TechReport: A graph-based method for interactive mapping revision

Weizhuo Li[1,2], Songmao Zhang[1,✉], Guilin Qi[3], Xuefeng Fu[4], and Qiu Ji[5]

[1] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China.
liweizhuo@amss.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China.
smzhang@math.ac.cn
[3] School of Computer Science and Engineering, Southeast University, Nanjing, China.
gqi@seu.edu.cn
[4] School of Information Engineering, Nanchang Institute of Technology, NanChang, China.
fxf@nit.edu.cn
[5] School of Modern Posts and Institute of Modern Posts, Nanjing University of Posts and Telecommunications, Nanjing, China.
qiuji@njupt.edu.cn

Abstract. SAMR a graph-based method for interactive mapping revision to reduce the manual efforts, in which ontologies are encoded into an integrated graph, and mappings are transformed into the mapping arcs are waiting for expert decision. A notion decision space is defined based on graph to detect and propagate implications of expert's decisions. After a decision is made by the expert in each interaction, those mappings entailed by the already confirmed mappings are automatically approved. Relatively, those mappings that would entail rejected mappings or lead to the graph incoherent are declined. Moreover, we define an impact function based on the constructed graph to select suitable mappings that will be displayed to the expert. In this way, the effort of manual evaluation could be reduced further. In this paper, we briefly introduce SAMR and present its results of two tracks (i.e.,Conference, Anatomy) on OAEI.

1 Presentation of the system

1.1 State, purpose, general statement

Up to now, the mainstream methods depend on eliminating inconsistencies among ontologies and mappings to improve the quality of alignments. Such mappings are referred to as incoherent as they yield unsatisfiable concepts and roles in source ontologies [1]. In these works, mappings are interpreted as sets of axioms in description logics (DL), their variants, or restricted logic programmings, and logical reasoners are applied to detect the inconsistency (see Alcomo [1], LogMap [2] and AMLR [3] for example). At the repair stage, some local or global strategies are adopted to remove some mappings so as to regain the consistency. Alternatively, there exist works that consider the uncertainty of mappings and model mapping revision as an optimization problem where probabilistic reasoning techniques are applied to solve the conflicts, as exemplified by ContraBovemRufum [4] and ELog [5].

Although can be effective in maintaining the consistency of mappings with source ontologies, these revision methods suffer from two limitations. Firstly, logical unsatisfiability is caused by erroneous mappings across ontologies, but not all wrong mappings will lead to unsatisfiability. Such mappings are often reserved by the revision systems in the final alignments. Generally, the detection of incoherent mappings depends on the presence of disjointness axioms. When there are fewer or none such axioms specified in source ontologies, the effectiveness of mapping revision will be affected. Secondly, after the incoherence detection that often identifies hundreds or thousands of minimal incoherence preserving subalignments (MIPSs) [1] for large and complex domain ontologies, tracking down the true negatives in MIPSs remains a challenge. To address it, heuristic principles are instantiated including consistency and conservativity principle, minimal change principle, and many others [1, 6, 7], whereas the accuracy and completeness are not guaranteed. This calls for manual revisions with domain expertise.

We propose a graph-based method for interactive mapping revision, call SAMR (Semi-Automatic Mapping Repair), which is able to reduce the number of human decisions. Firstly, source ontologies are encoded into an integrated graph, where its mapping arcs are obtained by transforming mappings to be evaluated by an expert. We specify the decision space for mapping revision and the corresponding operations that can be applied in graph. After a decision is made by the expert in each interaction, the mapping arcs in the integrated graph will be updated automatically. For every mapping confirmed manually, we identify mappings that can be entailed as correct ones whereas mappings in conflict with the evaluated mapping are discarded. On the other hand, for every mapping declined manually, we identify mappings that can entail it and recognize them as incorrect ones. The whole update process modeled in decision space can be finished in polynomial time. Moreover, we define an impact function based on the integrated graph so as to show the most influential mapping to human in each interaction.

1.2 Specific techniques used

Constructing graphs to represent ontology and their mappings According to the works in [8–10], DL-Lite ontologies and their mappings can be encoded into a directed graph without any loss of information. Concretely, we first construct two subgraphs $\mathcal{G}_{\mathcal{T}_i} = (N_i, E_i)$ and $\mathcal{G}_{\mathcal{T}_j} = (N_j, E_j)$ for ontologies O_i and O_j, respectively, in which $N_k(k \in \{i, j\})$ represents the concepts and roles of ontologies, and $E_k(k \in \{i, j\})$ represents the inclusion relationships among them. Then, we transform their mappings $\mathcal{M}$ into mapping arcs for connecting two subgraphs. The constructed graph about ontologies O_i, O_j and $\mathcal{M}$, called the integrated graph, is denoted as $\mathcal{G} = (N, E \cup E_{\mathcal{M}})$. Here, $N = N_i \cup N_j$ and $E = E_i \cup E_j$ are the union of the nodes and arcs in subgraphs $\mathcal{G}_{\mathcal{T}_i}$ and $\mathcal{G}_{\mathcal{T}_j}$. $E_{\mathcal{M}}$ represents a set of mapping arcs, that are bridges linking those nodes across $\mathcal{G}_{\mathcal{T}_i}$ and $\mathcal{G}_{\mathcal{T}_j}$.

In a directed graph, the transitive closure of a graph $\mathcal{G} = (N, E)$ is a graph (N, E^*) such that there is an arc $\langle s, t\rangle$ in E^* iff there exists a directed path from node s to t in $\mathcal{G}$.

Definition 1 *[10] (Path-Unsatisfiability). Let $\mathcal{G} = (N, E \cup E_{\mathcal{M}})$ be the integrated graph constructed from two DL-Lite ontologies O_i and O_j and their mappings $\mathcal{M}$. A* *node $S \in N$ is path-unsatisfiable if there exist two paths in $\mathcal{G}$ starting from S and ending with node S' and node $\neg S'$, respectively. Graph $\mathcal{G}$ is incoherent iff there exists at least one path-unsatisfiable node in $\mathcal{G}$.*

Example 1 *Ontologies O_1 and O_2 describe the domain of the conference management systems, whose axioms are listed as follows:*

$$\mathit{Meta\text{-}Review}_1 \sqsubseteq \mathit{Review}_1 \qquad \exists \mathit{hasName}_1^- \sqsubseteq \mathit{Conference}_1$$
$$\mathit{Conference}_1 \sqsubseteq \neg \mathit{Regular\_Author}_1$$
$$\mathit{Author\_of\_paper}_2 \sqsubseteq \mathit{Author}_2 \qquad \exists \mathit{hasname}_2^- \sqsubseteq \mathit{Author}_2$$

Their alignment $\mathcal{M}$ consists of the following mappings:

$$m_1 = (\mathit{Regular\_Author}_1, \mathit{Author}_2, \equiv, 0.8)$$
$$m_2 = (\mathit{Regular\_Author}_1, \mathit{Author\_of\_paper}_2, \sqsupseteq, 0.7)$$
$$m_3 = (\exists \mathit{hasName}_1^-, \exists \mathit{hasname}_2^-, \equiv, 0.8)$$
$$m_4 = (\mathit{Review}_1, \mathit{Reviewing\_event}_2, \equiv, 0.6)$$
$$m_5 = (\mathit{Meta\text{-}Review}_1, \mathit{Reviewing\_event}_2, \sqsupseteq, 0.7)$$

Fig 1 shows an integrated graph constructed from the ontologies and their mappings in Example 1. One can see that the nodes $\exists \mathrm{hasName}_1^-$ and $\exists \mathrm{hasname}_2^-$ are path-unsatisfiable w.r.t $O_1 \cup O_2 \cup \mathcal{M}$ because there exist two paths starting from them to node $\mathrm{Regular\_Author}_1$ and node $\neg\mathrm{Regular\_Author}_1$, respectively.

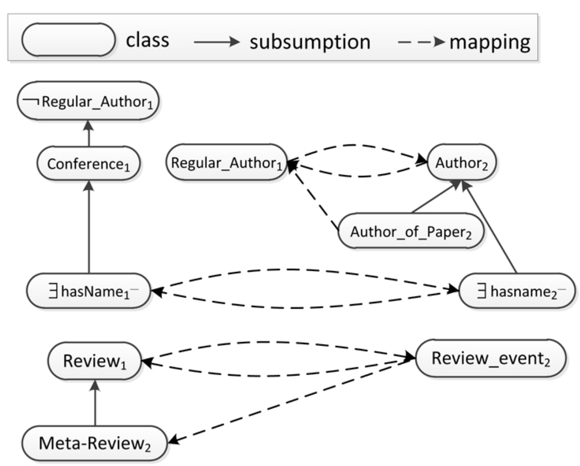

Fig. 1: The integrated graph constructed from ontologies and mappings

Graph-based Revision State and Graph-based Decision Space In parallel with the revision state and decision space proposed in [11] for interactive ontology revision, we define counterparts as follows for interactive mapping revision based on graph. For convenience, arcs α, β and γ represent three different arcs in the constructed graph.

Definition 2 *(Graph-based Revision State). A graph-based revision state is defined as a tuple* $(E \cup E_{\mathcal{M}}, E^{\models}, E^{\not\models})$ *of arcs in the graph* $\mathcal{G} = (N, E \cup E_{\mathcal{M}})$ *with* $E^{\models} \subseteq E \cup E_{\mathcal{M}}$*,* $E^{\not\models} \subseteq E_{\mathcal{M}}$ *and* $E^{\models} \cap E^{\not\models} = \emptyset$*. A graph-based revision state is complete, if* $E \cup E_{\mathcal{M}} = E^{\models} \cup E^{\not\models}$*. The closure of a graph-based revision state is denoted by clos(*$(E \cup E_{\mathcal{M}}, E^{\models}, E^{\not\models})$*)=* $(E \cup E_{\mathcal{M}}, E_c^{\models}, E_c^{\not\models})$ *with* $E_c^{\models} = \{\alpha \in E \cup E_{\mathcal{M}} | \alpha \in (E^{\models})^*\}$ *and* $E_c^{\not\models} = \{\alpha \in E_{\mathcal{M}} | \beta \in (E^{\models} \cup \{\alpha\})^*$ *and* $\beta \in E^{\not\models}\}$*.*

Definition 3 *(Graph-based Decision Space). Given a graph-based revision state* $(E \cup E_{\mathcal{M}}, E^{\models}, E^{\not\models})$ *with* $E^{\not\models} \neq \emptyset$*, the graph-based decision space* $\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\models}, E^{\not\models})} = (E_{\mathcal{M}}^?, \mathcal{E}, \mathcal{C})$ *contains the set* $E_{\mathcal{M}}^? = (E \cup E_{\mathcal{M}}) \setminus (E_c^{\models} \cup E_c^{\not\models})$ *of unevaluated mapping arcs and two binary relations,* $\mathcal{E}$*(entails) and* $\mathcal{C}$*(conflicts) defined by* $\alpha \mathcal{E} \beta$ *iff* $\beta \in (E^{\models} \cup \{\alpha\})^*$ *and* $\alpha \mathcal{C} \beta$ *iff* $\gamma \in (E^{\models} \cup \{\alpha, \beta\})^*$ *for some* $\gamma \in E^{\not\models}$*.*

Considering the requirement that $E^{\not\models} \neq \emptyset$, we can add a special kind of arcs $\{\langle C_k, \neg C_k \rangle\}$ and $\{\langle R_k, \neg R_k \rangle\}$ ($k \in \{i, j\}$) to express inconsistency, where C_k and R_k represent concept and role in DL-Lite ontologies. The operations about $\mathcal{E}$ and $\mathcal{C}$ in $(E_{\mathcal{M}}^?, \mathcal{E}, \mathcal{C})$ w.r.t mappings can be reduced to the reachability of two nodes as follows.

1. Given two arcs α, β and $\beta = \langle S_i, S_j \rangle$, $\alpha \mathcal{E} \beta$ iff $E^{\models} \cup \{\alpha\}$ contains at least one path from node S_i to S_j.
2. Given two arcs α, β, $\alpha \mathcal{C} \beta$ iff $E^{\models} \cup \{\alpha, \beta\}$ contains at least one path from node S_i to S_j, where $\langle S_i, S_j \rangle$ is an arc that belongs to $E^{\not\models}$.

For convenience, we denote $\uparrow \alpha$=$\{\beta | \alpha \mathcal{E} \beta\}$, $\downarrow \alpha$=$\{\beta | \beta \mathcal{E} \alpha\}$ and $\wr\alpha$=$\{\beta | \alpha \mathcal{C} \beta\}$. Then, we can obtain following properties of the integrated graph.

Impact Function

Definition 4 *Let* $(E \cup E_{\mathcal{M}}, E^{\models}, E^{\not\models})$ *be a graph-based revision state and* $\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\models}, E^{\not\models})} = (E_{\mathcal{M}}^?, \mathcal{E}, \mathcal{C})$ *be the graph-based decision space. For one mapping arc* α*, we define its approval impact, impact*$^+(\alpha)$*, its decline impact, impact*$^-(\alpha)$ *and its impact, impact*(α) *as follow.*

1. *impact*$^+(\alpha)$*=map*$(\uparrow \alpha)$*+map*$(\wr\alpha)$
2. *impact*$^-(\alpha)$*=map*$(\downarrow \alpha)$
3. *impact*(α)*=max(impact*$^+(\alpha)$*, impact*$^-(\alpha)$*)*

where map(·) return the number of the entailed arcs belonging to $E_{\mathcal{M}}$*.*

We define the impact function to measure the influence of a mapping in terms of how many mappings are in entailment or conflicting relationships with this mapping so as to further reduce the number of expert's decisions. In each interaction, the most influential mapping is shown to the expert for his/her decision. Note that if a mapping specifies an equivalence correspondence between two concepts across ontologies, we represent it by two mapping arcs and add up their impacts.

1.3 Graph-based Framework for Interactive Mapping Validation

Algorithm **??** presents the graph-based framework for interactive mapping revision. Concretely, we firstly employ the construction rules designed in [10] to encode ontologies into an integrated graph and obtain the mapping arcs from mappings. Then, we initiate the revision state and decision space in Steps 4–5. Steps 6–13 are a concrete realization of an interactive mapping revision: in each interaction, a mapping arc m_0 with the largest impact is selected for an expert to make a decision. If an expert confirms m_0, the decision space is updated based on the confirmed m_0 by Algorithm 2; otherwise, the decision space is updated based on the declined m_0 by Algorithm 3. When the correctness of all mappings is decided, either manually or by reasoning. Finally, Algorithm 4 transforms confirmed mapping arcs into the format of mappings.

Note that there may be multiple mapping arcs with the largest impact value. We can refine the selection according to the weights of mappings, i.e., if the impact of mapping arcs comes from approval impacts, the one with the highest weight is selected; otherwise the lowest weight.

Algorithm 1: Graph-based Framework for Interactive Mapping Revision.

Input: Two DL-Lite ontologies O_i and O_j, and mappings $\mathcal{M}$.
Output: The repaired mapping $\mathcal{M}'$.
1 Construct two sub-graphs $\mathcal{G}_i = (N_i, E_i)$ and $\mathcal{G}_j = (N_j, E_j)$ of ontologies O_i and O_j according to rules in [10];
2 Transform mappings $\mathcal{M}$ into mapping arcs $E_{\mathcal{M}}$ according to rules in [10];
3 Initial an integrated graph $\mathcal{G} = (N, E)$, where $N = N_i \cup N_j$ and $E = E_i \cup E_j$;
4 $E^{\vDash} \leftarrow E$, $E^{\nvDash} \leftarrow \{\langle B_i, \neg B_i \rangle\} \cup \{\langle P_i, \neg P_i \rangle\}$, $E^{?}_{\mathcal{M}} \leftarrow E_{\mathcal{M}}$;
5 $\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\vDash}, E^{\nvDash})} = (E^{?}_{\mathcal{M}}, \mathcal{E}, \mathcal{C})$;
6 **while** $E^{?}_{\mathcal{M}} \neq \emptyset$ **do**
7   **for** *each mapping arc* $m \in E^{?}_{\mathcal{M}}$ **do**
8   Calculate the impact(m);
9     $m_0 \longleftarrow$ Select one mapping arc with the largest impact;
10   **if** *expert confirms* m_0 **then**
11         $\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\vDash}, E^{\nvDash})} \leftarrow$ UDSC($\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\vDash}, E^{\nvDash})}, m_0, E^{?}_{\mathcal{M}}$);
12   **else**
13         $\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\vDash}, E^{\nvDash})} \leftarrow$ UDSD($\mathbb{D}_{(E \cup E_{\mathcal{M}}, E^{\vDash}, E^{\nvDash})}, m_0, E^{?}_{\mathcal{M}}$);
14 $\mathcal{M}' \longleftarrow$ TransformSoucreMappings($E^{\vDash} \cap E_{\mathcal{M}}$);
15 **return** $\mathcal{M}'$;

Example 2 *(Example 1 cont'd). Table 1 summarizes the intermediate results for Example 1 in each interaction of Algorithm 1. The second column lists the mappings with their impacts and shows the selected one in bold. The third column shows the expert's decision of the selected mapping, and the last two columns list the entailed mappings and left mappings in each loop. For five mappings in Example 1, two are evaluated by*

Algorithm 2: UDS-C($\mathbb{D}_{(E\cup E_{\mathcal{M}},E^{\vDash},E^{\nvDash})}, \alpha, E^{?}_{\mathcal{M}},$).

Input: The decision space $\mathbb{D}_{(E\cup E_{\mathcal{M}},E^{\vDash},E^{\nvDash})}$, confirmed mapping arc α.
Output: The updated decision space $\mathbb{D}_{(E\cup E_{\mathcal{M}},E^{\vDash}\cup\uparrow\alpha,E^{\nvDash}\cup\wr\alpha)} = (E^{?}_{\mathcal{M}}, \mathcal{E}, \mathcal{C})$.

```
E^⊨ ← E^⊨ ∪ {α};
for each mapping arc β ∈ (E ∪ E_M) \ (E^⊨ ∪ E^⊭) do
    if β ∈ (E^⊨ ∪ {α})* then
        E^⊨ ← E^⊨ ∪ {β};
    for each mapping arc γ ∈ E^⊭ do
        if γ ∈ (E^⊨ ∪ {α, β})* then
            E^⊭ ← E^⊭ ∪ {β};
E^?_M ← E_M \ (E^⊨ ∪ E^⊭);
return D_(E∪E_M, E^⊨∪↑α, E^⊭∪≀α) = (E^?_M, ℰ, 𝒞);
```

Algorithm 3: UDS-D($\mathbb{D}_{(E\cup E_{\mathcal{M}},E^{\vDash},E^{\nvDash})}, \alpha, E^{?}_{\mathcal{M}}$).

Input: The decision space $\mathbb{D}_{(E\cup E_{\mathcal{M}},E^{\vDash},E^{\nvDash})}$, declined mapping arc α.
Output: The updated decision space $\mathbb{D}_{(E\cup E_{\mathcal{M}},E^{\vDash},E^{\nvDash}\cup\downarrow\alpha)} = (E^{?}_{\mathcal{M}}, \mathcal{E}, \mathcal{C})$.

```
E^⊭ ← E^⊭ ∪ {α};
for each mapping arc β ∈ (E ∪ E_M) \ (E^⊨ ∪ E^⊭) do
    if β ∈ (E^⊨ ∪ {α})* then
        E^⊭ ← E^⊭ ∪ {β};
E^?_M ← E_M \ (E^⊨ ∪ E^⊭);
return D_(E∪E_M, E^⊨, E^⊭∪↓α) = (E^?_M, MIPPs^?, ℰ, 𝒞);
```

Algorithm 4: TransformSoucreMappings.

Input: A set of mapping arcs $Arcs$ in $\mathcal{G}$;
Output: A set of inclusions Axioms;

```
Axioms ⟵ ∅;
for each mapping arc ⟨B1, B2⟩ ∈ Arcs do
    if B1 is in the form of ∃R1 and B2 the form of ∃R2 then
        Axioms ⟵ Axioms ∪{(R1, R2, ⊑)};
    else
        Axioms ⟵ Axioms ∪{(B1, B2, ⊑)};
return Axioms;
```

human whereas the others are automatically decided by our algorithms, and the final correct mappings are listed as follows:

$$m_1 = (\textit{Regular\_Author}_1, \textit{Author}_2, \equiv, 0.8)$$
$$m_2 = (\textit{Regular\_Author}_1, \textit{Author\_of\_paper}_2, \sqsupseteq, 0.7)$$

Table 1: The intermediate results for Example 1 in each interaction of Algorithm 1

Iteration	Selected mapping in $E^?_{\mathcal{M}}$	Expert decision	Entailed mappings	Unlabeled mappings
1	**impact(m_1)=3** impact(m_2)=2 impact(m_3)=2 impact(m_4)=2 impact(m_5)=2	Confirm	m_2, m_3	m_4, m_5
2	**impact(m_4)=2** impact(m_5)=2	Decline	m_5	None

1.4 Link to the system and parameters file

The latest version of SAMR including the data sets and results can be seen on https://github.com/liweizhuo001/SAMR. In addition, we also provide a web version for user-interactive repairing with a linkage in https://github.com/liweizhuo001/SAOR.

2 Experimental evaluation

In this section, we present the experimental results of our method. Our algorithms were implemented with the OWLAPI[6], a tool for managing OWL ontologies. We compared our method with the method proposed by Christian et al. proposed in [12]. To our knowledge, it is the only interactive mapping revision work that can save human decisions. Concretely, we have implemented a total of four systems for comparison, as shown in Table 2.

1. **1:1-Repair** manually repairs mappings with 1:1 constraint where no automated reasoning is applied. When an expert accepts a mapping, all the mappings sharing concepts with this mapping will be treated as declined ones [1]. **1:1-Repair** is used as the baseline in our experiment.
2. **DDL-Repair** employs distributed description logics (DDL) to detect and propagate implications of expert decisions on the correctness of mappings, as proposed by Christian et al. in [12].

[6] http://owlapi.sourceforge.net/

3. **Graph-Repair1** implements our algorithms with impact function defined in Definition 5 proposed by Christian et al. in [12].
4. **Graph-Repair2** implements our algorithms with impact function defined in Definition 4.

Table 2: A summary of the repair systems compared in the evaluation

System	1:1 constraint	Reasoning on confirmed mapping	Reasoning on declined mapping	Impact function in Definition 4	Impact function in Definition 5
1:1-Repair	✓				
DDL-Repair		✓			✓
Graph-Repair1		✓	✓		✓
Graph-Repair2		✓	✓	✓	

Definition 5 *[12](Potential impact of a bridge rule). The potential impact of a bridge rule from TBox $\mathcal{T}_1$ to TBox $\mathcal{T}_2$ denoted as imp($\mathcal{T}_1$, $\mathcal{T}_2$, $1 : C \xrightarrow{R} 2 : D$) is defined as*

$$
\begin{cases}
sub(\mathcal{T}_1, C) \cdot (super(\mathcal{T}_2, D) + dis(\mathcal{T}_2, D)) & \text{if } R = \sqsubseteq \\
super(\mathcal{T}_1, C) \cdot (sub(\mathcal{T}_2, D) + dis(\mathcal{T}_2, D)) & \text{if } R = \sqsupseteq \\
imp(\mathcal{T}_1, \mathcal{T}_1, 2 : C \xrightarrow{\sqsubseteq} 2 : D) + imp(\mathcal{T}_1, \mathcal{T}_2, 1 : C \xrightarrow{\sqsupseteq} 2 : D) & \text{if } R = \equiv
\end{cases}
$$

where $sub(\mathcal{T}, C)$ returns the number of all subclasses of concept C in $\mathcal{T}$, $super(\mathcal{T}, C)$ returns the number of all superclasses of concept C in $\mathcal{T}$, and $dis(\mathcal{T}, C)$ returns the number of all classes that are disjoint with C.

All the experiments were performed on a desktop computer with Intel® Core^{TM} i7-2600 (3.4GHz) and 32GB RAM in Java 1.8.

2.1 Data set and evaluation criteria

For repair tasks, we selected Conference Track, Anatomy Track from OAEI[7] (Ontology Alignment Evaluation Initiative) in our experiments. The alignments are automatically generated between all pairs of conference ontologies by applying the matching system HMatch [13] and ASMOV [14]. In contrast to the majority of existing systems limited to the discovery of " $\equiv$ " mapping precisely, HMatch can generate lots of 1:n mappings and ASMOV is additionally capable of finding " $\sqsubseteq$ " and " $\sqsupseteq$ " relations with few inconsistency cases. It is suitable for us to compare repair systems and analyze their inherent properties.

In order to measure the repair performance, we statistic the number of confirm and decline of the expert, which are represented by "Appd" and "Rej" in the following tables. The efficiency of manual revision (denoted as $Saved$) can be revealed by the

[7] http://oaei.ontologymatching.org/

Table 3: The automatic evaluation result of alignments generated by HMatch

Repair Task	$\|\mathcal{M}\|$	MIPS	1:1-Repair			DDL-Repair			Graph-Repair1			Graph-Repair2		
			Appr/Rej	Corr	Saved	Appr/Rej	Corr	Saved	Appr/Rej	Corr	Saved	Appr/Rej	Corr	Saved
cmt-Conference	24	16	8/14	8	8.3%	8/14	8	8.3%	8/12	8	16.7%	**8/9**	**8**	**29.2**%
cmt-confOf	13	10	5/8	5	0.0%	4/6	4	23.1%	**4/5**	**4**	**30.8**%	4/6	4	23.1%
cmt-edas	22	20	9/10	9	13.6%	8/4	8	45.5%	**7/4**	**7**	**50**%	7/5	7	45.5%
cmt-ekaw	29	28	7/19	7	10.3%	**6/14**	**6**	**31.0**%	**6/14**	**6**	**31.0**%	**6/14**	**6**	**31.0**%
cmt-iasted	9	2	4/5	4	0.0%	4/5	4	0.0%	4/5	4	0.0%	4/5	4	0.0%
cmt-sigkdd	20	16	9/9	9	10.0%	9/8	9	15%	9/8	9	15%	**9/5**	**9**	**30.0**%
Conference-confOf	25	18	9/14	9	8.0%	**8/11**	**8**	**26.9**%	**8/11**	**8**	**26.9**%	**8/11**	**8**	**26.9**%
Conference-edas	38	32	8/27	8	7.9%	7/19	7	31.6%	**6/19**	**6**	**34.2**%	7/21	7	26.3%
Conference-ekaw	43	26	18/19	18	14.0%	18/20	18	11.6%	**17/17**	**17**	**20.9**%	16/20	16	16.3%
Conference-iasted	13	0	5/8	5	0.0%	5/8	5	0.0%	5/8	5	0.0%	5/8	5	0.0%
Conference-sigkdd	27	4	9/18	9	0.0%	9/17	8	3.7%	**9/16**	**8**	**7.4**%	9/17	8	3.7%
confOf-edas	29	19	12/15	12	6.9%	**10/11**	**10**	**27.6**%	**10/11**	**10**	**27.6**%	10/12	10	24.1%
confOf-ekaw	26	13	14/12	14	0.0%	13/10	13	11.5%	13/10	13	11.5%	**13/9**	**13**	**15.4**%
confOf-iasted	8	0	4/4	4	0.0%	4/4	4	0.0%	4/4	4	0.0%	4/4	4	0.0%
confOf-sigkdd	11	8	4/7	4	0.0%	**4/4**	**4**	**27.3**%	**4/4**	**4**	**27.3**%	**4/4**	**4**	**27.3**%
edas-ekaw	49	50	16/31	16	4.1%	13/22	13	26.8%	11/21	11	34.7%	**11/16**	**11**	**44.9**%
edas-iasted	16	7	**7/9**	**7**	**0.0**%	**7/8**	**7**	**6.3**%	**7/8**	**7**	**6.3**%	7/9	7	0.0%
edas-sigkdd	28	19	7/16	7	17.9%	7/13	7	28.6%	**6/13**	**6**	**32.1**%	7/13	7	28.6%
ekaw-iasted	7	0	6/1	6	0.0%	6/1	6	0.0%	6/1	6	0.0%	6/1	6	0.0%
ekaw-sigkdd	21	16	7/11	7	14.3%	**7/7**	**7**	**33.3**%	**7/7**	**7**	**33.3**%	7/8	7	28.6%
iasted-sigkdd	34	6	14/19	14	2.9%	**14/17**	**14**	**8.8**%	**14/17**	**14**	**8.8**%	14/18	14	5.9%
human-mouse	2752	–	–	–	–	–	–	–	–	–	–	–	–	–

fraction of mappings without manual evaluation. Moreover, we utilize the number of correct mappings (denoted as $Corr$) to evaluate the quality of repair. Given a alignment $\mathcal{M}$, a reference alignment $\mathcal{R}$ and the number of confirm "Appd" and decline "Rej" of the expert, the formulas about $Saved$ and $Corr$ are defined as follows.

$$Saved = 1 - \frac{|Appd| + |Rej|}{|\mathcal{M}|} \qquad Corr = |Appd \cap \mathcal{R}|$$

2.2 Evaluation on Automatic Repairing

For automate repairing, we utilize the reference alignment to act as an expert. Concretely, if the selected mapping belong to the reference alignment, repair systems make a confirm decision. Otherwise, systems executes the reject operation.

Table 3 lists the evaluation result of twenty-one alignments across ontologies generated by HMatch. Overall, all the reasoning-based revision methods can significantly save the decisions. In terms of the efficiency of manual revision, Graph-Repair1 is better than DDL-Repair with the same impact function shown in Definition 5, which achieves the best results in twelve out of twenty-one repair tasks, but DDL-Repair only achieves the six out of twenty-one. The main reason is that our graph-based method models the entailment process when expert declined mappings, which may be suitable for alignments generated by 1:n matchers. Nevertheless, it may lead to some correct mappings removed indirectly such as cmt-edas alignment repaired by the graph methods. Note that Graph-Repair2 equipped with our defined function are not well, but it also achieve the six out of twenty-one as the same as DDL-Repair. Both of these two impact functions have their own advantages. It makes sense to combine them for selecting mappings, we leave this issue for future work.

Table 4: The automatic evaluation result of alignments generated by ASMOV

Repair Task	$\|\mathcal{M}\|$	MIPS	1:1-Repair			DDL-Repair			Graph-Repair1			Graph-Repair2		
			Appr/Rej	Corr	Saved	Appr/Rej	Corr	Saved	Appr/Rej	Corr	Saved	Appr/Rej	Corr	Saved
cmt-Conference	16	0	7/9	7	0.0%	7/9	7	0.0%	7/9	7	0.0%	7/9	7	0.0%
cmt-confOf	9	0	4/5	4	0.0%	4/5	4	0.0%	4/5	4	0.0%	4/5	4	0.0%
cmt-edas	15	6	8/7	8	0.0%	7/6	7	13.3%	7/6	7	13.3%	**5/6**	**5**	**26.7%**
cmt-ekaw	15	0	6/9	6	0.0%	6/9	6	0.0%	6/9	6	0.0%	6/9	6	0.0%
cmt-iasted	15	0	3/12	3	0.0%	3/12	3	0.0%	3/12	3	0.0%	3/12	3	0.0%
cmt-sigkdd	17	0	7/10	7	0.0%	7/10	7	0.0%	7/10	7	0.0%	7/10	7	0.0%
Conference-confOf	24	0	8/16	8	0.0%	8/16	8	0.0%	8/16	8	0.0%	8/16	8	0.0%
Conference-edas	25	6	6/19	6	0.0%	**9/18**	**6**	**4.0%**	**9/18**	**6**	**4.0%**	**9/18**	**6**	**4.0%**
Conference-ekaw	27	2	12/15	11	0.0%	12/15	11	0.0%	12/15	11	0.0%	**12/14**	**12**	**3.7%**
Conference-iasted	29	0	5/24	5	0.0%	5/24	5	0.0%	5/24	5	0.0%	5/24	5	0.0%
Conference-sigkdd	30	0	10/20	10	0.0%	10/20	10	0.0%	10/20	10	0.0%	10/20	10	0.0%
confOf-edas	21	2	10/11	9	0.0%	10/11	9	0.0%	10/11	9	0.0%	10/11	9	0.0%
confOf-ekaw	23	7	15/18	15	0.0%	14/8	14	4.3%	14/8	14	4.3%	**14/6**	**14**	**13.0%**
confOf-iasted	32	3	7/25	7	0.0%	7/25	7	0.0%	7/25	7	0.0%	7/25	7	0.0%
confOf-sigkdd	14	5	4/10	4	0.0%	**4/8**	**4**	**14.3%**	**4/8**	**4**	**14.3%**	**4/8**	**4**	**14.3%**
edas-ekaw	36	9	12/24	12	0.0%	10/24	10	5.6%	10/24	10	5.6%	**9/23**	**9**	**11.1%**
edas-iasted	39	2	11/28	11	0.0%	**11/28**	**10**	**2.6%**	**11/28**	**10**	**2.6%**	**11/28**	**10**	**2.6%**
edas-sigkdd	24	0	9/15	9	0.0%	9/15	9	0.0%	9/15	9	0.0%	9/15	9	0.0%
ekaw-iasted	45	0	8/37	8	0.0%	8/37	8	0.0%	8/37	8	0.0%	8/37	8	0.0%
ekaw-sigkdd	28	2	7/21	7	0.0%	7/21	7	0.0%	7/21	7	0.0%	**7/20**	**7**	**3.6%**
iasted-sigkdd	35	0	13/22	13	0.0%	13/22	13	0.0%	13/22	13	0.0%	13/22	13	0.0%
human-mouse	1206	49	**1041/161**	**1041**	**0.3%**	1042/163	1042	0.1%	1042/163	1042	0.1%	1042/163	1042	0.1%

Table 4 lists the evaluation result of the alignments generated by ASMOV. Compared the alignments generated by HMatch, there exist few MIPSs [1] that cause concepts or roles unsatisfiable because ASMOV has corrected the alignment by anti-patterns before its final delivery [14], which weaken the efficiency of reasoning-based revision methods. Of note, reason-based methods may not be effective for all alignments including incoherent mappings because all the mappings in MIPSs could be wrong such as confOf-iasted alignment. Benefited from our defined impact function, Graph-Repair2 achieve the best results in eight out of twenty-one repair tasks. but DDL-Repair and Graph-Repair1 only achieve the three out of twenty-one. Compared with a static impact function in Definition 5, our impact function is dynamic one that could be adaptive according to the current state of decision space. It is suitable for improving the efficiency of repair results with entailment process of declined mappings.

2.3 Evaluation on User-Interactive Repairing

For user-interactive repairing, we only selected Conference Track for evaluation since mapping among ontologies of Anatomy Track need strong expertise. In order to ensure the fairness of the experiment, we require that the expert must evaluate alignments with 1:1-Repair method firstly, and maintain the same judgments in subsequent repair methods. In order to perform user evaluation on the proposed approach, we also developed a user interface for the implementation of our proposed approach. This yields a user friendly tool for repairing mapping shown in Figure 2.

Table 5 lists the evaluation result of twenty-one alignments across ontologies generated by HMatch. Overall, all the reasoning-based revision methods can keep most correct mappings and significantly save the decisions made by expert. In terms of the efficiency of manual revision, our two graph-based methods are better than DDL-Repair,

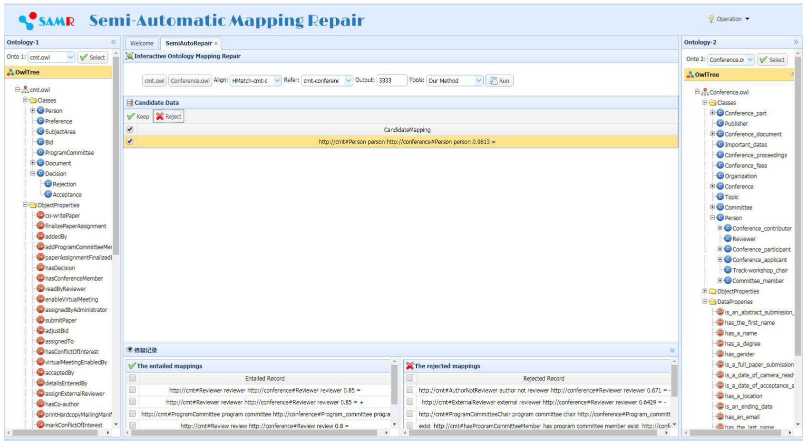

Fig. 2: A snapshot for interactive mapping revision, which implements the proposed approach

which achieve the best results in nine out of twenty-one repair tasks, but DDL-Repair only achieve the five out of twenty-one.

Table 5: The interactive evaluation result of alignments generated by HMatch

Repair Task	$\|\mathcal{M}\|$	MIPS	1:1-Repair			DDL-Repair			Graph-Repair1			Graph-Repair2		
			Appd/Rej	Corr	Saved	Appd/Rej	Corr	Saved	Appd/Rej	Corr	Saved	Appd/Rej	Corr	Saved
cmt-Conference	24	16	10/9	7	20.8%	9/9	6	25.0%	9/9	6	25.0%	**9/6**	**7**	**37.5**%
cmt-confOf	13	10	6/7	5	0.0%	5/4	4	30.8%	**5/3**	**4**	**38.5**%	5/4	4	30.8%
cmt-edas	22	20	9/11	9	9.1%	8/4	8	45.5%	**7/4**	**7**	**50**%	7/5	7	45.5%
cmt-ekaw	29	28	10/19	7	0.0%	**9/11**	**6**	**31.0**%	**9/11**	**6**	**31.0**%	**9/11**	**6**	**31.0**%
cmt-iasted	9	2	5/4	4	0.0%	5/4	4	0.0%	5/4	4	0.0%	5/4	4	0.0%
cmt-sigkdd	20	16	10/8	9	10.0%	10/7	9	15%	10/7	9	15%	**10/4**	**9**	**30.0**%
Conference-confOf	25	18	9/16	8	0.0%	7/13	7	20.0%	7/13	7	20.0%	**7/10**	**6**	**30.0**%
Conference-edas	38	32	10/26	6	5.3%	8/19	6	28.9%	**7/19**	**5**	**31.6**%	**8/18**	**5**	**31.6**%
Conference-ekaw	43	26	20/12	14	25.6%	20/17	14	14.0%	19/14	14	23.3%	**18/13**	**12**	**27.9**%
Conference-iasted	13	0	6/7	5	0.0%	6/7	5	0.0%	6/7	5	0.0%	6/7	5	0.0%
Conference-sigkdd	27	4	**9/8**	**8**	**37.0**%	9/17	8	3.7%	9/16	8	7.4%	9/17	8	3.7%
confOf-edas	29	19	16/13	12	0.0%	**13/8**	**10**	**27.6**%	**13/8**	**10**	**27.6**%	13/9	10	24.1%
confOf-ekaw	26	13	14/11	13	3.8%	12/7	11	26.9%	12/7	11	26.9%	**12/6**	**11**	**30.8**%
confOf-iasted	8	0	4/4	4	0.0%	4/4	4	0.0%	4/4	4	0.0%	4/4	4	0.0%
confOf-sigkdd	11	8	4/7	4	0.0%	**4/4**	**4**	**27.3**%	**4/4**	**4**	**27.3**%	**4/4**	**4**	**27.3**%
edas-ekaw	49	50	16/29	14	8.2%	14/19	12	32.7%	12/17	10	40.8%	**11/16**	**10**	**44.9**%
edas-iasted	16	7	**10/5**	**7**	**6.3**%	**10/5**	**7**	**6.3**%	**10/5**	**7**	**6.3**%	10/7	7	0.0%
edas-sigkdd	28	19	8/12	7	28.6%	8/12	7	28.6%	**7/12**	**6**	**32.1**%	8/12	7	28.6%
ekaw-iasted	7	0	6/1	6	0.0%	6/1	6	0.0%	6/1	6	0.0%	6/1	6	0.0%
ekaw-sigkdd	21	16	7/10	7	19.0%	**7/7**	**7**	**33.3**%	**7/7**	**7**	**33.3**%	7/8	7	28.6%
iasted-sigkdd	34	6	15/18	13	2.9%	**15/16**	**13**	**8.8**%	**15/16**	**13**	**8.8**%	15/17	13	5.9%

Table 6 lists the evaluation result of the alignments generated by ASMOV. Overall, reason-based methods may not be effective for all alignments including incoherent mappings. Benefited from our defined impact function, Graph-Repair2 achieve the best results in eight out of twenty-one repair tasks. but DDL-Repair and Graph-Repair1 only achieve the five out of twenty-one.

Table 6: The interactive evaluation result of alignments generated by ASMOV

Repair Task	$\lvert\mathcal{M}\rvert$	MIPS	1:1-Repair			DDL-Repair			Graph-Repair1			Graph-Repair2		
			Appd/Rej	Corr	Saved	Appd/Rej	Corr	Saved	Appd/Rej	Corr	Saved	Appd/Rej	Corr	Saved
cmt-Conference	16	0	8/8	7	0.0%	8/8	7	0.0%	8/8	7	0.0%	8/8	7	0.0%
cmt-confOf	9	0	5/4	4	0.0%	5/4	4	0.0%	5/4	4	0.0%	5/4	4	0.0%
cmt-edas	15	6	8/7	8	0.0%	7/6	7	13.3%	7/6	7	13.3%	**5/6**	**5**	**26.7**%
cmt-ekaw	15	0	7/8	6	0.0%	7/8	6	0.0%	7/8	6	0.0%	7/8	6	0.0%
cmt-iasted	15	0	4/11	3	0.0%	4/11	3	0.0%	4/11	3	0.0%	4/11	3	0.0%
cmt-sigkdd	17	0	9/8	7	0.0%	9/8	7	0.0%	9/8	7	0.0%	9/8	7	0.0%
Conference-confOf	24	0	9/15	8	0.0%	9/15	8	0.0%	9/15	8	0.0%	9/15	8	0.0%
Conference-edas	25	6	9/16	6	0.0%	**9/14**	**6**	**8.0**%	**9/14**	**6**	**8.0**%	**9/14**	**6**	**8.0**%
Conference-ekaw	27	2	15/12	11	0.0%	**14/12**	**10**	**3.7**%	**14/12**	**10**	**3.7**%	**14/12**	**10**	**3.7**%
Conference-iasted	29	0	6/23	5	0.0%	6/23	5	0.0%	6/23	5	0.0%	6/23	5	0.0%
Conference-sigkdd	30	0	14/16	10	0.0%	14/16	10	0.0%	14/16	10	0.0%	14/16	10	0.0%
confOf-edas	21	2	10/11	9	0.0%	10/11	9	0.0%	10/11	9	0.0%	10/11	9	0.0%
confOf-ekaw	23	7	15/18	14	0.0%	13/8	12	8.7%	13/8	12	8.7%	**13/7**	**13**	**13.0**%
confOf-iasted	32	3	8/24	7	0.0%	8/24	7	0.0%	8/24	7	0.0%	8/24	7	0.0%
confOf-sigkdd	14	5	4/10	4	0.0%	**4/8**	**4**	**14.3**%	**4/8**	**4**	**14.3**%	**4/8**	**4**	**14.3**%
edas-ekaw	36	9	14/22	12	0.0%	12/22	10	5.6%	12/22	10	5.6%	**11/21**	**9**	**11.1**%
edas-iasted	39	2	13/26	10	0.0%	**12/26**	**9**	**2.6**%	**12/26**	**9**	**2.6**%	**12/26**	**9**	**2.6**%
edas-sigkdd	24	0	10/14	9	0.0%	10/14	9	0.0%	10/14	9	0.0%	10/14	9	0.0%
ekaw-iasted	45	0	12/33	8	0.0%	12/33	8	0.0%	12/33	8	0.0%	12/33	8	0.0%
ekaw-sigkdd	28	2	9/19	7	0.0%	9/19	7	0.0%	9/19	7	0.0%	**9/18**	**7**	**3.6**%
iasted-sigkdd	35	0	15/20	12	0.0%	15/20	12	0.0%	15/20	12	0.0%	15/20	12	0.0%

3 Conclusion

In this paper, we have presented SAMR and its results of two tracks (i.e.,Conference and Anatomy) on OAEI. The results show that indicated that SAMR could save lots of decisions made by the expert and outperform the method proposed by Meilicke et al. in most cases.

In the future, we will explore the case that provides more than one mapping for interactive mapping revision. Correspondingly, we need to optimize our impact function for this case so as to achieve better effects of manual evaluation globally. In addition, we would like optimize our method to efficiently repair the mappings among large biomedical ontologies.

References

1. Christian Meilicke. *Alignment incoherence in ontology matching*. PhD thesis, Universitätsbibliothek Mannheim, 2011.
2. Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. Logmap: Logic-based and scalable ontology matching. In *Proceedings of the 10th International Semantic Web Conference*, pages 273–288. Springer, 2011.
3. Emanuel Santos, Daniel Faria, Catia Pesquita, and Francisco M Couto. Ontology alignment repair through modularization and confidence-based heuristics. *PloS one*, 10(12):1–19, 2015.
4. Silvana Castano, Alfio Ferrara, Davide Lorusso, Tobias Henrik Näth, and Ralf Möller. Mapping validation by probabilistic reasoning. In *Proceedings of the 5th European Semantic Web Conference*, pages 170–184. Springer, 2008.
5. Jan Noessner, Heiner Stuckenschmidt, Christian Meilicke, and Mathias Niepert. Completeness and optimality in ontology alignment debugging. In *Proceedings of the 9th International Conference on Ontology Matching*, pages 25–36. CEUR-WS. org, 2014.

6. Guilin Qi, Qiu Ji, and Peter Haase. A conflict-based operator for mapping revision. In *Proceedings of the 8th International Semantic Web Conference*, pages 521–536. Springer, 2009.
7. Alessandro Solimando, Ernesto Jiménez-Ruiz, and Giovanna Guerrini. Minimizing conservativity violations in ontology alignments: Algorithms and evaluation. *Knowledge and Information Systems*, 51(3):775–819, 2017.
8. Domenico Lembo, Valerio Santarelli, and Domenico Fabio Savo. Graph-based ontology classification in owl 2 ql. In *Proceedings of the 10th European Semantic Web Conference*, pages 320–334, 2013.
9. Xuefeng Fu, Guilin Qi, Yong Zhang, and Zhangquan Zhou. Graph-based approaches to debugging and revision of terminologies in dl-lite. *Knowledge-Based Systems*, 100:1–12, 2016.
10. Weizhuo Li, Songmao Zhang, and Guilin Qi. A graph-based approach for resolving incoherent ontology mappings. *Web Intelligence*, 16(1):15–35, 2018.
11. Nadeschda Nikitina, Sebastian Rudolph, and Birte Glimm. Interactive ontology revision. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12:118–130, 2012.
12. Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. Supporting manual mapping revision using logical reasoning. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, pages 1213–1218. AAAI Press, 2008.
13. Silvana Castano, Alfio Ferrara, and Stefano Montanelli. Dealing with matching variability of semantic web data using contexts. In *Proceedings of the 22nd international conference on Advanced information systems engineering*, pages 194–208. Springer-Verlag, 2010.
14. Yves R Jean-Mary, E Patrick Shironoshita, and Mansur R Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235–251, 2009.