Ontology Debugging
—— Debugging Semantic Defects

Zlatan Dragisic and Qiang Liu

Ontologies and Ontology Engineering, VT 2011
Linköpings universitet, Sweden

Outline

Ontologies based on DL

- TBox – terminological axioms
  - Concepts: C ⊑ D, C ≡ D
  - Roles: hasMember
- ABox – assertion component
  - Individuals: C(a), D(b), R(a, b)

- Knowledge base: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

OWL DL

- OWL ontology maps to DL knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$
- OWL DL – based on *SHOIN(D)*

(ALC + transitive roles, →S
role hierarchies, →H
nominals, →O
inverse roles, →I
number restrictions) →N
data type properties, data values or data types →(D)

Ontology defects

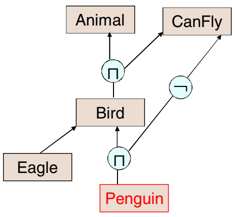
Three main categories of defects:
- Syntactic defects
- Semantic defects
- Modeling defects

Semantic defects

- Contradictions in TBox
  - Unsatisfiable concepts & incoherent ontologies
- Contradictions in ABox
  - Inconsistent ontologies

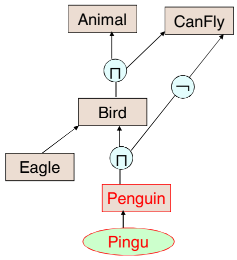
Unsatisfiable concepts

- A concept is unsatisfiable iff $A^I = \varnothing$ for all models I of the ontology
- Unsatisfiable concepts can not have any individuals
- Ontology is incoherent if has an unsatisfiable concept



Inconsistent Ontologies

- An ontology is inconsistent if it has no models
- Contradiction in the ABox



Outline

- Background
- **Explanations for the Semantic Defects**
  - Unsatisfiable Concepts & Incoherent Ontologies
  - Inconsistent Ontologies
- Find the Explanations for the Unsatisfiable Concepts
  - Glass-box approach
  - Black-box approach
- Repairing the Unsatisfiable Concepts
  - Diagnosis
  - Pinpointing
  - Hitting sets with axiom ranking
- Summary

Example : an Incoherent Ontology

Consider the following TBox $\mathcal{T}^*$, where A, B and C are primitive and $A_1, \ldots, A_7$ defined concept names:

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$	$ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$
$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5$	$ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$
$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B$	$ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$
$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$	

⇩

DL Reasoner ⇨ The ontology is incoherent!
The set of unsatisfiable concepts are : $\{A_1, A_3, A_6, A_7\}$.

What are the root causes of these defects?

Explain the Semantic Defects

- We need to identify the sets of axioms which are necessary for causing the logic contradictions.

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$	$ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$
$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5$	$ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$
$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B$	$ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$
$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$	

- For example, for the unsatisfiable concept "A_1", there are two **(and only two)** sets of axioms rendering **$A_1 = \bot$**

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$
$ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$
$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5$
$ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$
$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B$

Minimal Unsatisfiability Preserving Sub-TBoxes (MUPS)

Definition 1 Let A be a concept which is unsatisfiable in a TBox $\mathcal{T}$. A set $\mathcal{T}' \subseteq \mathcal{T}$ is a *minimal unsatisfiability-preserving sub-TBox (MUPS)* of $\mathcal{T}$ if

- A is unsatisfiable in $\mathcal{T}'$, and
- A is satisfiable in every sub-TBox $\mathcal{T}'' \subset \mathcal{T}'$.

We will abbreviate the set of MUPS of $\mathcal{T}$ and A by $mups(\mathcal{T}, A)$.

$$mups(\mathcal{T}^*, A_1) = \{\{ax_1, \cancel{ax_2}\}, \{\cancel{ax_1}, ax_3, ax_4, ax_5\}\}$$

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$
$ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$
$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5$
$ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$
$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B$

The MUPS of an unsatisfiable concept implies the solutions for repairing.

Remove at least one axiom from each axiom set in the MUPS

Example

- For all the unsatisfiable concepts in the ontology, we have

$$mups(\mathcal{T}^*, A_1)= \{\{\cancel{ax_1}, ax_2\}, \{\cancel{ax_1}, \cancel{ax_3}, \cancel{ax_4}, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_3)= \{\{\cancel{ax_3}, \cancel{ax_4}, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_6)= \{\{\cancel{ax_1}, ax_2, \cancel{ax_4}, ax_6\}, \{\cancel{ax_1}, \cancel{ax_3}, \cancel{ax_4}, ax_5, ax_6\}\}$$
$$mups(\mathcal{T}^*, A_7)= \{\{\cancel{ax_4}, ax_7\}\}$$

- Possible ways of repairing all the unsatisfiable concepts in the ontology:

$\{ax_1, ax_3, ax_4\}$

How to represent all these possibilities?

Minimal Incoherence Preserving Sub-TBox (MIPS)

Definition 2 Let $\mathcal{T}$ be an incoherent TBox. A TBox $\mathcal{T}' \subseteq \mathcal{T}$ is a *minimal incoherence-preserving sub-TBox (MIPS) of $\mathcal{T}$* if

- $\mathcal{T}'$ is incoherent, and
- every sub-TBox $\mathcal{T}'' \subset \mathcal{T}'$ is coherent.

$$mups(\mathcal{T}^*, A_1)= \{\{ax_1, \cancel{ax_2}\}, \{ax_1, ax_3, \cancel{ax_4}, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_3)= \{\{ax_3, \cancel{ax_4}, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_6)= \{\{ax_1, \cancel{ax_2}, \cancel{ax_4}, ax_6\}, \{ax_1, ax_3, \cancel{ax_4}, ax_5, ax_6\}\}$$
$$mups(\mathcal{T}^*, A_7)= \{\{\cancel{ax_4}, \cancel{ax_7}\}\}$$

We will abbreviate the set of MIPS of $\mathcal{T}$ by $mips(\mathcal{T})$. For $\mathcal{T}^*$ we get three MIPS:

$$mips(\mathcal{T}^*) \ = \{\{ax_1, ax_2\}, \{ax_3, ax_4, ax_5\}, \{ax_4, ax_7\}\}$$

A possible repairing is $\{ax_i\} \cup \{ax_j\} \cup \{ax_k\}$, where

- $ax_i \in \{ax_1, ax_2\}$
- $ax_j \in \{ax_3, ax_4, ax_5\}$
- $ax_k \in \{ax_4, ax_7\}$

Example : an Inconsistent Ontology

$$\mathcal{T}^* = \{ \quad ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3 \qquad ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$$
$$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5 \qquad ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$$
$$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B \qquad ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$$
$$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$$

$$\mathcal{A}^* = \{ \quad ass_1: i_1 : A_1 \qquad ass_2: i_2 : A_4$$
$$ass_3: i_3 : \neg B \qquad ass_4: s(i_2, i_3)$$
$$ass_5: i_4 : A_4 \sqcap A_5 \qquad ass_6: i_2 : \forall s.B$$

⇩

DL Reasoner ⇨ The ontology is inconsistent !

Minimal Inconsistent Preserving Sub-Ontologies (MISO)

Definition 4 Let $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ be an inconsistent ontology. An ontology $\mathcal{O}' = (\mathcal{T}', \mathcal{A}')$ where $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathcal{A}' \subseteq \mathcal{A}$ is a *minimal inconsistency preserving sub-ontology (MISO) of $\mathcal{O}$* if, and only if,

- $\mathcal{O}'$ is inconsistent, and
- every sub-ontology $\mathcal{O}'' = (\mathcal{T}'', \mathcal{A}'')$, where $\mathcal{T}'' \subset \mathcal{T}'$ and $\mathcal{A}'' \subseteq \mathcal{A}'$ is consistent

$$\mathcal{T}^* = \{ \quad ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3 \qquad ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$$
$$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5 \qquad ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$$
$$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B \qquad ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$$
$$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$$

$$\mathcal{A}^* = \{ \quad ass_1: i_1 : A_1 \qquad ass_2: i_2 : A_4$$
$$ass_3: i_3 : \neg B \qquad ass_4: s(i_2, i_3)$$
$$ass_5: i_4 : A_4 \sqcap A_5 \qquad ass_6: i_2 : \forall s.B$$

There are a number of MISOs, for example,

$\mathcal{O}_1^* = (\{ax_1, ax_2\}, \{ass_1\}),$

$\mathcal{O}_2^* = (\varnothing, \{ass_3, ass_4, ass_6\})$

...

Outline

The Example Seen Before

$$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3 \qquad ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$$
$$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5 \qquad ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$$
$$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B \qquad ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$$
$$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$$

⇩

DL Reasoner ⇨ The ontology is incoherent!
The set of unsatisfiable concepts are : $\{A_1, A_3, A_6, A_7\}$.

⇨

$$mups(\mathcal{T}^*, A_1)= \{\{ax_1, ax_2\}, \{ax_1, ax_3, ax_4, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_3)= \{\{ax_3, ax_4, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_6)= \{\{ax_1, ax_2, ax_4, ax_6\}, \{ax_1, ax_3, ax_4, ax_5, ax_6\}\}$$
$$mups(\mathcal{T}^*, A_7)= \{\{ax_4, ax_7\}\}$$

How to make the program to find MUPS?

Glass-box Approach

- By modifying the internals of a DL reasoner for extract and reveal the cause for unsatisfiability.

- It gives explanation to why the error occurs.
  - e.g. find out the axioms in the ontology responsible for the logic contradictions.

19

Glass-box Approach

- Phase 1 : Clash detection procedure
  - Use tableaux algorithm by applying transformation rules to individuals in the ontology until no more rules are applicable or an individual has a clash.
    - For example, an individual belongs to a concept and its complement.
- Phase 2 : Tableaux trace procedure
  - Trace back to find the source axioms supporting the clash and determine the minimal sets of support (i.e. MUPS).

20

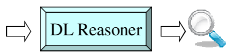
Glass-box Approach

- Advantage
  - By tightly integrating the debugging with the reasoning procedure, precise results can be obtained.
- Disadvantage
  - The reasoner needs to maintain extra data structures to track the source and its dependencies, which introduces additional memory and computation consumption.

21

Black-box Approach

- By using DL reasoner as an oracle for a certain set of questions (standard inferences such as satisfiability, subsumption, etc.)



- Give explanation to the dependencies between unsatisfiable concepts.
  - i.e. distinguish *root* from *derived* unsatisfiable concepts.

$Student \sqsubseteq (\geq 2) hasAdvisor \sqcap (\leq 1) hasAdvisor$

$GraduateStudent \sqsubseteq Student$

22

Black-box Approach

- Dependency detection algorithm
  - For each unsatisfiable concept in the ontology, the algorithm returns all its parent dependency unsatisfiable concepts along with the corresponding axioms that link this concept to the parent.

e.g. Unsatisfiable concept A is derived if:

(1) $A = (B \sqcap C \sqcap \ldots \sqcap D)$, and one of $B, C, \ldots, D$ is unsatisfiable

(2) $A = (B \sqcup C \sqcup \ldots \sqcup D)$, and all $B, C, \ldots, D$ are unsatisfiable

(3) $A \sqsubseteq (\geq 1 p)$, $domain(p) = B$, and B is unsatisfiable

(4) $A = \exists(p, B)$ and B is unsatisfiable

23

Black-box Approach

- Advantage
  - Reasoner independence, i.e., you do not need a specialized, explanation generating reasoner.
  - Avoid the performance penalty (memory consumption) of glass box techniques.
- Disadvantage
  - Since it does not follow the standard reasoning procedure, the results are sound but not complete.

24

Outline

- Background
- Explanations for the Semantic Defects
  - Unsatisfiable Concepts & Incoherent Ontologies
  - Inconsistent Ontologies
- Find the Explanations for the Unsatisfiable Concepts
  - Glass-box approach
  - Black-box approach
- **Repairing the Unsatisfiable Concepts**
  - Diagnosis
  - Pinpointing
  - Hitting sets with axiom ranking
- Summary

25

Diagnosis

- Generic method – applicable to a number of domains
- System represented as the set (SD, COMP)
  - SD – system description
  - COMP – set of system components

Example

- SD

$$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3 \qquad ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$$
$$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5 \qquad ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$$
$$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B \qquad ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$$
$$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$$

- COMP – ax_1, ax_2, ax_3, ax_4, ax_5, ax_6, ax_7

26

Diagnosis

- Observations made about the system – set OBS
  - Example: A_1, A_3, A_6, A_7 – are unsatisfiable

Definition 2.4. A *diagnosis* for (SD, COMPONENTS, OBS) is a minimal set $\Delta \subseteq$ COMPONENTS such that

$$\text{SD} \cup \text{OBS} \cup \{\text{AB}(c) \mid c \in \Delta\} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$$

is consistent.

Diagnoses for our example:
$\{ax_1, ax_4\}$, $\{ax_1, ax_5, ax_7\}$, $\{ax_2, ax_4\}$,…

27

Diagnosis

Naïve approach:

Generate subsets Δ of COMP and check for consistency of $\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPONENTS} - \Delta\}$

- Problem: complexity

28

Conflict sets and hitting sets

Definition 4.1. A *conflict set* for (SD, COMPONENTS, OBS) is a set $\{c_1, \ldots, c_k\} \subseteq$ COMPONENTS such that

$$\text{SD} \cup \text{OBS} \cup \{\neg\text{AB}(c_1), \ldots, \neg\text{AB}(c_k)\}$$

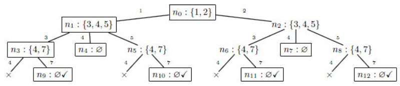
Conflict sets for our example:
$\{ax_1, ax_2\}$, $\{ax_3, ax_4, ax_5\}$, $\{ax_4, ax_7\}$

Definition 4.3. Suppose C is a collection of sets. A *hitting set* for C is a set $H \subseteq \bigcup_{S \in C} S$ such that $H \cap S \neq \{\ \}$ for each $S \in C$. A hitting set for C is *minimal* iff no proper subset of it is a hitting set for C.

29

Hitting set example

Conflict sets for our example:
$\{ax_1, ax_2\}$, $\{ax_3, ax_4, ax_5\}$, $\{ax_4, ax_7\}$



30

Theorem 4.4. *$\Delta \subseteq$ COMPONENTS is a diagnosis for* (SD, COMPONENTS, OBS) *iff Δ is a minimal hitting set for the collection of conflict sets for* (SD, COMPONENTS, OBS).

How to find conflict sets?

Compute all refutations of $\mathrm{SD} \cup \mathrm{OBS} \cup \{\neg \mathrm{AB}(c) \mid c \in \mathrm{COMPONENTS}\}$

Record the AB instances entering the refutation

If $\{\neg \mathrm{AB}(c_1), \ldots, \neg \mathrm{AB}(c_k)\}$ is set of of AB instances used in the refutation then $\{c_1, \ldots, c_k\}$ is a conflict set.

Pinpointing

1 Diagnosis with hitting sets still computationaly too expensive – NP complete

1 Compute set of axioms which need to be removed/fixed

  1 does not have to be minimal

Algorithm:

1. Find unsatisfiable concepts
2. Search for MUPS for unsatisfiable concepts
3. Calculate MIPS from MUPS
4. Calculate the pinpoint from MIPS
5. Remove axioms in the pinpoint from the ontology

Pinpointing

1. Find unsatisfiable concepts

$ax_1: A_1 \dot{\sqsubseteq} \neg A \sqcap A_2 \sqcap A_3$ $\quad ax_2: A_2 \dot{\sqsubseteq} A \sqcap A_4$

$ax_3: A_3 \dot{\sqsubseteq} A_4 \sqcap A_5$ $\quad ax_4: A_4 \dot{\sqsubseteq} \forall s.B \sqcap C$

$ax_5: A_5 \dot{\sqsubseteq} \exists s.\neg B$ $\quad ax_6: A_6 \dot{\sqsubseteq} A_1 \sqcup \exists r.(A_3 \sqcap \neg C \sqcap A_4)$

$ax_7: A_7 \dot{\sqsubseteq} A_4 \sqcap \exists s.\neg B$

⇩

DL Reasoner ⇨ The ontology is incoherent!
The set of unsatisfiable concepts are : $\{A_1, A_3, A_6, A_7\}$.

Pinpointing

2. Search for MUPS for unsatisfiable concepts
3. Calculate MIPS from MUPS

$$mups(\mathcal{T}^*, A_1) = \{\{ax_1, ax_2\}, \{ax_1, ax_3, ax_4, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_3) = \{\{ax_3, ax_4, ax_5\}\}$$
$$mups(\mathcal{T}^*, A_6) = \{\{ax_1, ax_2, ax_4, ax_6\}, \{ax_1, ax_3, ax_4, ax_5, ax_6\}\}$$
$$mups(\mathcal{T}^*, A_7) = \{\{ax_4, ax_7\}\}$$

$$mips(\mathcal{T}^*) = \{\{ax_1, ax_2\}, \{ax_3, ax_4, ax_5\}, \{ax_4, ax_7\}\}$$

Pinpointing

4. Calculate the pinpoint from MIPSs
5. Remove axioms in pinpoint from the ontology

To do this we need to calculate cores.
Cores are sets of axioms occuring in several incoherent TBoxes.
  Core arity – number of incoherent TBoxes an axiom appears in
  Core size – number of axioms in the core

We are interested in cores of maximal arity (and size)

$$mips(\mathcal{T}^*) = \{\{ax_1, ax_2\}, \{ax_3, ax_4, ax_5\}, \{ax_4, ax_7\}\}$$

⬇

$$P(\mathcal{O}) = \{\{ax_1\}, \{ax_4\}\}$$

Hitting set with axiom ranking

1 Hitting sets generated from MUPS

1 Ranking criteria introduced into the algorithm

1 Optimality based on minimal path rank

Axiom Ranking strategies

Ranking based on:

- Impact analysis
- User test cases
- Provenance information regarding change
- Syntactic relevance

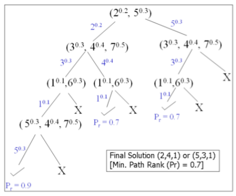
Further improvements

- Focus on root unsatisfiable concepts

37

Hitting set with axiom ranking

$C = \{\{2,5\},\{3,4,7\},\{1,6\},\{4,5,7\},\{1,2,3\}\}$



$r(1) = 0.1$
$r(2) = 0.2$
$r(3) = 0.3$
$r(4) = 0.4$
$r(5) = 0.3$
$r(6) = 0.3$
$r(7) = 0.5$

38

Outline

- Background
- Explanations for the Semantic Defects
  - Unsatisfiable Concepts & Incoherent Ontologies
  - Inconsistent Ontologies
- Find the Explanations for the Unsatisfiable Concepts
  - Glass-box approach
  - Black-box approach
- Repairing the Unsatisfiable Concepts
  - Diagnosis
  - Pinpointing
  - Hitting sets with axiom ranking
- **Summary**

39

Summary

- In this presentation, we introduced the existing research work on ontology debugging, focusing on one kind of ontological defects —— semantic defects.
- We reviewed different kinds of semantic defects and explanations for these semantic defects.
- We reviewed the ontology debugging techniques for
  - Finding explanations for the unsatisfiable concepts
  - Repairing the unsatisfiable concepts

40

References

[1] Schlobach S, **Debugging and semantic clarification by pinpointing**, Proceedings of the 2nd European Semantic Web Conference - ESWC05, LNCS 3532, 226-240, 2005.

[2] Kalyanpur A, Parsia B, Sirin E, Cuenca-Grau B, **Repairing unsatisfiable concepts in OWL ontologies**, Proceedings of the 3rd European Semantic Web Conference - ESWC06, LNCS 4011, 170-184, 2006.

[3] Kalyanpur A, Parsia B, Sirin E, Hendler J, **Debugging unsatisfiable classes in OWL ontologies**, Journal of Web semantics, 3(4):268-293, 2005.

[4] Schlobach S and Huang Z, **Inconsistent Ontology Diagnosis and Repair**, SEKT Deliverable D3.6.3, 2007.

[5] R. Reiter. **A theory of diagnosis from first principles**. 1987. Artificial Intelligence 32:57-95.

41

Thank you!

42