An Institutional Perspective of Semantic Web Stack

Dorel Lucanu
Joint work with Yuan Fang Li and Jin Song Dong from NUS
{dlucanu}@info.uaic.ro

Universitatea “Alexandru Ioan Cuza”, Iași
Outline

- institutions
- institution independent logic programming
- applications to SW
  - logic programming views on web ontologies
  - institutional meaning of RDF layer
  - institutional meaning of layering
- conclusion
The jungle of Semantic Web Languages

SWRL FOL

OWL Full

OWL

F-Logic

OWL Light

OWL Light

RDF Schema

OWL Full

Description Logic (DL)

Datalog

SWRL

OWL DL

DHL

OWL DL

DLP

OWL Flight

WRL

DAML-OIL

DERI, Innsbruck, Austria, 06.12.2005
Motivation

- an integrating mathematical structure for Semantic Web Languages (SWL)
- translating Web ontologies into other formalisms
- a safe way to walk in the jungle
- disputes on layering of SWL
- Open World Assumption (OWA) vs Closed World Assumption (CWA)
- soundness of the reasoners for Web ontologies
- finding the real meaning of Semantic Web Stack
Institutions

- formalize the notion of "a logic"
- study the properties of a logic
  - representation
  - implementation
- translation of logics
Institutions: ingredients

- **signatures**: formalize vocabularies

- **models**: structures interpreting the symbols (names) from a signature

- **sentences**: formulas built with symbols from signature expressing specific properties

- **satisfaction relation**: says when a given sentence holds in a given model (both correspond to the same signature)
The architecture of an institution
Institutions: signatures

Horn Logic: $\Sigma = (PtN, FtN, CtN)$
$PtN =$ predicate names, $FtN =$ function names, $CtN =$ constant names

Description Logic: $\Sigma = (CN, PN, IN)$
$CN =$ class names, $PN =$ property names, $IN =$ individual names
$PtN, FtN, CtN$ are pairwise disjoint

OWL: $\Sigma = (CN, PN, IN)$
$PtN, FtN, CtN$ are pairwise disjoint only for OWL DL and its dialects
Institutions: models

- **Horn Logic:** \( \Sigma\text{-model } \mathfrak{I} = (\mathcal{D}_\mathfrak{I}, \_\mathfrak{I}) \)
  \[
  \rho_\mathfrak{I} \subseteq \mathcal{D}_\mathfrak{I}^{\text{arity}(p)}, \quad f_\mathfrak{I} : \mathcal{D}_\mathfrak{I}^{\text{arity}(f)} \rightarrow \mathcal{D}_\mathfrak{I}, \quad a_\mathfrak{I} \in \mathcal{D}_\mathfrak{I}
  \]

- **Description Logic:** \( \Sigma\text{-model } \mathfrak{I} = (\Delta_\mathfrak{I}, [[\_]]_\mathfrak{I}) \)
  \( \Delta_\mathfrak{I} \) domain of the interpretation
  
  \[
  [[cn]]_\mathfrak{I} \subseteq \Delta_\mathfrak{I}, \quad [[pn]]_\mathfrak{I} \subseteq \Delta_\mathfrak{I} \times \Delta_\mathfrak{I}, \quad [[in]]_\mathfrak{I} \in \Delta_\mathfrak{I}
  \]

- **OWL:** \( \Sigma\text{-model } \Pi = (R_\Pi, S_\Pi, \text{ext}_\Pi) \)
  
  \[
  S_\Pi : CN \cup PN \cup IN \rightarrow R_\Pi
  \]
  \[
  \text{ext}_\Pi(cn) \subseteq R_\Pi, \quad \text{ext}_\Pi(pn) \subseteq R_\Pi \times R_\Pi
  \]
Institutions: sentences

- Horn Logic: Horn rules $p_1(u_1), \ldots, p_n(u_n) \rightarrow p_0(u_0)$
  
  # hasAuthor $(p, a) \land \text{citedBy} (p, q) \rightarrow \text{CitedAuthor} (a)$

- Description Logic:

  $C ::= \bot \mid \top \mid cn \mid C \cap C \mid C \cup C \mid \neg C$
  
  $\mid \forall pn.C \mid \exists pn.C \mid \leq n pn \mid \geq n pn$

  $F ::= C \sqsubseteq C \mid C \equiv C$

  $\mid pn^+ \sqsubseteq pn \mid pn \sqsubseteq pn' \mid pn \equiv pn'$

  $\mid o : C \mid (o, o') : pn$

**Author $\sqsubseteq$ Person**

**Book $\sqsubseteq (\geq 1 \text{hasAuthor})$ (each book has at least one author)**
Institutions: sentences

OWL:
each book has at least one author

```
<owl:Class rdf:ID="Author">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasAuthor" />  
      <owl:minCardinality rdf:datatype="#&xsd;nonNegativeInteger">1
    </owl:minCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Class(Author partial restriction(hasAuthor minCardinality(1)))
Institutions: satisfaction relation

relates the models and the sentences: \( M \models \Sigma \varphi \)
where \( M \) is \( \Sigma \)-model and \( \varphi \) is a \( \Sigma \)-sentence

it is the subject of the satisfaction condition which
expresses the invariance of truth under change of notation

\[
M' \models_{\Sigma'} \phi(\varphi) \iff M' \models_{\phi} \models_{\Sigma} \varphi
\]

where \( \phi : \Sigma \to \Sigma' \), \( M' \) is a \( \Sigma' \)-model, and \( \varphi \) is a \( \Sigma \)-sentence

DL:

\[
I \models_{\Sigma} A \subseteq \forall P.B \iff [A]_I \subseteq \{x \mid (\forall y)(x, y) \in [P]_I \Rightarrow y \in [B]_I\}
\]
Institutions: Specifications and Theories

- A specification is a pair \((\Sigma, F)\), where \(\Sigma\) is a signature and \(F\) is a set of sentences.

- Semantical consequences: \((\Sigma, F) \models \varphi\) iff
  \[ (\forall M)(M \models_\Sigma F \Rightarrow M \models_\Sigma \varphi) \]

- A theory is a specification \((\Sigma, F)\) s.t.
  \[ (\forall \varphi)((\Sigma, F) \models \varphi \Rightarrow \varphi \in F) \]

- The inclusion \(\text{Th} \rightarrow \text{Spec}\) is an equivalence of categories.

- Theoroidal (spec-oidal) institutions:
  - Signatures are theories (specifications).
  - A \((\Sigma, F)\)-sentence is a \(\Sigma\)-sentence.
  - \((\Sigma, F)\)-models are \(\Sigma\)-models satisfying \(F\).
  - \(M \models (\Sigma, F) \varphi\) iff \(M \models_\Sigma \varphi\).
Relating Institutions

- **morphism**: capture the way in which a “richer” institution is built over a “simpler” one

- **comorphism**: capture the way in which a “simpler” institution is embedded (encoded) into a “richer” one

  both are the subject of a corresponding satisfaction condition

- there exist a variety of definitions for morphisms and variety of definitions for comorphisms in literature

- a prover from the target logic can be used to prove properties from the source logic only if certain conditions are fulfilled
**Closed World Assumption**

*If we cannot prove \((\Sigma, F) \models \varphi\), then we add \(\neg \varphi\) to \(F\) or, equivalently, restrict the class of \((\Sigma, F)\)-models to those satisfying \(\neg \varphi\).*

The negation can be defined in an institution independent way:

\[
M \models_\Sigma \neg \varphi \iff M \not\models_\Sigma \varphi
\]

\(\Sigma\) *satisfies the closed world assumption (CWA)* iff for each specification \((\Sigma, F) \in \text{Spec},\)

\[
(\Sigma, F) \models \neg \varphi \iff (\Sigma, F) \not\models \varphi
\]
variables as signature morphisms
\((\forall x)p(x)\)
Institution independent quantifiers

variables as signature morphisms

$$(\forall x)p(x)$$

$x : \Sigma = (PtN, FtN, CtN) \rightarrow \Sigma' = (PtN, FtN, CtN \cup \{x\})$

$I \models_{\Sigma} (\forall x)p(x)$ iff

$$(\forall I' \Sigma'\text{model})I'|_x = I \Rightarrow I' \models_{\Sigma'} p(x)$$
variables as signature morphisms
\[(\forall x)p(x)\]
\[x : \Sigma = \{PtN, FtN, CtN\} \rightarrow \Sigma' = \{PtN, FtN, CtN \cup \{x\}\}\]
\[M_0 \models (\forall x)p(x) \iff (\forall M' \Sigma'\text{model})M'\models_x M \models \Sigma' \models (\forall x)p(x)\]

\[\Sigma \rightarrow \Sigma'\]
variables as signature morphisms

\((\forall \, x)\, p(x)\)

\(x : \Sigma = (PtN, FtN, CtN) \rightarrow \Sigma' = (PtN, FtN, CtN \cup \{x\})\)

\(L \models_{\Sigma} (\forall \, x)\, p(x) \iff (\forall \, I' \Sigma'\text{\ model})I'|_x = L \Rightarrow L' \models_{\Sigma'} p(x)\)

\(X : \Sigma \rightarrow \Sigma'\)

A \(\Sigma\)-sentence \((\forall \, X)\varphi'\) is the universal quantification of the \(\Sigma'\)-sentence \(\varphi'\) iff

\(M \models_{\Sigma} (\forall \, X)\varphi' \iff (\forall \, M' \text{ a } \Sigma'\text{-model})M'|_x = M \Rightarrow M' \models_{\Sigma'} \varphi'\)
Institution independent Horn Logic

$\chi : \Sigma \rightarrow \Sigma'$ is *representable* iff there is a $\Sigma$-model $M_\chi$ s.t.

1. any $\Sigma$-model $M'$ is an expansion along $X$ of a $\Sigma$-model $M$, i.e. $M = M'|_X$, and

2. there is a morphism $M_\chi \rightarrow M$

A set $F$ of $\Sigma$-sentences is *basic* iff there is a $\Sigma$-model $M_F$ s.t. $M \models_\Sigma F$ iff there is a homomorphism $M_F \rightarrow M$.

If $M_F \rightarrow M$ is unique, then $F$ is *epic basic*.

A *Horn clause*: $(\forall X)F \rightarrow F'$ s.t. $F$ is epic basic, $F'$ is basic, and $X : \Sigma \rightarrow \Sigma'$ is representable. (Diaconescu, 2004)
Institution independent Horn Logic

DL

- $cn(in)$ is epic basic ($cn$ class name, $in$ individual name)
- $cn_1 \sqcup cn_2(in)$ is not basic
  \[
  \Delta_{I_1} = \{a\}, \; [[in]]_{I_1} = a, \; [[cn_1]]_{I_1} = \{a\}
  \]
  \[
  \Delta_{I_2} = \{b\}, \; [[in]]_{I_1} = b, \; [[cn_2]]_{I_2} = \{b\}
  \]
  $I_1$ and $I_2$ cannot be related via a homomorphism

- Lloyd-Topor transformations define a comorphism

  \[(\Phi, \beta, \alpha) : \text{ExtHL} \rightarrow \text{HL}^\wedge\]

  $\text{HL}^\wedge$ is HL enriched s.t. any conjunction of rules is a sentence
Institution independent Logic Programming

**OWA Logic Programming institution** with base institution $\mathcal{S}$, $\mathcal{LP}(\mathcal{S})$, is defined as follows:

1. the signatures are Horn specifications $\text{HSpec}$;
2. the model functor is $\text{Mod}(\mathcal{S})$ extended to Horn specifications;
3. the $\Sigma$-sentences are *Horn queries* $(\exists X)\varphi$ with $\varphi$ basic;
4. the satisfaction is given by $M \models_{(\Sigma,F)} (\exists X)\varphi$ iff $M \models_{\Sigma} (\exists X)\varphi$, i.e., the satisfaction is inherited from the base institution.
CWA Logic Programming institution with base institution \( \mathcal{S} \), \( \mathcal{LP}^\Delta(\mathcal{S}) \), is defined similarly to \( \mathcal{LP}(\mathcal{S}) \) except the model functor which associates a category of logically equivalent canonical models \( \Delta(\Sigma, F) \) with each Horn specification \( (\Sigma, F) \).

Herbrand Theorem (Diaconescu, 2004). In an arbitrary institution consider a specification \( (\Sigma, F) \) which has an initial model \( 0_{\Sigma,F} \). Then for each query \( (\exists X)\varphi \)

\[
(\Sigma, F) \models (\exists X)\varphi \text{ iff } 0_{\Sigma,F} \models (\exists X)\varphi
\]
Institution independent Logic Programming

The institution of a Logic Programming with Knowledge Bases, $\mathcal{LP}^{kb}(\mathcal{S})$, is defined as follows:

1. a signature is a pair $((\Sigma, F), KB)$, where $(\Sigma, F)$ is a Horn specification (the logic program), and $KB$ is a $(\Sigma, F)$-knowledge base (Grothendieck construction);

2. the model functor maps each signature $((\Sigma, F), KB)$ into a category $\Delta(\Sigma_{KB}, F \cup F_{KB})$;

3. the sentences are Horn queries;

4. the satisfaction is given by $\Delta(\Sigma, F, KB) \models_{(\Sigma,F)} (\exists X) \varphi$ iff $\Delta(\Sigma, F, KB) \models_{\Sigma} (\exists X) \varphi$. 
Logic Programming Views

- OWLDLP is OWA LP over OWLDesLog
- OWLDLP is CWA LP over OWLDesLog
- OWLDLP\(^{kb}\) is LP with KB over OWLDesLog

- DLP is OWA LP over DHL
- DLP is CWA LP over DHL
- DLP\(^{kb}\) is LP with KB over DHL

- DATAHLP is OWA LP over DATAHORN
- DATAHLP is CWA LP over DATAHORN
- DATALOG is LP with KB over DATAHORN
Adding rules to Web Ontology Languages . . .

\[ \mathcal{LP}^{kb}(\text{OWLDesLog}_-^-) \xrightarrow{\text{CO}} \mathcal{LP}^{kb}(\text{DHL}) \xrightarrow{\text{CO}} \text{DATAHLP}^{kb} \]

\[ \mathcal{LP}^\Delta(\text{OWLDesLog}_-^-) \xrightarrow{\text{CO}} \mathcal{LP}^\Delta(\text{DHL}) \xrightarrow{\text{CO}} \text{DATAHLP}^{\Delta} \]

\[ \mathcal{LP}(\text{OWLDesLog}_-^-) \xrightarrow{\text{CO}} \mathcal{LP}(\text{DHL}) \xrightarrow{\text{CO}} \text{DATAHLP} \]

\[ \mathcal{HL}(\text{OWLDesLog}_-^-) \xrightarrow{} \mathcal{HL}(\text{DHL}) \xrightarrow{} \text{DATAHORN}^{\wedge} \]
RDF Serialization

From Semantic Web talk by Tim Berners-Lee at XML 2000
Bare RDF logic

- signatures: $RR$ a set of resources references
- models: $\models = (R_\models, P_\models, S_\models, \text{ext}_\models)$, where $R_\models$ is a set of resources, $P_\models \subseteq R_\models$ - the set of properties, $S_\models : RR \rightarrow R_\models$ 
  $\text{ext}_\models : P_\models \rightarrow \mathcal{P}(R_\models \times R_\models)$ is an extension function mapping each property to a set of pairs of resources that it relates
- $RR$-sentences are triples of the form $(sn, pn, on)$, where $sn, pn, on \in RR$
- satisfaction is defined as follows:

$$\models \models_{RR} (sn, pn, on) \iff (S_\models(sn), S_\models(on)) \in \text{ext}_\models(\text{ext}_\models(pn)),$$
RDF theory \( \text{RDF} = (\text{RDFVoc}, \mathcal{T}_{\text{RDF}}) \)

\text{RDFVoc} includes the following items:

- \text{rdf:type}, \text{rdf:Property}, \text{rdf:value},
- \text{rdf:Statement}, \text{rdf:subject}, \text{rdf:predicate}, \text{rdf:object},
- \text{rdf:List}, \text{rdf:first}, \text{rdf:rest}, \text{rdf:nil},
- \text{rdf:Seq}, \text{rdf:Bag}, \text{rdf:Alt}, \text{rdf:li} \text{ rdf:_1 rdf:_2} ...

and \( \mathcal{T}_{\text{RDF}} \) is

\[
\{ \\
(rdf:type, rdf:type, rdf:Property), \\
(rdf:subject, rdf:type, rdf:Property), \\
(rdf:predicate, rdf:type, rdf:Property), \\
(rdf:object, rdf:type, rdf:Property), \\
(rdf:value, rdf:type, rdf:Property), \\
(rdf:first, rdf:type, rdf:Property), \\
(rdf:rest rdf:type rdf:Property), \\
(rdf:nil rdf:type rdf:List), \\
(rdf:_1 rdf:type rdf:Property), \\
(rdf:_2 rdf:type rdf:Property), \\
\ldots
\}
\]
RDF theory $\text{RDF} = (\text{RDFVoc}, T_{\text{RDF}})$

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

The prefix rdf: is used for the namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#

We suppose that there is given a set $R_{\text{RDF}}$ of RDF resources and a function $S_{\text{RDF}} : \text{RDFVoc} \rightarrow R_{\text{RDF}}$ which associates a resource with each RDF symbol. It is easy to see that $R_{\text{RDF}}$ and $S_{\text{RDF}}$ can be extended to a RDF-model RDF.
signatures: theory morphisms \( f : \text{RDF} \rightarrow (RR, T) \)

models: \((RR, T)\)-models \( \mathcal{I} \) such that

1. \( R_\mathcal{I} \) includes \( R_{\text{RDF}} \) and the restriction of \( S_\mathcal{I} \) to \( \text{RDFVoc} \) coincides with \( S_{\text{RDF}} \),
2. if \( p \in P_\mathcal{I} \) then \((p, \text{rdf}: \text{Property}) \in \text{ext}_\mathcal{I}(\text{rdf}: \text{type}) \)

\( f \)-sentences are \( RR \)-sentences

satisfaction: \( \mathcal{I} \models_f (sn, pn, on) \text{ iff } \mathcal{I} \models_{RR} (sn, pn, on) \)
RDF Schema theory $\text{RDFS} = (\text{RDFS Voc}, \ T_{\text{RDFS Voc}})$

$\text{RDFS Voc}$ is RDF Voc together with

- `rdfs:domain`, `rdfs:range`, `rdfs:Resource`,
- `rdfs: Literal`, `rdfs: Datatype`,
- `rdfs:Class`, `rdfs:subClassOf`,
- `rdfs:subPropertyOf`,
- `rdfs:member`, `rdfs:Container`, `rdfs:ContainerMembershipProperty`

and the sentences $\ T_{\text{RDFS}}$

\[
\begin{align*}
\{ & \text{rdf:type}, \text{rdfs:domain}, \text{rdfs:Resource}, \\
& \text{rdfs:domain}, \text{rdfs:domain}, \text{rdf:Property}, \\
& \text{rdfs:range}, \text{rdfs:domain}, \text{rdf:Property}, \\
& \text{rdfs:subPropertyOf}, \text{rdfs:domain}, \text{rdf:Property}, \\
& \text{rdfs:subClassOf}, \text{rdfs:domain}, \text{rdf:Class}, \\
& \text{rdf:subject}, \text{rdfs:domain}, \text{rdf:Statement}, \\
& \ldots
\end{align*}
\]
RDF Schema theory $\text{RDFS} = (\text{RDFSVoc}, T_{\text{RDFSVoc}})$

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
...

We suppose that there is given a set $R_{\text{RDFS}}$ of RDF Schema resources and a function $S_{\text{RDFS}} : \text{RDFSVoc} \rightarrow R_{\text{RDFS}}$ which associates a resource with each RDF Schema symbol and that satisfies $S_{\text{RDFS}}|_{\text{RDFSVoc}} = S_{\text{RDF}}$.
RDF Schema logic \( \text{RDFS} \)

- **signatures:** theory morphisms \( f : \text{RDFS} \to (RR, T) \)
- **models:** \( (RR, T) \)-models \( \mathcal{I} \) such that
  - \( R_\mathcal{I} \) includes \( R_{\text{RDFS}} \) and the restriction of \( S_\mathcal{I} \) to \( \text{RDFSVoc} \) coincides with \( S_{\text{RDFS}} \)
  - \( \text{ext}_\mathcal{I}(\text{rdfs:Resource}) = R_\mathcal{I} \)
  - \( (\forall x, y, u, v \in R_\mathcal{I})(x, y) \in \text{ext}_\mathcal{I}(\text{rdfs:domain}) \land (u, v) \in \text{ext}_\mathcal{I}(x) \Rightarrow u \in \text{ext}_\mathcal{I}(y) \)
  - \( (\forall x, y, u, v \in R_\mathcal{I})(x, y) \in \text{ext}_\mathcal{I}(\text{rdfs:range}) \land (u, v) \in \text{ext}_\mathcal{I}(x) \Rightarrow v \in \text{ext}_\mathcal{I}(y) \)
  - \( (\forall x, y \in R_\mathcal{I})(x, y) \in \text{ext}_\mathcal{I}(\text{rdfs:subClassOf}) \Rightarrow \text{ext}_\mathcal{I}(x) \subseteq \text{ext}_\mathcal{I}(y) \)
  - \( (\forall x \in \text{ext}_\mathcal{I}(\text{rdf:Class}))(x, \text{rdfs:Resource}) \in \text{ext}_\mathcal{I}(\text{rdfs:subClassOf}) \)
  - \( (\forall x, y \in R_\mathcal{I})(x, y) \in \text{ext}_\mathcal{I}(\text{rdfs:subPropertyOf}) \Rightarrow \text{ext}_\mathcal{I}(x) \subseteq \text{ext}_\mathcal{I}(y) \)
  - \( (\forall x \in \text{ext}_\mathcal{I}(\text{rdfs:ContainerMembershipProperty})) \\
    (x, \text{rdfs:member}) \in \text{ext}_\mathcal{I}(\text{rdfs:subPropertyOf}) \)
- \( f \)-sentences are \( RR \)-sentences
- **satisfaction:** \( \mathcal{I} \models_f (sn, pn, on) \iff \mathcal{I} \models_{RR} (sn, pn, on) \)
RDF and Web Ontology Vocabulary layers

Web Ontology Vocabulary layer

RDF layer
OWL DL theory

\[
\text{OWDL} = (\text{OWDLVoc}, T_{\text{OWDL}}), \text{ where } \text{OWDLVoc} \text{ is } \text{OWLVoc} \text{ together with }
\]

- owl:DeprecatedClass, owl:DisjointClasses, owl:SubClassOf,
- owl:Functional, owl:InverseFunctional, owl:Transitive,
- owl:SameIndividual, DifferentIndividuals, owl:someValues,
- owl:Thing, owl:Nothing,
- owl:intersectionOf, owl:unionOf, owl:complementOf, owl:oneOf,
- owl:someValues, owl:hasValue, owl:maxCardinality

and \( T_{\text{OWDL}} \) is

\[
\{
\text{(owl:intersectionOf, rdf:type, rdf:Property),}
\text{(owl:intersectionOf, rdfs:domain, owl:Class),}
\text{(owl:intersectionOf, rdfs:range, rdf:List),}
\text{(owl:equivalentClass, rdf:type, rdf:Property),}
\text{(owl:equivalentClass, rdfs:subPropertyOf, rdfs:subClassOf),}
\text{(owl:equivalentClass, rdfs:domain, owl:Class),}
\text{(owl:equivalentClass, rdfs:range, owl:Class),}
\text{(owl:disjointWith, rdf:type, rdf:Property),}
\text{(owl:disjointWith, rdfs:domain, owl:Class),}
\text{(owl:disjointWith, rdfs:range, owl:Class),}
\ldots
\}
\]
RDF Serialization of OWL DL: \textbf{owldl}

- signatures: theory morphisms \( f : \text{owldl} \rightarrow (RR,T) \)

- models: \((RR,T)\)-models \( \mathcal{I} \) such that
  - \( R_\mathcal{I} \) includes \( R_{\text{owldl}} \) and the restriction of \( S_\mathcal{I} \) to \( \text{owldlVoc} \) coincides with \( S_{\text{owldl}} \).

- Restrictions expressing the intended meaning of the new features.

- \( f \)-sentences are \( RR \)-sentences

- satisfaction: \( \mathcal{I} \models_f (sn, pn, on) \) iff \( \mathcal{I} \models_{RR} (sn, pn, on) \)
The meaning of layering
Institutions: Main references

- Introducing Institutions, by J. Goguen and R. Burstall, 1984
- Institutions: Abstract model theory for specification and programming, by J. Goguen and R. Burstall, 1992
- Structuring theories on consequence, by J. Fiadeiro and A. Sernadas - 1988
- May I Borrow Your Logic?, by M. Cerioli and J. Meseguer, 1993
- Moving Between Logical Systems, Andrzej Tarlecki, 1995
- Institution Morphisms, by J. Goguen and Gr. Rosu, 2002
- Grothendieck Institutions, by R. Diaconescu, 2002
- Herbrand Theorem, by R. Diaconescu, 2004
Institutions offer

- A rigorous and systematic approach of the logics underlying SW languages
- An important step towards structuring and re-using ontology parts
- A solid framework for relating SW languages with other formalisms and for proving the soundness of the reasoners
Questions?

Thank you!