderen Eigenschaften zu erstellen. Bei diesem Konzept spielt der Datasource Client als Grundgerüst für die einzelnen iPlugs eine wesentliche Rolle.

Ziel der iPlugs ist es, die unterschiedlichen Datentöpfe, die über den iBus recherchierbar sein sollen, den Anfrage-Schnittstellen zur Verfügung zu stellen.

Dadurch ist der Anwender in der Lage Informationen zu recherchieren, die organisatorisch zu einem oder mehreren Partnern gehören, in jedem Fall aber bei verschiedenen Anbietern vorliegen. Hierfür können vom Administrator der Installation beliebige, unterschiedlich strukturierte Datenquellen eingebunden und dem Anwender für Recherchen zur Verfügung gestellt werden. Dies beinhaltet Webseiten (inkl. Themenseiten, Aktuelles), Adressen, Fachinformationssysteme, Forschungsdatenbanken und Metadaten (UDK/CSW).

Um ein erweiterbares und skalierbares System aufzubauen, wurden bei InGrid die iPlugs eingeführt. Durch diesen Ansatz ist die Software InGrid 1.0 hoch flexibel und offen für jegliche zukünftige Erweiterungen.

2.1 Allgemeine Anforderung an die Suche

Die Suche ist ein zentrales Element in der Gesamtarchitektur InGrid und den iPlugs. Daher wird im folgendem Absatz genauer auf die allgemeinen Anforderungen an die Suche eingegangen.

Neben der einfachen (auf Schlagworten basierenden) Suche bietet InGrid 1.0 eine Reihe von zusätzlichen Suchfunktionen. Neben der oben erwähnten Einschränkung auf den Suchraum umfasst die detaillierte Suche im Wesentlichen Funktionen, die der genaueren Definition der Suchanfrage dienen.

Dies beinhaltet die Formulierung der Suchanfrage mittels einer erweiterten Syntax (Phrase, Boolean, Klammerung, etc.). Eine weitere Spezifizierung kann durch Anreicherung der Query mittels so genannter key:value-Erweiterungen erfolgen. Durch den Key wird hier das Indexfeld spezifiziert, in dem der angegebene Value enthalten sein muss, um ein Dokument als Treffer zu erkennen. Die Keys sind hierbei vorgegeben und entsprechen in der Regel dem Namen des Indexfeldes. Key:Value Erweiterungen können vom Benutzer direkt textbasiert angegeben oder benutzerfreundlich über Formulare oder andere GUIs (z.B. Karten) generiert werden.

Spezielle Suchfunktionen , deren Syntax laut Leistungsbeschreibung nicht in dem Formular "Erweiterte Suche" generiert werden muss, die aber durch den Funktionsumfang von Lucene / Nutch zur Verfügung stehen (z.B. Fuzzy Search) werden dem versierten Nutzer durch die Online-Hilfe vermittelt.

Um dem Benutzer eine komfortablere Suche anzubieten, wird der ‚Semantic Network Service‘ des Umweltbundesamtes in die Anwendung integriert. Die Integration findet auf verschiedenen inhaltlichen Ebenen statt und wird technisch als iPlug realisiert, welches den SNS mit Hilfe des SNS-Webservice anspricht. Aus Performance-Gründen ist geplant, dass der Betreiber den Webservice eventuell erweitert, um mehrere Anfragen in einer Anfrage abbilden zu können.

Die für die inhaltliche Integration nötigen Daten sind zum größten Teil schon heute im SNS verfügbar, eventuell fehlende Daten wie beispielsweise Bounding Boxes werden sukzessiv hinzugefügt, jedoch in der technischen Realisierung schon heute beachtet.

Die Spezifizierungen zur genaueren Suchanfragenformulierung beinhalten:

2.1.1 Suche im zeitlichen Bezug

Der Nutzer kann die Trefferliste auf Ergebnisse einschränken, denen ein zeitlicher Bezug zugeordnet wurde. Hierbei werden sowohl Zeitpunkte als auch Zeitspannen unterstützt. Weiterhin kann der Anwender zwischen verschiedenen Modi wie "innerhalb", "Einschluss" und "Überschneidung" des Zeitbezugs wählen.

Zur Generierung dieser Metadaten im Index wird die Funktionalität des SNS genutzt.

2.1.2 Suche im räumlichen Bezug

Der Nutzer kann die Trefferliste auf Ergebnisse einschränken, denen ein räumlicher Bezug zugeordnet wurde. Auch hier wird der Index mit Metadaten (Koordinaten, Gemeindekennzahlen und geografischen Begriffen) angereichert, die aus dem SNS bezogen werden. Außer durch direkte Angabe von Koordinaten kann der Nutzer den Raumbezug mittels Termen spezifizieren. Eine Übersetzung in Koordinaten erfolgt hier ebenfalls durch den SNS. Als dritte Möglichkeiten steht dem Nutzer die Verwendung von kartenbasierten Services zur Verfügung, die die Query mit Koordinaten oder je nach Suchmodi mit administrative Einheiten anreichern.

2.1.3 Ähnliche Begriffe

Jede Query kann durch den SNS analysiert werden, um zusammen mit der Ergebnisliste "ähnliche Begriffe"⁷ zu präsentieren. Der Nutzer kann die Suche unter Einbindung von "ähnlichen Begriffen" wiederholen, um das Suchfeld entsprechend auszuweiten.

⁷ Unter „ähnliche Begriffe“ werden Thesaurus-Deskriptoren verstanden, die vom SNS automatisch zu den einzelnen Suchbegriffen der Anfrage generiert wurden.

2.1.4 Suche in Attributen

Es ist möglich, nur in definierten Feldern der Indizes zu suchen. So kann z.B. eine Suchanfrage auf das Feld Nachname in einem Adress-Datensatz beschränkt werden.

Weiterhin unterstützt die InGrid-Suche eine Reduzierung der Begriffe auf den Wortstamm (Stemming), um Abarten der Begriffe in die Suche einzuschließen. Der Anwender kann außerdem wählen, ob nur ganze Worte oder auch Teilzeichenketten in den Suchresultaten zulässig sind. Eine Erkennung der Sprache der Daten bereits während der Indizierung ermöglicht es, die Suche auf eine bestimmte Sprache einzuschränken.

Die Ergebnisse aus unterschiedlichen Datenquellen werden, soweit möglich, in einer gemischten und neugerankten Trefferliste präsentiert. Das Ranking kann sowohl seitens des Administrators als auch des Anwenders beeinflusst werden. Dem Nutzer stehen zusätzlich Funktionen wie Sortierung (Aktualität) und Gruppierung (Partner) zur Beeinflussung der Ergebnisliste zur Verfügung.

3 iPlugs

Im folgenden Kapitel werden zunächst die möglichen Komponenten eines iPlug beschrieben, um dann auf die einzelnen iPlugs, die für InGrid 1.0 erstellt werden, näher einzugehen.

3.1 iPlug-Architektur

Um ein erweiterbares, flexibles und damit auch skalierbares System zu erlangen, das die Ansprüche von InGrid erfüllt, muss das System in einzelne Komponenten zerlegt werden. Besonders durch die hohen Anforderungen an Rankingqualität, Geschwindigkeit und der verteilten Struktur kommt nur eine Architektur der horizontalen Skalierung in Frage. Beim Ansatz der horizontalen Skalierung werden die Arbeitsaufgaben in Teilaufgaben geteilt und diese parallel bearbeitet. Diese Anforderungen führen zum iPlug-Konzept, das im Bereich der Suchmaschinen bereits seit längerem erfolgreich eingesetzt wird.

Durch die klar definierte Schnittstelle des iBus ist es so möglich, jederzeit weitere unterschiedliche iPlugs in InGrid einzusetzen. Durch die Nutzung von Peer-to-Peer Technologie für die Kommunikation zwischen den iPlugs ist es möglich, die iPlugs direkt beim Dateninhaber einzurichten, soweit dies die Gegebenheiten der Infrastruktur zulassen⁸. Dadurch kann alternativ zu einer weitgehend zentralen Installation auch ein voll verteiltes System erzeugt werden. Dieser Ansatz hat den Vorteil, dass die angestrebte Performance erfüllt werden kann. Hinzu kommt, dass die technische Kaskadierung von mehreren hintereinander geschalteten iBus-Installationen nicht mehr notwendig ist.

⁸ Derzeit ist der dezentrale Ansatz in erster Linie für das Datasource Client iPlug (DSC-iPlug) sowie für das UdkDb-iPlug geplant

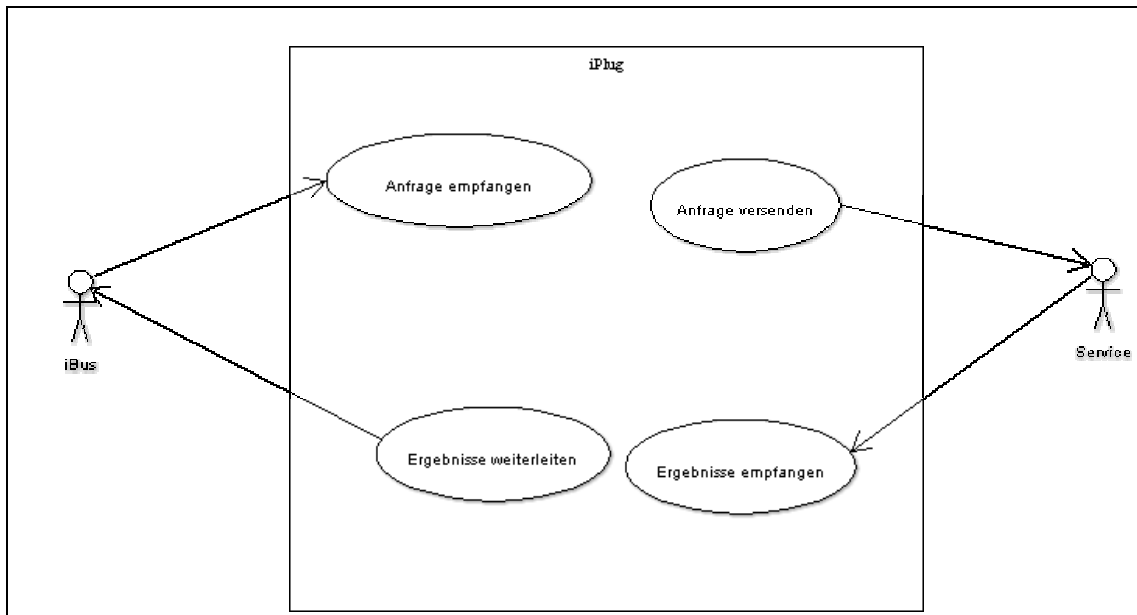


Abbildung 1: Anwendungsfälle eines iPlugs

Abbildung 1 stellt die Anwendungsfälle eines iPlugs mit den dazugehörigen Akteuren dar. Auf der linken Seite befindet sich der Akteur –„iBus“, er übergibt dem iPlug die Anfrage. Ein weiterer Anwendungsfall ist die Weiterleitung der Ergebnisse an den iBus.

Das Gegenstück ist rechts mit dem Akteur-Service dargestellt. Dieser empfängt vom iPlug die Anfrage und sendet die Ergebnismenge wieder zurück.

Jeder iPlug setzt ebenfalls wie der iBus auf JMX auf. Außerdem werden die iPlugs in kleine logische Komponenten untergliedert. Dies ermöglicht, einzelne Bausteine für andere iPlugs wieder zu verwenden oder diese zu einem späteren Zeitpunkt durch neuere Technologien zu ersetzen. Der Austausch von einzelnen Komponenten ist ein wichtiger Aspekt, z.B. in Hinsicht auf die Weiterentwicklung der eingesetzten Standards (OGC/ISO).

Im Folgenden werden die einzelnen möglichen Bausteine eines iPlugs beschrieben. Diese Komponenten werden in Kapitel 3.2 bis 3.9 verwendet, um die entsprechenden iPlugs zu beschreiben.

3.1.1 iPlugInterface

Das iPlugInterface ist das Gegenstück zum iPlugConnector im iBus. Über die Schnittstellen iPlugInterface und iPlugConnector findet die Kommunikation mit den beiden Komponenten statt. Die Kommunikation selbst findet über JXTA-Sockets statt.

Das iPlugInterface nimmt die Anfragen des iBus entgegen und übergibt diesem die Trefferliste als Objekt-Array verpackt in einem Datatransfer-Objekt⁹ zurück.

⁹ <http://martinfowler.com/eaCatalog/dataTransferObject.html>

3.1.2 Pre-/PostProcessingPipeline

Pipelines sind ein generelles Konzept, um Daten in einer Anzahl an Schritten vor und/oder nachzubearbeiten. Es gibt verschiedene Prozessoren, die an unterschiedlichsten Stellen zum Einsatz kommen. So können Daten, bevor diese in einen Index geschrieben werden, um Metadaten ergänzt werden oder Daten, nachdem diese durch eine Suchanfrage aus dem Index gelesen wurden, in ein anderes Format umgerechnet werden.

Daher gibt es zwei Pipelines, eine um Daten, die in den Index geschrieben werden, zu bearbeiten. Hier werden die Daten pre-processed – also bearbeitet, bevor sie in den Index geschrieben werden. Die Postprocessing Pipeline kommt zum Einsatz um Daten, die aus dem Index gelesen wurden, nachzubearbeiten.

Auch hier ist ein Prozessor die Implementierung eines Java-Interfaces. Eine beliebige Anzahl an Prozessoren kann in Reihe geschaltet werden.

Während der Indizierung durch das SearchEngine-iPlug wird z.B. der SNS genutzt, um zusätzliche Metadaten zu indizieren. Neben der durch Verschlagwortung erhaltenen Begriffe sind das ebenfalls Raum- und Zeitbezüge. Die gewonnenen Metadaten werden pro Metadattentyp in separate Indexfelder überführt.

Ein anderes Beispiel ist die Indizierung von Metadatenbanken, bei der spezielle Metadaten wie Raum- und Zeitbezüge in separate Indexfelder überführt werden.

3.1.3 Transformationsschicht

Die Transformationsschicht analysiert die übergebene Anfrage und setzt je nach iPlug die Logik der zu verwendenden Spezifikation um. Da InGrid unterschiedliche Dienste ansprechen muss, wird es für jede Schnittstelle eine eigene Implementierung geben. Dies sind folgende drei Schnittstellen:

- UDK 5.0 SOAP-Schnittstelle
- CSW 2.0
- g2k

3.1.4 Index

Die aus der Quelle geladenen Daten werden nach Verarbeitung in der Prozessor-Pipeline mit Hilfe des Index in eine hoch performante Datenstruktur umgewandelt und abgelegt. Als Bibliothek zur Indizierung wird das oben vorgestellte Apache Lucene verwendet. Während des Indizierens werden die durch den Administrator vorgegebenen Gewichtungen zur Vorberechnung eines Ranking-Wertes verwendet.

3.1.5 QueryBuilder

Der QueryBuilder erzeugt unter Berücksichtigung der Spezifikationen (UDK, CSW, g2k) die korrekte Suchanfrage. Diese werden dem ServiceAdapter oder dem QuellenAdapter zur Verfügung gestellt um so den entsprechenden Adapter ansprechen zu können.

3.1.6 ServiceAdapter

Der ServiceAdapter ist wiederum die Schnittstelle zu den externen Diensten. Über diese Komponente verläuft die komplette Kommunikation zwischen iPlug und Service (z.B. CSW). Der ServiceAdapter versendet die erstellte Anfrage des QueryBuilder an den Dienst und empfängt entsprechend die Ergebnisliste. Diese Ergebnisse werden an das iPlugInterface weitergereicht. Der ServiceAdapter kommuniziert über HTTP/SOAP. Hierbei wird es für folgende drei XML-Profile eine eigene Implementierung geben:

- UDK 5.0 SOAP-Schnittstelle
- CSW 2.0 und Application Profiles
- G2k

3.1.7 QuellenAdapter

Über die Datenquellen-Adapter werden die Daten aus der Quelle geladen. Der QuellenAdapter muss mit dem von der Datenquelle vorgegebenen Protokoll interagieren können. Daher ist diese Schicht leicht auswechselbar. Es wird verschiedene Implementierungen geben: JDBC, Hibernate, Tamino (XQuery) und Nutch.

JDBC

Über das Kommunikationsprotokoll JDBC (Java Database Connectivity) wird es möglich sein, alle SQL fähigen Datenbanken über den Data-Source-Client anzuschließen.

Hibernate

Hibernate¹⁰ ist ein Open-Source-Persistenz-Tool, welches basierend auf XML-Konfigurationsdateien, so genannten Mappings, das Bindeglied zwischen JavaBeans und einer Datenbank darstellt. Derzeit werden von Hibernate 16 unterschiedliche Datenbanken unterstützt, dazu zählen Oracle, DB2, MySQL, MS SQL Server sowie PostgreSQL. Mit Hibernate wird das InGrid-Metadatenchema abgebildet. Diese Funktionalität wird auch für die Import-Schnittstelle des iBus verwendet.

Tamino (XQuery)

Eine weitere zu implementierende Kommunikationsschicht ist der Anschluss an Tamino¹¹. Tamino ist eine XML Datenbank, die anstelle von SQL eine andere Abfragesyntax (XQuery) unterstützt. Der Anschluss an den Server findet über die Tamino-API für Java statt.

Nutch

Nutch extrahiert die Inhalte von Webseiten für den Index mittels Parser-iPlugs. Derartige iPlugs sind für verschiedene Dateiformate (siehe 3.3.2) bereits im Lieferumfang der Distribution enthalten.

¹⁰ <http://www.hibernate.org/>

¹¹ <http://www1.softwareag.com/de/products/tamino/default.asp>

3.1.8 Zeitgesteuerte Jobs

Sowohl für angeschlossene Datenbanken als auch für zu indizierende Webseiten ist eine regelmäßige Aktualisierung des Index notwendig. Neben dem manuellen Start ist eine automatische, zeitgesteuerte Indizierung sinnvoll. Dazu wird ein Job-Scheduling-System benötigt, wobei die Komponente Quartz zum Einsatz kommt.

Quartz

Quartz¹² ist ein Open-Source-Job-Scheduling-System für J2SE und J2EE Anwendungen. Dabei speichert Quartz die Jobs im RAM oder Datenbanken. Einsatzmöglichkeiten wäre das Verwalten eines Systems oder das Steuern eines Workflows.

Mit Quartz können die Zeitabstände der Neuindizierung sehr flexibel gesteuert werden. So ist es z.B. möglich, einen Prozess täglich außer Freitags auszuführen oder in einem Zeitabstand X zu überprüfen, ob eine Bedingung erfüllt ist, die eine Neuindizierung zur Folge hat.

3.2 Datasource Client-iPlug

Den Datasource Client (DSC) gibt es im Gesamtsystem in gleicher Zahl wie es angeschlossene Datenquellen gibt, die in ein gemeinsames Ranking eingebunden sind. Hierbei ist der Datasource Client als Grundgerüst für alle vorhandenen iPlugs zu sehen, d.h. alle vorhandenen iPlugs für InGrid bauen auf dem Datasource Client auf und sind somit als Spezialisierung des Datasource Clients anzusehen. Dabei kann der DSC direkt als Adapter an der Datenquelle betrieben werden, aber auch zentralisiert und nur als Protokollübersetzer fungieren.

Der DSC übersetzt im Wesentlichen das Protokoll der Datenquelle (z.B. JDBC) in das einheitliche und performante Protokoll des InGrid Such-Moduls. Darüber hinaus stellt der DSC die für ein gemeinsames Ranking notwendigen Funktionen zur Verfügung. So werden Daten der Datenquelle in ein Format umgewandelt und vorgehalten, welches in der Lage ist, auch unter einer Last von mehreren hundert Anfragen pro Minute, diese jeweils innerhalb von wenigen Millisekunden zu beantworten.

Als Richtgröße der vorzuhaltenden freien Speicherkapazität für den Index kann die Größe der zu indizierenden Daten in Textform angenommen werden. Je nach Anzahl zusätzlicher Metadatenfelder kann der Index aber auch z.T. erheblich größer werden als die indizierte Textmenge. Letzteres kommt insbesondere zum Tragen, wenn SNS zur Generierung von Metadaten eingebunden wird (gilt nur für das SE-iPlug, siehe unten) .

Darüber hinaus stellt der DSC administrative Funktionen zur Verfügung, die es dem Administrator erlauben, das Ranking der Datenquelleninhalte zu beeinflussen.

¹² <http://www.opensymphony.com/quartz/>

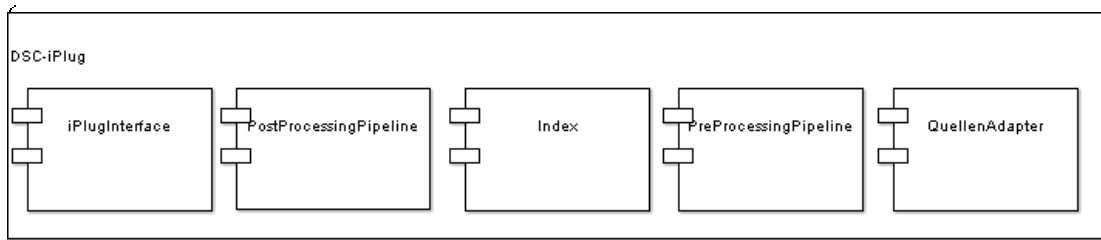


Abbildung 2: Komponentendiagramm des DSC-iPlug

In Abbildung 2 werden die einzelnen Komponenten des DSC-iPlug dargestellt. Hierzu gehört die Kommunikationsschicht zum iBus, das iPlugInterface, der Index, die Pre-/PostProcessingPipeline und der QuellenAdapter in den einzelnen Varianten, die im Folgenden aufgeführt werden:

JDBC

Die JDBC Parameter, die zum Verbindungsaufbau nötig sind, werden während der Installationsroutine abgefragt und sind auch später noch in einem Administrations-Interface änderbar. Das Mapping der Datenquellenstruktur zur planaren Indexstruktur kann durch den Administrator vorgenommen werden und bietet neben einfachem Mapping auch Möglichkeiten zum Einsatz von SQL Queries für den professionellen Administrator.

Die Daten werden nach dem Setup der Datenbankverbindung zeitgesteuert von der Datenbank ausgelesen und in einen hochperformanten Index geschrieben.

Felder einer Datenbank können als indizierter Text oder aber auch als eine Art Schlüssel in den Index überführt werden. Vollindizierte Felder können nach Wörtern, Wortgruppen etc. durchsucht werden. Inhalte von Feldern, die als Schlüssel abgelegt werden, sind jedoch nur durch Angabe des kompletten Schlüsselwertes auffindbar.

Beispiel: Während eine Suche nach „Umwelt Recht“ auf den Volltext Index alle Dokumente findet, in denen der Suchbegriff enthalten ist, führt eine Suche auf das Metadatenfeld „Thema“ nur zu einem Ergebnis, wenn als Thema exakt „Umwelt Recht“ hinterlegt ist. Eine Suche nach z.B. nur „Recht“ auf das Feld „Thema“ würde nicht als Treffer gewertet werden.

Tamino

Eine weitere zu implementierende Kommunikationsschicht für den Datasource Client ist der Anschluss an Tamino. Tamino ist eine XML-Datenbank, die statt SQL andere Abfragesyntax wie XQuery unterstützt.

Der Anschluss an den Datasource Client findet über die Tamino API für Java statt.

Die zu indizierenden Daten werden durch eine während der Installation angegebene und später änderbare XQuery ausgewählt und werden in definierten Zeitabständen im Index abgelegt.

Ein Mapping der in Tamino abgelegten XML-Dokumente kann zum Einen durch definierte XPath realisiert werden, alternativ kann ein XSL-Template angegeben werden, welches das XML-Dokument in eine für den Index verarbeitbare XML-Struktur bringt.

Auch bei der Tamino Anbindung können XML-Attribute oder Node Values, ähnlich Datenbankfeldern, voll indiziert oder als Schlüssel abgelegt werden.

Die von Tamino bezogenen Dokumente werden in ein planares Format überführt und in der erforderlichen Indexstruktur abgelegt.

3.2.1 Administration

Die Administrationsoberfläche wird beim Datasource Client als Webinterface bei der Installation mit eingerichtet. Diese wird durch einen in JMX integrierten „Mini“ Webcontainer zur Verfügung gestellt.

Die Administrationsoberfläche wird bei allen iPlugs sowohl zentral (im Portal) als auch dezentral (beim iPlug) zugreifbar sein. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber
 - Name der Behörde
 - Ansprechpartner (Name, Telefon, Email)
 - ID des Anbieters
- Festlegen des Mutter-iBus (siehe Feinkonzept iBus – Peer-to-Peer und InGrid)
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- Angaben zur Datenbankverbindung
 - Treiber
 - Login/Passwort
 - IP:Port
- Art der Daten, die der Dienst zurückliefert (Link, Linkliste oder Metadaten)¹³
- Pfadangaben für den Index
- Zeitsteuerung der Indexerstellung
- Mapping der Daten

¹³ Es werden drei Formen der Rückgabe unterstützt:

1. Rückgabe eines Links auf eine vom externen System zur Verfügung gestellte Ergebnisliste einschließlich der Angabe der gefundenen Treffer
2. Rückgabe einer Linkliste, wobei jeder Link auf jeweils eine Detaildarstellung des Treffers im externen System verweist
3. Rückgabe einer Liste von Treffern, wobei jeder Treffer auf die interne über InGrid definierte Detaildarstellung der UDK-Metadaten verweist.

- Rankingfaktor (einstellen der Gewichtung von einzelnen Feldern/Termen)

3.3 SearchEngine-iPlug

Apache Nutch wird vollständig in den Datasource Client integriert und ist daher ein besonderer Fall des Datasource Clients. Im Gegensatz zu den anderen Typen des Datasource Clients kommen im Nutch zusätzliche Funktionalitäten wie Crawling und Webseiten-Parsing zum tragen, da pure Textdaten aus den Quellen extrahiert werden. Außerdem wird der Index über den SNS mit Metadaten für Thema, Raum- und Zeit angereichert.

Die Kommunikation zum iBus wird über die Module des Datasource Clients realisiert. Die Anfragen werden jedoch dann an Nutch weitergeleitet und beantwortet. Darüber hinaus wird das Administrationsinterface des DSC um die spezifischen Funktionen von Nutch erweitert.

Die Integration von Nutch in den DSC geschieht durch das gemeinsame Deployment der Komponenten des Datasource Clients und Nutch als JMX MBeans in einem gemeinsamen JMX Container. Dies bedingt die Portierung aller Nutch Tools als MBeans. Die Kommunikation erfolgt über die von JMX zur Verfügung gestellten Tools.

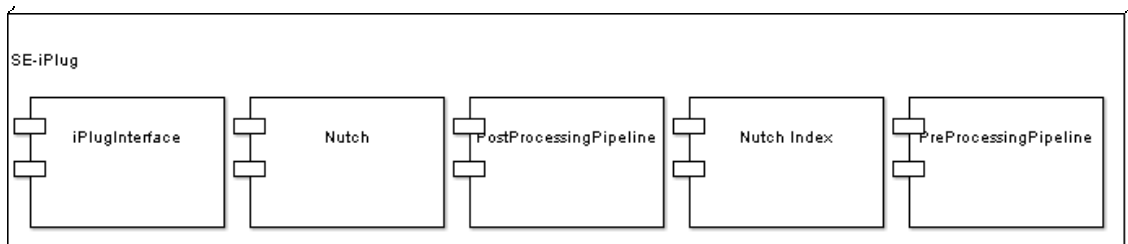


Abbildung 3: Komponentendiagramm des SE-iPlug

In Abbildung 3 werden die einzelnen Komponenten des SE-iPlugs aufgezeigt, hierbei handelt es sich um das iPlug-Interface und die zwei Nutch-spezifischen Komponenten Nutch und den Nutch Index. Im Kontext des Datasource Clients tritt Nutch als Blackbox auf und verhält sich wie ein normaler Index. Wie auch bei den ‚normalen‘ Datasource Clients ist eine Postprozessing Pipeline vor dem Nutchindex vorhanden die Daten beispielsweise umrechnet und eine Preprozessingpipeline nach dem Nutchindex die Daten die in den Index geschrieben werden beispielsweise mit Metadaten aus dem SNS anreichert.

3.3.1 Administration

Nutch besteht in seiner aktuellen Version 0.7 aus einer Anzahl von Command-Line Tools, die nicht über ein Web Interface gestartet, gestoppt, oder per Monitoring betrachtet werden können. Um eine derartige Administration zu ermöglichen, wird Nutch wie bereits erwähnt auf einen JMX-Sockel portiert. Die im Folgenden beschriebenen Funktionen werden in einem webbasiertem Administrationsbereich zur Verfügung gestellt.

Zustandsmonitor

Der Administrator kann einsehen, welche Teilprozesse der Web-Suchmaschine gerade aktiv sind (Generierung neuer Index-Segmente, Generierung von Link-Listen zum Crawlen, Crawlen der Webseiten, Schreiben von extrahierten Links in die Nutch-Web-DB, Entfernen von Link-Duplikaten aus der Web-DB, Indizierung). Ebenfalls ist es möglich, einzelne Teilprozesse manuell zu starten oder zu stoppen.

URL-Tester

Aus verschiedenen Gründen kann der Crawling-Prozess fehlschlagen. Das sind z.B. die Nicht-Verfügbarkeit des Servers, zu viele Anfragen pro Sekunde, die Nicht-Zulassung des Crawlers durch robot.txt, etc. Mit Hilfe eines URL Testers kann überprüft werden, ob nach dem Crawlen und Indizieren einer URL sinnvolle Ergebnisse vorliegen.

Start-URLs, URL-Filter

Der Crawler benötigt einen Einstiegspunkt für seine Tätigkeit im Web. Dieser kann durch eine beliebige Anzahl von Start-URLs definiert werden. Aus gecrawlten Seiten werden alle Links extrahiert, die dann im nächsten Crawling-Prozess weiterverfolgt werden. Um dem Administrator eine genaue Steuerung der weiteren Crawling-Prozesse zu ermöglichen ist es notwendig, bestimmte URL-Muster definieren zu können, die entweder gecrawlt (positive URL-Filter) oder nicht gecrawlt (negative URL-Filter) werden sollen. Diese Muster können durch beliebige reguläre Ausdrücke (regular expressions) angegeben werden. So ist es z.B. möglich, das Crawling auf einzelne Domains (z.B. <http://www.xyz.de/>*) zu begrenzen. Ebenso können Content-Typen definiert werden die gecrawlt werden sollen, z.B. nur *.html, *.htm und *.pdf oder alles außer *.exe und *.swf. Durch den Funktionsumfang der regulären Ausdrücke ist über genannte Beispiele hinaus eine wesentlich feinere Granulierung möglich.

Anzahl CrawlerThreads

Legt die Anzahl von Threads fest, die der Crawler benutzt, d.h. wie viele URLs gleichzeitig gecrawlt werden können.

Anzahl Crawler Threads per Host

Legt die Anzahl von Threads fest, die sich gleichzeitig mit einem Host verbinden dürfen.

Anzahl erneuter Anfragen

Legt die Anzahl fest, wie oft der Crawler im gleichen Indexlauf eine Anfrage nach einer URL wiederholt, wenn Fehler aufgetreten sind.

Abstand zwischen verschiedenen Anfragen an einen Server

Einige Webserver lassen nur eine begrenzte Anzahl von Anfragen pro IP und Zeiteinheit zu. Der Zeitabstand von Anfragen kann angegeben werden.

Maximale Dateigröße

Maximalgröße der herunterzuladenden Inhalten einer Website. Beispielsweise kann so die (unverhältnismäßig zeitaufwändige) Indizierung von mehreren Megabyte großen PDFs vermieden werden.

Anzahl von Umleitungen

Maximale Anzahl von Umleitungen (redirects) einer Anfrage, die verfolgt werden.

Proxy

Verwendung eines Proxy-Servers statt einer direkten Verbindung zum Internet.

Indizierung zusätzlicher Felder

Die Struktur des Webseiten-Index kann hier erweitert werden.

Zusätzlich zu den Standard-Index-Feldern (Title, Content, Anchor, URL) ist das Indizieren von weiteren Metadaten möglich. Durch Nutch werden hier bereitgestellt: Content-Typ, Datum, Host und die Sprache des Inhalts.

Zusätzlich muss zu einer Themenseite auch das jeweilige Thema im Index hinterlegt werden können.

3.3.2 Dateiformate

Im Folgenden werden die von Nutch unterstützten Dateiformate aufgeführt.

parse-html

Mit Hilfe des "parse-html"-iPlugs können HTML-Dokumente geparkt werden. Dabei wird die Java-Bibliothek NekoHtml genutzt, welche HTML-Dokumente einliest und wenn nötig in "sauberes" XHTML transformiert. NekoHtml ist in der Version 0.9.4 erhältlich und steht unter der "CyberNeko Software License, Version 1.0".

(Hinweis: Der Crawler crawlt HTML-Dokumente, wobei es nicht von Bedeutung ist, ob dies *.jsp, *.php oder *.asp Seiten sind.)

parse-msword

Das iPlug "parse-msword" ist in der Lage, Microsoft-Word-Dokumente in den Versionen 97-2002 zu parsen. Auch Tests mit Word 2003 Dokumenten verliefen erfolgreich. Die Java-Bibliothek POI¹⁴ stellt die dazu nötigen Funktionalitäten zur Verfügung. Weitere Funktionalitäten sind das Lesen von Microsoft-Excel-Dokumenten in den Versionen 97-2002 als auch von OLE2-Objekten. POI ist in der Version 2.5.1 erhältlich und steht unter der Apache License.

parse-mspowerpoint

Mit "parse-mspowerpoint"-iPlug ist es möglich, Microsoft-Powerpoint-Dokumente zu analysieren. Wie bei dem "parse-msword"-iPlug, wird die Java-Bibliothek POI verwendet.

(Hinweis: Dieses iPlug ist noch nicht in der Nutch-Distribution enthalten, steht aber bereits zum Download zur Verfügung.)

<http://jakarta.apache.org/poi/>

parse-pdf

Dieses iPlug parst mit Hilfe der Java-Bibliothek PDFBox PDF-Dateien. Ebenfalls wird das Verschlüsseln und Entschlüsseln von Dokumenten unterstützt. PDFBox ist in der Version 0.7.1 erhältlich und steht unter der "BSD License" zur Verfügung.

parse-text

Ermöglicht das Extrahieren von ASCII-Text aus Textdateien.

parse-rtf

Das iPlug "parse-rtf" extrahiert Text aus Rich-Text-Format Dokumenten.

3.4 UdkDB-iPlug

Der UdkDB-iPlug ist die Schnittstelle zu den in der InGrid-Metadatenbank abgelegten (z.B. aus dem Windows-UDK 5.0 importierten) Metadaten. Der UdkDB-iPlug erfüllt die Kriterien der aktuellen UDK 5.0 SOAP-Schnittstellen-Spezifikation.

Die Architektur des UdkDB-iPlugs basiert auf der des Datasource Clients, jedoch kommen noch weitere Komponenten hinzu. Diese dienen zur Abfrage der UDK Besonderheiten. Außerdem wird der QuellenAdapter basierend auf Hibernate eingesetzt, den die InGrid-Metadatenbank als Abstraktions-Layer bereitstellt.

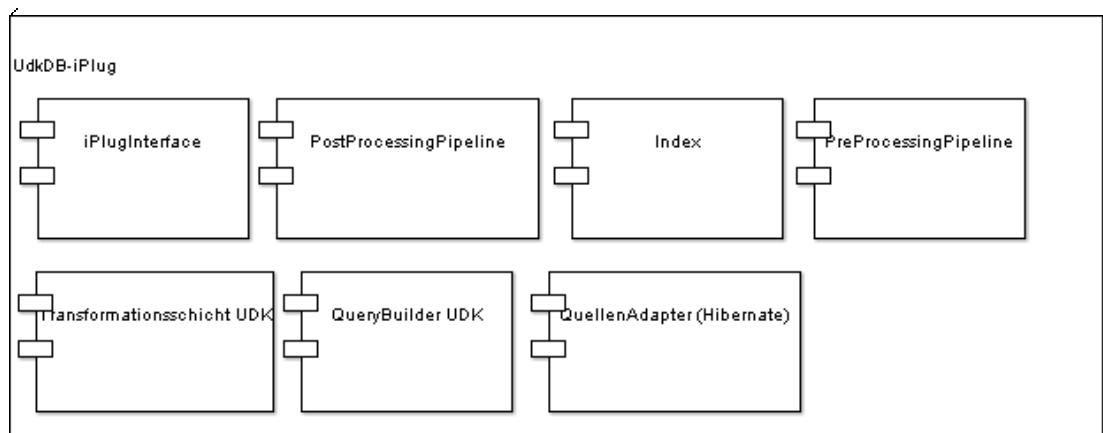


Abbildung 4: Komponentendiagramm des UdkDB-iPlug

In Abbildung 4 sind die einzelnen Komponenten des UdkDB-iPlugs dargestellt. Hierbei handelt es sich um das iPlugInterface, die postProcessingPipeline, den Index, die Transformationsschicht (UDK), den QueryBuilder (UDK) und den QuellenAdapter basierend auf Hibernate. Der funktionale Ablauf des UdkDB-iPlug ist identisch mit dem des DSC-iPlug, jedoch wird dieser um zwei spezifische Bausteine erweitert, die die Anforderungen an die UDK Spezifikation umsetzen.

Bei der Indizierung und der Generierung der Detailansicht wird der UdkDB-iPlug insbesondere die UDK-internen Steuerungsflags zur Sichtbarkeit von UDK-Objekten im Internet oder Intranet (UDK-Feld „Veröffentlichung“), zur Anzeige von Links im Internet

und/oder Intranet (UDK-Feld „URL-Typ“) und den Delete-Flag (logisch gelöschter Datensatz) berücksichtigt.

3.4.1 Administration

Die Administrationsoberfläche wird beim UdkDB-iPlug als Webinterface bei der Installation mit eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber
 - Name der Behörde
 - Ansprechpartner (Name, Telefon, Email)
 - ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- Angaben zur Datenbankverbindung
 - Treiber
 - Login/Passwort
 - IP:Port
- Pfadangaben für den Index
- Zeitsteuerung der Indexerstellung
- Rankingfaktor (einstellen der Gewichtung von einzelnen Feldern/Termen)

3.5 CswDB-iPlug

Der CswDB-iPlug stellt ebenso wie der UdkDB-iPlug die Schnittstelle zur InGrid Metadatenbank dar. Allerdings werden bei den Anfragen, die über den iBus an den iPlug weitergeleitet werden, die OGC-Spezifikationen berücksichtigt. Hierbei handelt es sich um die CSW 2.0 Spezifikation bzw. das darauf aufbauende Application Profile¹⁵. Welche Kriterien der CswDB-iPlug erfüllt, wird in dem Feinkonzept zum Thema Schnittstellen beschrieben.

Der CswDB-iPlug wird nur bei Anfragen verwendet, die von der iBus-CSW-Schnittstelle kommen. Alle anderen Anfragen vom Portal oder von der UDK-Schnittstelle des iBus werden über den UdkDB-iPlug beantwortet.

¹⁵ https://portal.opengeospatial.org/files/?artifact_id=8305

Auch beim CswDB-iPlug dient der Datasource Client als Grundlage. Er erweitert ihn um die Komponenten Transformationsschicht (CSW), QueryBuilder (CSW) und QuellenAdapter (Hibernate). Die einzelnen Komponenten sind in Abbildung 5 dargestellt. Der QuellenAdapter (Hibernate) ist identisch mit dem des UdkDB-iPlug, da das gleiche Datenbankschema zum Einsatz kommt.

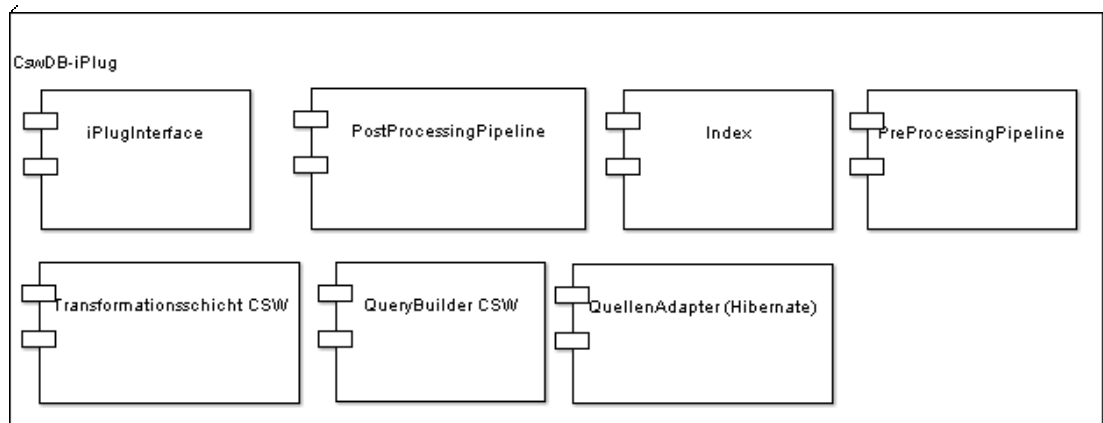


Abbildung 5: Komponentendiagramm des CswDB-iPlug

In Abbildung 5 sind die einzelnen Komponenten des CswDB-iPlugs dargestellt. Hierbei handelt es sich um das iPlugInterface, die postProcessingPipeline, den Index, die perProcessingPipeline, die Transformationsschicht (CSW), den QueryBuilder (CSW) und den QuellenAdapter basierend auf Hibernate. Der Workflow des iPlugs ist identisch mit dem des Datasource Clients, jedoch kommen zwei spezifische Komponenten hinzu, die die Richtlinien und Logik des Catalogue Service Web umsetzen.

3.5.1 Administration

Die Administrationsoberfläche wird beim CswDB-iPlug als Webinterface bei der Installation mit eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber
 - Name der Behörde
 - Ansprechpartner (Name, Telefon, Email)
 - ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- Angaben zur Datenbankverbindung
 - Treiber

- Login/Passwort
- IP:Port
- Pfadangaben für den Index
- Zeitsteuerung der Indexerstellung

3.6 UDK-iPlug

Der UDK-iPlug dient zur Anbindung externer Meta-Informationssysteme, die die UDK 5.0 SOAP-Schnittstellen-Spezifikation unterstützen. Diese Schnittstelle bietet kein Ranking an, da kein „direkter“ Zugriff auf die Daten möglich ist um diese zu indizieren. Der UDK-iPlug versendet die Anfrage an die vorhandene SOAP-Schnittstelle des Datenanbieters. Die Ergebnisse werden entsprechend der SOAP-Schnittstellen Spezifikation an den iPlug zurückgeliefert.

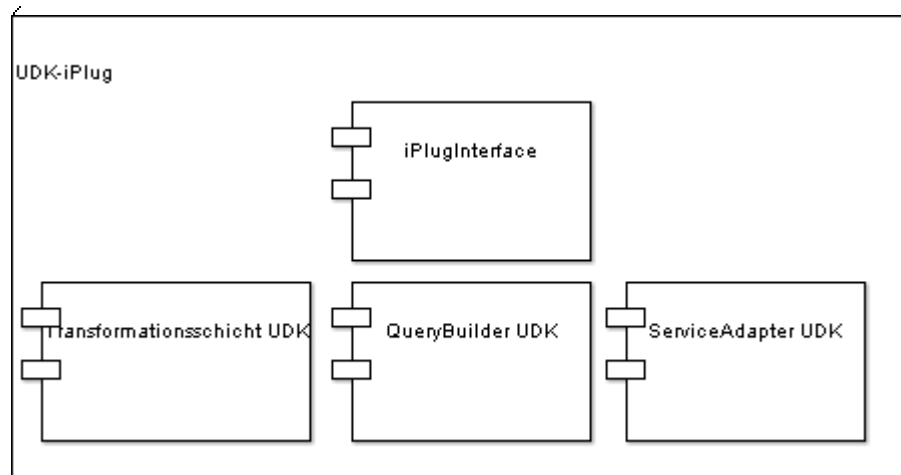


Abbildung 6: Komponentendiagramm des UDK-iPlug

Bei diesem iPlug entfällt die Komponente Index. Diese wird nicht benötigt, da die Daten nicht indiziert werden können. Die Transformationsschicht (UDK), der QueryBuilder (UDK) und der ServiceAdapter (UDK) werden zum Datasource Client hinzugefügt. Das iPlugInterface enthält vom iBus die Anfrage und leitet diese über die Transformationsschicht und den QueryBuilder an den ServiceAdapter weiter. Der ServiceAdapter versendet die erstellte XML-Anfrage über SOAP an den angeschlossenen UDK-Web Service.

3.6.1 Administration

Die Administrationsoberfläche wird beim UDK-iPlug als Webinterface bei der Installation mit eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber

- Name der Behörde
- Ansprechpartner (Name, Telefon, Email)
- ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- URL des Dienstes der abgefragt wird
- Proxy-Eintrag, falls dies für die Kommunikation notwendig ist

3.7 CSW-iPlug

Der CSW-iPlug wird zum Anschluss von OGC-konformen Katalogsystemen an den iBus genutzt. Die Architektur des iPlugs ist mit dem des UDK-iPlugs identisch, jedoch wird beim CSW-iPlug die Transformationsschicht und der ServiceAdapter an die Ansprüche der OGC/ISO-Spezifikationen angepasst.

Der CSW-iPlug wird bei einer Anfrage immer dann berücksichtigt, wenn die Suchabfrage auch die UDK-Klassen "Geoinformation/Karte" bzw. "Dienst/Anwendung/Informationssystem" einschließt.

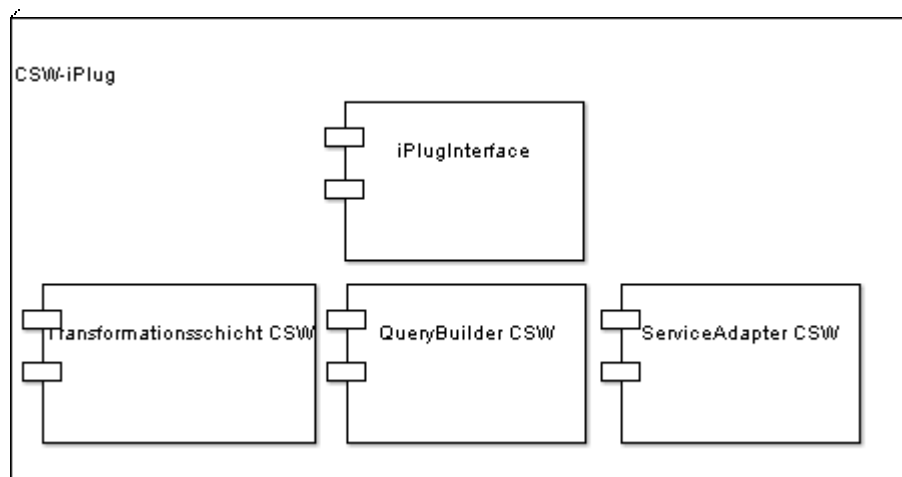


Abbildung 7: Komponentendiagramm des CSW-iPlug

Abbildung 7 zeigt die Komponenten des CSW-iPlug, er besteht aus den Komponenten iPlugInterface, postProcessingPipeline, Transformationsschicht (CSW), QueryBuilder (CSW) und ServiceAdapter (CSW). Der Ablauf innerhalb des CSW-iPlug ist identisch mit dem UDK-iPlug. Die Indizierung von Daten über Harvesting-Technologie ist nicht vorgesehen.

3.7.1 Administration

Die Administrationsoberfläche wird beim CSW-iPlug als Webinterface bei der Installation mit eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber
 - Name der Behörde
 - Ansprechpartner (Name, Telefon, Email)
 - ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- URL des Dienstes der abgefragt wird
- Proxy-Eintrag, falls dies für die Kommunikation notwendig ist

3.8 g2k-iPlug

Der g2k-iPlug dient dazu diverse Fachinformationssysteme über das bereits vorhandene g2k-Profil anzuschließen. Da kein Index erstellt wird, ist auch bei diesen Schnittstellen kein Ranking möglich. Dies spiegelt sich entsprechend in den Komponenten des g2k-iPlugs wider. Der Index des Datasource Clients entfällt, dafür werden die Transformationsschicht (g2K), der QueryBuilder (g2k) und der ServiceAdapter (g2k) hinzugefügt (siehe Abbildung 8).

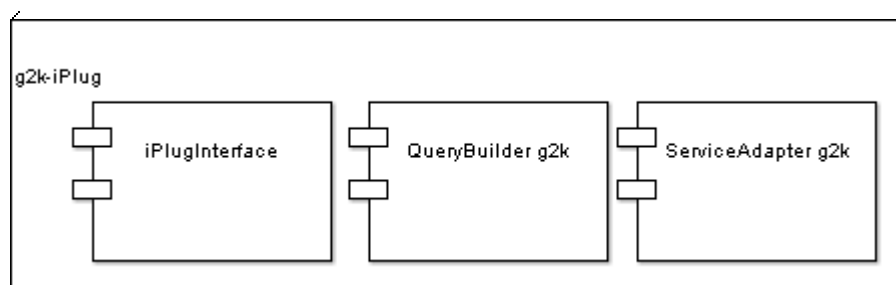


Abbildung 8: Komponentendiagramm des g2k-iPlug

Das iPlugInterface leidet die Suchanfrage an den QueryBuilder weiter. Dieser wandelt die Anfrage in eine einfache g2k-Anfrage um und übergibt sie dem ServiceAdapter. Der ServiceAdapter versendet die Anfrage und empfängt die Antwort. Die Trefferliste wird über das iPlugInterface an den iBus übergeben.

3.8.1 Administration

Die Administrationsoberfläche wird beim g2k-iPlug als Webinterface bei der Installation mit eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber
 - Name der Behörde
 - Ansprechpartner (Name, Telefon, Email)
 - ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- URL des Dienstes der abgefragt wird
- Art der Daten, die der Dienst zurückliefert (Anzahl der Treffer mit URL zur Trefferliste oder Metadaten)
- Proxy-Eintrag, falls dies für die Kommunikation notwendig ist

3.9 SNS-iPlug

Der SNS-iPlug dient zum Anschluss des Semantic-Network-Service des Umweltbundesamtes. Der SNS¹⁶ setzt sich aus dem Umweltthesaurus (UmThes), dem Geo-Thesaurus-Umwelt (GTU) und der Umwelt-Chronologie zusammen. Er wird im Rahmen von InGrid in die Suchoberfläche integriert und zur Anfrageerweiterung genutzt.

Neben der Unterstützung bei der thematischen Suche kommt der SNS auch bei der räumlichen Suche zum Einsatz. Insbesondere werden vom SNS administrative Einheiten oder Bounding Boxes abgefragt, um diese bei Indizierung und Recherche zu integrieren.

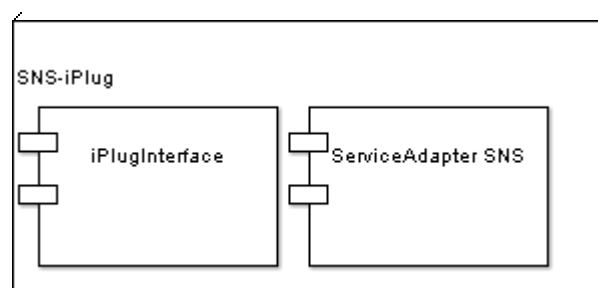


Abbildung 9: Komponentendiagramm des SNS-iPlug

¹⁶ <http://www.semantic-network.de>

Der SNS-iPlug besteht aus zwei Komponenten, dem iPlugInterface zum Anschluss an den iBus und dem spezifischen ServiceAdapter um den Web Service des SNS anzusprechen. Die Kommunikation mit dem SNS findet über SOAP statt.

3.9.1 Verwendung des SNS in Portal-U und InGrid 1.0

Der SNS wird für folgende Funktionalitäten genutzt:

- Analyse der Query, um "ähnliche Begriffe" auf der Ergebnisseite anzeigen zu können
- Definition von Raum- und Zeitfiltern in der Erweiterten Suche
- Browsen von Thesaurusbegriffen in der Erweiterten Suche
- Indizierung von Metadaten (Verschlagwortung, Raum- und Zeitbezüge)
- Einbindung von "Events" in Portal-U ("heute vor x Jahren", Umweltchronik)

Wie oben ersichtlich, kann die anfragende Komponente sowohl das Portal als auch der Index des Datasource Clients sein.

Analyse der Query (Portal)

Auf der Suchergebnis-Seite werden dem Nutzer ähnliche Begriffe angeboten (Synonyme, Ober- und Unterbegriffe) unter deren Einbeziehung er die Suche erneut ausführen kann, um das Suchfeld entsprechend auszuweiten.

Der SNS bietet bereits mit findTopic() und autoClassify() die Möglichkeit, für Begriffe vorhandene Topics bzw. Deskriptoren zu liefern. Weiterhin können mit getPSI() entsprechende Nicht-Deskriptoren, Ober- und Unterbegriffe zurückgegeben werden.

Browsen des Thesaurus (Portal)

Der Nutzer erhält die Möglichkeit, Thesaurusbegriffe zu browsen. Ähnlich der unter „Analyse der Query“ beschriebenen Funktionalität liegt der Sinn in einer Auslieferung von ähnlichen Begriffen, mit denen die Query erweitert werden kann.

Durch Vorlage eines Begriffs wird versucht, einen Deskriptor zu ermitteln, um von diesem Synonyme, Ober- und Unterbegriffe und verwandte (Themen-)Begriffe zu ermitteln. In der Oberfläche wird jedes Wort der Ergebnisliste (mit Ausnahme der Synonyme) wiederum anklickbar sein, um eine neue SNS Anfrage stellen zu können und so ein Browsing zu ermöglichen.

Ermittlung von Raumbezügen (Portal)

Der Benutzer kann auf begrifflicher Ebene herausfinden, ob dem SNS Raumbeziehungen bekannt sind. Für administrative Einheiten wird eine Gemeindeganziffer mit ausgeliefert. Eine Ergänzung der Rückgabewerte auf Seiten des SNS um Koordinaten (Bounding Boxes) ist erforderlich. Letztere sind vor allem auch notwendig, um eine kartenbasierte Suche zu ermöglichen, da in diesem Fall Koordinaten im Index hinterlegt sein müssen.

Während es zu anderen Suchfiltern jeweils genau ein korrespondierendes Indexfeld gibt und sich daraus eine eindeutige Suchstrategie ergibt (Filter -> korrespondierendes Indexfeld), werden im Unterschied dazu für Raumbezüge mehrere Felder verfügbar sein. Aus diesem Grund werden nachfolgend die Suchstrategien für eine raumbezogene Suche erläutert.

Wird vom Benutzer eine Karte benutzt, um den Raumbezug zu definieren, reicht diese die Query durch Koordinaten an, die durch die Größe und Lage der aufgezogenen Boundingbox definiert sind. Die Suche filtert dann mittels der in den 4 Koordinatenfeldern (Nord, Süd, Ost, West) hinterlegten Werten. Der Raumbezug wird in der Anfrage als Range Query formuliert.

Alternativ kann der Nutzer Raumbezüge auf begrifflicher Ebene mittels SNS ermitteln. Für administrative Einheiten können für Terme Gemeindekennziffern zurückgegeben werden. Der Suchfilter bezieht sich dann auf das Indexfeld, in dem die Gemeindekennziffern hinterlegt sind. Da diese Schlüssel hierarchisch entsprechend einer administrativen Gliederung vergeben wurden, kann die Anfrage als Wildcard Query formuliert werden. Dadurch können z.B. bei einem Raumbezug „Bayern“ auch alle Dokumente als Treffer gewertet werden, zu denen die Gemeindekennziffer von z.B. „München“ hinterlegt ist.

Da SNS für nicht-administrative Einheiten weder Koordinaten noch Gemeindekennziffern zurückliefern kann, beschränkt sich die Suche für derartige Räume auf die durch Verschlagwortung gewonnenen Raum-Deskriptoren.

Ermittlung von Zeitbezügen (Portal)

Auch die Ermittlung von Zeitbezügen findet auf begrifflicher Ebene statt. Vom SNS wird ermittelt, ob zu einem Begriff ein Zeitbezug im Sinne eines Datums oder Zeitraums ausgeliefert werden kann.

Verschlagwortung (Indizierung)

Die durch Verschlagwortung eines Textes erhaltenen Begriffe werden in einem separaten Feld des Indexes mitindiziert. Dies ermöglicht zum einen eine einschränkende Suche auf Metadatenebene, zum anderen kann das Ranking eines Dokumentes manipuliert werden, wenn der Suchbegriff als verschlagworteter Begriff in einem Dokument hinterlegt ist.

Auf eine Mitindizierung von Synonymen (Nicht-Deskriptoren) soll verzichtet werden, da eine manuelle Einbeziehung seitens des Nutzers die zu erwartende Ergebnismenge transparenter hält.

Raumbezüge (Indizierung)

Um die verschiedenen Suchmodi abbilden zu können, sollten Raum-Deskriptoren und soweit verfügbar Gemeindekennziffern und Bounding Boxes in den Metadaten der Indizes hinterlegt werden. Diese werden in getrennten Feldern des Index hinterlegt.

Zeitbezüge (Indizierung)

Können einzelnen Begriffen im Dokument Zeitbezüge zugeordnet werden, wird das entsprechend zugeordnete Datum / Zeitraum in den Metadaten hinterlegt. Dafür stehen die beiden Felder „von“ und „bis“ bereit. Im Falle eines Zeitpunktes sind die Werte von „von“ und „bis“ identisch. Die Anfrage wird als Range Query formuliert.

Die genannten Funktionalitäten können zum größten Teil bereits heute vom SNS abgedeckt werden. Gespräche mit den Verantwortlichen des SNS über kleinere Änderungen und Anpassungen werden zur Zeit geführt.

3.9.2 Administration

Die Administrationsoberfläche wird beim SNS-iPlug als Webinterface bei der Installation mit eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber
 - Name der Behörde
 - Ansprechpartner (Name, Telefon, Email)
 - ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- URL des SNS
- Proxy-Eintrag, falls dies für die Kommunikation notwendig ist

3.10 Forschungsportal.net-iPlug

Der Forschungsportal.net-iPlug (FPN-iPlug) dient zur Einbindung der Suchmaschine „Forschungsportal.net“. Die Suchmaschine verfügt über eine HTTP-GET-Schnittstelle. Aus diesem Grund wird für diese Suchmaschine ein eigenständiger iPlug umgesetzt, der die Suchanfragen des iBusses in den entsprechenden GET-Request umwandelt. Hierbei werden die Operatoren „und (+)“, „oder (|)“, „nicht (-)“ und Wildcardsuche (Begriff*) unterstützt. Es wird nur die Suchmethode „Textsuche“ von Forschungsportal.net sowie die Dateitypen „HTML“ und „PDF“ unterstützt. Alle weiteren Funktionen der Suchmaschine werden nicht unterstützt.

Eine Anfrage könnte wie folgt aussehen:

```
http://www.forschungsportal.net/cgi-bin/fp-search-3?q=wasser&dub=2&dub=ein&pdf=1&html=1&all=1
```

Die folgende Tabelle listet die Parameter und ihr Bedeutung auf, die der FPN-iPlug bei der Suchanfrage verwendet.

Parameter	Bedeutung
q	Query, der eigentliche Suchstring
art	Art der Suche. Der FPN-iPlug wird HTML und PDF unterstützen.
dub	Dublettenfilter. Der Dublettenfilter wird bei der Suche aktiviert sein.
all	Liefert alle 200 Treffer, wenn der Parameter auf 1 gesetzt ist. Dies wird bei InGrid der Fall sein.

Tabelle 1: Auflistung der Parameter, die der FPN-iPlug bei der Suche verwendet

Die Ergebnisse werden von Forschungsportal.net an den FPN-iPlug als HTML-Seite zurückgeliefert. Diese HTML-Seiten werden vom FPN-iPlug geparkt und an den iBus weitergereicht. Es wird bei jeder Suchanfrage die Trefferanzahl sowie die einzelnen Trefferbeschreibungen mit URL aus dem HTML-Code entnommen (siehe Abbildung 10). Ein Ranking der Daten wird nicht möglich sein.

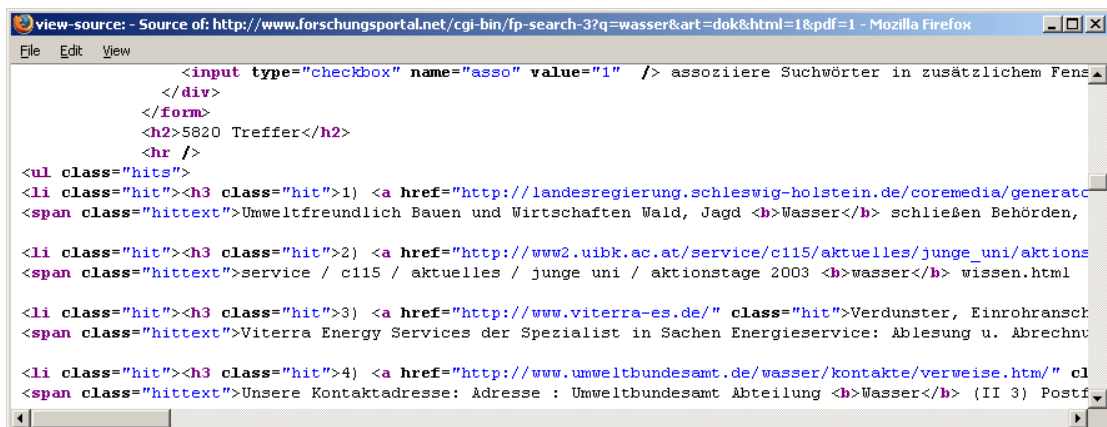


Abbildung 10: Beispiel Screenshot vom Forschungsportal.net HTML-Code

Durch die Gegebenheiten der Schnittstelle, ist es notwendig, dass die Struktur der Antworten nicht geändert wird. Falls dennoch Änderungen durchgeführt werden kann dies wiederum zur Folge haben, dass der FPN-iPlug entsprechend angepasst werden muss.

3.10.1 Administration

Die Administrationsoberfläche wird beim FPN-iPlug als Webinterface bei der Installation des iPlug eingerichtet. Folgende Einstellungen und Funktionen sind auf der Administrationsoberfläche möglich:

- Angaben zum Betreiber

- Name der Behörde
- Ansprechpartner (Name, Telefon, Email)
- ID des Anbieters
- Festlegen des Mutter-iBus
- Auswahl, ob der iPlug in der Peer-to-Peer-Group veröffentlicht werden soll, oder nur vom Mutter-iBus verwendet werden darf.
- URL der Suchmaschine (Default <http://www.forschungsportal.net/cgi-bin/fp-search-3?>)
- Proxy-Eintrag, falls dies für die Kommunikation notwendig ist

3.11 Sonderfall Forschungsdatenbanken

Die in der Leistungsbeschreibung zum Projekt gein® 2.0 aufgelisteten "Forschungsdatenbanken" stellen eine relativ heterogene Sammlung von Informationsquellen dar. Insbesondere kann man die Informationsquellen technisch in drei Kategorien unterteilen:

1. Einfache Internetseiten, auf denen die Informationen zu den Forschungsprojekten dargestellt werden.
2. Datenbanken, in denen die Informationen zu Forschungsprojekten unter unterschiedlichen Attributen abgelegt sind.
3. Suchmaschine (nur www.forschungsportal.net), die einen Index über verschiedene Server erstellt, auf denen Beschreibungen und Ergebnisse von Forschungsprojekten lagern.

Das in Kapitel 3.11.1 beschriebene Suchkonzept bietet den Kompromiss einerseits alle Informationsquellen über die einfache Suche zu berücksichtigen, andererseits gezielte Eingrenzungen des Suchergebnisses über die erweiterte Suche zu ermöglichen, wobei allerdings nur die Datenquellen berücksichtigt werden können, die über ihre Struktur als Datenbank im technischen Sinne dies zulassen.

Im Kapitel 3.11.2 wird kurz auf die Entwicklung der Filtermöglichkeiten eingegangen.

3.11.1 Suchkonzepte

Einfache Suche

Die einfache Suche findet über alle genannten Datenbanken statt. Dazu werden vier Plugins verwendet:

1. SE-iPlug: Suche über auf Forschungsdatenbanken eingeschränkten Suchraum; es werden alle als für Umweltforschung relevant gemeldeten Internetseiten durchsucht

2. g2k-iPlug: Anschluss von Forschungsdatenbanken über altes g2k-Profil; Suche im Volltext-Modus (Simple-Search); soll nur übergangsweise für bereits an gein@ angeschlossene Datenbanken genutzt werden
3. FPN-iPlug: Spezielles Plugin zum Anschluss von Forschungsportal.net; Meta-Suche in diesem speziellen Portal
4. DSC-iPlug: Anschluss von Forschungsdatenbanken per DataSourceClient; Suche im Volltext-Modus; zukünftig bevorzugter Anschluss

Die Ergebnisse für 1. und 4. werden gemeinsam in der 1. gemeinsam gerankten Suchergebnisliste dargestellt. Die Ergebnisse aus 2. und 3. werden anwendungsbezogen in der 2. Suchergebnisliste dargestellt.

Erweiterte Suche

Die Erweiterte Suche für Forschungsdatenbanken wird in InGrid parallel zur Erweiterten Suche in den anderen Suchräumen definiert. Es wird auch hier eine Volltextsuche angeboten, deren Ergebnismenge aber durch Filtermöglichkeiten eingeschränkt werden kann.

Die erweiterte Suche wird in Anlehnung an die Erweiterte Suche in anderen Suchräumen über einen "Suchassistenten" geführt, der die Filtermöglichkeiten der speziellen Suche über Forschungsdatenbanken abbildet. Die Ergebnismenge kann über die folgenden Attribute gefiltert werden:

Filter	Beschreibung
Titel	Text; Suche nach Titeln von Forschungsvorhaben
Institution	Text; Suche nach Namen der durchführenden Institutionen
Projektleiter	Text; Suche nach Namen von Projektleitern
Beteiligte	Text; Suche nach Namen von beteiligten Institutionen und Personen
Projektträger	Text; Suche nach Namen von finanzierenden Institutionen
Laufzeit	Zwei Datumfelder; "von" und "bis"

Die einzelnen Filter ergeben sich als kleinster gemeinsamer Nenner aus der Untersuchung der gemeldeten Datenbanken (siehe Kapitel 3.11.2).

In den Textfeldern wird wie in der Standardeinstellung nach ganzen Worten gesucht.

Die Filtermöglichkeiten werden so definiert, dass bei einer Nichtunterstützung einer Filtermöglichkeit durch eine Datenbank ein Nullergebnis geliefert wird. Das bedeutet konkret für die genannten Anschlussmöglichkeiten, dass bei einer Nutzung nur einer Filtermöglichkeit die über die Punkte 1. - 3. angeschlossenen Datenquellen nicht berücksichtigt werden. Nur die über den DSC-iPlug angeschlossenen Datenquellen können entsprechend gefilterte Ergebnisse liefern. Der DSC-iPlug wird entsprechend ausgebaut.

3.11.2 Untersuchung einiger Datenbanken

Die in untenstehender Tabelle aufgeführten Datenbanken lagen zur Untersuchung vor. Der Vergleich wurde aufgrund der Vorlage der Attribute der UFORDAT angelegt. In der Regel sind in den anderen untersuchten Datenbanken noch weitere Attribute

vorhanden. Diese wurden aber ignoriert, da ein Filtern nach diesen Attributen Ergebnisse der UFORDAT explizit ausschließen würden. Die Inhalte der zusätzlichen Attribute werden aber bei der Volltext-Recherche berücksichtigt.

Zum Vergleich wurde den Attributen der untersuchten Datenbanken die Attribute der UDK-Klasse "Vorhaben/Projekt/Programm" gegenüber gestellt. Auf diese Attribute soll die Abbildung der Originalfelder der Forschungsdatenbanken im DSC-iPlug erfolgen.

	UDK (Vorhaben/...)	UFORDAT	Umweltforschung Bayern	Umweltforschung BW
Titel	Objektname	Originalthema	Titel	Titel
Institution	Indirekt über die dem Projektleiter zugeordnete Institution	Institution	Behörde	Institution
Projektleiter	Projektleiter	Projektleiter	Bearbeiter	Projektleiter
Beteiligte	Beteiligte	Beteil. Pers.	Auftragnehmer	Nicht vorhanden
Laufzeit	Von / bis	Laufzeit	Laufzeit-Gep Von / LaufzeitGep Bis	Beginn / Ende
Beschreibung	Beschreibung	Deutsch	Kurzbeschreibung	Beschreibung
Schlagwort	Suchbegriffe	Schlagworte	Nicht vorhanden	Schlagwort
Raumbezug	Staat/Region/...	Geogr. Deskr.	Nicht vorhanden	Nicht vorhanden
Umweltklassen	Umweltklassifikation	Umweltklassen	Nicht vorhanden	Nicht vorhanden
Projekträger	Indirekt über Adressverweis "Projekträger"	Finanzierung	KostenGesamt	Finanzierende Institution
Zus.arb.Inst.	Nicht explizit	Zus.arb.Inst.	Nicht explizit	Nicht explizit

Die Volltext soll grundsätzlich über alle Textfelder geführt werden. Welche Felder hierfür im Einzelfall in Frage kommen, soll vom Betreiber der Datenbank bei Installation des DSC-iPlug definiert werden.

Bei dem Vergleich der einzelnen Felder wurden diejenigen, die von den meisten Datenbanken unterstützt werden, als Kandidaten für die Filtermöglichkeiten in der erweiterten Suche betrachtet.

Die beschreibenden Felder wie Beschreibung, Schlagworte, etc. wurden als wenig geeignet für einen Filter angesehen und sollen stattdessen über die Volltextsuche abgedeckt werden.

Es wird davon ausgegangen, dass Felder, die nicht bei allen Datenbanken auftreten, aber inhaltlich mit anderen Feldern zusammengefasst werden können, wie z.B. "Zus.arb.Inst." und "Beteil. Pers." im DSC-iPlug gemeinsam auf ein Attribut wie z.B. "Beteiligte" projiziert werden.

4 Anhang

4.1 Abbildungsverzeichnis

<i>Abbildung 1: Anwendungsfälle eines iPlugs</i>	15
<i>Abbildung 2: Komponentendiagramm des DSC-iPlug</i>	19
<i>Abbildung 3: Komponentendiagramm des SE-iPlug</i>	21
<i>Abbildung 4: Komponentendiagramm des UdkDB-iPlug</i>	24
<i>Abbildung 5: Komponentendiagramm des CswDB-iPlug</i>	26
<i>Abbildung 6: Komponentendiagramm des UDK-iPlug</i>	27
<i>Abbildung 7: Komponentendiagramm des CSW-iPlug</i>	28
<i>Abbildung 8: Komponentendiagramm des g2k-iPlug</i>	29
<i>Abbildung 9: Komponentendiagramm des SNS-iPlug</i>	30
<i>Abbildung 10: Beispiel Screenshot vom Forschungsportal.net HTML-Code</i>	34

4.2 Literatur

- [1] Java Management Extension (JMX)
<http://java.sun.com/products/JavaManagement/>
- [2] Management Extension for Java (MX4J)
<http://www.mx4j.org>
- [3] Juxtapose (JXTA)
<http://www.jxta.org>
- [4] Apache Lucene
<http://lucene.apache.org>
- [5] Apache Nutch
<http://incubator.apache.org/nutch/>
- [6] Page Rank
<http://citeseer.ist.psu.edu/page98pagerank.html>
- [7] Data Transfer Object
<http://martinfowler.com/eaCatalog/dataTransferObject.html>
- [8] Hibernate
<http://www.hibernate.org/>
- [9] Software AG Tamino
<http://www1.softwareag.com/de/products/tamino/default.asp>

- [10] Quartz
<http://www.opensymphony.com/quartz/>
- [11] Semantischer Netzwerk Service (SNS)
<http://www.semantic-network.de>
- [12] Open Geospatial Consortium
<http://www.opengeospatial.org/>