

---

# Semantische Annotation

Gerald Reif

Institut für Informatik, Universität Zürich, 8057 Zürich, Schweiz  
`reif@ifi.unizh.ch`

**Zusammenfassung** In diesem Kapitel wird zuerst der Begriff *Semantische Annotation* eingeführt und es werden Techniken besprochen um die Annotationen mit dem ursprünglichen Dokument zu verknüpfen. Weiters wird auf Probleme eingegangen, die sich beim Erstellen der Annotationen ergeben. Im Anschluss daran werden Software Tools vorgestellt, die einen Benutzer beim Annotierungsprozess unterstützen. Zum Abschluss werden Methoden diskutiert, die den Annotierungsvorgang in den Entwicklungsprozess einer Web Applikation integrieren.

## 1 Einleitung

Tim Berners-Lee definiert das Semantic Web als eine Erweiterung des existierenden World Wide Web, in dem Informationen eine wohldefinierte Bedeutung zugeordnet ist, um die Zusammenarbeit zwischen Menschen und Computern zu erleichtern [4]. Um die Bedeutung (auch Semantik genannt) der Informationen formalisiert festzuhalten, werden im Semantic Web Meta-Daten im RDF Format verwendet. Die Meta-Daten erlauben es Computern auf die Bedeutung eines Dokuments zuzugreifen. Diese Grundidee des Semantic Web ermöglicht die Entwicklung von Anwendungsprogrammen, die den Benutzer bei der Bewältigung von Wissensproblemen unterstützen, die weit über die Fähigkeiten traditioneller Anwendungen hinausgehen. Informationen können applikations- und herstellerunabhängig zwischen Anwendungen ausgetauscht werden, ohne dass dabei die Semantic verloren geht. Beispiele solcher Anwendungen sind in Kapitel ?? angeführt.

Die Existenz von RDF Meta-Daten ist eine Grundvoraussetzung für die Realisierung des Semantic Web. Den Prozess des Hinzufügens von semantischen Meta-Daten zu Dokumenten, die den Inhalt eines Dokuments in maschinen-verarbeitbarer<sup>1</sup> Form beschreiben, nennt man *Semantische Anno-*

---

<sup>1</sup> In der Literatur wird häufig der Begriff *maschinen-verständlich* verwendet. In diesem Kapitel wollen wir der Argumentation in [3] folgen und von maschinen-

*tation* (engl. Semantic Annotation). Dabei kann es sich bei den Dokumenten um Informationsquellen wie z.B. Web-Seiten, Bilder, Audio oder Video Dateien handeln. Da es sich bei den Dokumenten im Web hauptsächlich um HTML Web-Seiten handelt, liegt der Schwerpunkt in diesem Kapitel bei der Annotation von Web-Seiten.

## 2 Verknüpfung zwischen Annotation und Web-Seite

Wie in der Einleitung bereits diskutiert, erhält man durch den Annotierungsprozess, zusätzlich zum eigentlichen Dokument Meta-Daten, die den Inhalt des Dokuments beschreiben. Das annotierte Dokument muss nun mit seinen Meta-Daten verknüpft werden. Dazu gibt es keine Standard Methode [14]. Abhängig vom Anwendungsfall stehen verschiedene Methoden zur Verfügung. Die Methoden können grundsätzlich in folgende Kategorien eingeteilt werden:

- Die Annotationen werden eingebettet in das Dokument.
- Das Dokument verweist auf ein externes Dokument mit den Annotationen.
- Die Annotationen verweisen auf das annotierte Dokument; das Dokument selbst enthält keine Hinweise auf Annotationen.

In den nachfolgenden Abschnitten werden diese Techniken und ihre Anwendungsgebiete vorgestellt. Für eine ausführliche Diskussion dieses Themas finden Sie in [14].

### 2.1 Eingebettete Annotationen

Bei dieser Technik werden die RDF Meta-Daten direkt in das zu annotierende Dokument eingebettet. Dies hat den Vorteil, dass man beim Datenaustausch nur ein Dokument weitergegeben werden muss. So kann auf die Meta-Daten des Dokuments auch zugegriffen werden, wenn man keinen Zugriff auf externe Ressourcen hat, zum Beispiel, wenn man offline arbeitet.

Beim Einbetten der RDF Meta-Daten muss darauf geachtet werden, dass ein Web Browser die Meta-Daten nicht am Bildschirm anzeigt. Dies kann durch das Einbetten von RDF im XML Syntax in den `<head>` der HTML Seite geschehen. Dabei wird der Abbreviated XML/RDF Syntax verwendet, da dieser Syntax die Meta-Daten in Attributen speichert, die von den meisten Web Browsern nicht dargestellt werden (im Gegensatz zum Inhalt von Elementen) [14]. Abbildung 1 zeigt ein Beispiel für diese Art der Einbettung der Meta-Daten.

---

verarbeitbaren Informationen sprechen. Maschinen müssen nicht notwendigerweise die Informationen verstehen, es reicht, wenn sie in der Lage sind, die Informationen effizient zu verarbeiten. Unter Umständen führt dies dazu, dass Menschen glauben, die Maschinen würden tatsächlich verstehen.

```

<head>
  <title>Some Page</title>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description rdf:about="http://www.w3.org/"
      dc:title="W3C Homepage"/>
  </rdf:RDF>
</head>

```

**Abbildung 1.** Einbetten von Abbreviated RDF/XML `<head>` der Web-Seite.

HTML bietet zwei Elemente um nicht HTML Medien in eine Web-Seite einzubinden: Das `<object>` und das `<script>` Element. Beim `<object>` Element kann das `data` Attribut dazu verwendet werden um die Meta-Daten einzubinden. Dazu werden die Meta-Daten base64 encoded, siehe Abbildung 2. Dies hat den Vorteil, dass auch andere RDF Serialisierungen ausser RDF/XML, wie zum Beispiel Notation 3, NTriples oder Turtle, verwendet werden können. Das `<object>` kann auch dazu verwendet werden um auf eine externe Meta-Daten Beschreibung zu verlinken. Diese Technik wird im nachfolgenden Abschnitt beschrieben.

```

<object data="data:application/rdf+xml;base64,PHJkZjpsREYgeG1sbnM
w0i8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIgGPg==" />

```

**Abbildung 2.** Einbetten von base64 encodeten Meta-Daten im `<object>` Element.

Der Inhalt des `<script>` Elements wird üblicherweise von Script Interpretern ausgeführt und wird nicht am Bildschirm dargestellt. In diesen Fall wird das Element dazu verwendet, die Meta-Daten im RDF/XML Format aufzunehmen. Dabei wird mit dem `type` Attribut der Name des Interpreters mit dem MIME-Typ `application/rdf+xml` angegeben, siehe Abbildung 3. Dieser MIME-Typ ist im RFC3870 definiert [20].

Eine weitere Technik ist die Meta-Daten mit Hilfe eines HTML Kommentars (`<!-- -->`) in die Web-Seite einzufügen.

Mit Ausnahme der Techniken mit dem `<object>` Element und dem HTML Kommentar haben die in diesem Abschnitt vorgestellten Techniken den Nachteil, dass das annotierte Dokument die Validierungsvorschriften von HTML 4.01 [16] und XHTML 1.0 [23] brechen.

## 2.2 Intern verlinkte Annotationen

Bei dieser Technik wird im annotierten Dokument lediglich ein Link auf eine externe Meta-Daten Beschreibung gespeichert. Die eigentlichen RDF Meta-Daten sind in einem eigenständigen Dokument gespeichert. Dabei wird davon

```

<head>
  <title>My Document</title>
  <script type="application/rdf+xml">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:dc="http://purl.org/dc/elements/1.1/">
      <rdf:Description rdf:about="http://www.w3.org/"
        dc:title="W3C Homepage"/>
    </rdf:RDF>
  </script>
</head>

```

**Abbildung 3.** Einbetten von RDF Meta-Daten im `<script>` Element.

ausgegangen, dass Programme die das Dokument verarbeiten auch Zugriff auf die Verlinkten Meta-Daten haben. Wenn es sich bei dem annotierten Dokument um eine Web-Seite handelt, stellt diese Voraussetzung kein Problem dar, da Web-Seiten im allgemeinen online verfügbar sind und zur Darstellung auch auf andere online Ressourcen wie Bilder, Stylesheets oder JavaScript Bibliotheken verwendet werden. Um offline zu arbeiten stellt diese Voraussetzung aber ein Hindernis dar.

Der Link wird durch das `<link>` Element im `<head>` der Web-Seite hergestellt. Abbildung 4 zeigt ein Beispiel für so einen Link. Bei dieser Technik handelt es sich um die Methode die auf der W3C RDF Issue Tracking List empfohlene Vorgehensweise.[17, 18]

```

<link rel="meta" type="application/rdf+xml"
      href="Meta-Data-for-Web-Page.rdf"/>

```

**Abbildung 4.** Verlinken auf eine externe Meta-Daten Beschreibung.

Das `rel` Attribut kann auch dazu verwendet werden um alternative Beziehungen auszudrücken, zum Beispiel um Clients die Meta-Daten in anderen RDF Serialisierungen anzubieten, wie in Abbildung 5 dargestellt ist.

```

<link rel="meta" type="application/rdf+xml" href="meta.rdf"/>
<link rel="alternate meta" type="application/n3" href="meta.n3"/>
<link rel="alternate meta" type="application/ntriples" href="meta.nt"/>

```

**Abbildung 5.** Anbieten von alternativen RDF Serialisierungen.

Alternativ kann auch das `data` Attribut im `<object>` Element dazu verwendet werden, um auf die Meta-Daten zu verlinken. Im Dokument `<body>` kann ein `<a>` Link dazu verwendet werden um auf die Meta-Daten zu verlinken, wie in Abbildung 6 gezeigt wird.

```
<body><p><a rel="meta" type="application/rdf+xml"
      href="meta.rdf">blargh</a>[...]
```

Abbildung 6. Verlinken aus dem Dokument `<body>`.

### 2.3 Extern verlinkte Annotationen

Will man ein Dokument annotieren, auf das man keine Zugriff hat, zum Beispiel weil das Dokument auf einem Web Server liegt, auf dem man keine Schreibrechte hat, so kann man die Meta-Daten in einem eigenständigen Dokument speichern und die Verknüpfung zwischen Meta-Daten und dem Dokument über einen externen Link herstellen. Die externen Links und die Meta-Daten werden auf einem Server öffentlich zugänglich gemacht. Der Link kann sich dabei auf das ganze Dokument beziehen, zum Beispiel der URL der Web-Seite, oder mittels XPointer [22] auf ein bestimmtes Fragment der Seite. Ein Programm das die Meta-Daten zu einem solchen Dokument verarbeiten soll, muss beim Server überprüfen, ob ein externen Link für dieses Dokument existiert und die entsprechenden Annotationen herunter laden.

## 3 Datenaufbereitung für die Semantischen Meta-Daten

Den Informationen in den semantischen Meta-Daten liegen meistens die Daten aus dem zu annotierenden Dokument zugrunde. Diese Daten können aber nicht in allen Fällen direkt als Meta-Daten weiterverwendet werden, sondern müssen erst nach den Erfordernissen der für die Annotation verwendeten Ontologien angepasst werden. In der folgenden Auflistung diskutieren wir Fälle, in denen eine solche Anpassung notwendig ist.

- **Unterschiedliche Granularität** – Die Granularität der Informationen in einem Dokument entspricht oft nicht der Granularität der Informationen die von der Ontologie für die Annotationen gefordert wird. Zum Beispiel ist auf einer Web-Seite die Telefonnummer ohne Trennzeichen zwischen der Landesvorwahl, der Ortsvorwahl und der lokalen Rufnummer dargestellt (+4155556729). Die zur Annotation verwendete Ontologie, verwendet aber getrennte Properties für die Speicherung der Teile der Telefonnummer (+41 / 555 / 56729). Beim Annotieren muss die Telefonnummer somit zuerst in ihre Bestandteile zerlegt werden, das heisst an die Granularität in der Ontologie angepasst werden.
- **Implizite Information** – Informationen die zur Annotation benötigt werden stehen unter Umständen nicht explizit zur Verfügung, können aber von den zur Verfügung stehenden Informationen abgeleitet werden. Zum Beispiel findet man auf der Web-Seite einer Theatervorstellung die Beginnzeit und die Dauer der Aufführung. Die Ontologie, die zur Annotierung der Seite verwendet wird, stellt zur Speicherung der Daten Properties für die

Beginn- und Endzeit zur Verfügung. Diese Daten sind in den zur Verfügung stehenden Informationen implizit enthalten und müssen beim Annotieren explizit gemacht werden.

- **Unterschiedliche Datenformate** – Die Informationen in einem Dokument und in den Meta-Daten haben unterschiedliche Zielgruppen und werden daher in einem unterschiedlichen Format repräsentiert. Das Dokument zielt auf menschliche Benutzer ab, während die Meta-Daten von Maschinen verarbeitet werden sollen. Zum Beispiel, ein Datum das auf einer Web-Seite für den Benutzer im Format "24. Dezember 2005, 12:00 Uhr" dargestellt wird, muss in den Meta-Daten im XML Schema `xsd:dateTime` Format dargestellt werden (`2005-12-24T12:00:00.000+01:00`). Das Datenformat der zur Verfügung stehen den Informationen muss in einem solchen Fall für die Meta-Daten in das von der Ontologie geforderte Datenformat übergeführt werden.

In den oben genannten Fällen müssen die Informationen aus dem zu annotierenden Dokument aufbereitet werden, um sie in der Meta-Daten Beschreibung verwenden zu können.

## 4 Annotationstools

Das semantische Annotieren eines Dokuments stellt einen zeitaufwändigen Prozess dar, der zusätzlich zum Erstellen des eigentlichen Dokuments durchgeführt werden muss. Auch ist die Person die ein Dokument annotiert, üblicherweise nicht die Person die von der Existenz der Annotationen profitiert. Das reduziert die Bereitschaft Dokumente zu annotieren. Daher werden Programme benötigt, die dem Benutzer den Annotierungsprozess so weit wie möglich zu erleichtern. An so ein Programm werden unterschiedlichste Anforderungen gestellt.

### 4.1 Anforderungen an ein Annotierungstool

Der komplexe Syntax von RDF/XML macht es Benutzern schwer RDF Meta-Daten in einem einfachen Texteditor zu verfassen. Ein Annotierungstool sollte die RDF Statements in Triple Form oder in einem graphischen Editor darstellen und automatisiert in den RDF/XML Syntax überführen. Der Benutzer sollte sich nicht um den Syntax kümmern müssen.

In RDF sind URIs allgegenwärtig. URIs werden dazu verwendet um XML Namespaces zu definieren, um Konzepte aus den Ontologien zu referenzieren und um Ressource identifizieren. Die häufige Verwendung von langen URIs bring ein großes Potential für Tippfehler mit sich. Annotierungstools sollten es dem Benutzer so weit wie möglich abnehmen URIs selbst einzugeben. Dies kann über einen Browser für die bekannten Namespaces und Ressource Identifikatoren erfolgen und durch einen Ontologie-Browser, in dem man ein

Konzept auswählen kann, ohne den URI eingeben zu müssen. Weiters sollte ein Annotierungstool sicherstellen, dass für eine Ressource immer der gleiche URI als Identifikator verwendet wird.

Betrachtet man Dokumente die semantisch annotiert sind, so stellt man fest, dass wichtige Teile des Dokuments zweimal gespeichert werden. Zum einem im eigentlichen Dokument für den menschlichen Benutzer, zum anderen in den Meta-Daten für die maschinelle Verarbeitung. Diese Redundanz führt zu einem Inkonsistenzproblem bei der Wartung des Dokuments. Ein Annotierungstool sollte unterstützen, dass Änderungen immer konsistent an beiden Arten von Informationen durchgeführt werden.

Ein Annotierungstool sollte auch die Anforderungen an die Datenaufbereitung, die in Abschnitt 3 diskutiert wurden, berücksichtigen.

Weiters sind viele Dokumente wie zum Beispiel Web-Seiten keine statischen Dokumente, sondern werden dynamisch generiert, zum Beispiel aus einer Datenbank. Beim Annotieren von dynamisch generierten Dokumenten müssen immer wieder die gleichen Schritte für ein bestimmtes Muster von Dokumenten durchgeführt werden. Ein Annotierungstool sollte es erlauben, dass man zur Annotierung von dynamischen Dokumenten einmalig ein Annotierungsmuster festlegt, sodass die Annotierung der Instanzen automatisiert erfolgen kann.

Weitergehende Annotierungstools sollten den Annotierungsprozess in den Erstellungsprozess des Dokuments integrieren. So ein Tool überwacht die Eingaben des Benutzers und bietet Vorschläge zur Annotation an. Zum Beispiel vergleicht ein solches Annotierungssystem die eingegebenen Zeichenketten mit den bereits vorhandenen Meta-Daten und erkennt, ob für die eingegeben Information bereits eine Instanz existiert und macht den Benutzer einen entsprechenden Annotierungsvorschlag.

Desweiteren gibt es Ansätze die auf maschinellen Lernen, nlp, Datamining, etc beruhen. todo

## 4.2 Beispiele für Annotierungstools

Das Bewusstsein für die oben angeführten Anforderungen an Annotierungstools hat sehr früh zu ersten Implementierungen geführt. Diese ersten Tools, wie zum Beispiel der SHOE Knowledge Annotator [9], konzentrierten sich hauptsächlich auf die syntaktisch korrekte Eingabe der Meta-Daten. Aktuelle Entwicklungen bewegen sich in die Richtung komplexer Annotierungs-Frameworks die über plug-ins an die Benutzeranforderungen angepasst werden können.

Bei den in diesem Abschnitt vorgestellten Annotierungstools handelt es sich um Forschungsprototypen, die Lösungsansätze für die oben angeführten Anforderungen aufzeigen. Eine ausführliche Liste mit Annotierungstools finden Sie unter [21].

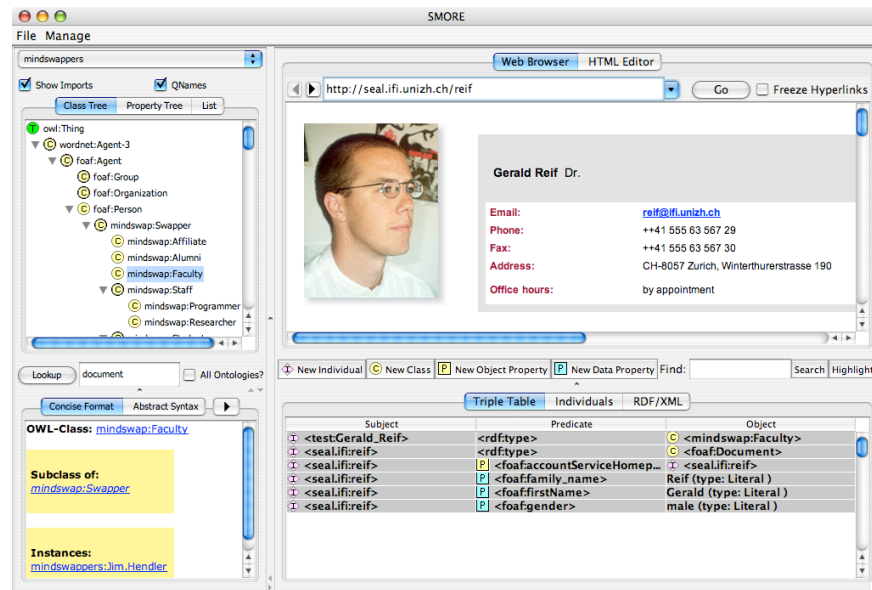


Abbildung 7. SMORE Annotationstool und Ontologie-Editor

## SMORE

SMORE [11] ist eine Java Applikation die im Rahmen des mindswap Projekts [13] entwickelt wird. SMORE bietet die Möglichkeit bestehende Web-Seiten via Drag and Drop zu annotieren und neue Web-Seiten im eingebauten HTML Editor zu erstellen. Der eingebaute Ontologie-Browser und Property-Editor ermöglicht es dem Benutzer Annotationen zu erstellen, ohne selbst URIs eingeben zu müssen. Weiters bietet SMORE die Möglichkeit die Ontologie um Klassen und Properties zu erweitern. Abbildung 7 zeigt links oben den Ontologie-Browser, darunter werden nähere Informationen zum Ausgewählten Ontologie Konzept angezeigt. Im Hauptframe auf der rechten Seite wird die zu annotierenden Web-Seite oder der HTML Editor angezeigt. Darunter werden die generierten RDF Statements dargestellt. SMORE unterstützt OWL Ontologien und speichert die generierten RDF Meta-Daten in einer eigenständigen Datei ab. Es wird keine Verknüpfung mit der Web-Seite hergestellt, wie dies in Abschnitt 2 diskutiert wurde.

## CREAM und der OntoMat-Annotizer

Bei CREAM [7] handelt es sich um ein umfangreiches Annotierungs Framework das verschiedene Annotierungsmethoden unterstützt. Der OntoMat-Annotizer ist eine Implementierung dieses Frameworks. Mit OntoMat können existierende Web-Seiten manuell annotiert werden. Ontomat bietet weiters



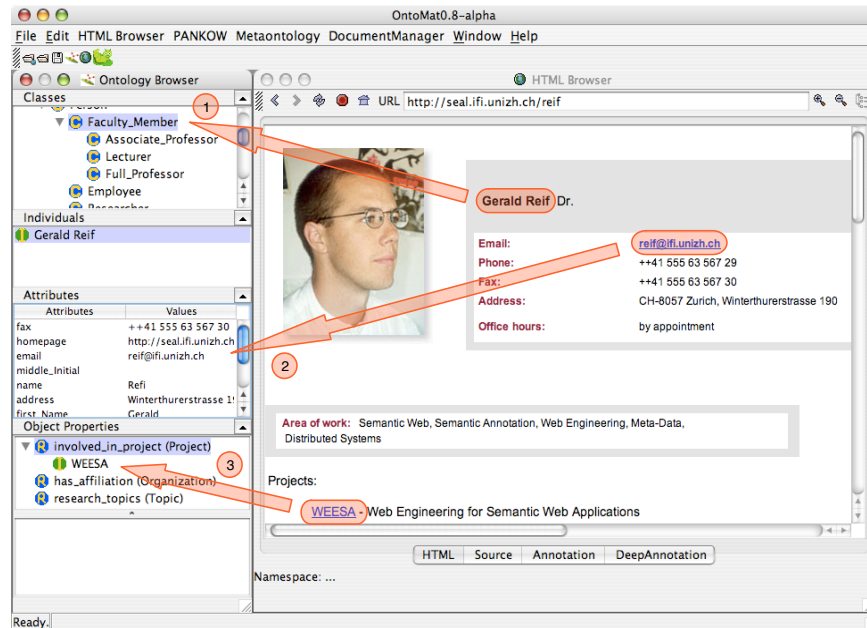


Abbildung 8. Der OntoMat-Annotizer

einen Editor der es erlaubt eine Web-Seite zu annotieren, während diese erstellt wird. Abbildung 8 zeigt das Erstellen einer Resource via Drag and Drop über den Ontologie-Browser (1), das Hinzufügen einer Properties (2) und Verknüpfen zweier Ressourcen (3). Die generierten Meta-Daten werden als XML Kommentar in die annotierte Web-Seite eingebaut. Die mit dem OntoMat erzeugten Annotationen werden mit Hilfe eines HTML Kommentars in die Web-Seite eingebettet.

Das semi-automatische Modul S-CREAM verwendet das Textextraktions-tool Amilcare um Texte automatisch zu annotieren. Amilcare lernt ausgehend von manuell annotierten Dokumenten Regeln die dann zur automatisierten Annotation versendet werden. Mit der "Deep Annotation" Methode von CREAM können auch dynamische Dokumente annotiert werden. Dabei werden Zuordnungsregeln zwischen der Datenbank und der verwendeten Ontologie erstellt. Diese Regeln werden dann verwendet, um ontologiebasierte Anfragen an die Datenbank stellen zu können. Die Zuordnungsregeln werden zur Zeit aber nicht dazu verwendet, um RDF Meta-Daten für die Web-Seite zu erstellen.

### Semi-automatische Tools

Neben den bisher vorgestellten manuellen Annotierungstools existieren auch semi-automatische Methoden die versuchen den Annotierungsprozess weitge-

hend zu automatisieren [5]. Die semi-automatischen Methoden basieren auf der Verwendung von Wrappern, wie diese in Kapitel ?? besprochen wurden und auf der Analyse von natürlichsprachigem Text (siehe Kapitel ??). Ein Beispiel für ein solches Annotierungstools ist das semi-automatische Modul S-CREAM [8] das im obigen Abschnitt beschrieben wurde.

Bei GATE [6] handelt es sich um ein Open Source Framework mit grafischer Benutzerschnittstelle zur Erstellung von Applikationen die natürlichsprachigen Text verarbeiten. Die Semantic Annotation Platform KIM [12] analysiert Texte und versucht Ressourcen (z.B. Personen, Organisationen, Orte, Zeiten) zu erkennen, diese mit bekannten Ressourcen zu vergleichen und daraus die Meta-Daten Beschreibung zu erzeugen. KIM baut auf dem GATE Framework auf.

Im nachfolgenden werden noch zwei weitere Systeme vorgestellt, bei denen es sich nicht direkt um Annotierungstools handelt.

### **Annotea**

Annotea [1, 10] ist ein W3C Projekt um Web-Seiten mit Kommentaren, Notizen, Erklärungen und anderen Bemerkungen zu versehen. Dabei werden die Annotationen nicht in die Web-Seite eingefügt, sondern auf einem Annotationsserver oder auf dem lokalen Rechner gespeichert. Mit der Hilfe von XPointern wird die Verknüpfung zur Annotierten Textpassage hergestellt. Diese Technik wurde in Abschnitt 2.3 vorgestellt. Bei den Annotationen handelt es sich um Texte die für menschliche Benutzer gedacht sind und sind daher für Maschinen schwer verarbeitbar. Als Datenrepräsentation wird RDF verwendet.

Annozilla [2] ist die Implementierung eines Annotea Clients für den Mozilla und Firefox Web-Browser. Abbildung 9 zeigt eine annotierte Web-Seite die geöffnete Side-Bar im Firefox Web-Browser.

### **Protège**

Bei Protège [15] handelt es sich um einen Ontologie Editor, der auch dazu verwendet werden kann um Instanzen zu erstellen. Protège bietet eine eigene Oberfläche um Instanzen neu anzulegen und zu verwalten. Abhängig von der Klasse zu der eine Instanz gehört, werden Eingabefelder für jene Properties angezeigt, die für diese Klasse definiert wurden. Diese Eingabemaske kann vom Benutzer konfigurieren werden. Weiters kann Protège durch Programmierung eines plug-ins an die eigenen Anforderungen angepasst werden. Die mit Protège erstellten Instanzen werden in einem eigenständigen Dokument abgespeichert.

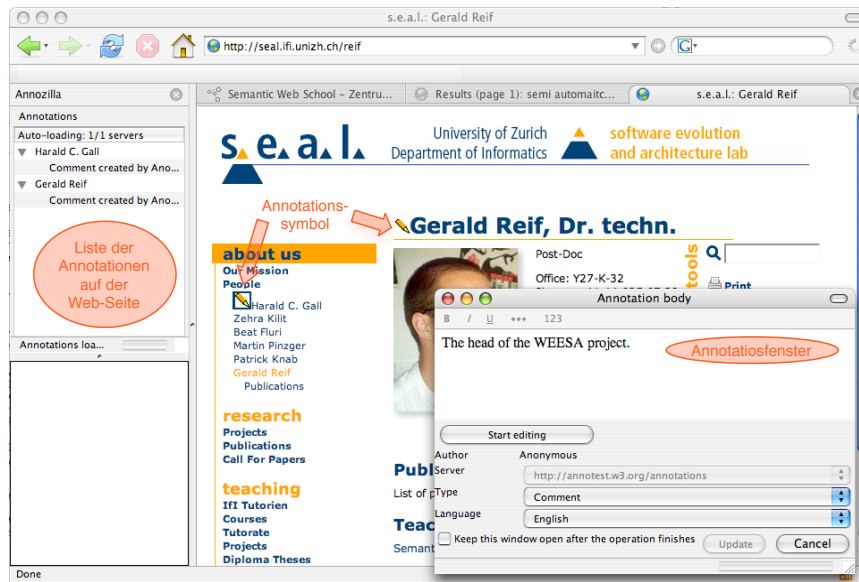


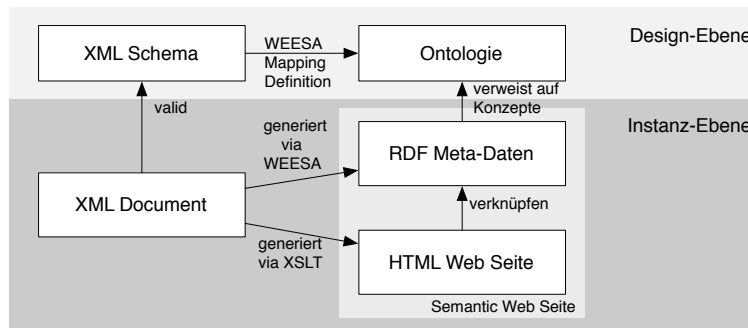
Abbildung 9. Annozilla Plug-in für den Mozilla Web-Browser.

## 5 Semantische Annotation beim Erstellen von Web Applikationen

In diesem Kapitel wurden bisher Ansätze vorgestellt um Web-Seiten zu annotieren, nachdem diese erstellt wurden. Es existieren jedoch auch Ansätze, den Annotierungsprozess bereits bei der Entwicklung der Web Applikation zu berücksichtigen. Dies bietet den Vorteil, dass beim Erstellen der Web Applikation üblicherweise Informationen in einen gut strukturierten Format vorliegen (zum Beispiel in einer Datenbank oder in XML Dokumenten). Diese Strukturinformation geht Großteils verloren, wenn die Daten als HTML Seite dargestellt werden. Aus diesem Grund wurden Techniken vorgeschlagen, den Annotierungsprozess in den Entwicklungsprozess einer Web Applikation zu integrieren.

WEESA [19] stellt eine Erweiterung für XML-basierte Web Entwicklungsmethoden dar, welche die Meta-Daten Generierung bereits in der Designphase berücksichtigt. XML-basierte Web Applikationen verwenden XSLT um XML Dokumente in HTML Web-Seiten zu transformieren. In WEESA werden diese XML Dokumente auch dazu verwendet um die RDF Meta-Daten Beschreibung für die Web-Seite zu generieren. Zu diesem Zweck wird ein Mapping von den Elementen und Attributen im XML Dokument zu den Konzepten in der Ontologie definiert. Dieses WEESA Mapping wird dann verwendet, um die RDF Meta-Daten aus dem XML Dokument zu erzeugen. Da das XML Dokument in WEESA sowohl als Datenquelle für die Meta-Datenerzeugung

und als auch für die Generierung der HTML Web Seite verwendet wird, ist sichergestellt, dass die Meta-Daten und die Web-Seite konsistent bleiben. Beim WEESA Mapping handelt es sich darüber hinaus nicht um ein einfaches eins-zu-eins Mapping, sondern erlaubt die Verwendung von Java Methoden zu Datenaufbereitung, wie dies in Abschnitt 3 diskutiert wurde.



**Abbildung 10.** Definition des WEESA Mappings in der Design-Ebene und Generierung der Meta-Daten in der Instanz-Ebene

In der Designphase der Web Applikation liegen üblicherweise noch keine XML Dokumente vor. Es ist jedoch die Struktur der XML Dokumente bekannt und in einem XML Schema definiert. Auf diese Information wird zurückgegriffen und in der Designphase das WEESA Mapping definiert. Im Betrieb der Web Applikation wird dieses Mapping dann verwendet, um aus XML Instanzen die HTML Seite und die RDF Meta-Daten zu generieren. Dieser Prozess ist in Abbildung 10 dargestellt.

## 6 Zusammenfassung

In diesem Kapitel wurden Fragen diskutiert, die das hinzufügen von Meta-Daten, die den Inhalt eines Dokuments in maschinen-verarbeitbarer Form beschreiben, betreffen. Dieser Vorgang wird *Semantische Annotation* genannt. Es wurden Techniken vorgestellt, mit denen das annotierte Dokument mit den zugehörigen Meta-Daten verknüpft wird, es wurden Probleme diskutiert, die sich bei der semantischen Annotation ergeben und es wurden die Anforderungen an Annotierungstools besprochen.

Bei den im Anschluss vorgestellten Annotierungstools handelt es sich um Forschungsprototypen die nicht für den Endanwender geeignet sind. Diese Tools zeigen jedoch Lösungsansätze auf, wie der Annotierungsprozess in künftige Anwendungen, wie zum Beispiel Content Management Systeme oder Dokument-Editoren, integriert werden kann. Es ist jedoch immer noch ein of-

fene Aufgabe Tools für den Endanwender zu entwickeln, die den Annotierungsprozess für den Anwender möglichst einfach gestalten und so den zusätzlichen Arbeitsaufwand minimieren.

Am Ende dieses Kapitels wurde weiters ein Ansatz vorgestellt, der den Annotierungsvorgang bereits in den Entwicklungsprozess einer Web-Applikation integriert wird. Dadurch ist es möglich die Struktur-Information der im Entwicklungsprozess vorliegenden Datenquellen (z.B. einer Datenbank) zum Annotieren der Web-Seiten zu nutzen. Dies erlaubt es, ohne grossen Mehraufwand Web-Applikationen semantisch zu annotieren.

## Literatur

1. Annotea project, Last visited November 2005. <http://www.w3.org/2001/Annotea/>.
2. The annozilla homepage, Last visited November 2005. <http://annozilla.mozdev.org/>.
3. Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer*. The MIT Press, Cambridge, Massachusetts; London, England, 2004.
4. Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific America*, 284(5):34–43, 2001.
5. Paul Buitelaar and Thierry Declerck. *Annotation for the Semantic Web*, volume 96 of *Frontiers in Artificial Intelligence and Applications*, chapter Linguistic Annotation for the Semantic Web, pages 93–111. IOS Press, Amsterdam, 2004.
6. Gate, a general architecture for text engineering, Last visited November 2005. <http://www.gate.ac.uk/>.
7. Siegfried Handschuh and Steffen Staab. Annotation of the shallow and the deep web. In Siegfried Handschuh and Steffen Staab, editors, *Annotation for the Semantic Web*, volume 96 of *Frontiers in Artificial Intelligence and Applications*, pages 25–45. IOS Press, Amsterdam, 2003.
8. Siegfried Handschuh, Steffen Staab, and Fabio Ciravegna. S-cream - semi-automatic creation of metadata. In *Proc. of the European Conference on Knowledge Acquisition and Management - EKAW-2002*, Madrid, Spain, October 1-4 2002.
9. Jeff Heflin and James Hendler. Searching the web with SHOE. In *Artificial Intelligence for Web Search. Papers from the AAAI Workshop*, pages 35–40, Menlo Park, CA, 2000. AAAI Press.
10. Josè Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the 10th International World Wide Web Conference*, pages 623–632, Hong Kong, Hong Kong, May 2001. ACM Press.
11. Aditya Kalyanpur, James Hendler, Bijan Parsia, and Jennifer Golbeck. SMORE - semantic markup, ontology, and RDF editor. Technical report, University of Maryland, 2003. <http://www.mindswap.org/papers/SMORE.pdf>.
12. Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Danyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2(1), 2005.

13. Mindswap project homepage, Last visited April 2005. <http://www.mindswap.org/>.
14. Sean B. Palmer. RDF in HTML: Approaches, June 2002. <http://infomesh.net/2002/rdfinhtml/index.html>.
15. The protégé ontology editor and knowledge acquisition system, Last visited November 2005. <http://protege.stanford.edu/>.
16. Dave Raggett, Arnaud Le Hors, and Ian Jacobs. HTML 4.01 specification. W3C recommendation, 24 December 1999. <http://www.w3.org/TR/html4/>.
17. W3C: Frequently Asked Questions about RDF: How do I put some RDF into my HTML pages?, September 2004. <http://www.w3.org/RDF/FAQ#How>.
18. W3C: RDF issue tracking: Issue faq-html-compliance: The suggested way of including RDF meta data in HTML is not compliant with HTML 4.01 or XHTML, January 2004. <http://www.w3.org/2000/03/rdf-tracking/#faq-html-compliance>.
19. Gerald Reif, Harald Gall, and Mehdi Jazayeri. WEESA - Web Engineering for Semantic Web Applications. In *Proceedings of the 14th International World Wide Web Conference*, pages 722–729, Chiba, Japan, May 2005.
20. RFC 3870: Application/rdf+xml media type registration. IETF RFC, September 2004. <http://www.ietf.org/rfc/rfc3870.txt>.
21. Semantic web - annotation & authoring. Homepage, Last visited November 2005. <http://annotation.semanticweb.org/>.
22. W3C Working Draft. *XML Pointer Language (XPointer)*, 16 August 2002. <http://www.w3.org/TR/xptr/>.
23. XHTML 1.0 the extensible hypertext markup language (second edition). W3C recommendation, 1 August 2002. <http://www.w3.org/TR/xhtml1/>.

---

## Index

Annotation, 1

Semantic Annotation, 1

Semantische Annotation, 1

