

Repairing the missing is-a structure of ontologies

Patrick Lambrix, Qiang Liu, He Tan
Linköpings Universitet
Sweden

Outline

- n Background
- n Theory of our approach
- n Repairing the structure of an ontology
- n Implemented system
- n Experiments
- n Conclusion & Future Work

Outline

n **Background**

- n Theory of our approach
- n Repairing the structure of an ontology
- n Implemented system
- n Experiments
- n Conclusion & Future Work

Ontologies

- n Developing ontologies is not an easy task.
Many ontologies have some underlying defects.

- n Such ontologies, although often useful, also lead to problems when used in semantically-enabled applications.
Wrong conclusions may be derived or valid conclusions may be missed.

Defects in ontologies

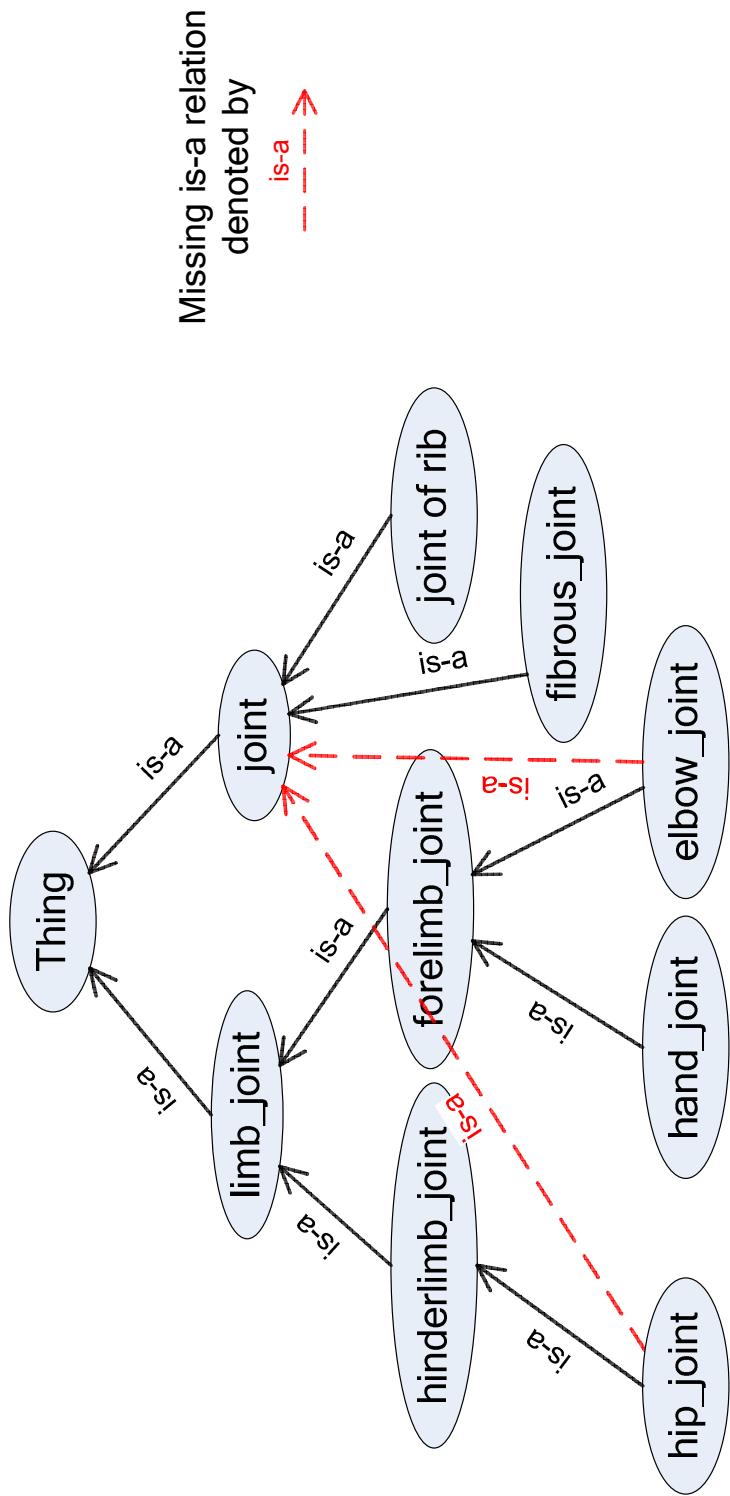
- Syntactic defects
 - ☒ eg. wrong tags or incorrect format
Easy to detect and resolve
- Semantic defects
 - ☒ eg. unsatisfiable concepts or inconsistent ontologies
 - There has been some work on detection
 - In recent years, there is some work on repairing
- Modeling defects
 - ☒ eg. wrong or missing relations
 - Solution requires domain knowledge.
 - There is some work on detection, but little work on repairing

Missing is-a relations

- n In 2008 Ontology Alignment Evaluation Initiative (OAEI)
 - Anatomy track, task 4
 - ¤ Ontology MA : Adult Mouse Anatomy Dictionary (2744 concepts)
 - ¤ Ontology NCI-A : NCI Thesaurus - anatomy (3304 concepts)
 - ¤ Partial reference alignment between them (988 mappings)
 - n 121 missing is-a relations in MA
 - n 83 missing is-a relations in NCI-A

Example

- A small piece of Ontology MA



Missing is-a relations

n Impact

- ☒ They undermine the applications and algorithms relying on it, eg. ontology alignment.
 - ☒ The completeness of the structure and the correct use of the structural relations within the ontologies has an important influence on the quality of the results.

--- *Anatomy Results of OAEI 2009 Campaign*

- ☒ They influence the finding of semantic defects.

Missing is-a relations

- n Our focus: repairing of missing is-a structure in an ontology.

Given a set of missing is-a relations, how to repair the missing is-a structure in an ontology?

- n Assumptions
 - ☒ The ontology is defined using named concepts and subsumption axioms.
 - ☒ The existing is-a relations in the ontology are correctly defined.

Outline

- n Background
- n **Theory of our approach**
- n Repairing the structure of an ontology
- n Implemented system
- n Experiments
- n Conclusion & Future Work

Structural repair

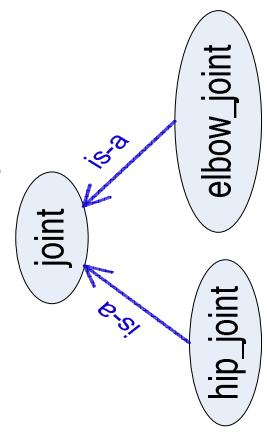
- n Structural repair for an ontology and a given set of missing is-a relations is a set of is-a relations that when added to the ontology allows us to infer the missing is-a relations.
- Structural repair repairs the missing is-a structure.
 - ¤ Is-a relations in structural repair are called **repairing actions**.
- n Set of missing is-a relations is a structural repair, but it is not always the only nor the best choice.

Example

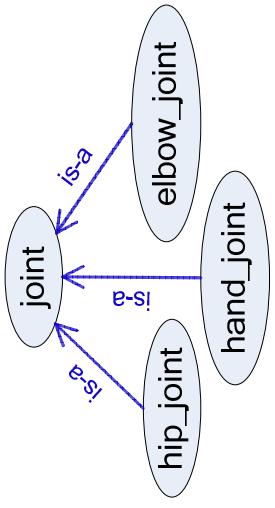
- Some possible structural repairs (SRs)

Repairing action
denoted by
 **is-a**

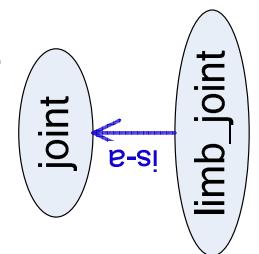
Structural Repair 1



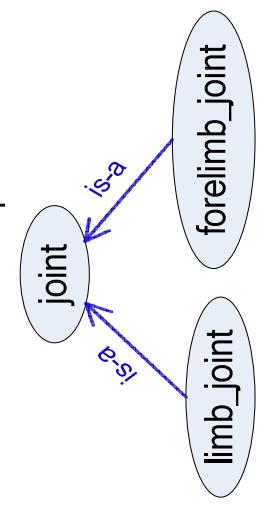
Structural Repair 2



Structural Repair 3



Structural Repair 4



Our goal: generate structural repairs
that are relevant for domain experts

Repair Preference I

Axiom-Preference

- Prefer to use structural repair without redundant or non-contributing repairing actions.

Definition 2. Let \mathcal{R}_1 and \mathcal{R}_2 be structural repairs for the ontology \mathcal{O} with respect to \mathcal{M} , then \mathcal{R}_1 is axiom-preferred to \mathcal{R}_2 (notation $\mathcal{R}_1 \ll_A \mathcal{R}_2$) iff $\mathcal{R}_1 \subseteq \mathcal{R}_2$.

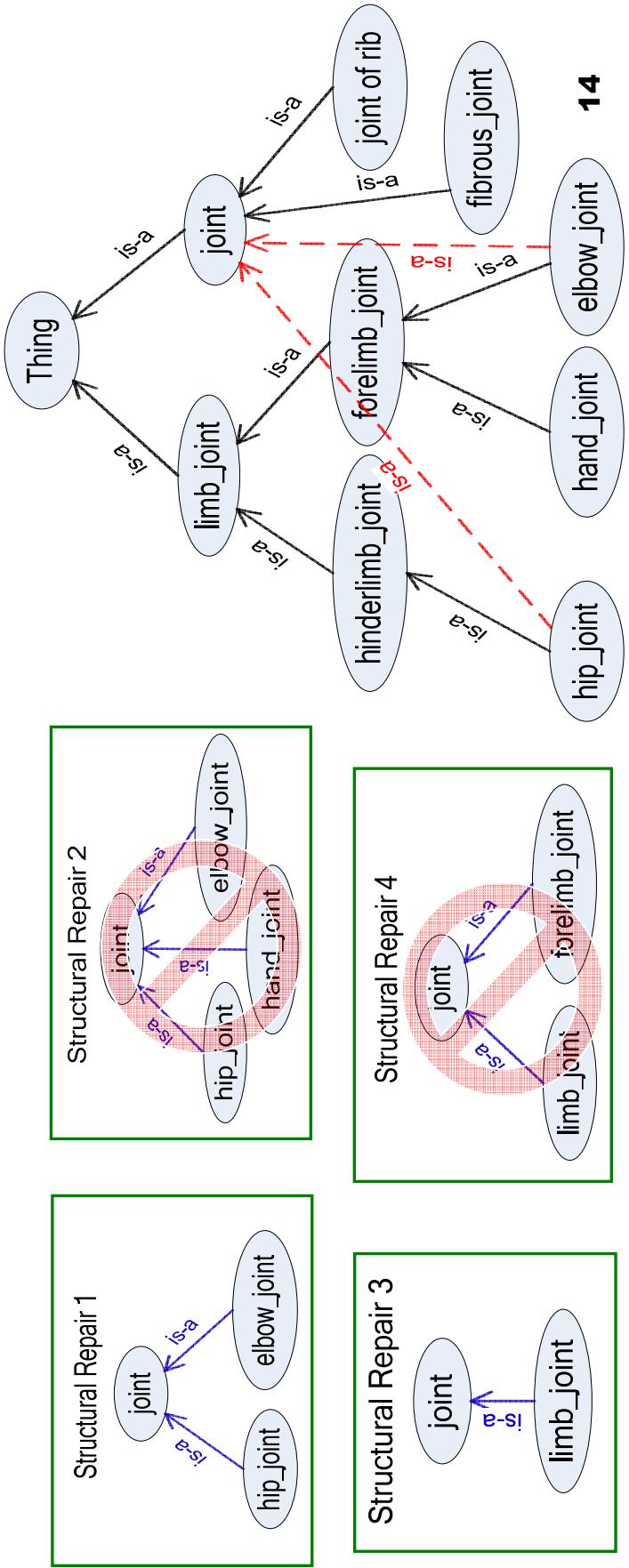
Example

n Prefer SR1 to SR2

In SR2, repairing action (**hand_joint, joint**) is non-contributing.

n Prefer SR3 to SR4

In SR4, repairing action (**forelimb_joint, joint**) is redundant w.r.t.
(limb_joint, joint).



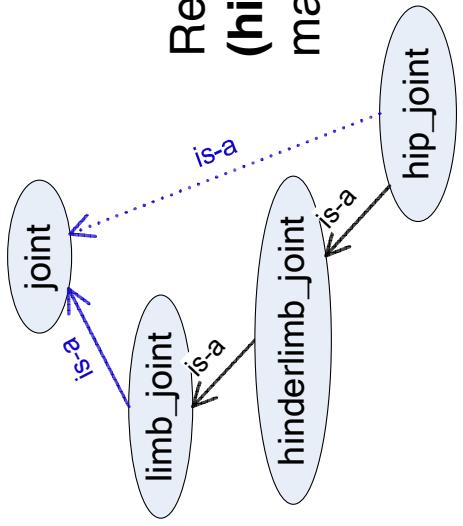
Repair Preference II

Information-Preference

- Prefer to use structural repair with more informative repairing actions.

Definition 3. We say that (X_1, Y_1) is more informative than (X_2, Y_2) iff $X_2 \rightarrow X_1$ and $Y_1 \rightarrow Y_2$.

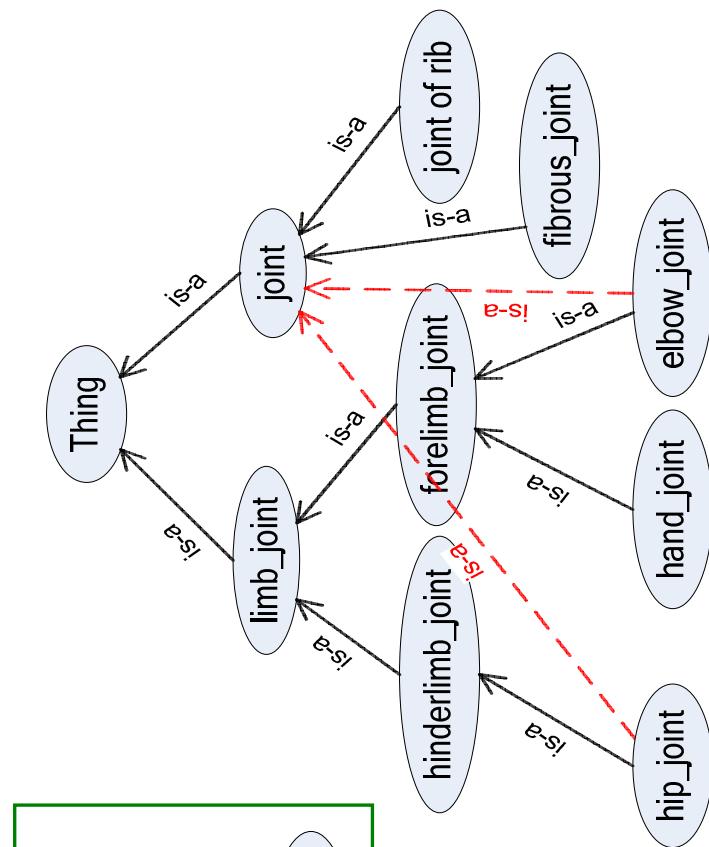
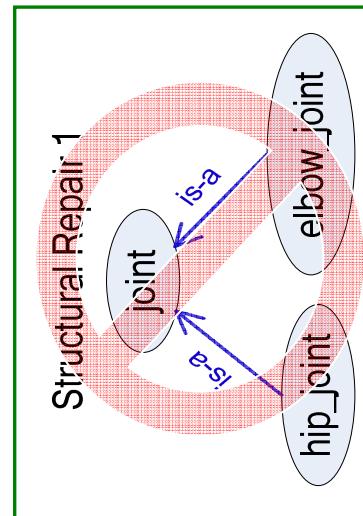
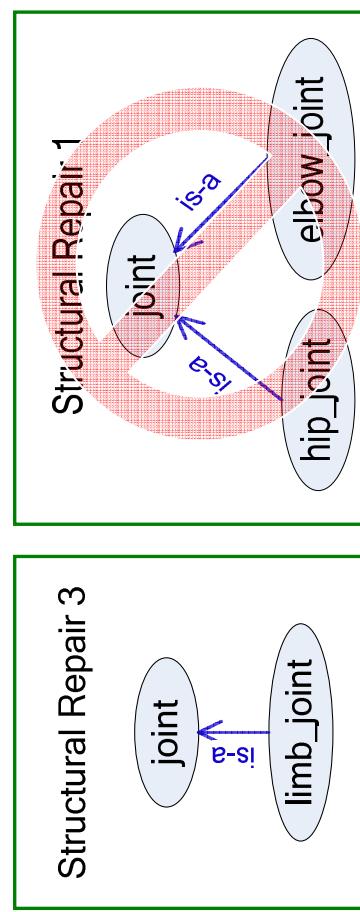
Let \mathcal{R}_1 and \mathcal{R}_2 be structural repairs for the ontology \mathcal{O} with respect to \mathcal{M} . Then \mathcal{R}_1 is **information-preferred** to \mathcal{R}_2 (notation $\mathcal{R}_1 \ll_I \mathcal{R}_2$) iff $\exists (X_1, Y_1) \in \mathcal{R}_1, (X_2, Y_2) \in \mathcal{R}_2$: (X_1, Y_1) is more informative than (X_2, Y_2) .



Repairing action **(limb_joint, joint)** is more informative than **(hip_joint, joint)**, in that adding the former to the ontology will make the latter derivable from the extended ontology.

Example

- Prefer SR3 to SR1
(limb_joint, joint) is more informative than both repairing actions in SR1



Note: Adding the given missing is-a relations is not always the best choice.

Repair Preference III

Strict-Hierarchy-Preference

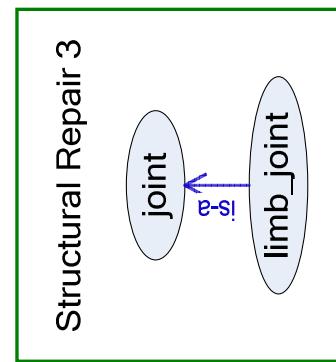
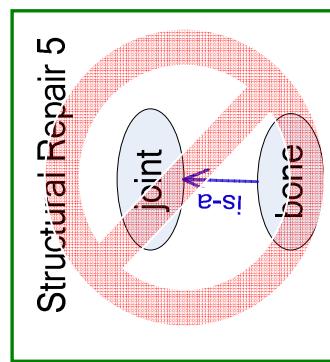
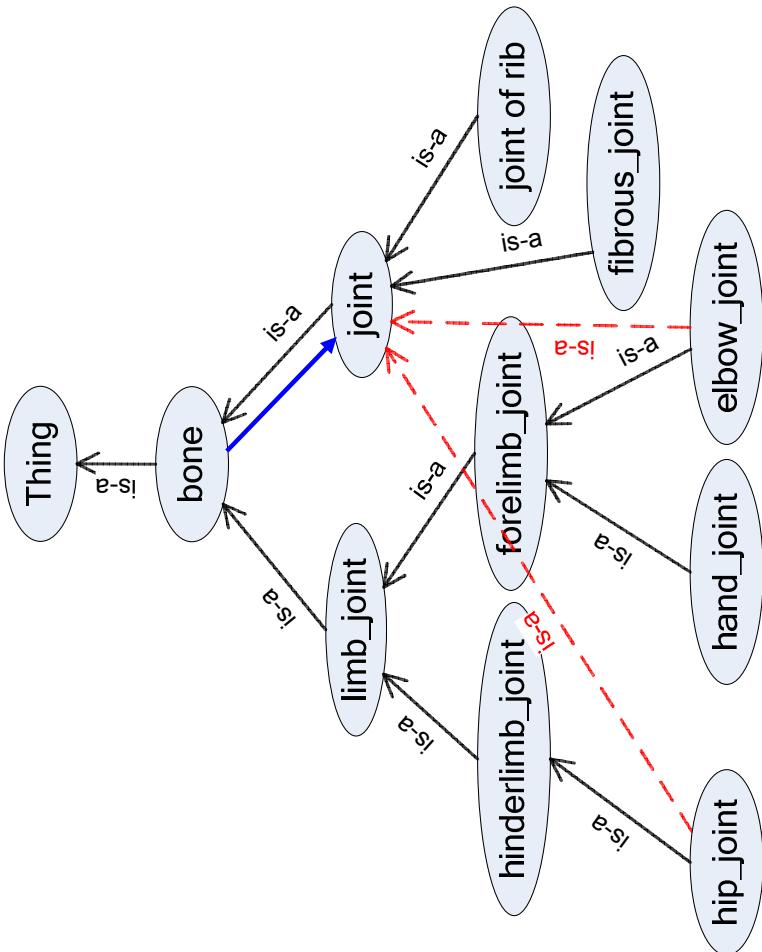
- Prefer to use the structural repair which does not change the existing is-a relations in the original ontology into equivalence relations.

Definition 4. Let \mathcal{R}_1 and \mathcal{R}_2 be structural repairs for the ontology $\mathcal{O} = (\mathcal{C}, \mathcal{I})$ with respect to \mathcal{M} . Then \mathcal{R}_1 is **strict-hierarchy-preferred** to \mathcal{R}_2 (notation $\mathcal{R}_1 \ll_{SH} \mathcal{R}_2$) iff $\exists A, B \in \mathcal{C}: (\mathcal{C}, \mathcal{I}) \models A \rightarrow B$ and $(\mathcal{C}, \mathcal{I}) \not\models B \rightarrow A$ and $(\mathcal{C}, \mathcal{I} \cup \mathcal{R}_1) \not\models B \rightarrow A$.

Example

Suppose we have a new concept 'bone' as shown in the ontology

- Prefer SR3 to SR5
(bone, joint) will introduce an equivalence relation between 'joint', and 'bone' .



Outline

- n Background
- n Theory of our approach
- n **Repairing the structure of an ontology**
- n Implemented system
- n Experiments
- n Conclusion & Future Work

Generating repairing actions

Algorithm 1 - basic algorithm

n Intuition

- Given a set of missing is-a relations, find possible repairing actions with consideration that all missing is-a relations will be repaired.

Input:

Source ontology, missing is-a relations.

Output

Repairing actions.

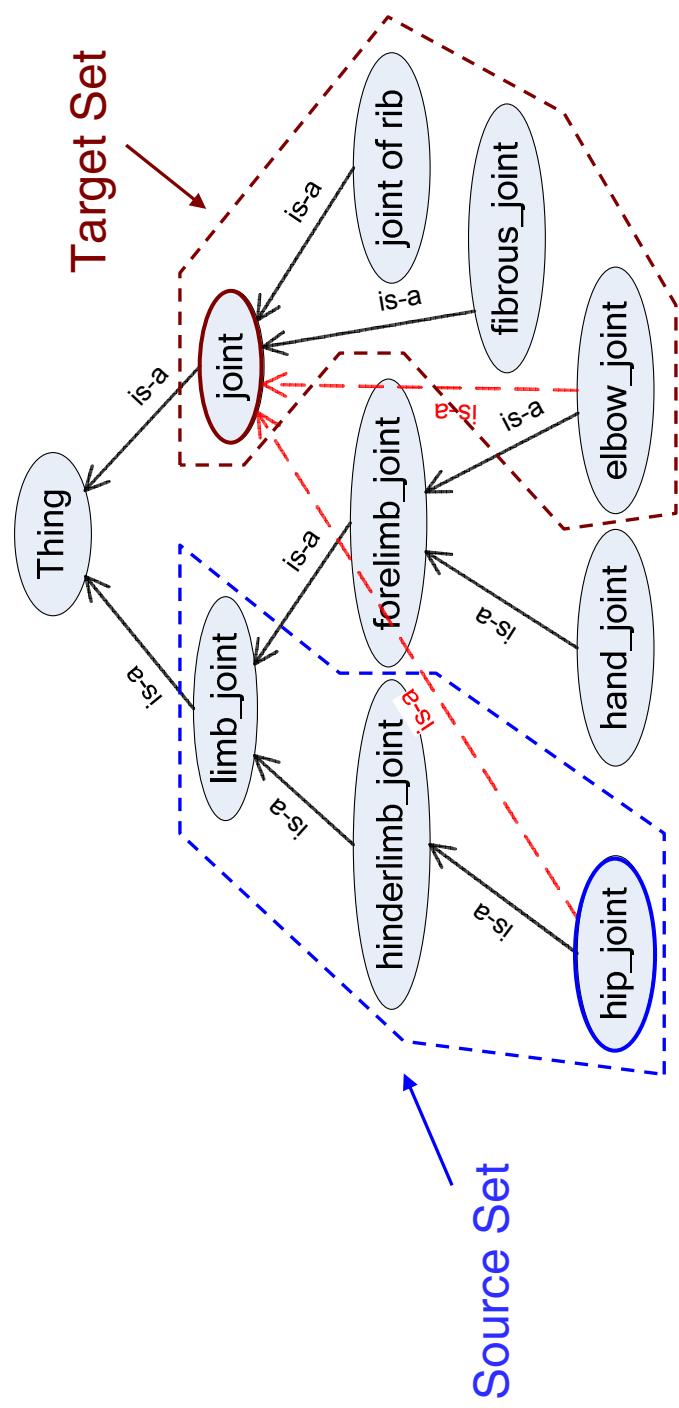
Algorithm

1. Initialize KB with ontology;
2. For every missing is-a relation (A_i, B_i) : add the axiom $A_i \rightarrow B_i$ to the KB;
3. For each (A_i, B_i) :
 - 3.1 $\text{Source}(A_i, B_i) := \text{super-concepts}(A_i) - \text{super-concepts}(B_i);$
 - 3.2 $\text{Target}(A_i, B_i) := \text{sub-concepts}(B_i) - \text{sub-concepts}(A_i);$
4. Missing is-a relation (A_i, B_i) can be repaired by choosing an element from $\text{Source}(A_i, B_i) \times \text{Target}(A_i, B_i)$.

Fig. 1. Algorithm for generating repairing actions - 1.

Example

For missing is-a relation (**hip_joint, joint**),
we get **3x4** repairing actions using algorithm 1.



Algorithm 1 - basic algorithm

- n Consistent with the preferences
 - ¤ For a repairing action (**S**, **T**) regarding missing is-a relation (**A**, **B**), it is guaranteed that
 - n since **A** → **S** and **T** → **B**
 - ¤ (**S**, **T**) is relevant for repairing (**A**, **B**) Axiom-Preference
 - ¤ (**S**, **T**) is more informative than (**A**, **B**) Information-Preference
 - n (**A**, **T**) and (**S**, **B**) will not introduce equivalence relations, where in the original ontology we have only is-a relations

Strict-hierarchy-Preference

- 3. For each (A_i, B_i):
 - 3.1 Source(A_i, B_i) := super-concepts(A_i) - super-concepts(B_i);
 - 3.2 Target(A_i, B_i) := sub-concepts(B_i) - sub-concepts(A_i);
- 4. Missing is-a relation (A_i, B_i) can be repaired by choosing an element from Source(A_i, B_i) × Target(A_i, B_i).

Algorithm 2 - extended algorithm

n Intuition:

- ☒ Taking into account influences of other missing is-a relations that are common to all possible choices for repairing actions of other missing is-a relations.

Input:

Source ontology, missing is-a relations.

Output

Repairing actions.

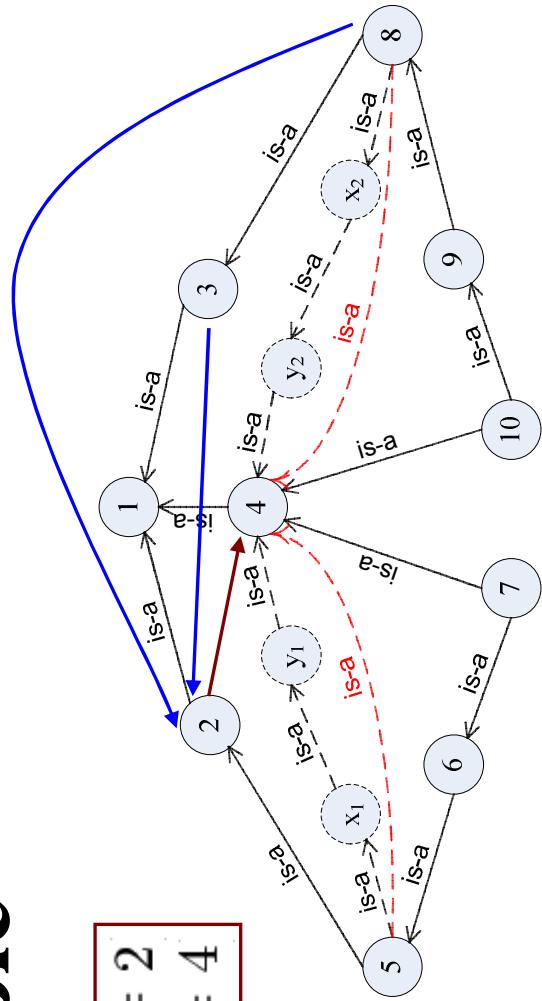
Algorithm

1. Initialize KB with ontology;
2. For every missing is-a relation $A_i \rightarrow B_i$:
 - 2.1 create new concepts X_i and Y_i in the KB;
 - 2.2 add the axioms $A_i \rightarrow X_i$, $X_i \rightarrow Y_i$, $Y_i \rightarrow B_i$ to the KB;
3. For each (A_i, B_i) :
 - 3.1 $\text{Source-ext}(A_i, B_i) := \text{super-concepts}(A_i) - \text{super-concepts}(X_i);$
 - 3.2 $\text{Target-ext}(A_i, B_i) := \text{sub-concepts}(B_i) - \text{sub-concepts}(Y_i);$
4. Missing is-a relation (A_i, B_i) can be repaired by choosing an original ontology element from $\text{Source-ext}(A_i, B_i)$ and an original ontology element from $\text{Target-ext}(A_i, B_i)$.

Fig.3. Algorithm for generating repairing actions - 2.

Example

$$\boxed{x_1 = 2 \\ y_1 = 4}$$



$$\begin{aligned}
 Source-ext(5, 4) &= \{5, 4, 1, 2, x_1, y_1\} - \{4, 1, x_1, y_1\} = \{5, 2\} \\
 Target-ext(5, 4) &= \{4, 8, 9, 10, 5, 6, 7, x_1, y_1, x_2, y_2\} - \{5, 6, 7, x_1, y_1\} \\
 &= \{4, 8, 9, 10, x_2, y_2\} \\
 Source-ext(8, 4) &= \{8, 4, 1, 3, x_2, y_2\} - \{4, 1, x_2, y_2\} = \{8, 3\} \\
 Target-ext(8, 4) &= \{4, 8, 9, 10, 5, 6, 7, x_1, y_1, x_2, y_2\} - \{8, 9, 10, x_2, y_2\} \\
 &= \{4, 5, 6, 7, \boxed{2} \}
 \end{aligned}$$

For instance, if we choose repairing action $(2, 4)$ for missing is-a relation $(5, 4)$, which means x_1 and y_1 will become equivalent to 2 and 4 respectively, the influence is that concept 2 will become a new element in $Target-ext(8, 4)$.

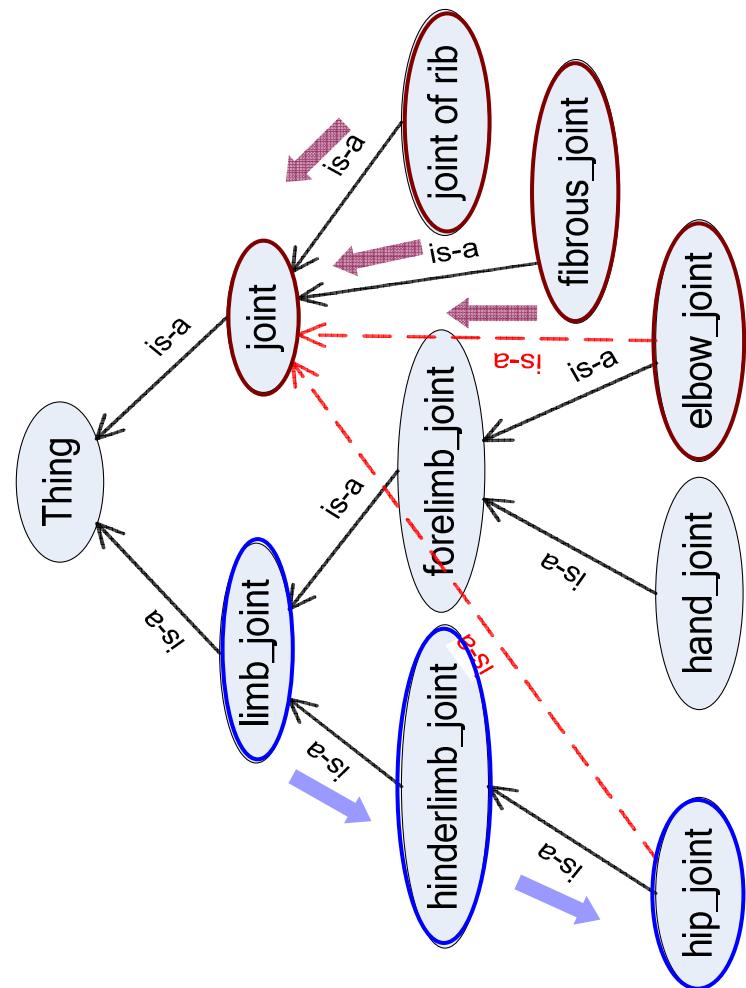
Recommending repairing actions

Recommendding algorithm

- We assume that we can query the external domain knowledge regarding subsumption of concepts
 - ✖ General thesauri
 - e.g. WordNet
 - ✖ Specialized domain-specific sources
 - e.g. UMLS (Unified Medical Language System)
- Algorithm
 - ✖ Given a missing is-a relation with possible repairing actions, recommend the most informative repairing actions that are supported by evidence in the domain knowledge.

Example

For missing is-a relation (**hip_joint, joint**),
we get the recommendation (**limb_joint, joint**).



Executing repairing actions

Executing repairing actions

n Intuition

- ¤ Every time a repairing action is chosen and executed, the repairing actions for the other missing is-a relations need to be recomputed based on the ontology extended with the chosen repairing action.
- ¤ In order to facilitate updates, we introduce an algorithm to keep track of the influences.

Executing repairing actions

n Intuition

- o After a repairing action (**X**, **Y**) is executed, for any other missing is-a relations, for example (**A**, **B**):
 - n **Source(A, B)** changes only when **A** or **B** is a sub-concept of **X**
 - n **Target(A, B)** changes only when **A** or **B** is a super-concept of **Y**

Ranking missing is-a relations

Ranking missing is-a relations

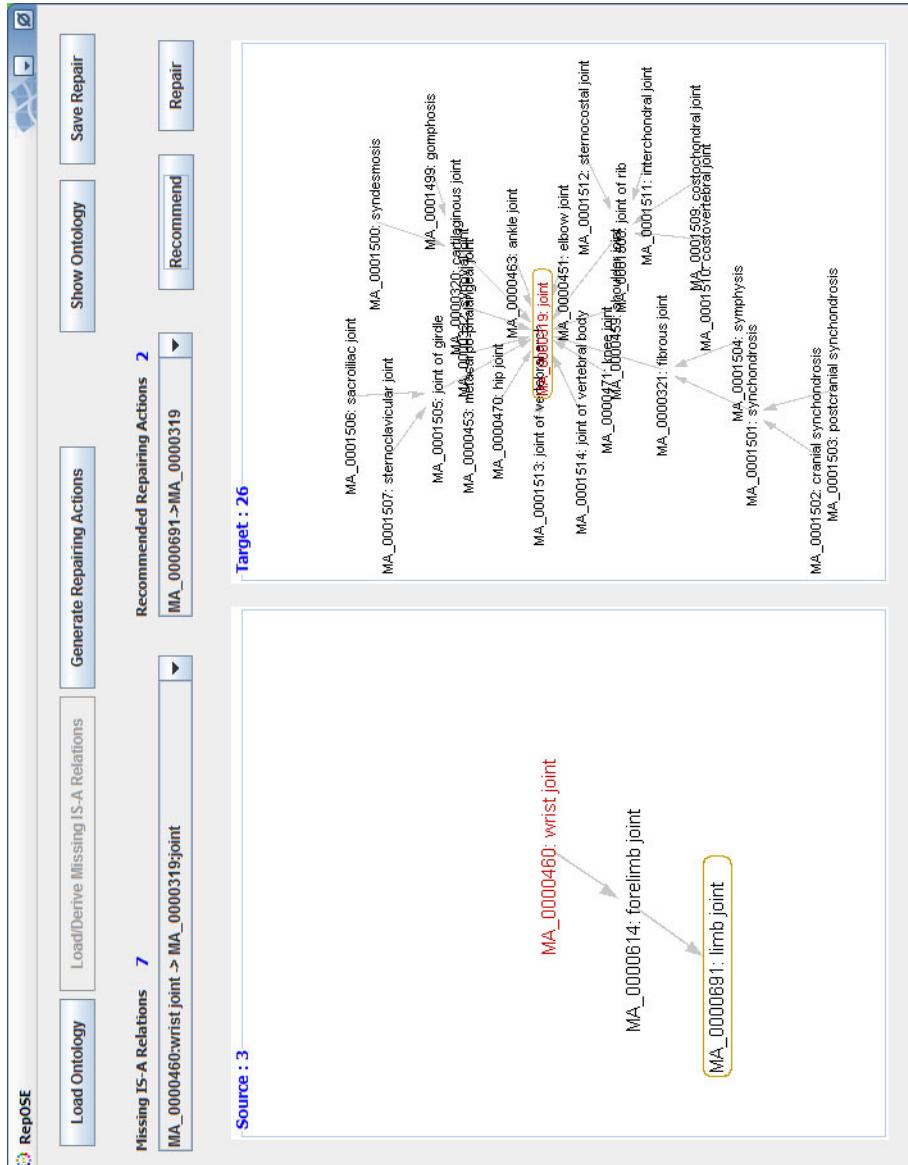
- n Idea
 - ¤ There may be many missing is-a relations that need to be repaired.
 - ¤ It may be easier for the user to start with the ones where there are the fewest choices.
 - ¤ So, we rank the missing is-a relations according to the number of possible repairing actions, which is “size of Source set” \times “size of Target set”

Outline

- n Background
- n Theory of our approach
- n Repairing the structure of an ontology
- n **Implemented System**
- n Experiments
- n Conclusion & Future Work

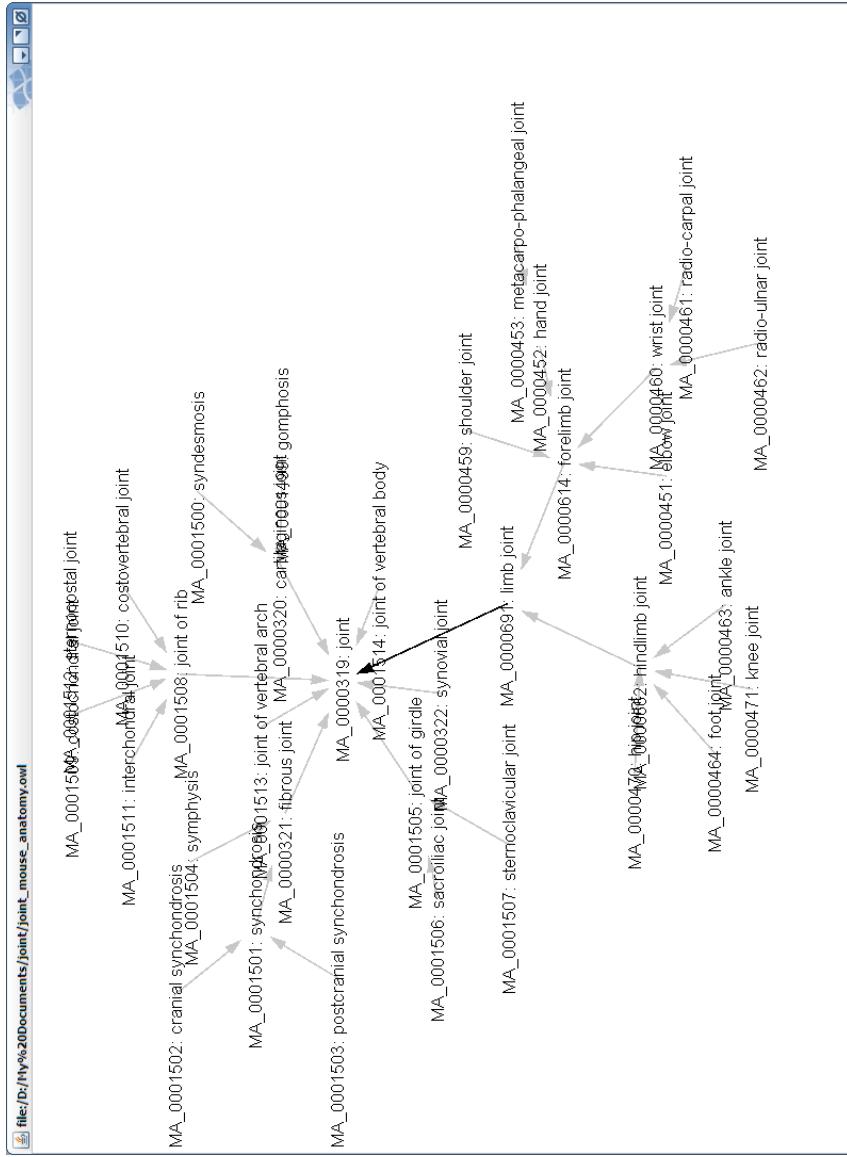
Implemented System

n RepOSE (*Repair of Ontological Structure Environment*)



Implemented System

- n Example of repaired ontology



Outline

- n Background
- n Theory of our approach
- n Repairing the structure of an ontology
- n Implemented system
- n **Experiments**
- n Conclusion & Future Work

DataSet

- n Anatomy track in OAEI 2008
 - ¤ Ontology MA : (2744 concepts)
 - ¤ Ontology NCI-A : (3304 concepts)
- n 121 missing is-a relations in MA
- n 83 missing is-a relations in NCI-A

Generating repairing actions

- The basic algorithm
 - ☒ For most of the missing is-a relations these sets are small and thus can be easily visualized in the panels of our system.
 - ☒ There are a number of missing is-a relations only having one repairing action which could be immediately repaired.
 - For MA the number is 15
 - For NCI-A the number is 8

	total	1	2-10	11-20	21-30	31-40	41-50	51-100	101-200	201-300	301-400	>400
MA - Source	121	76	45	0	0	0	0	0	0	0	0	0
MA - Target	121	17	50	5	9	4	6	5	18	3	0	4
NCI-A - Source	83	28	55	0	0	0	0	0	0	0	0	0
NCI-A - Target	83	11	52	6	2	0	0	5	4	1	2	0

Fig. 11. Sizes of Source and Target sets.

Generating repairing actions

- The extended algorithm
 - ☒ There are influences for most missing is-a relations
 - For MA the number is 92
 - For NCI-A the number is 67
 - ☒ We found some clusters of missing is-a relations where the influence only exists within the cluster.

	total	1	2	3	4	5	6	7	8	9	10	11-15	16-35	ST
MA	92	23	5	3	0	25	9	9	0	4	0	13	0	1
NCI-A	67	15	21	3	1	2	0	0	0	0	0	6	6	13

Fig. 12. Influence between repairing actions of different missing is-a relations - in Source or Target. Column (ST) shows the number of missing is-a relations where x's and y's of other missing is-a relations occur in both Source and Target sets

Recommending repairing actions

- n We use WordNet as domain knowledge.
- n The running time for generating recommendations for all missing is-a relations was
 - ☒ Circa 40 minutes for MA
 - ☒ Circa 1 hour for NCI-A
- NOTE: In our tool, we generate recommendations on demand for a particular missing is-a relation, rather than for all at once, and this goes quite fast.
- n Concerning the number of recommendations
 - ☒ MA: 22 receive 1; 12 receive 2; 2 receive 3.
 - ☒ NCI-A: 5 receive 1.

Executing repairing actions

- n We use the tool to repair MA and NCI-A with the basic algorithm, using [2] as our reference.
- n Concerning the time
 - ☒ About 90 minutes for MA
 - ☒ About 40 minutes for NCI-A
- n In some cases, the target set is too large to have a good visualization
 - ☒ For the last 28 out of 121 missing is-a relations in MA
 - ☒ For the last 13 out of 83 missing is-a relations in NCI-A

[2] F Feneis and W Dauber. *Pocket Atlas of Human Anatomy, 4th ed.* Thieme Verlag, 2000.

Outline

- n Background
- n Theory of our approach
- n Repairing the structure of an ontology
- n Implemented system
- n Experiments
- n **Conclusion & Future Work**

Conclusion

- n We introduced algorithms and a tool for repairing missing is-a relations in an ontology.
- n We defined the notion of structural repairs and developed algorithms for generating, recommending and executing repairing actions.
- n We discussed an experiment for repairing the two ontologies of the Anatomy track of OAEI.

Future Work

- n Extend the current work to the repairing of both ontology structure and ontology mappings.
- n Investigate the possible influence between semantic defects and modeling defects.
- n Investigate ways to partition the set of missing is-a relations into parts that can be processed independently.
- n Explore new ways to visualize Source and Target sets.
- n Explore ways to detect missing is-a relations.

Thank you!

Questions & Answers

