Semantic Matching of Interaction Rules

(Semantischer Abgleich von Interaktionsregeln)

Thesis of

Matthias Ferdinand

08.07.2002

Structuring

- Problem Situation
- Goals and Proceeding
- B2B E-Commerce with RosettaNet
- Semantic Web
  - Vision
  - Ontologies
  - Languages

Problem Situation

- important for growth of XML based B2B E-Commerce via Internet: widespread adoption of standards for business processes and documents
- major obstacle are **integration costs** for business partners

- **focus on RosettaNet** B2B Framework
  - partners must **manually analyze** each standard document and consult with their internal processes and IT systems
  - then form an agreement on how to use
  - some document fields are optional or may be used in a 'creative' way
  - takes up to three months to set up a new trading relationship
  - **cost** is prohibitive except for large companies
  - document specifications are complex
  - all work is done manually
  - lack of reusability, captured information can only be used by humans

Goals

- **automization** of the definition and agreement of/on business document usage
- help to reduce time and cost to set up a new RosettaNet connection

- develop a **language** to express **business rules**
  - stating constraints for the use of RosettaNet documents
  - considering different application contexts

- define **semantics** to specify
  - application context of rules
  - field contents

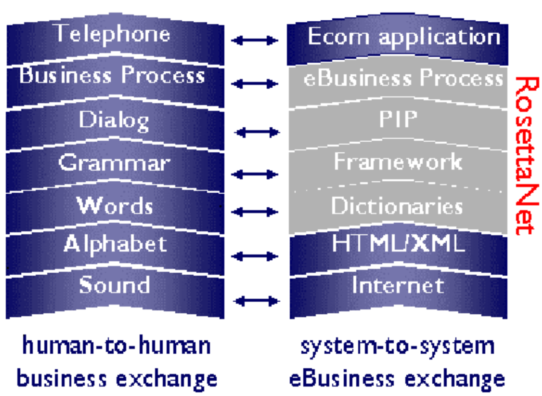
- develop a way to **match** two sets of rules
  - finding differences in the rule logic
  - semantic matching of rule terms using Semantic Web technology and ontologies

Proceeding

- analysis of RosettaNet Architecture and of business/technical requirements
- **investigation, analysis and evaluation** of
  - Semantic Web concepts, languages
  - concepts, languages, systems to express & handle (business) rules
  - problems and options concerning semantic matching with (multiple) ontologies
  - existing APIs, systems, platforms
- **development** of
  - a **general concept** & framework to express and handle rules
  - a **language** to describe rules
  - **algorithms** for semantic matching and for finding rule logic differences + implementation
- analysis of problems, contraints and benefits of the solution

RosettaNet
Introduction

- RosettaNet is a **non-profit consortium** of >400 companies of the IT, electronic components and semiconductor manufacturing industry, founded 1998
- wants to automate interactions between IT supply chain partners
- creates, implements and promotes **open B2B standards** for processes and data based on XML

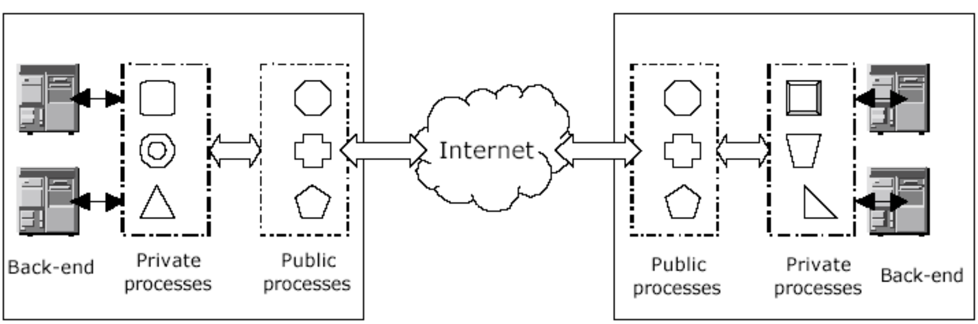


RosettaNet
Components

- Words → **Dictionaries**: provide common vocabulary
  - Business Dictionary defines the terms used in basic business activities
  - Technical Dictionary provides properties and a simple taxonomy to define products and services
- Grammar → **Implementation Framework**: provides exchange protocols, specifies information exchange incl. transport, routing, packaging, security
- Dialog → **Partner Interface Processes (PIP)**: specialized system-to-system XML dialogs, define business processes between trading partners
- additional **Product and partner codes**

RosettaNet
Processes

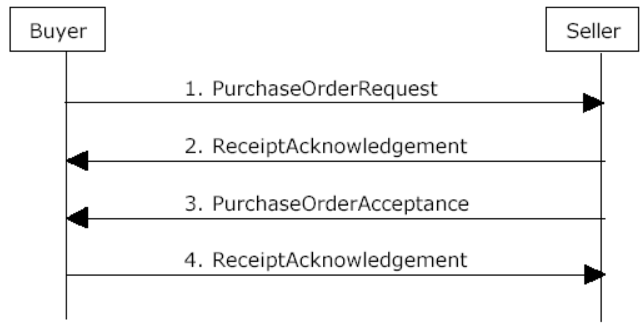
- Business Process → **eBusiness Process**
  - Private Processes: internal to the organization
  - Public Processes: visible interactions with trading partners, implement RosettaNet PIP specifications



RosettaNet
PIPs

- PIPs are organized in clusters (core business processes) and segments, e.g. “Service and Support”, “Order Management”, “Manufacturing”
- each specification includes
  - structure and content of exchanged documents
  - a process definition with the choreography of the message dialog
  - constraints for time, performance, security

Sample PIP interaction diagram:



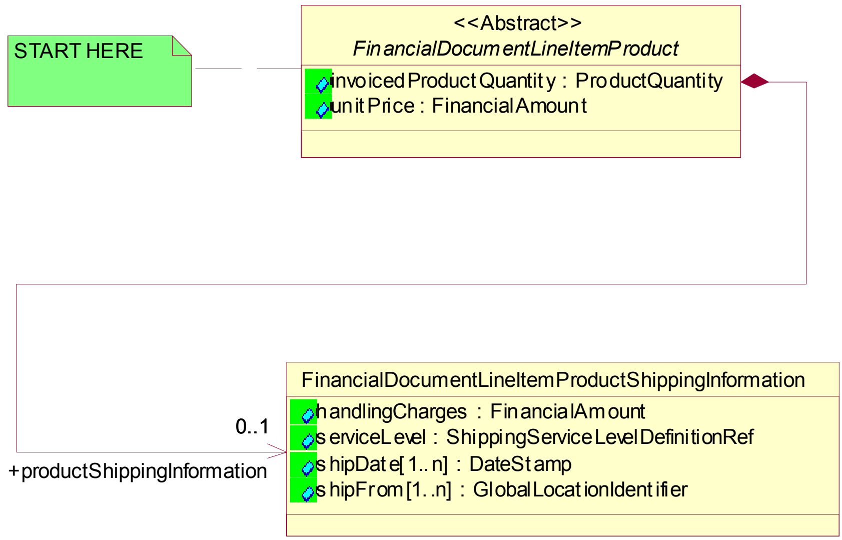
RosettaNet
NextGen PIPs

- single XML schema defines document
- UML used to document the design, generates the schema (“Specification Guide”)
- reuse of common data structures, machine-readable specifications

Schema example:

```
<xs:complexType name="FinancialDocumentLineItemProduct" abstract="false">
 <xs:annotation>
  <xs:documentation>A collection of business properties that describe
                    a financial document entry for a prod</xs:documentation>
 </xs:annotation>
 <xs:sequence>
  <xs:element name="invoicedProductQuantity" type="primitives:ProductQuantity"/>
  <xs:element name="productShippingInformation"
              type="financialdoc:FinancialDocumentLineItemProductShippingInformation" minOccurs="0"/>
  <xs:element name="unitPrice" type="primitives:FinancialAmount"/>
 </xs:sequence>
</xs:complexType>
```

UML example:



RosettaNet
NextGen PIPs

Business Document Structure (spreadsheet) example:

1	PIP3C3_LineItem.product : PIP3C3_FinancialDocumentLineItemProduct
1	PIP3C3_FinancialDocumentLineItemProduct.componentReference : PurchaseOrderLineItemComponentReference
1	PurchaseOrderLineItemComponentReference.purhcaseOrderLineItemIdentifier : ProprietaryDocumetnIdentifier
1	FinancialDocumentLineItemProduct.invoicedProductQuantity : ProductQuantity
1	ProductQuantity.description : String
1	ProductQuantity.quantity : AbstractQuantity (Choice: BulkQuantity, CountableQuantity)
	ProductQuantity.quantity : BulkQuantity
1	BulkQuantity.bulkQuantity : double
	ProductQuantity.quantity : CountableQuantity
1	CountableQuantity.productCount : Integer
1	FinancialDocumentLineItemProduct.unitPrice : FinancialAmount
1	FinancialAmount.globalCurrencyCode : CurrencyRef
1	FinancialAmount.monetaryAmount : MonetaryAmount
0..1	FinancialDocumentLineItemProduct.productShippingInformation : FinancialDocumentLineItemProductShippingInformation
1	FinancialDocumentLineItemProductShippingInformation.serviceLevel : ShippingServiceLevelDefinitionRef
1..n	FinancialDocumentLineItemProductShippingInformation.shipDate : DateStamp
1..n	FinancialDocumentLineItemProductShippingInformation.shipFrom : GlobalLocationIdentifier
1	FinancialDocumentLineItemProductShippingInformation.handlingCharges : FinancialAmount
1	FinancialAmount.globalCurrencyCode : CurrencyRef
1	FinancialAmount.monetaryAmount : MonetaryAmount

Situation today in the WWW:

- exponential growth
- handwritten and machine-generated HTML pages
- HTML is a markup language for display/rendering purposes
- web pages are made for direct human consumption & use
- content is primarily presented in natural language
- → it's a **web for humans**
- today's clients only transmit and present information
- difficult or impossible for machines to process content, especially semantics
- lack of meta-data, a **"syntactic web"**

- search engines only rely on (syntactic) keyword matching, often imprecise
- shopping agents must parse and extract information from web pages texts (screen scraping): hardwired implementation, hard to maintain

Vision of the Semantic Web:

- idea of Tim Berners-Lee 1998:

  "extension of the current Web in which information is given **well-defined meaning**, better enabling computers and people to work in cooperation"

  "allows **data** to be **shared** and processed by automated tools as well as by people"

- a web with machine-usable content, **machine-accessible semantics** of information
- explicit representation of the semantics underlying data, programs, pages and other web resources

Semantic Web
Vision

- meet the computer 'half-way':
  annotate data with **semantic markup** (meta-data)

- markup links information on the pages to
  semantic concepts defined in **ontologies**

- XML is not sufficient:
  - only allows a data format for structured documents
  - but does not imply specific interpretation of data
- XML tag names do not provide semantics, only implicit semantic agreements

Semantic Web
Ontologies

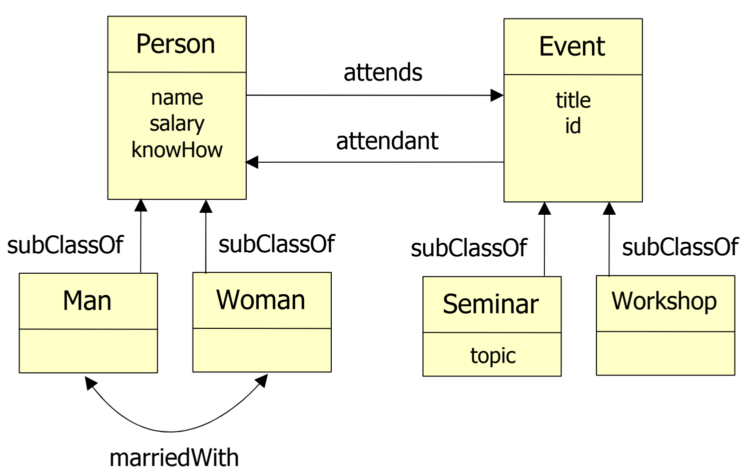
- ontologies are a popular research topic since the 1990s
- important in AI, knowledge representation, natural language processing, multi-agent systems etc.

- def.: *"an ontology is a formal, explicit specification of a shared conceptualization"* (Gruber 1993)
- *formal:* should be **machine-understandable**
- *shared:* should capture consensual knowledge accepted by communities
- *explicit:* type of **concepts** and constraints on their use are explicitly defined
- *conceptualization:* **abstract model** of (phenomena in) the real world

- enables to share common understanding of the structure of information among people or software agents that can be communicated ("a common language")
- enables reuse of domain knowledge
- makes domain assumptions explicit

- ontology typically consists of
  - important **concepts** in a domain (classes)
  - hierarchical **relations** among them
  - descriptions of **properties** of each concept (slots)
  - restrictions on properties
  - axioms, rules
- can be classified along different dimensions:
  formality, purpose, domain, task, level of detail, generality, language
- generalizations of ERM diagrams, OO designs, taxonomies, thesauries
- examples:
  - WordNet: large thesaurus for English language
  - RosettaNet Technical Dictionary: simple taxonomy of electronic equipment
  - Yahoo! directory, amazon.com catalog

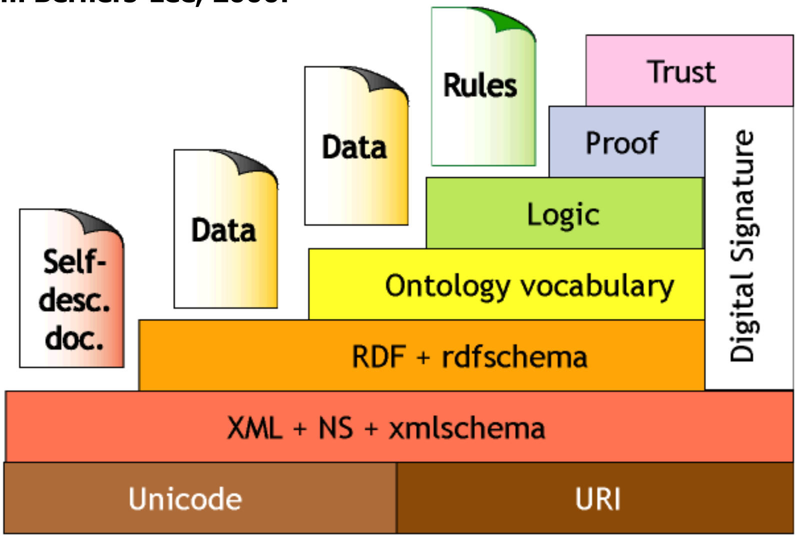
Semantic Web
Ontologies



axioms:
- 'attendants' and 'attendant' are inverse
- marriedWith is symmetric

Semantic Web
Languages

from Tim Berners-Lee, 2000:



Semantic Web
Resource Description Framework

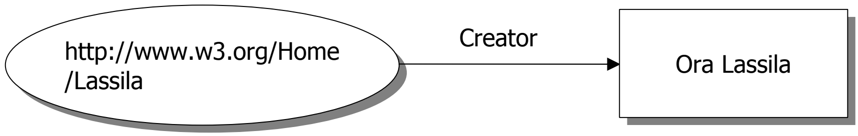
- XML provides a structure for data
- RDF tells something about data, i.e. give meaning to it
- RDF is a foundation for representing and exchanging **meta-data** on the web
- provides meta-data **interoperability** between applications, developed by the W3C

- defines a **model** and an encoding syntax for machine-accessible semantics
- statements about resources
  - resources can be anything in the world with an associated URI
  - statements are always a triple with:
    - **subject** (resource)
    - **predicate** (property)
    - **object** (resource or literal)

  - or "object-attribute-value"

Example:

- *Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila*




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

  <Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </Description>

</RDF>
```

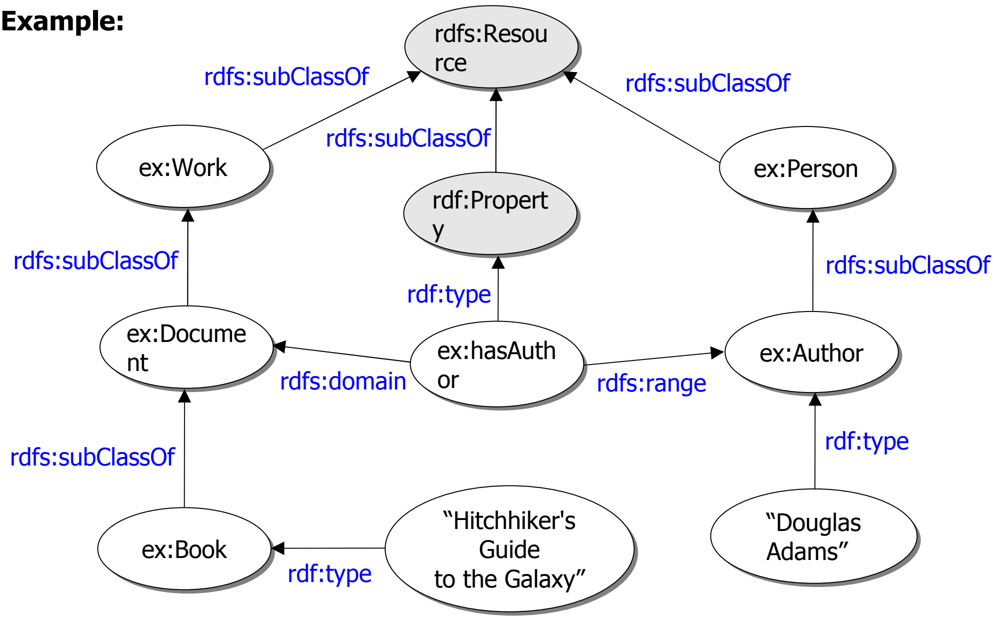

- RDF data model provides no mechanisms for declaring specific types or classes of resources or for meaningful use of properties

- RDF Schema is a simple, object-oriented **type system** on top of RDF
- RDFS is a vocabulary description language, introduces **basic ontological modeling primitives**
- used to describe properties of other RDF resource (incl. properties) to define domain-specific vocabularies
- primitives:
  - classes, types and properties definitions
  - range and domain constraints on properties
  - subclass and subproperty relations

- RDFS enables sharing, reuse and extensibility of meta-data definitions
- RDF & RDFS provide a simple knowledge representation mechanism for web resources

Semantic Web
RDF Schema

Example:



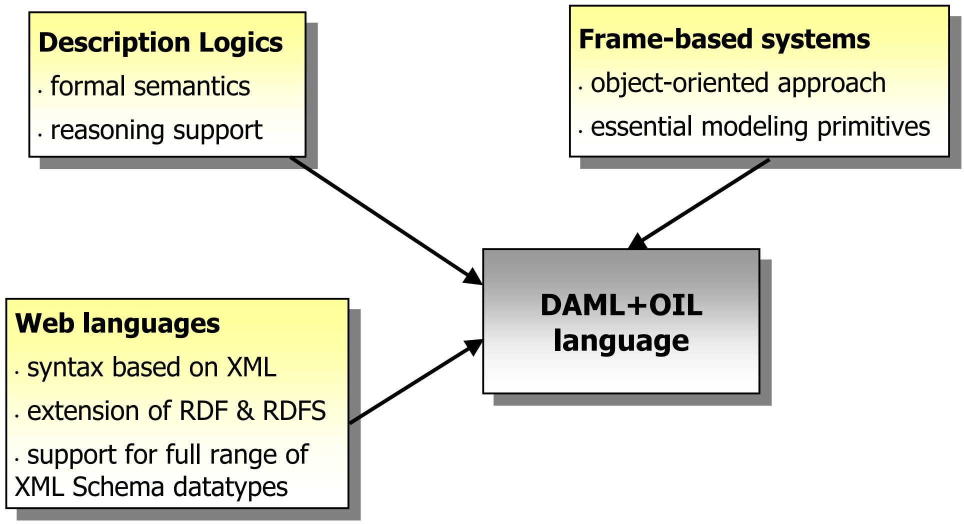
Semantic Web
DAML+OIL

- more expressive power is necessary to describe resources in sufficient detail
- automated **reasoning** over descriptions is desirable to determine semantic relationships and to derive new knowledge

- this has led to development of **DAML+OIL** based on RDFS
- result of merger in 2001 between
  - DARPA Agent Markup Language (USA)
  - Ontology Inference Layer (EU)

- an ontology definition and general-purpose markup language for the Semantic Web
- provides a set of intuitive and **rich modelling** primitives
- has well-defined **formal semantics**
- basis for the future Web Ontology Language by W3C

Roots of DAML+OIL:



Class constructors:

Constructor	DL Syntax	Example
`intersectionOf`	$C_1 \sqcap \ldots \sqcap C_n$	$\text{Human} \sqcap \text{Male}$
`unionOf`	$C_1 \sqcup \ldots \sqcup C_n$	$\text{Doctor} \sqcup \text{Lawyer}$
`complementOf`	$\neg C$	$\neg\text{Male}$
`oneOf`	$\{x_1 \ldots x_n\}$	$\{\text{john}, \text{mary}\}$
`toClass`	$\forall P.C$	$\forall\text{hasChild.Doctor}$
`hasClass`	$\exists P.C$	$\exists\text{hasChild.Lawyer}$
`hasValue`	$\exists P.\{x\}$	$\exists\text{citizenOf.}\{\text{USA}\}$
`minCardinalityQ`	$\geq n P.C$	$\geq 2\text{hasChild.Lawyer}$
`maxCardinalityQ`	$\leq n P.C$	$\leq 1\text{hasChild.Male}$
`cardinalityQ`	$= n\, P.C$	$= 1\,\text{hasParent.Female}$

Axioms:

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
sameClassAs	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \equiv P_2$	cost $\equiv$ price
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ $\neg${peter}
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
uniqueProperty	$\top \sqsubseteq \leq 1 P$	$\top \sqsubseteq \leq$1hasMother
unambiguousProperty	$\top \sqsubseteq \leq 1 P^-$	$\top \sqsubseteq \leq$1isMotherOf$^-$

Example:

```
<daml:Class rdf:about="#Person">
  <rdfs:comment>every person is a man or a woman
  </rdfs:comment>
  <daml:disjointUnionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Man"/>
    <daml:Class rdf:about="#Woman"/>
  </daml:disjointUnionOf>
</daml:Class>


<Person rdf:ID="Adam">
  <rdfs:label>Adam</rdfs:label>
  <rdfs:comment>Adam is a person.</rdfs:comment>
  <age><xsd:integer rdf:value="13"/></age>
  <shoesize><xsd:decimal rdf:value="9.5"/></shoesize>
</Person>
```

- DAML+OIL is equivalent to the very **expressive description logic** *SHIQ* DL
- exploits efficient algorithms for automated reasoning about ontologies

- key inference problems are decidable
  - **consistency**: detect logically inconsistent classes
  - **subsumption**: detect implicit subsumption relationships, new concept positions

- highly optimized **DL inference engines** can be used
  - FaCT (University of Manchester)
  - RACER (University of Hamburg, KOGS)

- many research challenges

Reasoning Example:

$$
\begin{aligned}
\text{Woman} &\equiv \text{Person} \sqcap \text{Female} \\
\text{Man} &\equiv \text{Person} \sqcap \neg\text{Woman} \\
\text{Mother} &\equiv \text{Woman} \sqcap \exists\text{hasChild}.\text{Person} \\
\text{Father} &\equiv \text{Man} \sqcap \exists\text{hasChild}.\text{Person} \\
\text{Parent} &\equiv \text{Father} \sqcup \text{Mother} \\
\text{Grandmother} &\equiv \text{Mother} \sqcap \exists\text{hasChild}.\text{Parent}
\end{aligned}
$$

- **Person** subsumes **Woman**
- **Woman**, **Parent** subsume **Mother**
- **Mother** subsumes **Grandmother**

Questions?

EMail: 6ferdina@informatik.uni-hamburg.de