

Schließen auf Usability-Probleme mit OWL 2 RL

Manuel Dudda

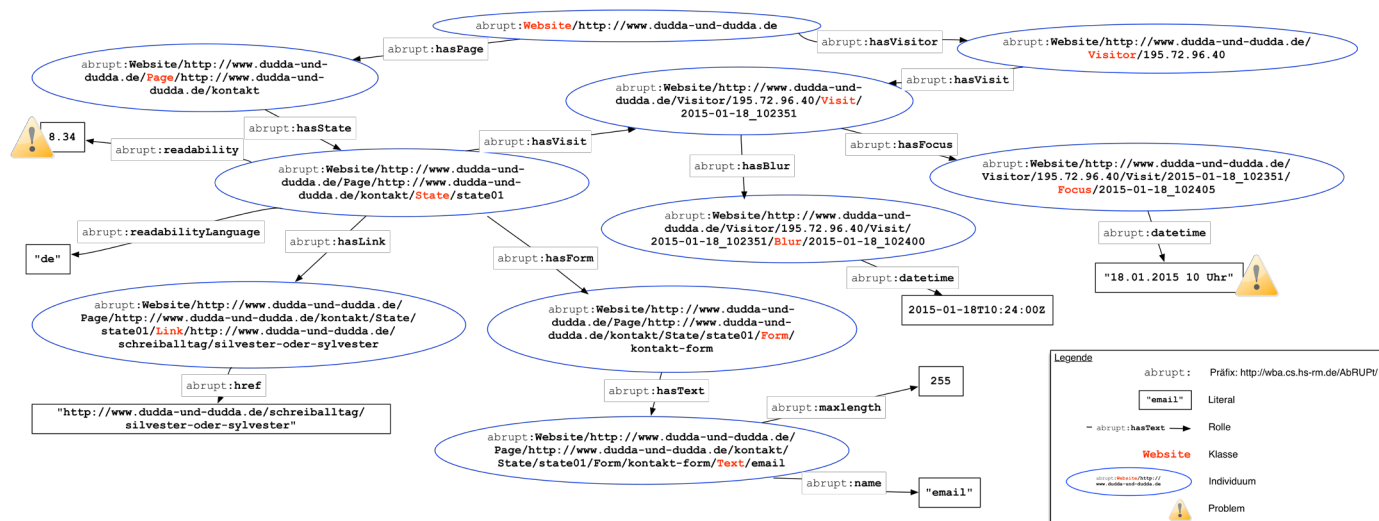


Fig. 1: Auszug eines Beispiel-Graphs

Modellierung & Vorgehensweise

Der Converter transformiert Website- und Besucherdaten in ein OWL 2 RL-Dokument. Dabei bezieht er terminologisches Basiswissen aus separat entwickelten Regeln und Vokabular (TBox).

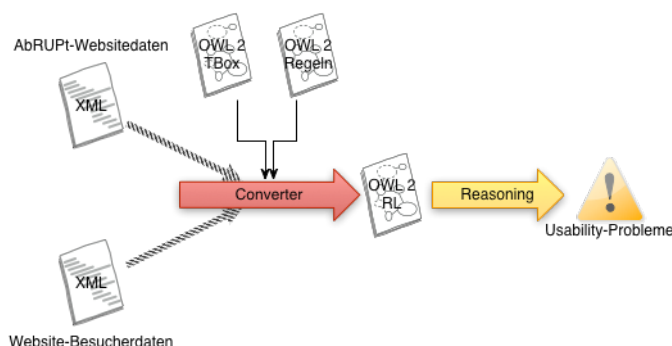


Fig. 2: Vorgehensweise

Die Klassen der Ontologie wurden gegenüber der Referenzarbeit wesentlich überarbeitet und erweitert. Für eine konfliktfreie URI-Struktur sorgt ein Nested-URL-Format.

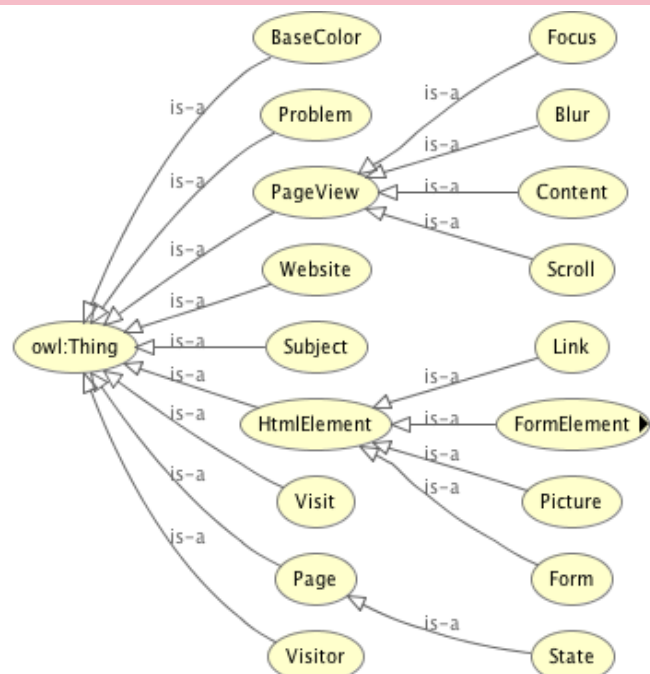


Fig. 3: Auszug der Klassenhierarchie

Bsp.: Inkonsistenz- und Datentypregeln

Für den Zeitstempel einer aufgezeichneten Besucheraktion ist das definierte Format `dateTime` aus dem XML-Schema angegeben. Das Dokument ist inkonsistent, falls ein anderer Datentyp vorliegt (Fig 1, rechts). Übersteigt der Lesbarkeitsindex den definierten Schwellwert von 8,0, liegt möglicherweise ein Usability-Problem vor (Fig 1, links).

```
@prefix abrupt: <http://wba.cs.hs-rm.de/AbRUpt/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

# Inkonsistenzregel
:datetime rdf:type owl:DatatypeProperty ;
           rdfs:range xsd:dateTime .

# Datentypregel
[ rdf:type owl:Restriction;
  rdfs:subClassOf abrupt:Problem;
  owl:onProperty abrupt:readability;
  owl:someValuesFrom [
    rdf:type rdfs:Datatype;
    owl:onDatatype xsd:decimal;
    owl:withRestrictions ( [ xsd:minInclusive 8.0] )
  ]
]
```

Listing 1: Inkonsistenz- und Datentypregel in turtle-Syntax

Bsp.: Produktions- und Listenregeln

Aus der Produktionsregel folgt ein Problem, wenn eine Seite kein HTML-Element beinhaltet. Ein HTML-Element kann ein Link, Bild oder Formular sein.

Aus der unten stehenden Kettenregel werden die hasState-Rollenaxiome von der jeweiligen Seite zu der zugehörigen Website abgeleitet.

```
# Produktionsregel
[ rdf:type owl:Restriction ;
  rdfs:subClassOf abrupt:Problem ;
  owl:onProperty abrupt:hasHtmlElement ;
  owl:cardinality 0
] .

# Listenregel
abrupt:hasState rdf:type owl:ObjectProperty;
  owl:propertyChainAxiom ( abrupt:hasPage abrupt:hasState ) .
```

Listing 2: Produktions- und Listenregel in turtle-Syntax

Prädikatenlogik 1. Stufe	Beschreibungslogik
$\forall x. readability(x, y) \wedge (y \geq 8.0) \rightarrow Problem(x)$	$readability.(\geq 8.0) \sqsubseteq Problem$
$name(x, agb) \rightarrow required(x, true)$	nicht möglich

Beschreibungslogik

OWL 2 RL basiert auf der Mächtigkeit der Beschreibungslogik $\mathcal{SHOIN}(\mathcal{D})$. Sie bildet eine Untermenge der Prädikatenlogik 1. Stufe. Im Gegensatz zu dieser ist das Schlussfolgern (Reasoning) entscheidbar. Durch kleine Einschränkungen gestaltet sich die Regelerstellung nicht ganz einfach und für die volle Ausdrucksstärke muss auf Erweiterungen zurückgegriffen werden.

PRO

- + Reasoning ist entscheidbar
- + Interoperabilität Dokumentena
- + fundierte theori Grundlage

KONTRA

- keine Rollen-Inferenz durch gegebene Klassenzugehörigkeiten
- keine Meta-Modellierung
- keine eingebauten Funktionen +, -, /, *, substr(), ...

Zusammenfassung & Fazit

Die Usability von Internetauftritten wird größtenteils subjektiv wahrgenommen. Es existiert zwar eine Normierung durch die ISO für die Anforderungen an die Gebrauchstauglichkeit einer Software (EN ISO 9241-11), diese trifft jedoch nur sehr vage Aussagen und orientiert sich stark am respektiven Nutzungskontext. Trotzdem gibt es Merkmale einer Website, die allgemein als negativ für die Eigenschaften der Usability eingestuft werden können, so zum Beispiel schlechte Kontrastverhältnisse, ein hoher Lesbarkeitssindex und Diskrepanzen zwischen erwartetem und tatsächlichem Nutzerverhalten. Das „AbRUPT“-System ist in der Lage, viele dieser Merkmale einer Website automatisiert zu erfassen. Durch die Konvertierung, Modellierung und Entwicklung eines durchdachten Regelsystems in OWL 2 RL entsteht ein System, welches Folgerungen auf Usability-Probleme ermöglicht. Doch das Format zeigt Grenzen auf. Im Allgemeinen ist die Regelerstellung und das Nachvollziehen des Reasonings für große Datenbestände gegenüber klassischen Programmier-Paradigmen eher unübersichtlich. Die Empfehlung des W3C ist noch relativ jung und findet eine stetig wachsende Community. Vor allem durch die gute Interoperabilität sowie Skalierbarkeit wird OWL 2 seine Daseinsberechtigung besitzen. Dieses Projekt kann die Prognose unterstreichen.