Recherchebericht

Inhaltsverzeichnis

1. Begriffe

1.1 URL

URLs (Uniform Resource Locator) dienen als Adresse zum Identifizieren und Lokalisieren von Ressourcen in Computernetzwerken. Sie werden meist auch „Webadressen“ genannt und dienen für den Zugriff auf bestimmte Websites (z.B.: http://www.olat.org). Sie können aber auch den Pfad für lokal gespeicherte Dateien oder Verzeichnisse angeben (z.B. file: ///C:/bsp/bsp.txt) .

1.2 URI

URIs (Uniform Resource Identifier) sind einheitliche Bezeichner von Ressourcen. Sie bestehen aus einer Zeichenfolge und dienen zur Identifizierung abstrakter oder physischer Ressourcen. Eine Unterart von URI ist URL (siehe: 1.1) .

1.3 Fragment Identifier

Ein Fragment Identifier (Fragmentbezeichner) kann einer URI (siehe: 1.2) hinzugefügt werden, um lokal Teile eines Dokuments zu adressieren. Er wird mit einem Zeichen (#) gekennzeichnet (z.B.: http://www.bsp.de/txt.pdf#seite3) .

1.4 W3C

Das W3C (World Wide Web Consortium) ist eine internationale Organisation, die Standards für das world wide web entwickelt. Neben forschenden Universitäten beteiligen sich viele große Firmen der Internetbranche an der Entwicklung.

1.5 HTML/XHTML

HTML (Hypertext Markup Language) ist eine textbasierte Sprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten. Es ist ein vom W3C (siehe 1.4) empfohlener Standard für Textdaten im Web. HTML wird von Webbrowsern gelesen und stellt diesen Informationen zum Dokument bereit, womit die Darstellung des Dokuments am Rechner beschrieben wird, z.B. Texte färben, Tabellen darstellen oder Links zu anderen HTML-Dokumenten angeben.
XHTML (Extensible HTML) nimmt darüber hinaus einen semantischen Bezug der Inhalte und genügt den Syntaxregeln von XML (Extensible Markup Language). HTML/XHTML-Dokumente sind die Basis eines Großteils der existierenden Websites und werden von Webbrowsern dargestellt.

1.6 JSP

JSP(JavaServer Pages) ist eine von Sun Microsystems entwickelte Programmiersprache zur dynamischen Erzeugung von HTML- und XML Ausgaben (siehe: 1.5) eines mit einer Java Laufzeitumgebung ausgestatteten Webservers.
Die statischen Teile einer Webseite werden dabei wie gewöhnt in normalem HTML/XML codiert und die dynamischen Teile werden durch spezielle JSP Tags und Anweisungen gekennzeichnet und dann von der JAVA Laufzeitumgebung zur Laufzeit dynamisch generiert. JSP steht damit in

direkter Konkurrenz zu anderen Webentwicklungssprachen wie zum Beispiel PHP.

1.7 CSS

CSS (Cascading Style Sheets) ist eine Sprache für Vorlagen (Stylesheets) von strukturierten Dokumenten, wie HTML (siehe 1.5) und XML.

1.8 Browser-Plugin

Ein Browser-Plugin („Erweiterungsmodul“) ist ein Programm, das die Funktionen des Browsers erweitert. Mit Hilfe des Plugins kann der Browser Daten verarbeiten, die keine browsertypische Dateiformate (z.B. HTML) haben. Sobald Nutzer das Plugin installieren, welches browser- und betriebssystemabhängig ist, können sie auf die zusätzlichen Inhalte zugreifen.

1.9 JavaScript

JavaScript ist eine Skriptsprache, die meist für Webanwendungen genutzt wird. Es handelt sich um eine Programmiersprache, die eher für kleinere Applikationen gedacht ist. Anwendungen, die in JavaScript geschrieben sind, werden als „Skripte“ bezeichnet.

1.10 SQL

SQL (häufig als Abkürzung für „Structured Query Language“ geführt) ist eine Sprache zur Kommunikation mit Datenbanken, die auf relationaler Algebra basiert.

1.11 PHP

PHP (rekursives Akronym für „PHP: Hypertext Preprocessor“) ist eine Skriptsprache, welche vorrangig zur Webprogrammierung verwendet wird. Stärken von PHP sind einerseits die breite Datenbankunterstützung und Internet-Protokolleinbindung, sowie die weite Verbreitung als Programmiersprache zum Erstellen von Websites (wodurch viele PHP- Bibliotheken verfügbar sind) . PHP wird als Open Source Software verbreitet.

1.12 AJAX

AJAX (*Asynchronous JavaScript and XML*) ist eine Programmiertechnik von Webseiten, die es der Seite ermöglicht Daten zu senden und zu empfangen, ohne das ein Nutzer zwingend eine Aktion ausgelöst hat oder die aktuelle Seite des Webauftritts gewechselt hat.
Es entsteht eine Interaktivität des Webauftritts, die der Nutzer sonst nur von Desktopanwendungen gewohnt ist und hebt somit einige Einschränkungen auf, die man lange von den doch recht starren HTML (siehe 1.5) Seiten aus der Anfangszeit des WWW gewohnt war.

1.13 Web Annotation

Eine Web Annotation ist in etwa eine Notiz oder Information, die ein Nutzer bezüglich einer beliebigen Webresource veröffentlicht, ohne jedoch die ursprüngliche Webresource zu verändern. Die Nutzer eines Annotationssystems verändern somit untereinander ihre Sicht auf das Web, indem sie einzelne Seiten um vermeintlich nützliche Zusatzinformationen anreichern und so im

Stile des Social Webs selber Einfluss nehmen können, wie Andere das Web wahrnehmen.

1.14 Web-Service

Ein Web-Service (oder Webdienst) ist eine auf der Client-Server-Architektur (siehe 2.5) basierende Kommunikation zwischen einem Internetserver und einem Clientprogramm. Ein Web-Service stellt dabei keine Internetseite im herkömmlichen Sinn zur Verfügung, sondern erlaubt die Kommunikation zwischen Anwendungsprogrammen. Ein Beispiel für eine solche Kommunikation kann der Datenabgleich zwischen einem Onlineshop und einem Warenwirtschaftssystem oder zwischen einem Musikplayer und der Onlinedatenbank des Musikanbieters sein. In Zeiten der engen Anbindung von Desktopsoftware an Clouddienste gewinnen Webdienste an Bedeutung.

1.15 Ontologie

Eine Ontologie ist eine formal geordnete Darstellung einer Menge von Begrifflichkeiten.
Sie beschreibt einen Wissensbereich mit Hilfe einer standardisierten Terminologie, sowie Beziehungen zwischen dort definierten Begriffen. Somit kann sie ein Netzwerk von Informationen, mit logischen Relationen, darstellen.

1.16 OWL

OWL (*Web Ontology Language)* ist ein vom W3C (siehe 1.4) vorgeschlagener Standard zur Definition von Ontologien im Semantic Web (siehe 3.2) . Er baut auf der Syntax von RDF/RDFS auf, mit Kernerweiterungen wie zum Beispiel Kardinalitätsbeschränkungen von Subklassen und Teilmengenbeziehungen (Durchschnitt, Disjunktheit etc.) - den in RDF/RDFS (siehe 3.1) hergestellten Beziehungen einzelner Klassen untereinander wird somit auch noch die Information hinzugefügt, welche Eigenschaften diese Beziehungen haben.

1.17 SPARQL

SPARQL (*Protocol and RDF Query Language)* ist ein vom W3C (siehe 1.4) vorgeschlagener Standard einer Anfragesprache für RDF-Daten (siehe 3.1). Die Syntax orientiert sich an der Turtle-Notation von RDF-Dokumenten und dem Grundgerüst von SQL (siehe 1.10). Die RDF-Tripel werden in der Reihenfolge Subjekt, Prädikat, Objekt notiert und durch einen Punkt abgeschlossen, wobei URIs (siehe 1.2) in spitzen Klammern < > und Literale in Anführungszeichen zu setzen sind. Dabei sind Turtle-Abkürzungen zulässig. Variablen werden durch ? oder $ gekennzeichnet und sind zulässig als Subjekt, Prädikat oder Objekt.

2. Konzepte

2.1 Kommentieren und Highlighten

Unter dem Kommentieren eines Inhaltes (z.B. Web Content) versteht man das Hinzufügen einer Anmerkung. Betrachtet man die Umsetzung des Kommentars innerhalb einer Website, lässt sich dies beispielsweise durch eine Sprechblase oder ein Pop-Up verwirklichen. Ein weiterer Aspekt der semantischen Einflussnahme auf Webinhalte (siehe 1.13 Web Annotation) ist das Highlighten. Das Highlighten dient zur visuellen Hervorhebung (Markierung) einzelner Fragmente aus einem größeren Zusammenhang. Zum Beispiel können ausgewählte Textabschnitte eingerahmt oder farblich manipuliert werden.

2.2 Fragment Link

Ein Fragment Link wird eine Erweiterung eines normalen Hyperlinks sein. Dieser wird die Möglichkeit bieten nicht nur als Querverweis zu einer Website zu dienen, sondern direkt zu einem speziellen Abschnitt einer Website weiterleiten zu können. Die Besonderheit dabei ist, dass jeder Nutzer einen solchen verlinkten Abschnitt selbst bestimmen, highlighten und kommentieren kann (siehe 2.1) . Da diese Funktionen nicht von herkömmlichen Browsern angeboten werden, werden zusätzliche externe Ressourcen (z.B. ein Plugin, siehe 1.8) benötigt, um einen solchen Fragment Link zu nutzen.

2.3 Objektorientierte Programmierung

Die Objektorientierte Programmierung (OOP) ist ein Paradigma der Softwareentwicklung und baut auf die prozedurale Programmierung auf. Sie spiegelt die elektronisch abzubildende Miniwelt durch Objekte mit Eigenschaften (Attribute) und Operatoren (Methoden) wieder. Die Entwicklung nach dem objektorientierten Paradigma bedeutet das Programmieren von Klassen, d.h. das Erzeugen von Methoden (Funktionen) und Attribute (Variablen), die dann zur Laufzeit des Programms zu Objekten instantiiert werden. Eine wichtige Eigenschaft der OOP ist die Vererbung, d.h. das Erweitern und Anpassen bestehender Klassen ohne diese direkt ändern zu müssen.
Die wichtigste Programmiersprache der objektorientierte Programmierung ist Java, jedoch wurden viele bekannte Programmiersprachen (C, PHP, JavaScript) um dieses Paradigma erweitert. OOP ist die Basis vieler weiterer Paradigmen und Technologien wie MVC (siehe 2.4).

2.4 Model-View-Controller-Architektur

Die Architektur einer Software nach MVC (Model-View-Controller) bedeutet die logische Kapselung bestimmter Programmteile in entsprechend benannte Klassen. Diese Architektur geht davon aus, dass sich nahezu jedes Programm in die folgenden Bestandteile zerlegen lässt:
Die Datenschicht repräsentiert die vom Programm zu bearbeitenden Daten. Das Model ist der Programmteil, der die Daten des Programms liest, schreibt und die Konsistenz sicher stellt. Die Modelklasse stellt der Programmlogik alle Daten durch get- und set-Methoden zur Verfügung und trennt diese Funktionen damit von der allgemeinen Programmlogik.
Die View-Funktionen erzeugen die optische Repräsentation des Programms. Durch die Trennung der Ausgabelogik von der eigentlichen Programmlogik sind Änderungen an der Darstellung ohne Eingriff in die Programmlogik möglich. Besonders in der Webentwicklung sind auf diese Weise gravierende Layoutänderungen ohne Verlust der Stabilität der Programmlogik möglich. Ferner

erlaubt diese Technologie die Nutzung mehrerer Interfaces für das gleiche Programm.
Der Controller repräsentiert die eigentliche Programmlogik. Er empfängt Nutzereingaben, lässt die Daten von der Model-Schicht verarbeiten und erzeugt mit Hilfe der View-Komponente die nächsten Ausgaben für den Benutzer.

2.5 Client-Server Modell

Das Client-Server Modell beschreibt ein Systemdesign, bei dem die Verarbeitung einer Anwendung in zwei voneinander getrennte Teile aufgespalten wird. Ein Teil läuft dabei auf dem Server (Backend-Komponente) und der andere auf einer Workstation (Client oder Front-End). Beide Teile werden dabei über Netzwerke zu einem System zusammengefügt. Der Server stellt dabei Dienste bereit und reagiert auf Anfragen. Der Client nimmt Leistungen des Servers in Anspruch und verarbeitet dessen Antworten. Die Kommunikation erfolgt dabei durch Transaktionen, die an bestimmte Kriterien gebunden sind, um z.B. einen konsistenten Datenbestand zu garantieren. Im Unterschied zu Host-basierten Architekturen sind die Server heute nicht mit der gesamten Datenverarbeitung beschäftigt, sondern geben die Daten zur weiteren Verarbeitung an den Client zurück. Je nachdem in welchem Umfang dies geschieht wird zwischen Thin Clients und Fat Clients unterschieden. Mögliche Anwendungen der Client-Server-Architektur sind Mail-Server, Print-Server oder Web-Server.

2.6 Annotation Server

Im Unterschied zu einem reinen (Web-)Service (siehe 1.14) stellt ein Annotation Server ein eigenständiges (physische oder virtuelle Maschine) System dar, dass ausschließlich zur Bewältigung einer Aufgabe genutzt wird: Annotation von Webinhalten. Annotationswünsche werden dem Server durch ein geeignetes Protokoll (vgl. PHP Webservice) übergeben, verarbeitet und aufbereitet. Daraufhin erhält der Nutzer beispielsweise eine fertige URL zum verschicken.
Ein solcher Server bietet die Möglichkeit der Speicherung aller notwendigen Informationen durch DBS und somit eine gewisse Persistenz der Annotationen.
Es ist zudem denkbar annotierte Webinhalte vor Veränderungen zu schützen indem annotierte Inhalte zwischengespeichert werden (Snapshots).

3. Aspekte

3.1 RDF/S

RDF (Resource Description Framework) ist ein vom W3C vorgeschlagener Standard zur Beschreibung von eindeutig durch URIs (siehe 1.2) identifizierbaren Webressourcen, deren Datenmodell und deren Beziehung untereinander.
Darauf aufbauend erweitert RDFS RDF so, dass einzelne Gruppen von Webressourcen in Klassen zusammengefasst und diese Klassen durch Unterklassenbeziehungen geordnet werden können.

3.2 Linked Data/ Semantic Web

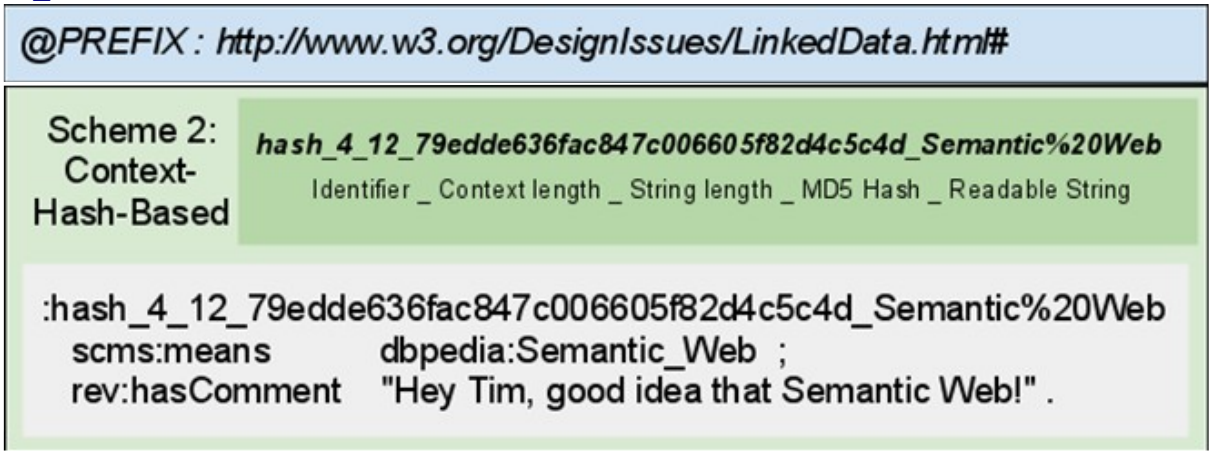
Das semantic web stellt ein Konzept zur Informationsbeschreibung im *world wide web (www)* dar. Im Gegensatz zur bisherigen Praxis, das Web als Sammlung von Dokumenten zu betrachten, sollen Inhalte im Web als Daten angesehen werden, die durch Maschinen bearbeitet werden können.
Wichtig hierbei ist z.B. die Betrachtung als Linked Data, womit auch nicht explizit genannte Verbindungen von verschiedenen Daten erfasst werden können.
Linked Data beschreibt verfügbare Daten in einer standardisierten Form und ihre Beziehungen zueinander.
Von der W3C (siehe 1.4) wird als Standard für Linked Data RDF (siehe 3.1) entwickelt, darauf aufbauend zur Untersuchung im semantic web OWL (siehe 16) als Ontologie (siehe 1.15) und SPARQL (siehe 1.17) als Anfragesprache auf OWL.

3.3 NIF

Das NLP Interchange Format, oder kurz NIF, ist ein im Rahmen des LOD2 EU Projektes entwickeltes Austauschformat, welches es ermöglicht, Sprachverarbeitungssoftware in Komponenten zu unterteilen oder mehrere dieser Komponenten zu kombinieren. Dadurch wird sowohl Wiederverwendbarkeit der Software als auch Interoperabilität zwischen Natural language Processing (NLP)-Tools, Sprachressourcen und Annotationen erhöht. Das Konzept basiert auf dem Resource Description Framework (RDF) und der darauf aufbauenden Web Ontology Language (OWL). Der Kern von NIF besteht dabei aus einem Vokabular, welches es ermöglicht, Strings als RDF-Ressourcen darzustellen. In diesem Zusammenhang wird ein spezielles URI-Design verwendet, um Annotationen zu einem bestimmten Teil eines Dokuments genau festzulegen. Diese URI's können dann dazu verwendet werden, um der jeweils betrachteten Folge von Zeichen beliebige Anmerkungen beizufügen. NIF besteht aus drei verschiedenen Komponenten, um das Problem der Interoperabilität von drei verschiedenen Seiten zu behandeln:
URI-Schemata werden genutzt, um Elemente in Hypertext zu identifizieren und Annotationen in Dokumenten mit Hilfe von Fragment Id's zu verankern. So wird erreicht, dass die annotierten Dokumententeile als Subjekte in RDF-Tripeln verwendet werden können. Dadurch wird erreicht, dass Annotationen im Web als Linked Data veröffentlicht und zwischen verschiedenen NLP-Tools und Anwendungen ausgetauscht werden können. An die URI Schemata werden drei wesentliche Anforderungen gestellt: Einheitlichkeit, Einfachheit bezüglich der Implementation und Stabilität bezüglich der Veränderung des Dokuments. Da es nicht ohne Weiteres möglich ist, diese unter einen Hut zu bringen, definiert NIF zwei verschiedene Schemata: offset-basierte und context-hash-basierte. Da die Offset-basierte Variante für unser Projekt aufgrund der Unbeständigkeit in Bezug auf die Verschiebung von Textteilen in einem Dokument keine Rolle spielt, soll an dieser Stelle nur auf die hashbasierte Variante eingegangen werden:
Eine Hash-basierte URI besteht aus 5 Teilen, die jeweils durch einen Unterstrich getrennt werden:

1) einen Identifier – im Beispiel der String „hash“
2) die Kontextlänge
3) die Gesamtlänge des zu adressierenden Strings
4) einen Auszug aus der Nachricht - einen 32 Zeichen umfassenden HEXIDIGIT md5 Hash, der aus dem zu adressierenden String und dem Kontext gebildet wird: Die Nachricht M hat dabei die Form „linkerKontext(String)rechterKontext“
5) einen 20 Zeichen langen Teil des zu adressierenden Strings (oder weniger, wenn der String kürzer ist) – URL-Encoding mit RFC 3986

Das folgende Beispiel nutzt einen Kontext der Länge 4. Der unter 4.) beschriebene Hash wird durch die Funktion md5(“ it (Semantic Web).<br“); ermittelt.
Die resultierende URI ist:
http://www.w3.org/DesignIssues/LinkedData.html#hash_4_12_79edde636fac847c006605f82d4c5c4d_Semantic%20Web



3.4 Bookmarklet

Bookmarklets (auch: Favelets genannt) sind kleine, in javaScript geschriebene Makros mit deren Hilfe die Funktionen eines Webbrowsers erweitert (siehe 1.8) werden können. Im Grunde genommen ist ein Bookmarklet ein einfaches Lesezeichen, das einen direkten Aufruf von JavaScript-Code im Browser erlaubt. Dies kann beispielsweise genutzt werden um Aussehen oder Funktionalität von Webseiten clientseitig zu verändern.
Üblicherweise werden eingebettete JavaScript Anweisungen durch den Aufruf einer URL oder bestimmte Events ausgeführt – wann und wie dies geschieht bestimmt dabei der Ersteller der Webseite. Im Gegensatz dazu wird im Falle von Bookmarklets (ähnlich wie bei der Firefox Erweiterung Greasemonkey) der enthaltene JavaScript-Code auf die aktuelle Webseite angewandt, wenn es vom Nutzer aktiviert/ ausgewählt wird.
Nun ist es durchaus denkbar Favelets zu nutzen um Webinhalte zu annotieren und diese Annotation mit anderen zu teilen. Beispielsweise könnte ein (sender-)Bookmarklet markierte Inhalte einer Webpage analysieren, um deren relative Position im Dokument zu ermitteln um dann nebst Kommentaren eben diese Inhalte durch eine korrespondierende Form von URL zugänglich machen.
Nutzer die eine solch formatierte URL erhalten, könnten nun mit Hilfe eines (Empfänger-)Bookmarklets die gewünschten Inhalte und Kommentare auf der Website hervorgehoben betrachten.
Die Installation von Favelets ist denkbar einfach und funktioniert wie das Anlegen eines typischen Lesezeichens in Browsern, wodurch der Zugang zu einer solchen Annotationsmöglichkeit für alle gängigen Browser gegeben ist.

Trotzdem hängt Unterstützung in den verschiedenen Browsern vom jeweiligen Support von JavaScript-URLs in den Bookmarks und deren expliziten Inhalt (evtl. besondere Bibliotheken) ab. Zusätzlich unterliegen Bookmarklets im Allgemeinen einer weiteren wichtigen Einschränkung: die maximal verwendbare Anzahl von Zeichen pro Bookmarklet unterscheidet sich je nach Browser. Unter Einbezug der (keineswegs vollständigen) Nachteile von Favelets kann von dieser Umsetzungsvariante abgesehen werden.

4. Quellen

http://de.wikipedia.org/wiki/Ontologie_(Informatik)
http://pcai042.informatik.uni-leipzig.de/swp/SWP-11/swp11-6/Recherchebericht.pdf
http://de.wikipedia.org/wiki/URL
http://de.wikipedia.org/wiki/Uniform_Resource_Identifier
http://de.wikipedia.org/wiki/Fragmentbezeichner
http://de.selfhtml.org/
http://de.wikipedia.org/wiki/HTML
http://de.wikipedia.org/wiki/XHTML
http://de.wikipedia.org/wiki/Cascading_Style_Sheets
http://www.w3.org
http://de.wikipedia.org/wiki/Plug-in
http://de.wikipedia.org/wiki/SQL
http://de.wikipedia.org/wiki/PHP
http://nlp2rdf.org/nif-1-0#toc-parameters
http://blog.aksw.org/2011/nlp-interchange-format-nif-1-0-spec-demo-and-reference-implementation/
http://nlp2rdf.org/nif-1-0#toc-parameters
http://www.slideshare.net/kurzum/nif-version-10
http://de.wikipedia.org/wiki/Client-Server-Modell
http://www.e-teaching.org/technik/vernetzung/architektur/client-server/
http://www.it-administrator.de/lexikon/client-server-architektur.html
NIF-An ontology -based and linked- data -aware NLP interchange Format (Sebastian Hellmann, Jens Lehmann, Sören Auer)