Kapitel WT:VIII (Fortsetzung)

VIII. Semantic Web

- WWW heute
- Semantic Web Vision
- RDF: Einführung
- RDF: Konzepte
- RDF: XML-Serialisierung
- RDF: Anwendungen
- RDFS: Einführung
- RDFS: Konzepte
- Semantik im Web
- Semantik von RDF/RDFS
- Ontologien
- OWL: Konzepte
- OWL: Logikhintergrund
- OWL: Anwendungen

©STEIN 2005-2014

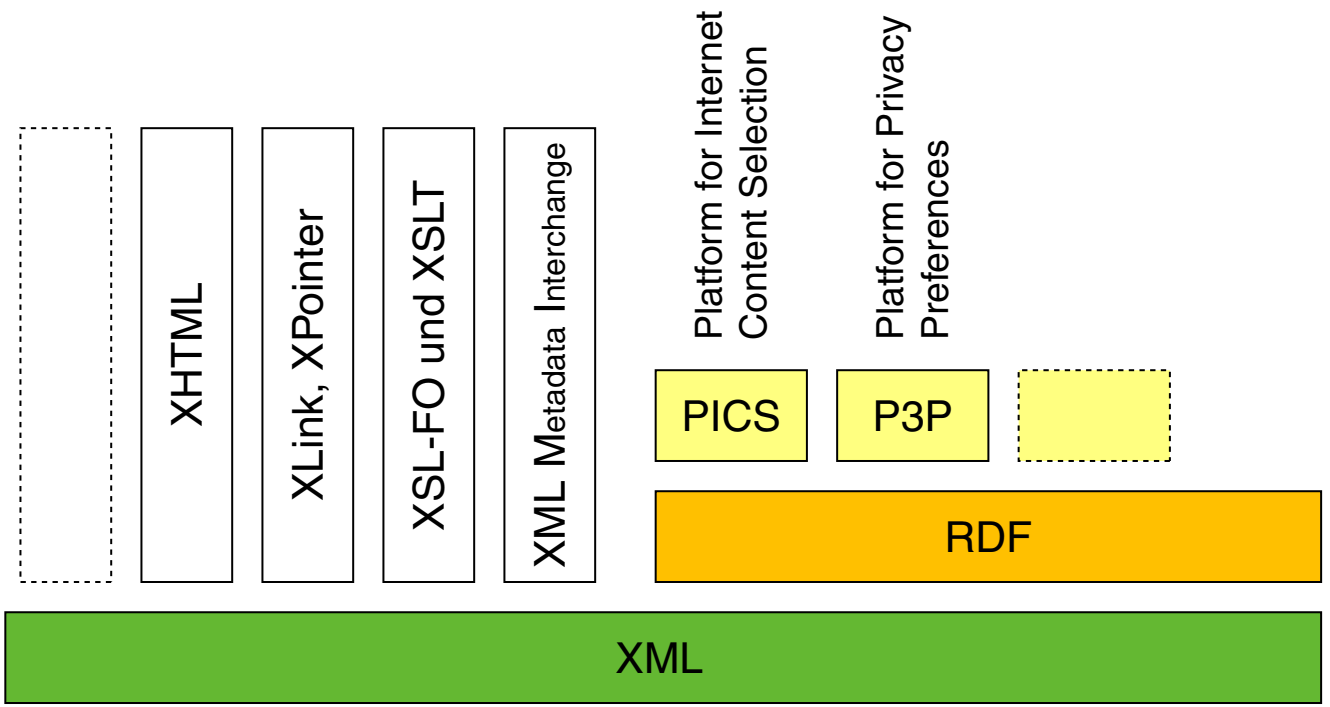
RDF: Einführung

©STEIN 2005-2014

RDF: Einführung

"The Resource Description Framework (RDF) is a general-purpose language for representing information in the Web."

[W3C]



[vgl. Hitzler 2005]

©STEIN 2005-2014

RDF: Einführung

Historie

1998 Berners-Lee. Plan für die Architektur eines Semantic Web.

1999 RDF Schema-Spezifikation. Proposed Recommendation.

2001 RDF/XML Syntax-Spezifikation. Working Draft.

2001 RDF Semantik-Spezifikation. Working Draft.

2002 OWL Web Ontology Language. Working Draft.

2004 RDF/XML Syntax Specification. [W3C]

2004 RDF Semantics. [W3C]

2004 RDF Vocabulary Description Language 1.0: RDF-Schema. [W3C]

2004 OWL. Web Ontology Language. [W3C]

2009 OWL 2. Web Ontology Language. [W3C]

©STEIN 2005-2014

Bemerkungen:

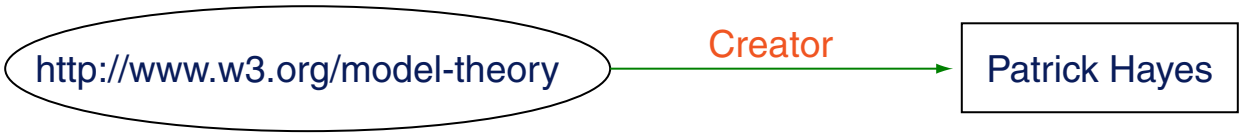
- Die OWL Web Ontology Language hat verschiedene Vorgänger, u.a. die DAML+OIL Ontologie-Sprache, die in der Historie nicht genannt sind. OWL wird als Revision und Weiterentwicklung dieser (zum Teil noch eingesetzten) Ontologie-Sprachen angesehen; wichtiges Ziel ist die Entwicklung eines einheitlichen Standards.

©STEIN 2005-2014

RDF: Einführung

Realistischer Use-Case für das Semantic Web

"The creator of resource `http://www.w3.org/model-theory` *is Patrick Hayes."*

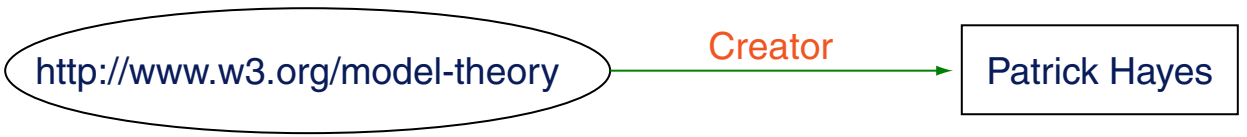


©STEIN 2005-2014

RDF: Einführung

Realistischer Use-Case für das Semantic Web

"The creator of resource `http://www.w3.org/model-theory` *is Patrick Hayes."*



Es gibt viele Möglichkeiten, diesen Sachverhalt in XML zu formulieren:

- ❑
```
<Creator>
  <uri>http://www.w3.org/model-theory</uri>
  <name>Patrick Hayes</name>
</Creator>
```

- ❑
```
<Document uri="http://www.w3.org/model-theory">
  <Creator>Patrick Hayes</Creator>
</Document>
```

- ❑
```
<Document uri="http://www.w3.org/model-theory"
    Creator="Patrick Hayes"/>
```

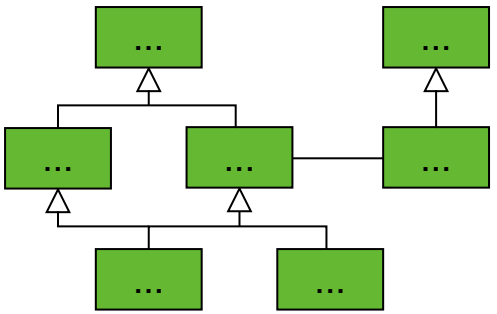
➜ Einigung auf das gleiche XML-Schema.

©STEIN 2005-2014

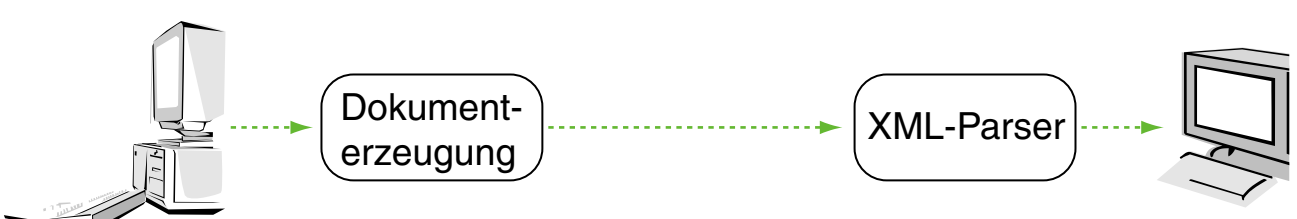
RDF: Einführung

Realistischer Use-Case für das Semantic Web (Fortsetzung)

"Möven sind Vögel.
Vögel sind Wirbeltiere.
Wirbeltiere sind Tiere.
..."



Konzeptuelles
Modell der Domäne



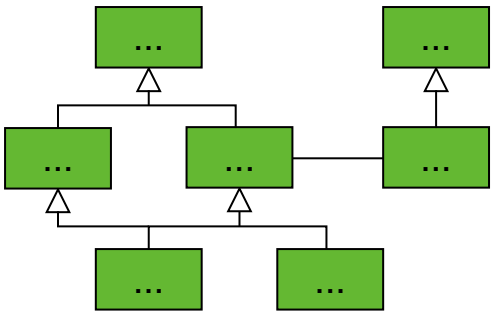
[vgl. Hitzler 2005]

©STEIN 2005-2014

RDF: Einführung

Realistischer Use-Case für das Semantic Web (Fortsetzung)

"Möven sind Vögel.
Vögel sind Wirbeltiere.
Wirbeltiere sind Tiere.
..."



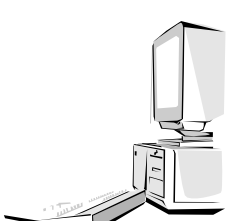
Konzeptuelles
Modell der Domäne

Abbildung

XML-Schema

```
<xsd:schema xmlns:xsd="http://..
      <xsd:annotation> Schema-A
      ...
</xsd:schema>
```

Anwendung



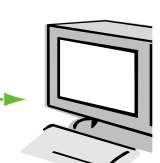
Dokument-
erzeugung

Verwendung
von Schema A

✓

XML-Parser

Verwendung
von Schema A



[vgl. Hitzler 2005]

©STEIN 2005-2014

Bemerkungen:

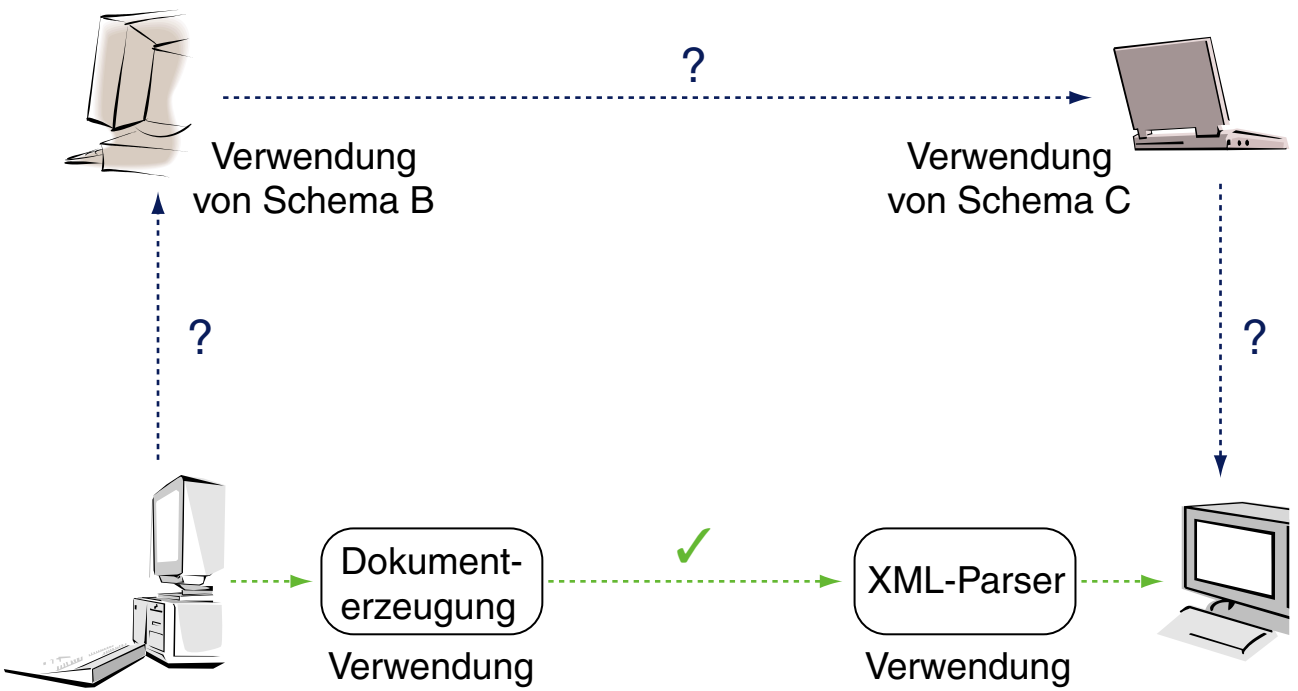
- Sender und Empfänger des Dokumentes haben sich auf eine gemeinsame Semantik verständigt: ihre (XML-basierte) Kommunikation greift auf das gleiche XML-Schema zurück.

©STEIN 2005-2014

RDF: Einführung

Realistischer Use-Case für das Semantic Web (Fortsetzung)

Viele Kommunikationspartner im World Wide Web sind einander zunächst unbekannt und haben sich *nicht* auf eine gemeinsame Semantik verständigt.



[vgl. Hitzler 2005]

©STEIN 2005-2014

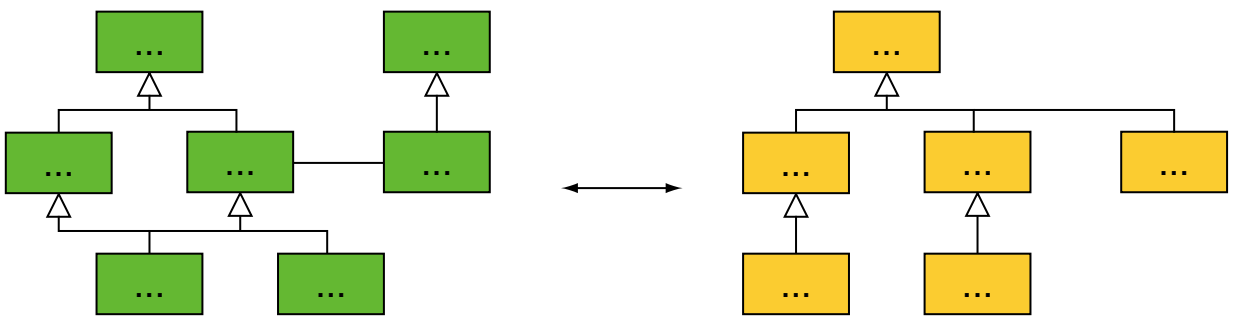
RDF: Einführung

Realistischer Use-Case für das Semantic Web (Fortsetzung)

Notwendige Schritte zur Modell- und Dokumentangleichung:

```
<xsd:schema xmlns:xsd="http://..
      <xsd:annotation> Schema-A
      ...
</xsd:schema>
```

```
<xsd:schema xmlns:xsd="http://..
      <xsd:annotation> Schema-B
      ...
</xsd:schema>
```

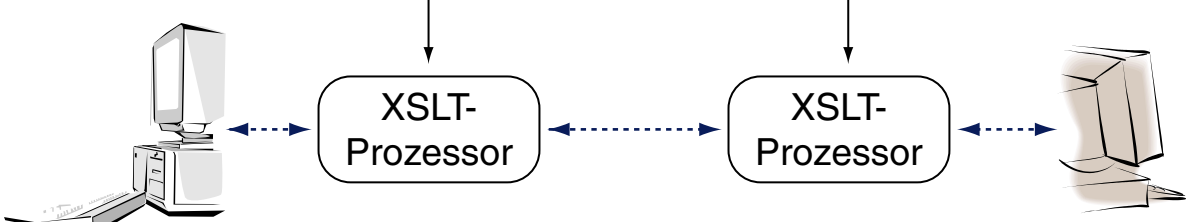


Abgleich des konzeptuellen Modells

```
<xsl:stylesheet ...
      <xsl:template match
      ...
</xsl:stylesheet>
```

```
<xsl:stylesheet ...
      <xsl:template match
      ...
</xsl:stylesheet>
```

Programmieren d. Transformations-spezifikation



Transformation der Dokumente

©STEIN 2005-2014

Bemerkungen:

- Der Abgleich konzeptueller Modelle ist schwierig; die Programmierung der Transformationsspezifikation ist kompliziert.
- Ausweg ist die Definition einer weiteren Sprachschicht oberhalb von XML, dem Resource Description Framework, RDF. Auf Basis seiner formalisierten Semantik soll RDF es ermöglichen, notwendige oder sinnvolle Anpassungen, Schematransformationen etc. automatisch zu erschlussfolgern und durchführen zu können.

©STEIN 2005-2014

RDF: Einführung

Anforderungen an RDF

RDF ist als Sprachschicht für Wissensrepräsentation im World Wide Web vorgesehen. Anforderungen an eine solche Sprachschicht:

- geeignet, um Daten und Metadaten zu beschreiben
- „Domänen-neutral“ – d.h., es werden keine Annahmen über einen Anwendungsbereich gemacht
- formulierbar (auch) in XML
- einfach und erweiterbar, um möglichst viele Anwendergruppen zu erreichen
- mit Mechanismen, um verteilte Ressourcen im als auch außerhalb des World Wide Web zu spezifizieren
- mit Konzepten zur Beschreibung von Metawissen, um Wissen, Meinungen und Aussagen *über* Wissen zu formulieren

©STEIN 2005-2014

RDF: Einführung

Einordnung von RDF

- XML

  Stellt die Syntax für strukturierte Dokumente bereit; keine Formulierung von Constraints bzgl. der Semantik von XML-Dokumenten möglich.

- XML-Schema

  Eine Sprache, um die Struktur von XML-Dokumenten vorzuschreiben; erweitert insbesondere auch die XML-Datentypen.

©STEIN 2005-2014

RDF: Einführung

Einordnung von RDF

- XML

  Stellt die Syntax für strukturierte Dokumente bereit; keine Formulierung von Constraints bzgl. der Semantik von XML-Dokumenten möglich.

- XML-Schema

  Eine Sprache, um die Struktur von XML-Dokumenten vorzuschreiben; erweitert insbesondere auch die XML-Datentypen.

- RDF

  Eine universelle Sprache (besser: Datenmodell) zur kanonischen Beschreibung von Ressourcen.

- RDF-Schema

  Vokabular zur Beschreibung von Klassenbeziehungen und Eigenschaften für Ressourcen; eine Art einfache Ontologie-Sprache.

©STEIN 2005-2014

RDF: Einführung

Einordnung von RDF

- XML
  Stellt die Syntax für strukturierte Dokumente bereit; keine Formulierung von Constraints bzgl. der Semantik von XML-Dokumenten möglich.
- XML-Schema
  Eine Sprache, um die Struktur von XML-Dokumenten vorzuschreiben; erweitert insbesondere auch die XML-Datentypen.

- RDF
  Eine universelle Sprache (besser: Datenmodell) zur kanonischen Beschreibung von Ressourcen.
- RDF-Schema
  Vokabular zur Beschreibung von Klassenbeziehungen und Eigenschaften für Ressourcen; eine Art einfache Ontologie-Sprache.

- OWL
  Vokabular, um eigene Terminologien für Ressourcen zu beschreiben: Relationen zwischen Klassen, Kardinalitäts-Constraints, Eigenschaftstypen, aufzählbare Klassen, etc.

©STEIN 2005-2014

Bemerkungen:

- XML-Tags sind eine Art „semantisches Markup“, wenn für menschliche Leser sinnvolle Bezeichner gewählt sind; aus Maschinensicht ist diese Semantik nicht gegeben. Der Einsatz von XML soll in erster Linie der maschinellen Generierung und Verarbeitung von Dokumenten dienen, z.B. zwischen verschiedenen Firmen und Anwendungen. Ein spezieller und begrenzter Ausweg sind branchenspezifische Festlegungen von gemeinsamen Vokabularen, DTDs und XML-Schemata.

- Die Verwendung von RDF, RDFS oder OWL geschieht durch die Formulierung von RDF-Statements (= SPO-Tripel), deren Elemente aus den RDF-, RDFS- und OWL-Vokabularen und der Menge aller URIs stammen. Für die Terme der Vokabulare ist eine bestimmte Semantik *intendiert*. Solche Vokabulare sollten möglichst klein sein: intendierte Semantik erwächst aus dem gemeinsamen Verständnis bzw. der gleichartigen Interpretation (= Operationalisierung) der Terme bei allen Verwendern des Vokabulars.

©STEIN 2005-2014

Bemerkungen: (Fortsetzung)

- Die Vokabulare, einschließlich Kommentar, Angabe der Signatur bei Prädikat-Ressourcen, Angabe der Oberklasse bei Subjekt-Ressourcen für RDF und RDFS, finden sich in den zugehörigen Namensräumen http://www.w3.org/1999/02/22-rdf-syntax-ns# bzw. http://www.w3.org/2000/01/rdf-schema#.

- Die Semantik der aus dem RDF-, RDFS- und OWL-Vokabularen gebauten Sätze bzw. Graphen ist in den Empfehlungen des W3C auf Basis der Modelltheorie definiert. Die Durchführung von Schlussfolgerungsprozessen und die Operationalisierung der Semantik geschieht in den Anwendungsprogrammen. Hintergründe und grundlegende Darstellung aus modelltheoretischer Sicht findet sich in www.w3.org/TR/rdf-mt.

- Aus Web-Browser-Sicht handelt es sich bei RDF-, RDFS- oder OWL-Code um XML-Code, der lediglich wohlgeformt sein muss.

©STEIN 2005-2014

RDF: Konzepte

Konzepte von RDF:

1. graphbasiertes Datenmodell
2. verwendete URIs gehören mit zum Vokabular
3. Datentypen
4. Zeichenkonstanten (Literale)
5. Syntax zur Serialisierung in XML
6. Basiskonstrukt zur Formulierung von Aussagen *(Statements)*
7. Inferenz auf Basis der Statements

[W3C]

©STEIN 2005-2014

RDF: Konzepte

RDF-Datenmodell

Elemente des RDF-Datenmodells:

1. Ressource

2. Property

3. Statement

Statement = (Ressource , Property (Ressource) , Wert (Ressource oder Literal))

Statement = (Subjekt , Prädikat , Objekt)

©STEIN 2005-2014

RDF: Konzepte

RDF-Datenmodell

Elemente des RDF-Datenmodells:

1. Ressource
   Etwas, das eine URI hat.

2. Property
   Eigenschaft einer Ressource. Die Property selbst ist auch eine Ressource, die ihrerseits wieder Eigenschaften haben kann.

3. Statement
   Tripel bestehend aus einer Ressource R, die beschrieben wird, einer Property von R, sowie die Ausprägung von R hinsichtlich der Property. Die drei Elemente eines Statements werden *Subjekt*, *Prädikat* und *Objekt* genannt.

Statement = (Ressource , Property (Ressource) , Wert (Ressource oder Literal))

Statement = (Subjekt , Prädikat , Objekt)

©STEIN 2005-2014

RDF: Konzepte

RDF-Datenmodell

Elemente des RDF-Datenmodells:

1. Ressource — Im Graph: Knoten
   Etwas, das eine URI hat.

2. Property — Im Graph: Kantenbeschriftung
   Eigenschaft einer Ressource. Die Property selbst ist auch eine Ressource, die ihrerseits wieder Eigenschaften haben kann.

3. Statement — Im Graph: gerichtete Kante
   Tripel bestehend aus einer Ressource R, die beschrieben wird, einer Property von R, sowie die Ausprägung von R hinsichtlich der Property. Die drei Elemente eines Statements werden *Subjekt*, *Prädikat* und *Objekt* genannt.

Statement = (Ressource , Property (Ressource) , Wert (Ressource oder Literal))

Statement = (Subjekt , Prädikat , Objekt)

©STEIN 2005-2014

Bemerkungen:

- RDF definiert ein Datenmodell für Objekte und Relationen zwischen diesen Objekten. Das Datenmodell von RDF ist sehr einfach; es basiert nur auf den Konzepten Ressource, Property und Statement.
- Datenmodelle in der Informatik dienen zur Erfassung und Darstellung der Informations*struktur*, nicht der Information selbst. Datenmodelle definieren die erlaubten Konzepte, um Information zu beschreiben; sie abstrahieren von einer konkreten Syntax.
  [vgl. DB:II relationales Datenmodell]

©STEIN 2005-2014

Bemerkungen: (Fortsetzung)

- Eine Ressource kann alles mögliche sein: ein real existierender Gegenstand, genauso wie eine Idee, eine Meinung oder etwas anderes, dem man eine URI zugeordnet hat. Beachte den Unterschied zwischen URI *(Universal Resource Identifier)* und URL *(Universal Resource Locator)*.
- Das Subjekt und das Prädikat eines Statements müssen eine Ressource oder ein anonymer Knoten *(Blank Node)* sein. Das Objekt kann eine Ressource, ein anonymer Knoten oder ein Literal sein.
- Andere Bezeichnungen für ein Statement sind: RDF-Tripel, Objekt-Attribut-Wert-Tripel, OAW- bzw. OAV-Tripel, SPO-Tripel, Fact
- Ein Statement definiert Rollen. Beachte, dass eine Ressource verschiedene Rollen einnehmen kann: als Subjekt, als Prädikat oder als Objekt. Es kann zur Verwirrung führen, dass mit „Ressource“ je nach Zusammenhang einmal die Rolle „Subjekt“ bezeichnet ist, das andere Mal etwas, dem eine URI zugeordnet werden kann (s.o.). Die letztere Verwendung ist grundsätzlicher.
- Das Datenmodell von RDF zeigt eine enge Verwandtschaft zu den Frame-Systemen und den semantischen Netzen aus dem Bereich der Künstlichen Intelligenz.

©STEIN 2005-2014

RDF: Konzepte

RDF-Datenmodell

Statement:

(a) *"The creator of resource* `http://www.w3.org/model-theory` *is Patrick Hayes."*

(b) *"Resource* `http://www.w3.org/model-theory` *has the creator Patrick Hayes."*

- Ressource bzw. Subjekt: `http://www.w3.org/model-theory`
- Property bzw. Prädikat: `http://www.w3.org/#Creator`
- Wert bzw. Objekt: „Patrick Hayes"

©STEIN 2005-2014

RDF: Konzepte

RDF-Datenmodell

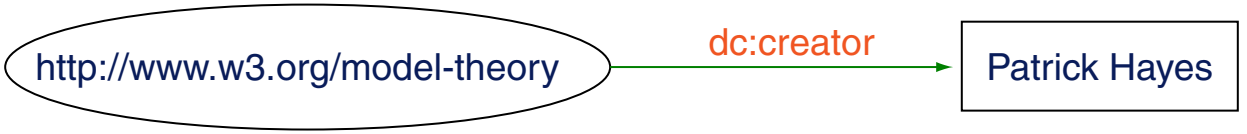
Statement:

(a) *"The creator of resource* `http://www.w3.org/model-theory` *is Patrick Hayes."*

(b) *"Resource* `http://www.w3.org/model-theory` *has the creator Patrick Hayes."*

- Ressource bzw. Subjekt: `http://www.w3.org/model-theory`
- Property bzw. Prädikat: `http://www.w3.org/#Creator`
- Wert bzw. Objekt: „Patrick Hayes"

Darstellung als gerichteter Graph:



Darstellung als Prädikat:

Creator(`http://www.w3.org/model-theory`, „Patrick Hayes")

©STEIN 2005-2014

Bemerkungen:

- Bei Aussage (b) ist das RDF-Subjekt auch das grammatikalische Subjekt des Satzes; bei Aussage (a) ist das grammatikalische Subjekt *"The creator"*. Beachte die Rolle von Agens und Patiens (siehe Lexikon unter culturitalia.uibk.ac.at/hispanoteca).
- Ressourcen werden durch Ellipsen dargestellt; Literale werden durch Kästchen dargestellt.
- In RDF ist nur die Definition binärer Prädikate möglich.
- In RDF gehört die Menge der URIs (innerhalb eines Graphen, innerhalb einer Anwendung) mit zum Vokabular.

©STEIN 2005-2014

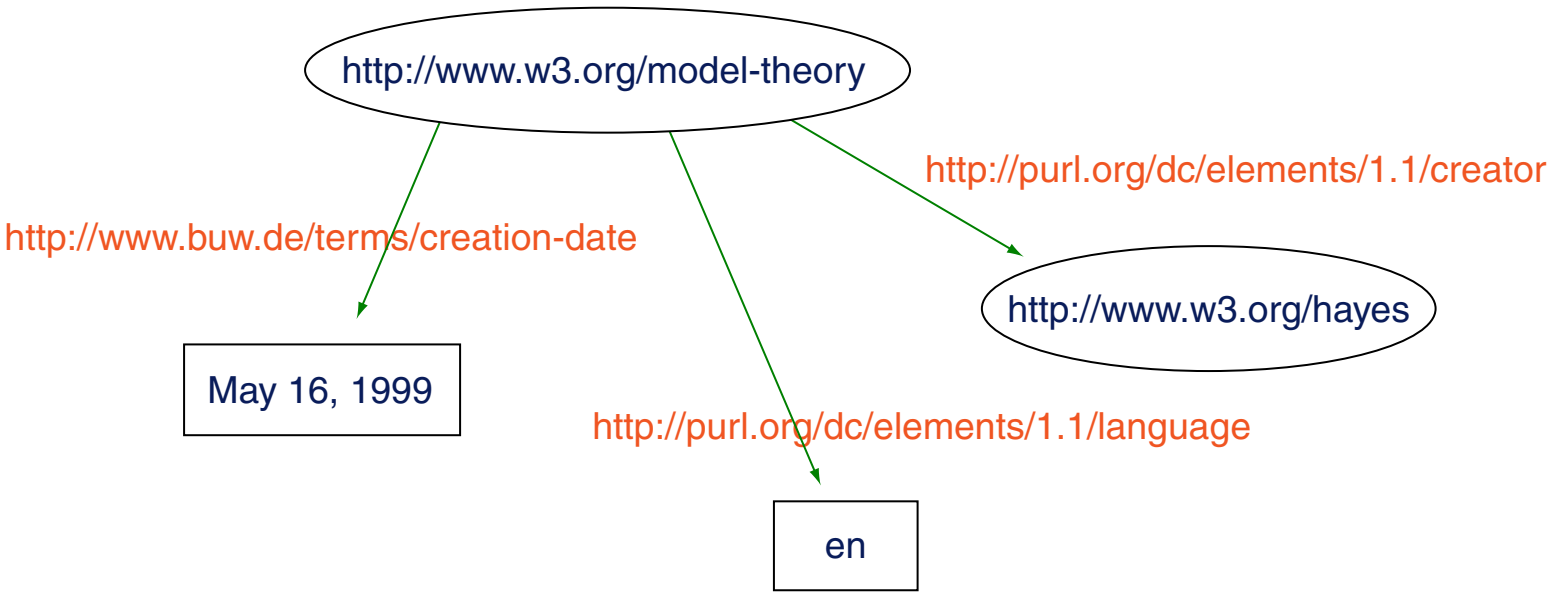
RDF: Konzepte

Semantisches Netz

Gleiche Elemente in Mengen von Statements werden unifiziert und bilden einen eventuell zusammenhängenden Graph.

Statements:

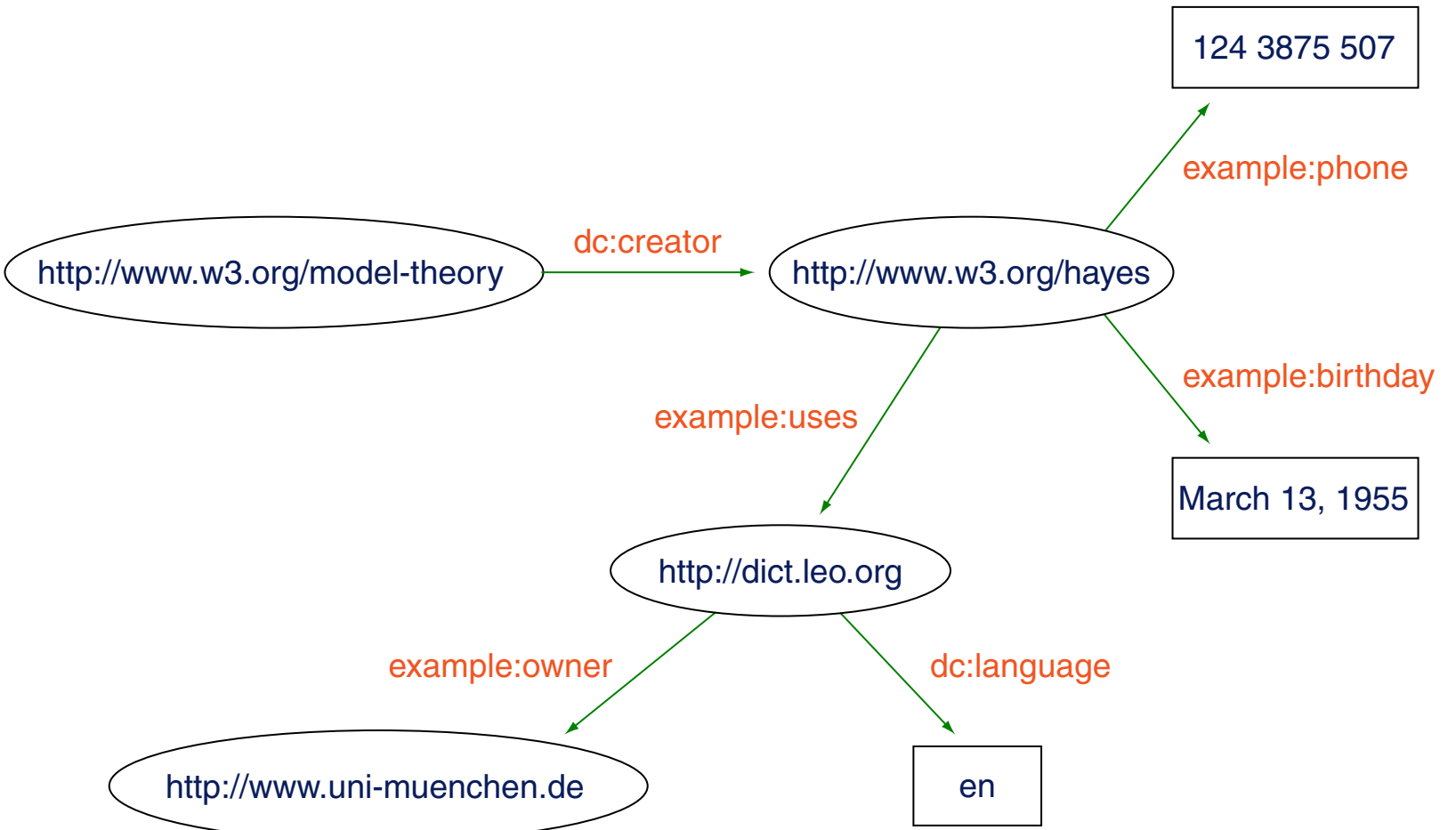
"The creator of `http://www.w3.org/model-theory` *is* `http://www.w3.org/hayes`*.*
`http://www.w3.org/model-theory` *has a creation-date whose value is May 16, 1999.*
`http://www.w3.org/model-theory` *has a language whose value is English."*



©STEIN 2005-2014

RDF: Konzepte

Semantisches Netz



©STEIN 2005-2014

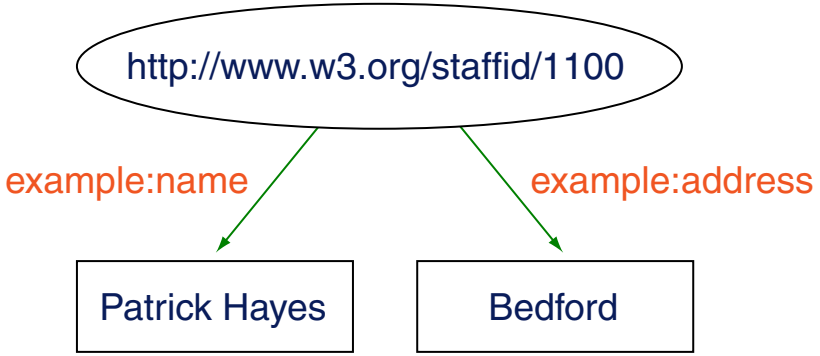
Bemerkungen:

- Der Prefix `dc:` steht für die Namensraum-URI http://purl.org/dc/elements/1.1/.
- Der Prefix `example:` steht für die Namensraum-URI `http://www.buw.de/example/`.

©STEIN 2005-2014

RDF: Konzepte

Binarization [W3C]

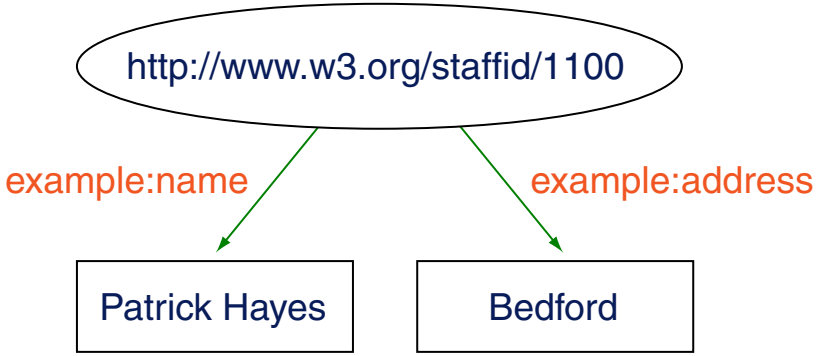


Als logische Formel: $\alpha = \mathit{Name}(1100, \text{"Hayes"}) \ \wedge \ \mathit{Address}(1100, \text{"Bedford"})$

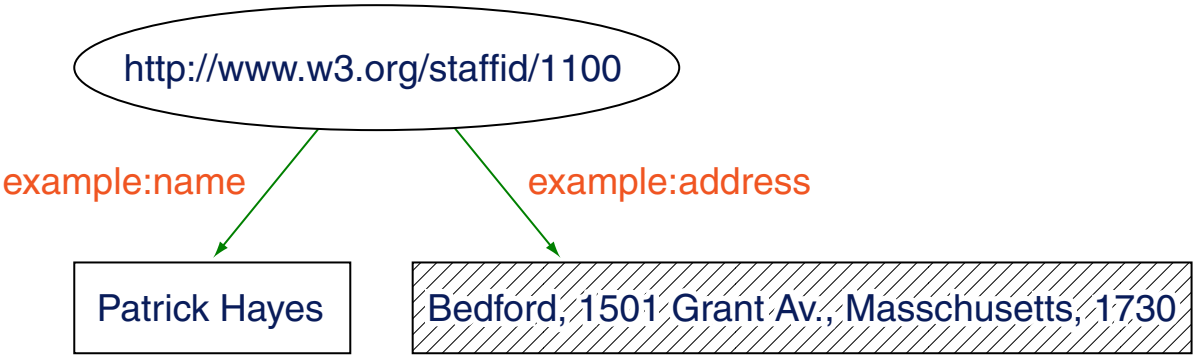
©STEIN 2005-2014

RDF: Konzepte

Binarization [W3C]



Als logische Formel: $\alpha = \mathit{Name}(1100, \text{"Hayes"}) \ \wedge \ \mathit{Address}(1100, \text{"Bedford"})$

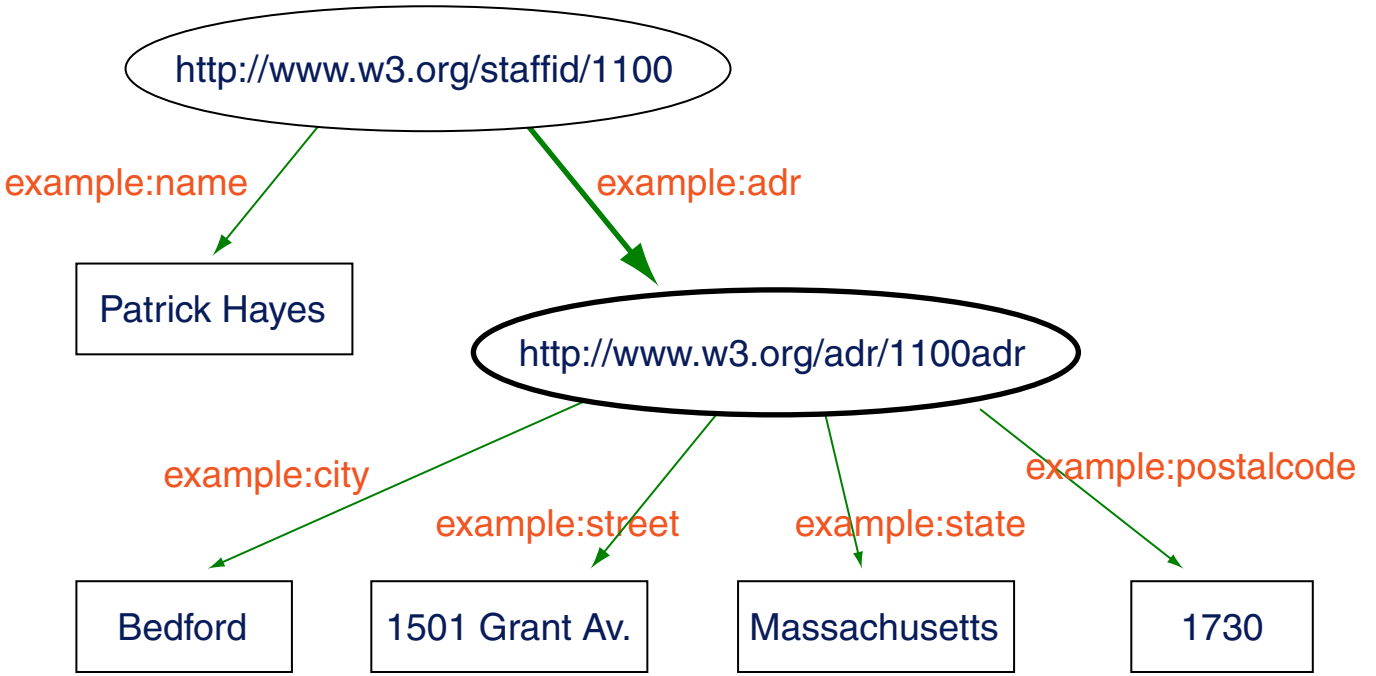


$$\beta = \ \mathit{Name}(1100, \text{"Hayes"}) \ \wedge$$
$$\mathit{Address}(1100, \text{"Bedford"}, \text{"1501 Grant Av."}, \text{"Massachusetts"}, 1730)$$

©STEIN 2005-2014

RDF: Konzepte

Binarization [W3C]



$$\gamma = \; \mathit{Name}(1100, \text{"Hayes"}) \;\wedge\; \mathit{Adr}(1100, 1100\text{adr}) \;\wedge$$
$$\mathit{City}(1100\text{adr}, \text{"Bedford"}) \;\wedge\; \mathit{Street}(1100\text{adr}, \text{"1501 Grant Av."}) \;\wedge$$
$$\mathit{State}(1100\text{adr}, \text{"Massachusetts"}) \;\wedge\; \mathit{PostalCode}(1100\text{adr}, 1730)$$

β und γ sind logisch äquivalent: $\beta \approx \gamma$

©STEIN 2005-2014

Bemerkungen:

- Die Reformulierung n-stelliger Prädikate als zweistellige Prädikate *(Binarization)* entspricht der Transformation eines Hypergraphen in einen ordinären Graphen. In einem Hypergraph kann eine Kante mehr als zwei Knoten verbinden.
- Beachte auch die Verwandtschaft zur Normalformbildung in relationalen Datenbankschemata.
- Binarization erfordert die Generierung eines neuen Knotens bzw. einer neuen Ressource. Diese Knoten dürfen anonym, also unbenannt bleiben – in RDF-Terminologie als Blank Node bezeichnet.

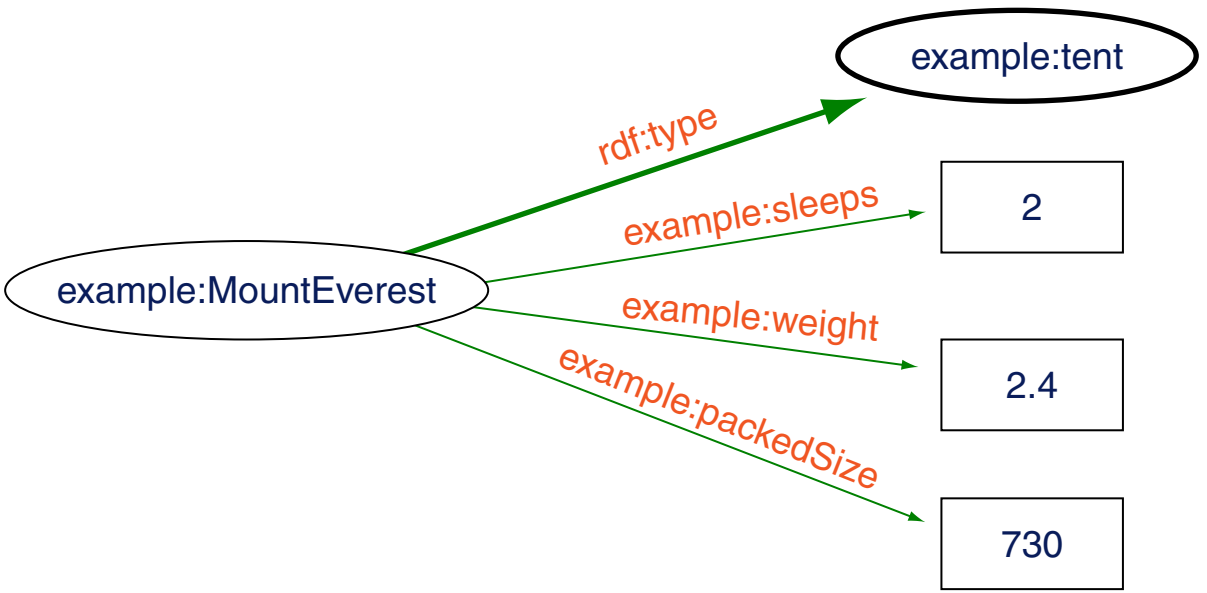
©STEIN 2005-2014

RDF: Konzepte

Getypte Ressourcen [W3C]

Mit der Property `rdf:type` wird ein Statement zur Angabe einer *Klasse* formuliert, zu der eine Ressource gehört.

- Der Wert (bzw. das Objekt) des Statements ist eine Web-Ressource, die eine Klasse repräsentiert.
- Zu einer Ressource sind mehrere `rdf:type`-Statements möglich.



©STEIN 2005-2014

Bemerkungen:

- Der Prefix `rdf:` steht für die Namensraum-URI http://www.w3.org/1999/02/22-rdf-syntax-ns#.
- Die Klassenangabe `rdf:type` besitzt nur deklarativen Charakter innerhalb eines RDF-Modells. Es ist Aufgabe der Anwendung, die ein RDF-Modell verarbeitet, die intendierte Semantik zu operationalisieren.

©STEIN 2005-2014

RDF: Konzepte

Container [W3C]

Container in RDF dienen dazu, eine Menge von Ressourcen oder Literalen in ihrer Gesamtheit zu behandeln. Für drei Container-Typen sind entsprechende Terme im RDF-Vokabular definiert:

- `rdf:Bag`
  Ungeordnete Liste von Ressourcen oder Literalen mit Mehrfachvorkommen.
- `rdf:Seq`
  Geordnete Liste von Ressourcen oder Literalen mit Mehrfachvorkommen.
- `rdf:Alt`
  Menge von alternativen Ressourcen oder Literalen ohne Mehrfachvorkommen.

Container stellen getypte Ressourcen dar, deren weitere Eigenschaften generische Properties sind. Die Werte der generischen Properties sind die Inhalte des Containers.

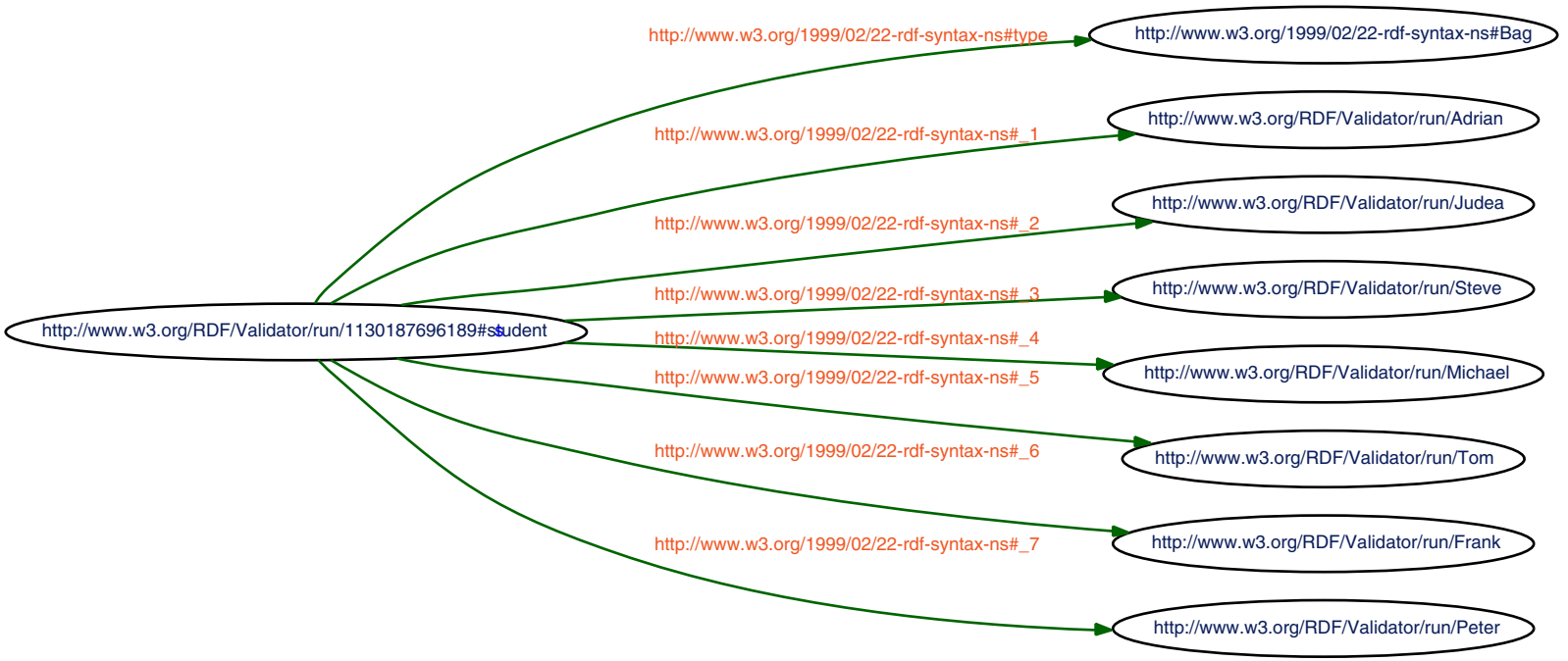
©STEIN 2005-2014

RDF: Konzepte

Container [Serialization]

Beispiel:

„Die Studenten (einer Vorlesung) sind Adrian, Judea, Steve, Michael, Tom, Frank, Peter.“



©STEIN 2005-2014

Bemerkungen:

- Die Bezeichner der generischen Properties sind entweder `rdf:_1`, `rdf:_2`, etc. oder alternativ `rdf:li`.
- Die drei Containertypen `rdf:Bag, rdf:Seq, rdf:Alt` besitzen nur deklarativen Charakter innerhalb eines RDF-Modells. Es ist Aufgabe der Anwendung, die ein RDF-Modell verarbeitet, die intendierte Semantik zu operationalisieren:

  "It is important to understand that while these types of containers are described using predefined RDF types and properties, any special meanings associated with these containers, e.g., that the members of an Alt container are alternative values, are only intended meanings. These specific container types, and their definitions, are provided with the aim of establishing a shared convention among those who need to describe groups of things."

  [W3C]

- Der Graph wurde mit dem W3C RDF-Validator automatisch aus einer XML-Serialisierung des RDF-Modells erzeugt. Siehe www.w3.org/RDF/Validator

©STEIN 2005-2014

RDF: Konzepte

Reification [W3C] [Serialization]

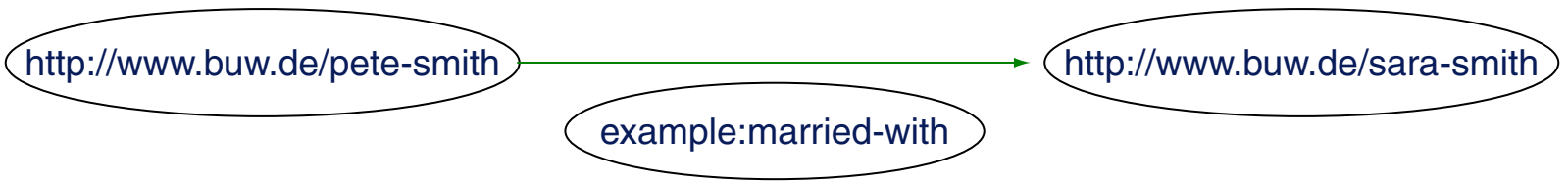
Idee: Für ein Statement wird eine neue Ressource erstellt, die wiederum in einem anderen Statement verwendet werden kann.

©STEIN 2005-2014

RDF: Konzepte

Reification [W3C] [Serialization]

Idee: Für ein Statement wird eine neue Ressource erstellt, die wiederum in einem anderen Statement verwendet werden kann.

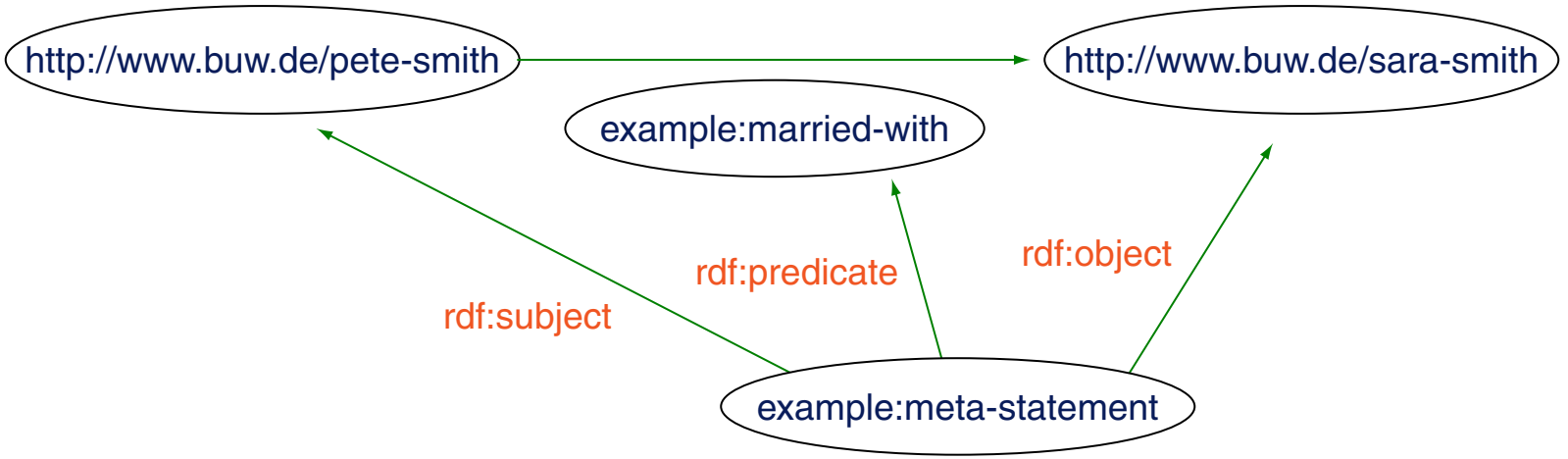


©STEIN 2005-2014

RDF: Konzepte

Reification [W3C] [Serialization]

Idee: Für ein Statement wird eine neue Ressource erstellt, die wiederum in einem anderen Statement verwendet werden kann.

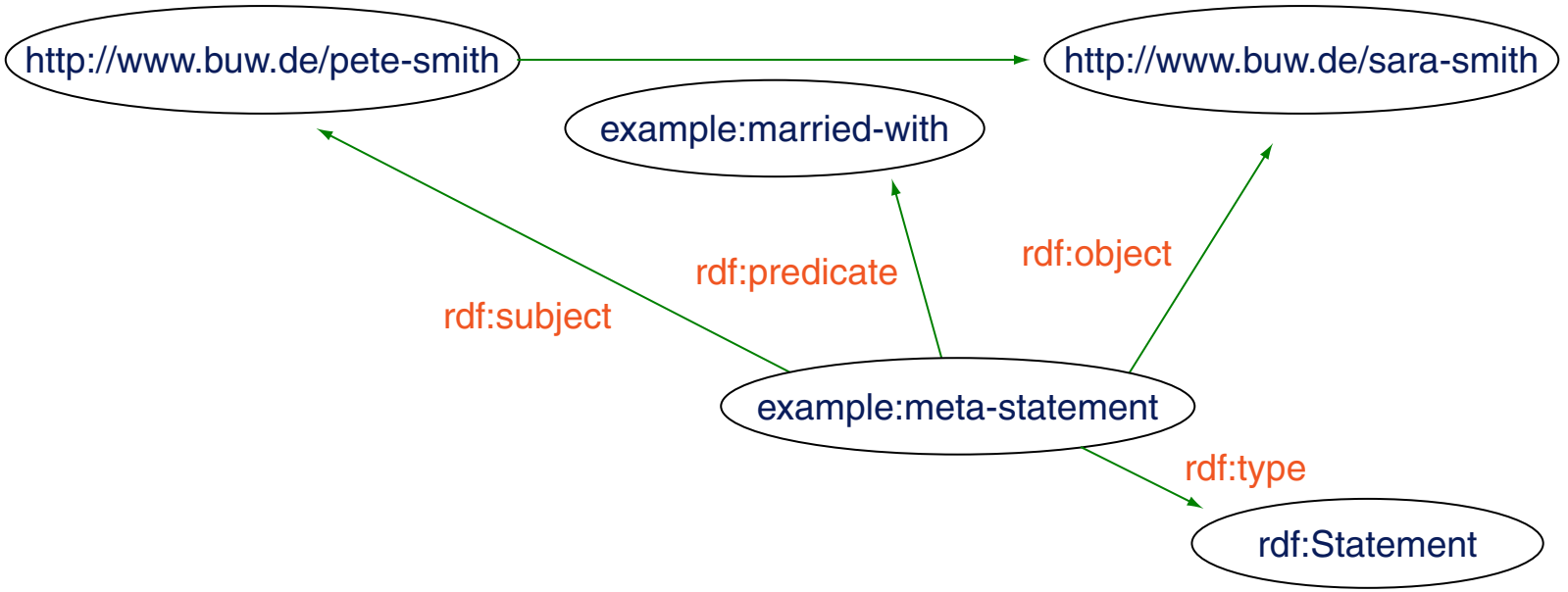


©STEIN 2005-2014

RDF: Konzepte

Reification [W3C] [Serialization]

Idee: Für ein Statement wird eine neue Ressource erstellt, die wiederum in einem anderen Statement verwendet werden kann.

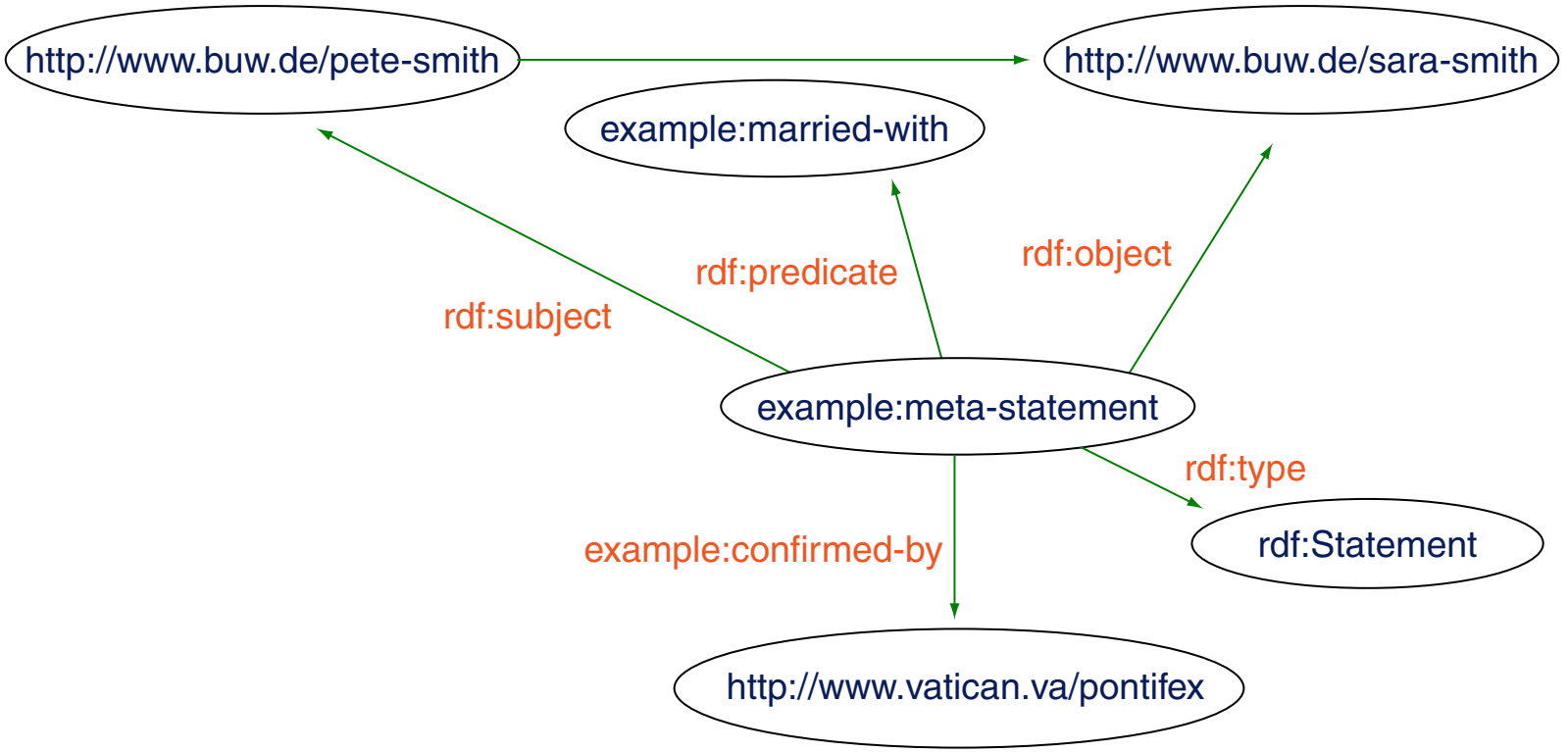


©STEIN 2005-2014

RDF: Konzepte

Reification [W3C] [Serialization]

Idee: Für ein Statement wird eine neue Ressource erstellt, die wiederum in einem anderen Statement verwendet werden kann.



©STEIN 2005-2014

Bemerkungen:

- Reification ist ein Mechanismus, um *Metawissen* zu formulieren:

  “The RDF reification vocabulary consists of the type rdf:Statement, and the properties rdf:subject, rdf:predicate, and rdf:object.” [W3C]

- Reification kann man mit „Vergegenständlichung“ übersetzen: ein Statement wird vergegenständlicht in dem Sinne, dass es Subjekt oder Objekt eines anderen Statements wird.

©STEIN 2005-2014

RDF: XML-Syntax

Definition 1 (RDF Document)

An RDF document is a serialization of an RDF graph into a concrete syntax.

[W3C]

RDF/XML ist eine Möglichkeit zur Serialisierung von RDF. W3C-Syntax:

- Ein RDF-Dokument speichert ein `<rdf:RDF>`-(Wurzel)element.
- Das `<rdf:RDF>`-Element enthält `<rdf:Description>`-Elemente.
- `<rdf:Description>`-Elemente *(Node Elements)* beschreiben Ressourcen.
- `<rdf:Description>`-Elemente können <*Property*>-Elemente enthalten.
- <*Property*>-Elemente wiederum können genau *ein* `<rdf:Description>`-Element oder Literal enthalten.

Der Abstieg in `<rdf:Description>`-Elemente entspricht dem Entlanggehen eines Pfades im zugehörigen RDF-Graph.

©STEIN 2005-2014

Bemerkungen:

- Eine Property beschriftet eine Kante, die im Graph zu genau einem Knoten führt. Deshalb enthalten <*Property*>-Elemente genau ein `<rdf:Description>`-Element oder Literal.
- Verwendete Namensräume (`rdf:`, `dc:`, etc.) müssen deklariert sein.
- Lange Zeichenketten lassen sich mit `Entity`-Deklarationen abkürzen.
- Es gibt eine Standardschreibweise und wenige syntaktische Varianten.
- Die XML-Serialisierung fügt dem RDF-Datenmodell (natürlich) keine weitere Semantik hinzu.

©STEIN 2005-2014

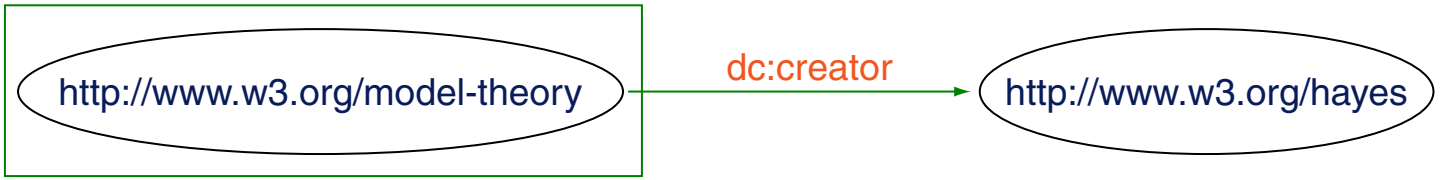
RDF: XML-Syntax

Prinzip

©STEIN 2005-2014

RDF: XML-Syntax

Prinzip

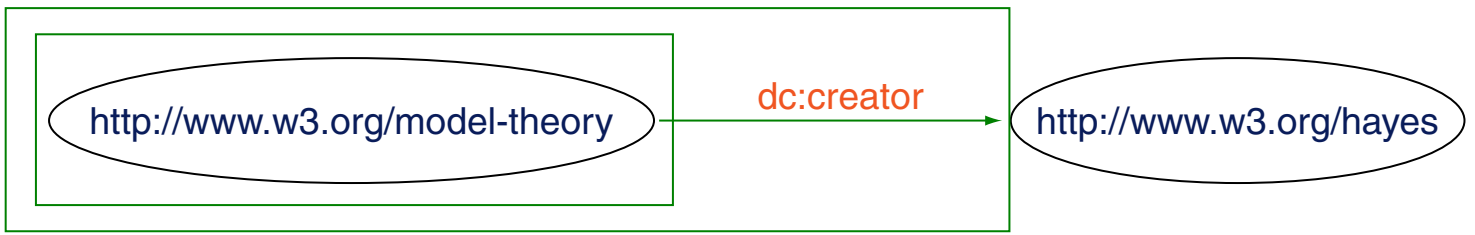



```
<rdf:Description rdf:about="http://www.w3.org/model-theory">

</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Prinzip

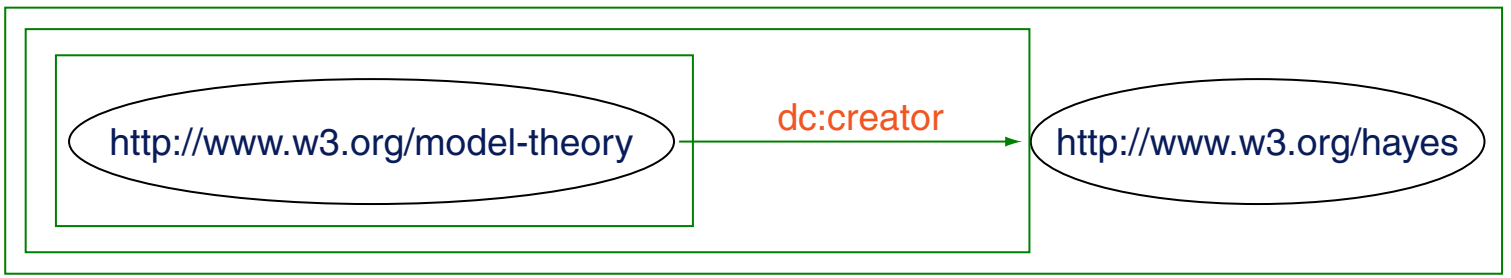



```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
   <dc:creator>



   </dc:creator>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Prinzip [W3C]




```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator>
    <rdf:Description rdf:about="http://www.w3.org/hayes">
    </rdf:Description>
  </dc:creator>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Objekt: Serialisierung als Kindelement oder Attribut

(a) Objekt als `<rdf:Description>`-Kindelement im Property-Element:

```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator>
    <rdf:Description rdf:about="http://www.w3.org/hayes">
    </rdf:Description>
  </dc:creator>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Objekt: Serialisierung als Kindelement oder Attribut

(a) Objekt als `<rdf:Description>`-Kindelement im Property-Element:

```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator>
    <rdf:Description rdf:about="http://www.w3.org/hayes">
    </rdf:Description>
  </dc:creator>
</rdf:Description>
```

≈ (b) Objekt als `rdf:resource`-Attribut im Property-Element:

```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator rdf:resource="http://www.w3.org/hayes" />
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Objekt: Ressource oder Literal

Objekt ist eine Ressource (eindeutig):

```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator>
    <rdf:Description rdf:about="http://www.w3.org/hayes">
    </rdf:Description>
  </dc:creator>
</rdf:Description>
```


©STEIN 2005-2014

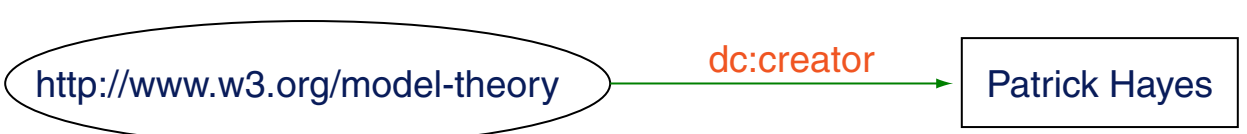
RDF: XML-Syntax

Objekt: Ressource oder Literal

Objekt ist eine Ressource (eindeutig):

```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator>
    <rdf:Description rdf:about="http://www.w3.org/hayes">
    </rdf:Description>
  </dc:creator>
</rdf:Description>
```

Objekt ist ein Literal (muss nicht eindeutig sein):




```
<rdf:Description rdf:about="http://www.w3.org/model-theory">
  <dc:creator>Patrick Hayes</dc:creator>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Gemeinsame Vorgängerknoten

Kanten, die im Graph des RDF-Modells von derselben Ressource ausgehen (= Statements mit demselben Subjekt) können als *einzelne* `<rdf:Description>`-Elemente oder *gemeinsam* serialisiert werden:

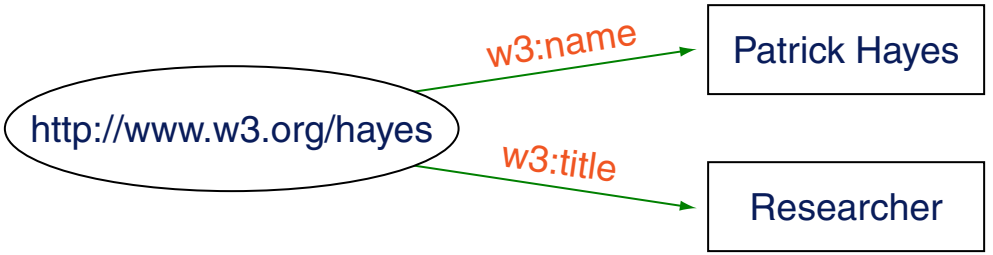
(a)
```
<rdf:Description rdf:about="http://www.w3.org/hayes">
  <w3:name>Patrick Hayes</w3:name>
</rdf:Description>

<rdf:Description rdf:about="http://www.w3.org/hayes">
  <w3:title>Researcher</w3:title>
</rdf:Description>
```

≈ (b)
```
<rdf:Description rdf:about="http://www.w3.org/hayes">
  <w3:name>Patrick Hayes</w3:name>
  <w3:title>Researcher</w3:title>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

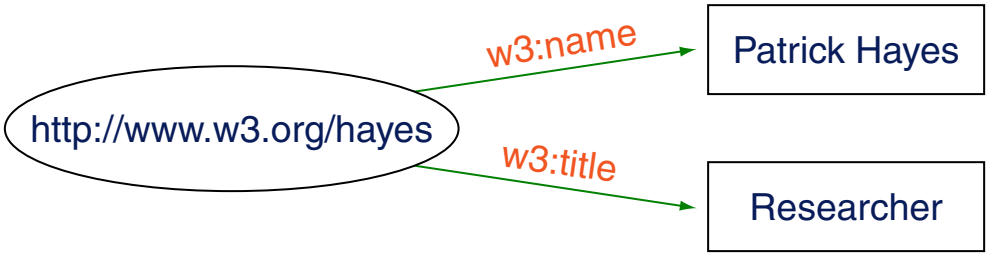
Beispiel



©STEIN 2005-2014

RDF: XML-Syntax

Beispiel

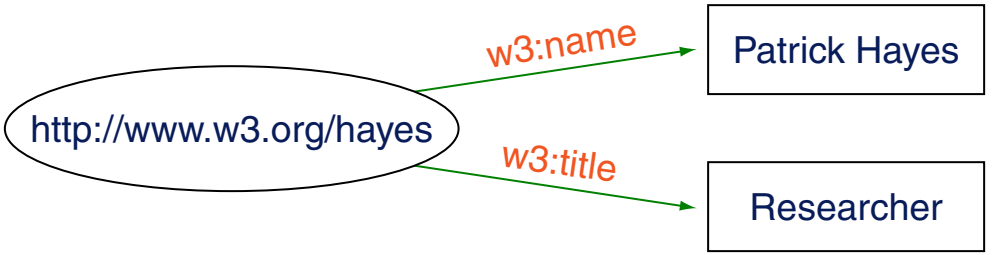



```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:w3="http://www.w3.org/">


  <rdf:Description rdf:about="http://www.w3.org/hayes">

    <w3:name>  Patrick Hayes  </w3:name>
    <w3:title> Researcher     </w3:title>

  </rdf:Description>


</rdf:RDF>
```


©STEIN 2005-2014

RDF: XML-Syntax

Beispiel

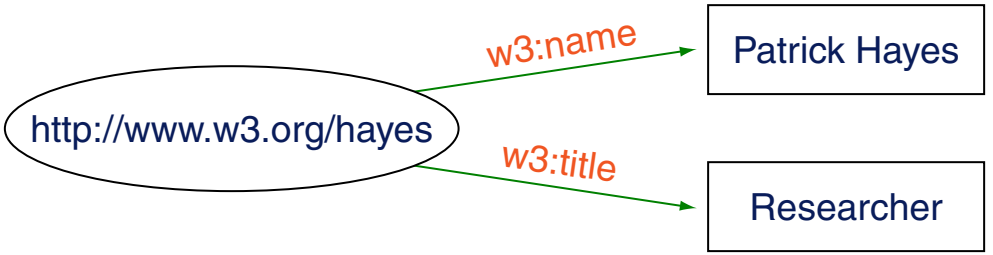



```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:w3="http://www.w3.org/">


  <rdf:Description rdf:about="http://www.w3.org/hayes">

    <w3:name>  Patrick Hayes  </w3:name>
    <w3:title> Researcher     </w3:title>

  </rdf:Description>


</rdf:RDF>
```


©STEIN 2005-2014

RDF: XML-Syntax

Beispiel




```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:w3="http://www.w3.org/">

  <rdf:Description rdf:about="http://www.w3.org/hayes">

    <w3:name> Patrick Hayes </w3:name>
    <w3:title> Researcher </w3:title>

  </rdf:Description>

</rdf:RDF>
```

Subjekt

Objekte

Statements

©STEIN 2005-2014

RDF: XML-Syntax

Geltungsbereich

Die folgenden zwei Deklarationen sind semantisch äquivalent:

(a)
```
<rdf:Description rdf:about="http://www.w3.org/hayes">
  <example:knows rdf:resource="http://www.buw.de/smith"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.buw.de/smith">
  <w3:name>Pete Smith</w3:name>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Geltungsbereich

Die folgenden zwei Deklarationen sind semantisch äquivalent:

(a)
```
<rdf:Description rdf:about="http://www.w3.org/hayes">
  <example:knows rdf:resource="http://www.buw.de/smith"/>
</rdf:Description>

<rdf:Description rdf:about="http://www.buw.de/smith">
  <w3:name>Pete Smith</w3:name>
</rdf:Description>
```

≈ (b)
```
<rdf:Description rdf:about="http://www.w3.org/hayes">
  <example:knows>
    <rdf:Description rdf:about="http://www.buw.de/smith">
      <w3:name>Pete Smith</w3:name>
    </rdf:Description>
  </example:knows>
</rdf:Description>
```


©STEIN 2005-2014

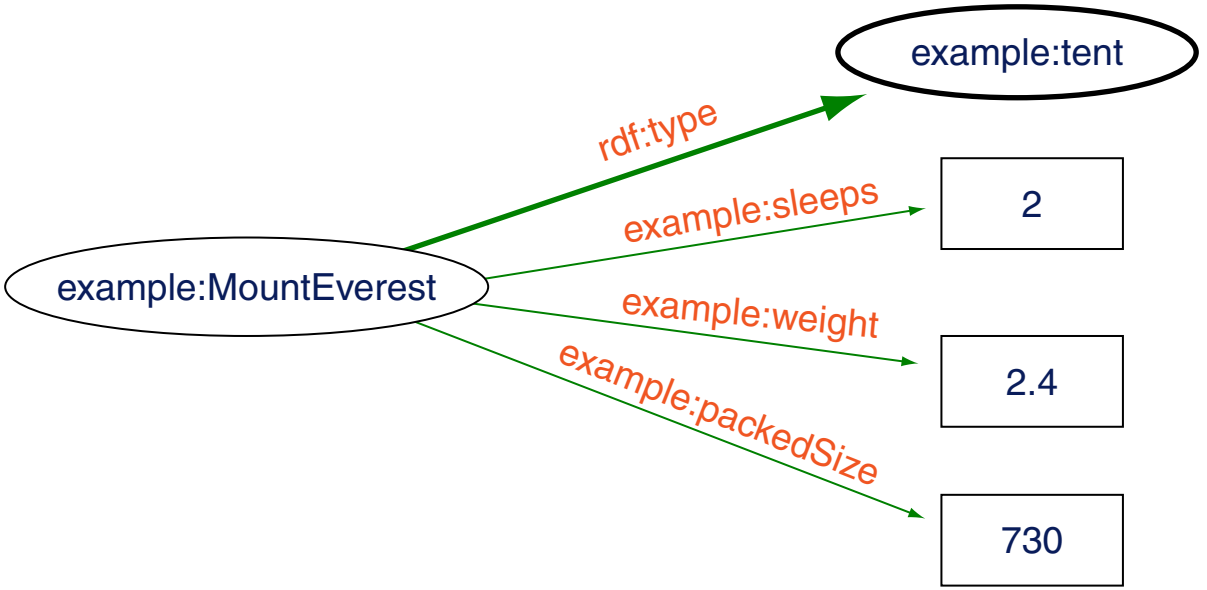
Bemerkungen:

- In einem (RDF-)Graph kann man nichts schachteln. Folglich ist der Geltungsbereich *(Scope)* eines `<rdf:Description>`-Elementes immer global.

©STEIN 2005-2014

RDF: XML-Syntax

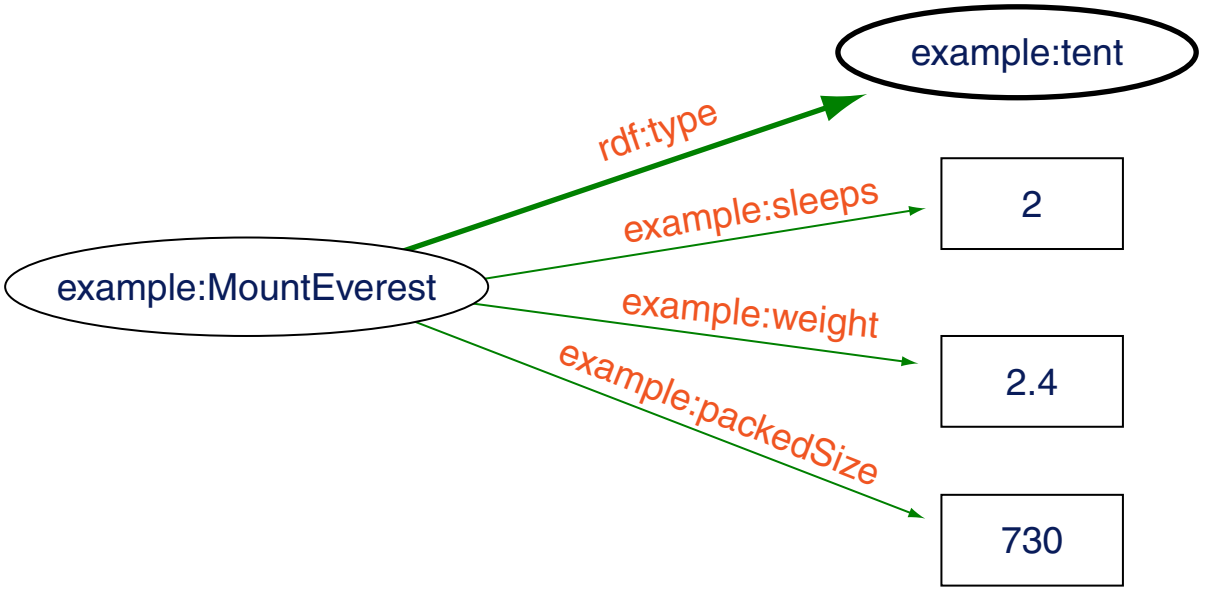
Getypte Ressourcen [W3C]



©STEIN 2005-2014

RDF: XML-Syntax

Getypte Ressourcen [W3C]




```
<rdf:Description rdf:about="http://www.buw.de/example/MountEverest">
  <rdf:type rdf:resource="http://www.buw.de/example/Tent"/>
  <example:sleeps>2</example:sleeps>
  <example:weight>2.4</example:weight>
  <example:packedSize>730</example:packedSize>
</rdf:Description>
```


©STEIN 2005-2014

Bemerkungen:

- Der Prefix `example:` steht für die Namensraum-URI `http://www.buw.de/example/`.
- `rdf:type` ist eine Property und ihr Wert kann auch als Kindelement serialisiert werden:

```
<rdf:Description rdf:about="...">
  </rdf:type>
    <rdf:Description rdf:about="http://www.buw.de/example/Tent"/>
    </rdf:Description>
  </rdf:type>
  <example:sleeps>2</example:sleeps>
  ...
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Abkürzende Schreibweisen

1. Property-Elemente ohne Kindelemente können im `<rdf:Description>`-Element als XML-Attribut definiert werden.
2. `<rdf:Description>`-Elemente, die eine `<rdf:type>`-Property beinhalten, können durch ein Element mit dem Namen des Typs ersetzt werden. Die `<rdf:type>`-Property kann dann entfallen.

Beispiel:

```
<rdf:Description rdf:about="courseMS21">
  <rdf:type rdf:resource="http://www.buw.de/course"/>
  <example:courseName>WebTec II</example:courseName>
  <example:taughtBy rdf:resource="http://www.buw.de/lecturers/l2305"/>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Abkürzende Schreibweisen

1. Property-Elemente ohne Kindelemente können im `<rdf:Description>`-Element als XML-Attribut definiert werden.
2. `<rdf:Description>`-Elemente, die eine `<rdf:type>`-Property beinhalten, können durch ein Element mit dem Namen des Typs ersetzt werden. Die `<rdf:type>`-Property kann dann entfallen.

Beispiel:

```
<rdf:Description rdf:about="courseMS21">
  <rdf:type rdf:resource="http://www.buw.de/course"/>
  <example:courseName>WebTec II</example:courseName>
  <example:taughtBy rdf:resource="http://www.buw.de/lecturers/l2305"/>
</rdf:Description>
```

$\approx$

```
<rdf:Description rdf:about="courseMS21" example:courseName="WebTec II">
  <rdf:type rdf:resource="http://www.buw.de/course"/>
  <example:taughtBy rdf:resource="http://www.buw.de/lecturers/l2305"/>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Abkürzende Schreibweisen (Fortsetzung)

1. Property-Elemente ohne Kindelemente können im `<rdf:Description>`-Element als XML-Attribut definiert werden.
2. `<rdf:Description>`-Elemente, die eine `<rdf:type>`-Property beinhalten, können durch ein Element mit dem Namen des Typs ersetzt werden. Die `<rdf:type>`-Property kann dann entfallen.

Beispiel:

```
<rdf:Description rdf:about="courseMS21" example:courseName="WebTec II">
  <rdf:type rdf:resource="http://www.buw.de/course"/>
  <example:taughtBy rdf:resource="http://www.buw.de/lecturers/l2305"/>
</rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Abkürzende Schreibweisen (Fortsetzung)

1. Property-Elemente ohne Kindelemente können im `<rdf:Description>`-Element als XML-Attribut definiert werden.
2. `<rdf:Description>`-Elemente, die eine `<rdf:type>`-Property beinhalten, können durch ein Element mit dem Namen des Typs ersetzt werden. Die `<rdf:type>`-Property kann dann entfallen.

Beispiel:

```
<rdf:Description rdf:about="courseMS21" example:courseName="WebTec II">
  <rdf:type rdf:resource="http://www.buw.de/course"/>
  <example:taughtBy rdf:resource="http://www.buw.de/lecturers/l2305"/>
</rdf:Description>
```

$\approx$

```
<buw:course rdf:about="courseMS21" example:courseName="WebTec II">
  <example:taughtBy rdf:resource="http://www.buw.de/lecturers/l2305"/>
</buw:course>
```


©STEIN 2005-2014

Bemerkungen:

- Der Prefix `buw:` steht für die Namensraum-URI `http://www.buw.de/`.

©STEIN 2005-2014

RDF: XML-Syntax

Abkürzende Schreibweisen bei Containern [RDF-Graph]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF
   [<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:about="students">
    <rdf:type rdf:resource="&rdf;Bag"/>
    <rdf:li rdf:resource="Adrian"/>
    ...
    <rdf:li rdf:resource="Peter"/>
  </rdf:Description>

</rdf:RDF>
```


©STEIN 2005-2014

RDF: XML-Syntax

Abkürzende Schreibweisen bei Containern [RDF-Graph]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF
   [<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Description rdf:about="students">
    <rdf:type rdf:resource="&rdf;Bag"/>
    <rdf:li rdf:resource="Adrian"/>
    ...
    <rdf:li rdf:resource="Peter"/>
  </rdf:Description>

</rdf:RDF>
```

$\approx$

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

  <rdf:Bag rdf:about="students">
    <rdf:li rdf:resource="Adrian"/>
    ...
    <rdf:li rdf:resource="Peter"/>
  </rdf:Bag>

</rdf:RDF>
```


©STEIN 2005-2014

RDF: XML-Syntax

Reification [RDF-Graph]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF [<!ENTITY example "http://www.buw.de/example/">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:example="http://www.buw.de/example/">

  <rdf:Description rdf:about="http://www.buw.de/pete-smith">
    <example:married-with rdf:resource="http://www.buw.de/sara-smith"/>
  </rdf:Description>
```


©STEIN 2005-2014

RDF: XML-Syntax

Reification [RDF-Graph]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE rdf:RDF [<!ENTITY example "http://www.buw.de/example/">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:example="http://www.buw.de/example/">

  <rdf:Description rdf:about="http://www.buw.de/pete-smith">
    <example:married-with rdf:resource="http://www.buw.de/sara-smith"/>
  </rdf:Description>

  <rdf:Statement rdf:about="&example;meta-statement">
    <rdf:subject rdf:resource="http://www.buw.de/pete-smith"/>
    <rdf:predicate rdf:resource="&example;married-with"/>
    <rdf:object rdf:resource="http://www.buw.de/sara-smith"/>
    <example:confirmed-by rdf:resource="http://www.vatican.va/pontifex"/>
  </rdf:Statement>

</rdf:RDF>
```


©STEIN 2005-2014

RDF: XML-Syntax

Referenzierung versus Definition

- Referenzierung einer existierenden Ressource:

```
<rdf:Description rdf:about="http://www.w3.org/hayes">
  ...
</rdf:Description>
```

- Definition einer neuen Ressource (in einer XML-Datei):

```
<rdf:Description rdf:ID="hayes">
  ...
</rdf:Description>
```

- Referenzierung einer in derselben XML-Datei definierten Ressource:

```
<rdf:Description rdf:about="#hayes">
  ...
</rdf:Description>
```

- Anonyme Ressource *(Blank Node)*:

```
<rdf:Description>
  ...
</rdf:Description>
```


©STEIN 2005-2014

Bemerkungen:

- Die Attribute `rdf:about` und `rdf:ID` besitzen nur deklarativen Charakter innerhalb eines RDF-Modells:

  "Formally speaking, the rdf:about-attribute is equivalent to the rdf:ID-attribute: A set of RDF statements together simply form a graph, and there is no such thing as defining an object in one place and referring to it elsewhere."

  [p.71, Antoniou/Harmelen 2004]

  Es ist Aufgabe der Anwendung, die ein RDF-Modell verarbeitet, die intendierte Semantik zu operationalisieren. Zum Beispiel lässt der RDF-Validator des W3C www.w3.org/RDF/Validator keine zwei gleichen IDs in einem RDF-Dokument zu.

- Enthält das `rdf:about`-Attribut keine URL, sondern nur einen XML-Namen, wird dieser um die Base-URL des Dokuments erweitert. Folglich referenzieren folgende Tags unterschiedliche Ressourcen:

```
<rdf:Description rdf:about="hayes">
                 ≠
<rdf:Description rdf:about="#hayes">
```


©STEIN 2005-2014

RDF: XML-Syntax

Serialisierungssyntax in EBNF

```
RDF ::=                 ['<rdf:RDF>'] description* ['</rdf:RDF>']
description ::=         '<rdf:Description' idAboutAttr? '>' propertyElt*
                           '</rdf:Description>'

idAboutAttr ::=         idAttr | aboutAttr
aboutAttr ::=           'about="' URIreference '"'
idAttr ::=              'ID="' IDsymbol '"'
propertyElt ::=         '<' propName '>' value '</' propName '>'
                           | '<' propName resourceAttr '/>'

propName ::=            Qname
value ::=               description | string
resourceAttr ::=        'resource="' URIreference '"'
Qname ::=               [ NSprefix ':' ] name

URIreference ::=        String, interpreted as URI.
IDsymbol, name ::=      Any legal XML name symbol.
NSprefix ::=            Any legal XML namespace prefix.
string ::=              Any XML text, with "<", ">", and "&" escaped.
```


©STEIN 2005-2014

RDF

Quellen zum Nachlernen und Nachschlagen im Web

- W3C RDF-Übersichtseite. www.w3.org/RDF
- D. Becketts RDF-Übersichtseite. planetrdf.com/guide
- F. Manola, E. Miller, Eds. *RDF Primer.*
  W3C Recommendation. www.w3.org/TR/rdf-primer
- ➜ G. Klyne, J. Carroll, Eds. *RDF: Concepts and Abstract Syntax.*
  W3C Recommendation. www.w3.org/TR/rdf-concepts
- D. Beckett, Ed. *RDF/XML Syntax Specification (Revised).*
  W3C Recommendation. www.w3.org/TR/rdf-syntax-grammar
- P. Hayes, Ed. *RDF Semantics.*
  W3C Recommendation. www.w3.org/TR/rdf-mt
- D. Brickley, R.V. Guha, Eds. *RDF Vocabulary Description Language 1.0: RDF Schema.*
  W3C Recommendation. www.w3.org/TR/rdf-schema
- R. Iannella. *An Idiot's Guide to the Resource Description Framework.*
  renato.iannella.it/paper/rdf-idiot

©STEIN 2005-2014