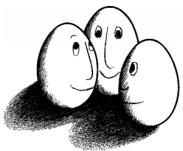
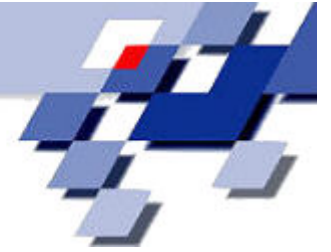




# Semantic Web Languages: RDF vs. SOAP Serialisation

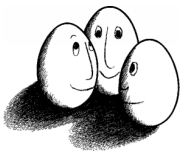
**Stefan Haustein**  
**University of Dortmund**  
**Computer Science VIII**  
**stefan.haustein@udo.edu**

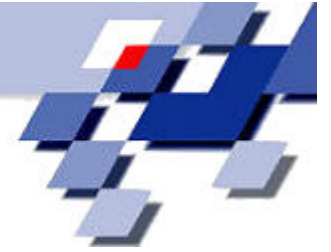




# Semantic Web Languages: RDF vs. SOAP Serialisation

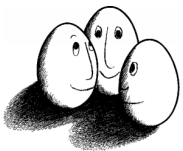
- **Why look at something else? Is RDF(S) not sufficient?**
- **What is SOAP? Why is SOAP important?**
- **Is SOAP Serialisation really an alternative to RDF(S)?**

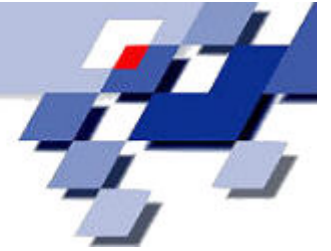




# RDF Issues

- **Syntax Variants**
  - Human readability?
  - XSLT Templates?
- **Verbosity**
  - Human readability?
- **Data model**





# RDF Syntax Variants

Resource description and type abbreviation:

```

<rdf:Description>
  <type resource="&fipaNS:#ApDescription" />
  ...
</rdf:Description>

```

→

```

<ApDescription id="1">
  ...
</ApDescription>

```

Using attributes instead of elements:

```

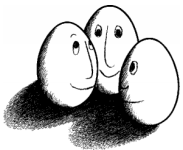
<ApDescription id="1">
  <name>paris.agentcities.org</name>
  <dynamic>true</dynamic>
  <mobility>true</mobility>
  ...
</ApDescription>

```

```

<ApDescription id="1"
  name="paris.agentcities.org"
  dynamic="true"
  mobility="true">
  ...
</ApDescription>

```





# RDF Syntax Variants

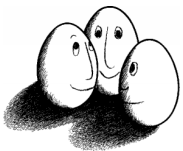
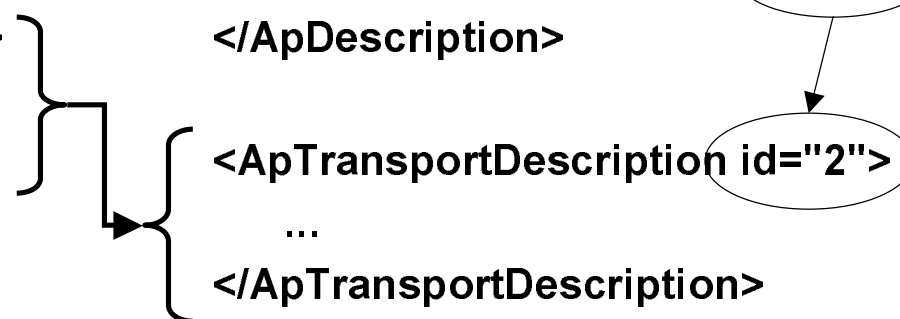
## Nesting vs. Linking:

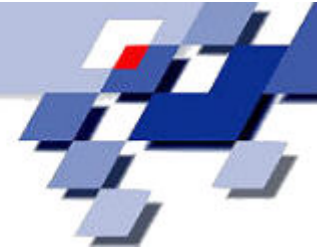
```

<ApDescription id="1">
  <name>paris.agentcities.org</name>
  <dynamic>true</dynamic>
  <mobility>true</mobility>
  <transportProfile>
    <ApTransportDescription id="2">
      ...
    </ApTransportDescription>
  </transportProfile>
</ApDescription>
  
```

```

<ApDescription id="1">
  <name>paris.agentcities.org</name>
  <dynamic>true</dynamic>
  <mobility>true</mobility>
  <transportProfile resource="#2" />
</ApDescription>
<ApTransportDescription id="2">
  ...
</ApTransportDescription>
  
```



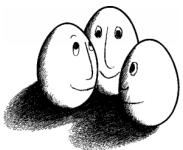
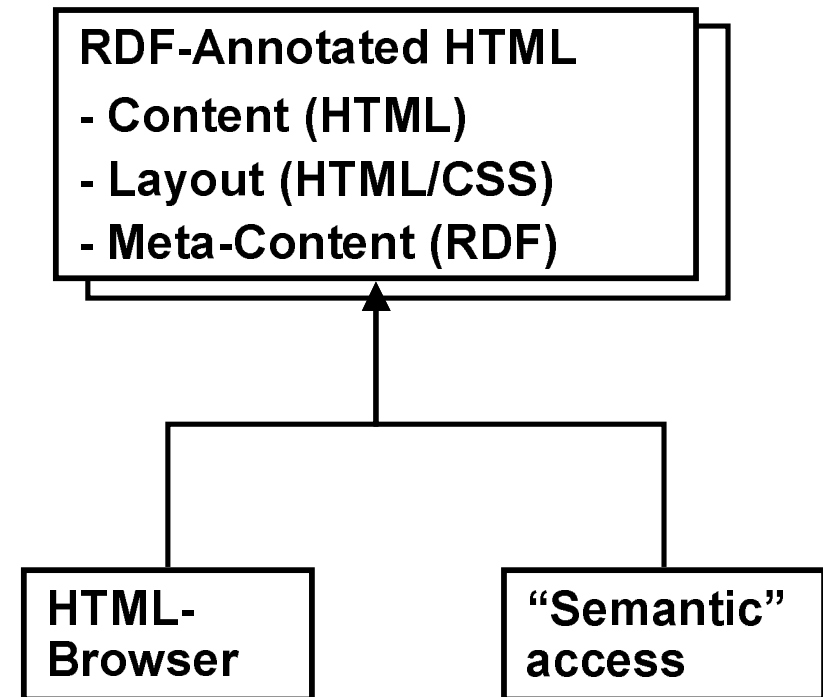


## RDF Syntax Variants: Motivation and Access Model

- **Browser-friendly annotation of human-readable information (HTML) with meta-information**
- **Abbreviations for convenience**

### Problems:

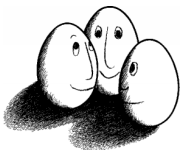
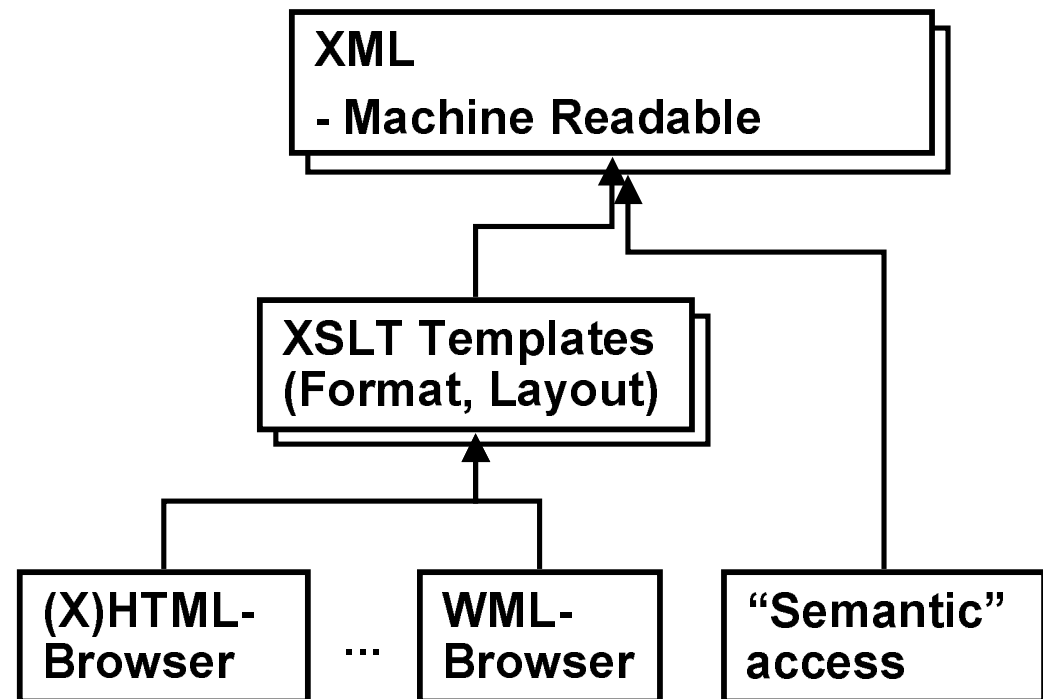
- **Redundancy???**
- **Bloated HTML (readability?)**
- **Separation of content and layout?**

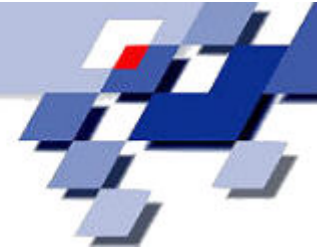




# XML Access Model

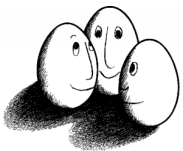
- Machine readable format in the first place
- How does it fit with RDF?
- If XML is replaced by RDF: XSLT templates for all syntax variants???



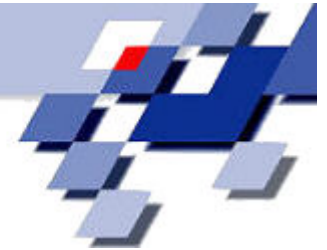


# RDFS Data Model

- **Not really object oriented but property centric**
- **Property names must be globally unique**
- **Fine granularity requires reification for statements about statements**



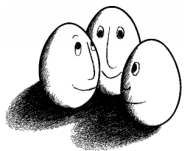
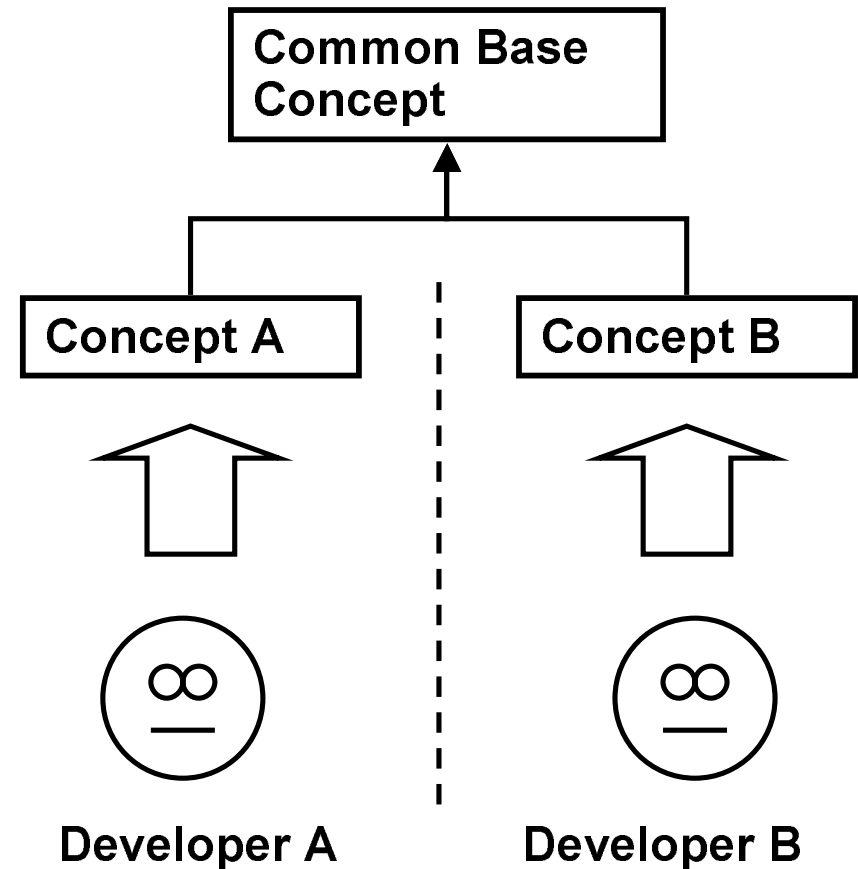


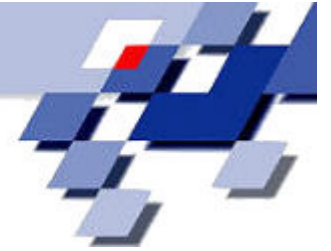


# Globally Unique Property Names vs. Schema Scalability

Consider two developers working on a RDF schema:

- Working in separate Namespaces requires remembering the Namespace of each property
- Working in one Namespace requires co-ordination for each property name



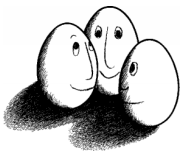


# What is SOAP? Why is SOAP important?

- SOAP is an XML based protocol for remote method invocation
- SOAP includes an XML serialisation format for object oriented data

## Advantages of SOAP serialisation:

- Support from Industry (MS, IBM, SUN...)
  - We cannot build the “Semantic Web” alone
- More compact and less ambiguous
  - comparable to “hand-made” XML languages
  - UML can be used for modelling XML languages
  - based on widely understood OO technologies



```

<ApDescription id="1">
  <name>paris.agencities.org</name>
  <dynamic>true</dynamic>
  <mobility>true</mobility>
  <transportProfile>
    <ApTransportDescription id="2">
      <availableMtps>
        <rdf:Bag>
          <rdf:li>
            <MtpDescription id="3">
              <mtpName>fipa.mts.mtp.iiop.std</mtpName>
              <addresses>
                <rdf:Bag>
                  <rdf:li>iiop://leap.crm-paris.com:9000/paris.agencities.org/acc</rdf:li>
                  <rdf:li>iiopname://leap.crm-paris.com:9000/paris.agencities.org/acc</rdf:li>
                </rdf:Bag>
              </addresses>
            </MtpDescription>
          </rdf:li>
          <rdf:li>
            <MtpDescription id="4">
              <mtpName>fipa.mts.mtp.http.std</mtpName>
              <addresses>
                <rdf:Bag>
                  <rdf:li>http://leap.crm-paris.com:8080/acc</rdf:li>
                </rdf:Bag>
              </addresses>
            </MtpDescription>
          </rdf:li>
        </rdf:Bag>
      </availableMtps>
    </ApTransportDescription>
  </transportProfile>
</ApDescription>

```



**RDF (abbreviated syntax!)**

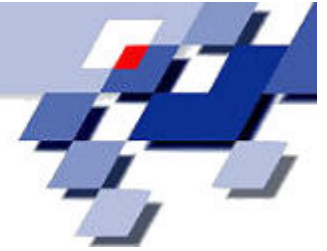
**SOAP (same content)**



```

<ApDescription>
  <name>paris.agencities.org</name>
  <dynamic>true</dynamic>
  <mobility>true</mobility>
  <transportProfile>
    <availableMtps>
      <MtpDescription>
        <mtpName>fipa.mts.mtp.iiop.std</mtpName>
        <addresses>
          <url>iiop://leap.crm-paris.com:9000/paris.agencities.org/acc</url>
          <url>iiopname://leap.crm-paris.com:9000/paris.agencities.org/acc</url>
        </addresses>
      </MtpDescription>
      <MtpDescription>
        <mtpName>fipa.mts.mtp.http.std</mtpName>
        <addresses>
          <url>http://leap.crm-paris.com:8080/acc</url>
        </addresses>
      </MtpDescription>
    </availableMtps>
  </transportProfile>
</ApDescription>

```

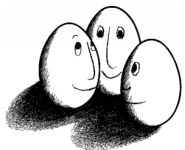
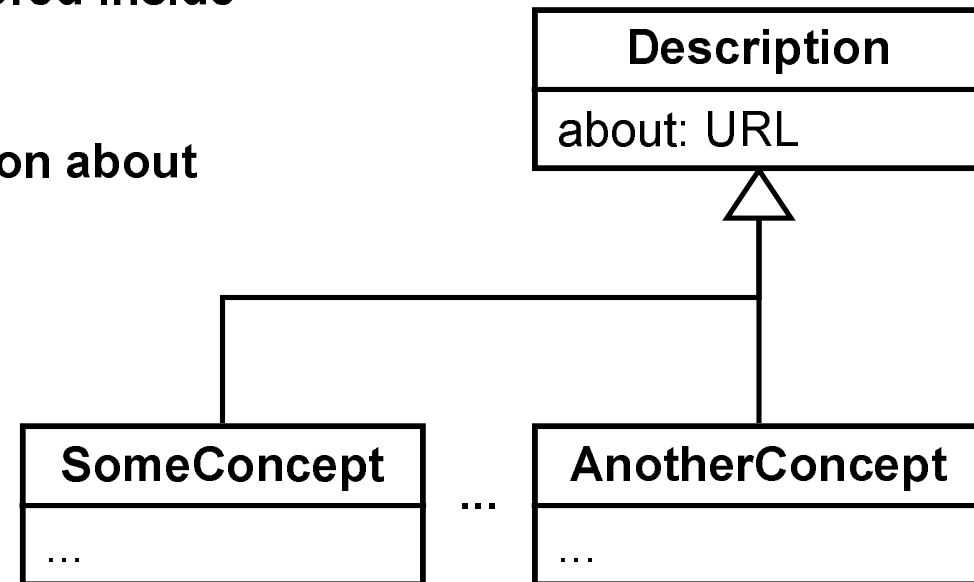


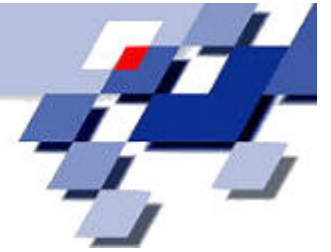
# Is SOAP Suitable for the Semantic Web?

Saying anything about anything:

“RDF is property centric because in OO, information about an object is stored inside that object. ”

- In OO, objects can hold information about other objects (or resources).
- Open issue: SOAP does not specify sufficiently how to assign URLs to objects.





## Integration with Existing Standards

	RDF	SOAP
<b>Serialisation</b>	RDF	SOAP
<b>HTML/WAP generation</b>	?	XSLT
<b>Query Language</b>	?	OQL
<b>Syntax Validation</b>	?	XML Schema
<b>Schema Serialisation</b>	RDFS	UML Meta-model
<b>Schema Modelling</b>	?	UML

