

# User Involvement for Large-Scale Ontology Alignment

Valentina Ivanova and Patrick Lambrix

Department of Computer and Information Science and the Swedish e-Science Research Centre  
Linköping University, 581 83 Linköping, Sweden

**Abstract.** Currently one of the challenges for the ontology alignment community is the user involvement in the alignment process. At the same time, the focus of the community has shifted towards large-scale matching which introduces an additional dimension to this issue. This paper aims to provide a set of requirements that foster the user involvement for large-scale ontology alignment tasks and a state of the art overview.

## 1 Motivation

The growth of the ontology alignment area in the past ten years has led to the development of a number of ontology alignment tools. The progress in the field has been accelerated by an annual evaluation initiative (Ontology Alignment Evaluation Initiative—OAEI) which has provided a discussion forum for developers and a platform for an annual evaluation of their tools. The number of systems participating in the evaluation increases each year, yet few provide a user interface and even fewer navigational aids or complex visualization techniques. Some systems provide scalable ontology alignment algorithms, however, for achieving high-quality alignments user involvement during the process is indispensable.

Nearly half of the challenges identified in [24] are directly related to user involvement. These include *explanation of matching results* to users, fostering the *user involvement* in the matching process and *social and collaborative matching*. Another challenge aims at supporting users' collaboration by providing *infrastructure and support* during all phases of the alignment process. All these challenges can be addressed by providing user interfaces in combination with suitable visualization techniques.

The demand for user involvement has been recognized by the alignment community and resulted in the introduction of the OAEI Interactive track in 2013. Quality measures for evaluation of interactive ontology alignment tools have been proposed in [20]. The results from the first edition of the track [3] show the benefits from introducing user interactions (in comparison with the systems' non-interactive modes) by means of increasing the precision for all (five) participants and the recall for three of them. The test cases presented in [9] show that simulating user interactions with 30% error rate during the alignment process has led to the same results as a non-interactive matching.

With the development of the ontology engineering field the size and complexity of the ontologies, the alignments and, consequently, the matching problems increase as emphasized in [24] by the *large-scale matching evaluation* challenge. This trend is demanding scalable and (perhaps) novel user interfaces and interactions which is going to impose even stricter scalability requirements towards the algorithms in order to provide

timely response to the users. For instance, graph drawing algorithms should not introduce delays in order for a tool to provide interactive visualization. Scalability, not only in terms of computation, but also in terms of interaction is one of the crucial features for the ontology alignment systems according to [9]. According to [22] user interactions are essential (in the context of large ontologies) for configuring the matching process, incremental matching and providing feedback to the system regarding the generated mapping suggestions.

Currently the alignment systems focus on their main task—ontology alignment—with little or no support for an infrastructure or functionalities which are not directly related to the alignment process. Coping with the increasing size and complexity of ontologies and alignments will require not only comprehensive visualization and user interactions but also supporting functionalities not directly related to them as discussed in subsection 2.2. For instance, the authors in [6] identify cognitive support requirements for alignment tools not directly related to the alignment process—interrupting/resuming the alignment process and providing a feedback on its state. Achieving collaborative matching, discussed above, is going to need a suitable environment.

This paper aims to provide requirements for ontology alignment tools that encourage user involvement for large-scale ontology alignment tasks (section 2). Several ontology alignment systems are evaluated in section 3 in connection with the requirements in section 2. Section 4 provides a discussion and section 5 concludes the paper.

## **2 Requirements for User Support in Large-Scale Ontology Alignment**

This section presents requirements for ontology alignment systems meant to foster user engagement for large-scale ontology alignment problems. Subsection 2.1 summarizes the requirements presented in [6] which address the cognitive support that should be provided by an alignment system to a user during the alignment process. While they are essential for every alignment system, the focus in the community has shifted towards large-scale matching since the time they have been developed. Thus other requirements to assist the user in managing larger and more complex ontologies and alignments are in demand (subsection 2.2). They may not always be directly related to visualization and user interactions but contribute to the development of a complete infrastructure that supports the users during large-scale alignment tasks. Those requirements are extracted from existing works and systems and from the authors' personal experience from the development of ontology alignment and debugging systems ([13], [12]). Since those requirements address user involvement as well they sometimes overlap with those in subsection 2.1 and can be considered complementary to them.

The requirements discussed in this section are crucial for large-scale alignment tasks, but also beneficial for aligning small and medium size ontologies. While the alignment of two medium size ontologies is feasible on a single occasion by a single user even without techniques for reducing user interventions, the alignment of large-scale ontologies without such techniques would be infeasible.

The authors in [7] identify requirements for supporting user interactions in alignment systems which can be seen as a subset of those in subsections 2.1 and 2.2. The

Dimensions	Requirements
Analysis and Generation Dimension	#3.1: automatic discovery of some mappings; #3.2: test mappings by automatically transforming instances between ontologies; #3.3: support potential interruptions by saving and returning users to given state; #3.4: support identification and guidance for resolving conflicts;
Representation Dimension	#4.1: visual representation of the source and target ontology; (I) #4.2: representation of a potential mapping describing why it was suggested, where the terms are in the ontologies, and their context; (I,E) #4.3: representation of the verified mappings that describe why the mapping was accepted, where the terms are in the ontologies, and their context; (I,E) #4.4: identify visually candidate-heavy regions; (I) #4.5: indicate possible start points for the user; (E) #4.6: progress feedback on the overall mapping process; (E) #4.7: feedback explaining how the tool determined a potential mapping; (E)
Analysis and Decision Making Dimension	#1.1: ontology exploration and manual creation of mappings; (I,M) tooling for the creation of temporary mappings; (M) #1.2: method for the user to accept/reject a suggested mapping; (M) #1.3: access to full definitions of ontology terms; (I) #1.4: show the context of a term when a user is inspecting a suggestion; (I)
Interaction Dimension	#2.1: interactive access to source and target ontologies; (I) #2.2: interactive navigation and allow the user to accept/reject suggestions; (I,M) #2.3: interactive navigation and removal of verified mappings; (I,M) #2.4: searching and filtering the ontologies and mappings; (I) #2.5: adding details on verified mappings and manually create mappings; (M)

Table 1. Cognitive support requirements adapted from [6].

same applies for those in [5] which lists requirements for alignment editors and visualizers relevant for individual and collaborative matching and explanation of the alignments.

## 2.1 Cognitive Support Requirements

The requirements identified in [6] are based on research in the area of cognitive theories and a small user study with four participants. They are grouped in four conceptual dimensions (table 1).

The *Analysis and Generation* dimension includes functions for automatic computation and trial execution of mapping suggestions (potential mappings), inconsistency detection/resolution and services for interrupting/resuming the alignment process. The mappings and mapping suggestions together with explanations why/how they are suggested/accepted are visualized by services in the *Representation* dimension. Other functions include interactions for overview and exploration of the ontologies and alignments and feedback for the state of the process. Requirements 1, 2 and 3 from [7] and the first and the third requirements in [5] focus on similar services. The *Analysis and Decision Making* dimension considers the users' internal decision making processes and involves exploration of the ontology terms and their context during the process of discovering

and creating (temporary) mappings, and validating mapping suggestions. During the *Interaction* dimension the user interacts with the system through its exploration, filtering and searching services in order to materialize his/her decisions by creating mappings and accepting/rejecting mapping suggestions. Requirements 4, 5 and 6 from [7] cover similar interactions. Such requirements are also identified in [5]. The requirements for the *Analysis and Decision Making* dimension can be considered to utilize the functionalities represented by the requirements in the *Interaction* dimension.

The requirements provided by the *Representation* and *Interaction* dimensions are involved in the human-system interaction and can be roughly separated in the following three categories—manipulation (M), inspection (I) and explanatory (E) requirements. Those in the first category include actions for transforming the mapping suggestions in an alignment—accept/reject mapping suggestions, add metadata and manually create mappings, etc. Similar functionalities are needed for the ontologies (#5.0), as well, since the user may need to, for instance, introduce a concept in order to provide more accurate mappings, as described in [16] as well. Those in the second category cover a broad set of actions for inspecting the ontologies and alignments—exploring the ontologies, mappings and mapping suggestions, search and filter by various criteria, zoom, overview, etc. The third category includes services for presenting information to the user, for instance, reasons to suggest/accept a mapping suggestion, how the tool has calculated it, hinting at possible starting points and showing the current state of the process.

## 2.2 Ontology Alignment in Large Scale

Various requirements arise from the tendency of increasing the size and complexity of the ontologies, alignments and alignment problems. They need to be supported by scalable visualization and interaction techniques as well. For instance, an introduction of a debugging phase during the alignment process (discussed below) will demand adequate presentation of the defects and their causes which is a problem of the same scale as the main problem discussed in this paper. This subsection does not discuss the techniques for large-scale matching identified in [22] or matching with background knowledge since they are not directly related to user involvement. However some of those techniques affect the interactivity of the systems and thus indirectly influence the user involvement.

Aligning large and complex ontologies cannot be handled on a single occasion. Thus the user should be able to suspend the process, preserve its state and resume it at another point in time (#3.3). Such **interruptions of the alignment process (#5.1)** may take place during different stages, for instance, during the computation of mapping suggestions, during their validation, etc. At the time of interruption the system may provide partial results which can be reused when the alignment process has been resumed. The SAMBO system [13] implements such approach introducing interruptible computation, validation and recommendation sessions. Requirement 9 in [7] can be seen as similar, but without saving and reusing already validated suggestions.

Another strategy to deal with large-scale tasks is to **divide them into smaller tasks (#5.2)**. This can be achieved by clustering algorithms or grouping heuristics. Smaller problems can be more easily managed by single users and devices with limited resources. Requirement 8 from [7] proposes distributing parts of the task among several

users. The authors of AlViz [15] highlight that clustering the graph improves the interactivity of the program (by reducing the size of the problem). Clustering of the ontologies and alignments will allow reusing visualization techniques that work for smaller problems. A fragment-based strategy is implemented in [4] where the authors also note that not all fragments in one schema would have corresponding fragments in another.

In the context of large-scale matching it is not feasible for a user to validate all mapping suggestions generated by a system, i.e., tools' developers should aim at **reducing unnecessary user interventions (#5.3)**. The authors in [20] define a measure for evaluating interactive matching tools based on the number and type of user interventions in connection with the achieved F-measure. LogMap2 [9] only requires user validation for problematic suggestions. In [13] the authors demonstrate that the session-based approach can reduce the unnecessary user interventions by utilizing the knowledge from previously validated suggestions. GOMMA [11] can reuse mappings between older ontology versions in order to match their newer versions. PROMPT [17] logs the operations performed for merging/aligning two ontologies and can automatically reapply them if needed. Reducing the user interventions, but at the same time effectively combining manual validation with automatic computations are two of the challenges identified in [19]. The authors in [2] and [23] discuss criteria for selecting mapping suggestions that are shown to the user and strategies for user feedback propagation in order to reduce the user-system interactions.

Matching large ontologies is a lengthy and demanding task for a single user. It can be relaxed by involving several users who can discuss together and decide on problematic mappings in a collaborative environment. The **social and collaborative matching (#5.4)** is still a challenge for the alignment community [24]. Requirement 7 in [7] addresses this open opportunity. It has potential to reduce the load of a single user and the number of incorrect mappings by building on the collective knowledge of a number of people who can review mappings created by other participants [5]. One of the quality aspects for ontology alignment discussed in [16] is the social aspect—it can be achieved by means of collaboration and information visualization techniques.

Another challenge insufficiently addressed [24] by the alignment community is related to the **environment (#5.5)** where such collaboration could happen. Apart from aligning ontologies it should also support a variety of functions for managing alignments such as storing/editing/retrieving/sharing alignments as explained in [5]. Accommodating different versions of alignments, for instance, would require an entire infrastructure on its own and probably a permanent storage similarly to GOMMA/COMA++. The environment should support services for communication between its members like discussion lists, wikis, subscriptions/notifications, messages, annotations, etc.

Providing **recommendations (#5.6)** is another approach to support the user during the decision making process. Such recommendations can be based on external resources, previous user actions, based on other users' actions (in a collaborative environment), etc. They can be present at each point user intervention is needed—choosing an initial matcher configuration [1], validating mapping suggestions [12], choosing a starting point, etc. The authors in [13] implement recommendation sessions which match small parts of the selected ontologies in order to recommend the best settings for matching them. Different weights can be assigned to the recommendations depending on their

sources. Suitable ranking/sorting strategies could be applied to present them in a particular order.

The outcome of the applications that consume alignments is directly dependent on the quality of the alignments. A direct step towards improving the quality of the alignments and, consequently, the results from such applications is an introduction of **a debugging step during the alignment process (#5.7)**. It was shown in [8] that a domain expert has changed his decisions regarding mappings he had manually created, after an interaction with a debugging system. Most of the alignments produced in the Anatomy, LargeBio and even Conference (which deals with medium size ontologies) tracks in OAEI 2013 [3] are incoherent which questions the quality of the results of the semantically-enabled applications utilizing them. According to [9] *reasoning-based error diagnosis* is one of the three essential features for alignment systems. Almost half of the quality aspects for ontology alignment defined in [16] address lack of correctness in the alignment in terms of *syntactic*, *semantic* and *taxonomic* aspects. The trends toward increasing the size and complexity of the alignment problem demand debugging techniques more than ever. In this context a debugging module should be present in every alignment system. The authors in [10] show that repairing alignments is feasible at runtime and improves their logical coherence when (approximate) mapping repairing techniques are applied. Since ontology debugging presents considerable cognitive complexity (due to the, potentially, long chains of entailments) adequate visual support to aid user interactions is a necessity.

In the field of ontology debugging there is already ongoing work that addresses explanation of defects to users. These techniques could be borrowed and applied in the ontology alignment to address the challenge for **explaining the matching results** to the users (#4.2, #4.7). The authors in [19] specify generating human understandable explanations for the mappings as a challenge as well. The authors in [1] implement advanced interfaces for **configuring the matching process (#5.8)** which provide the users with insights of the process and contribute to the understanding of the matching results.

**Trial execution of mappings (#5.9.1)** (what-if) mentioned above in the context of confirming user's expectations (#3.2) will be of even greater help during the debugging and alignment by aiding the user in the propagation of the consequences of his/her actions. Additionally **support for temporary decisions (#5.9.2)** in general, including temporary mappings (#1.1), list of performed actions and undo/redo actions, will help the user to explore the effects of his/her actions (and reduce the cognitive load).

### 3 Overview of Ontology Alignment Systems

The systems in this literature study are selected because they have mature interfaces, often appear in user interface evaluations and accommodate features addressing the alignment of large ontologies.

#### 3.1 AlViz

AlViz [15] is a Protégé plug-in which uses the linking and brushing paradigm for connecting multiple views of the same data where navigation in one of the views changes

*User Involvement for Large-Scale Ontology Alignment*

	Requirements	AIViz	SAMBO	PROMPT	CogZ	RepOSE	AML	COMA
manipulate	#2.5;1.1 create mapping manually	✓(*)	✓	✓	✓	+	-	✓(*)
	#2.2;1.2 accept/reject suggestion	✓(*)	✓	✓	✓	✓	-	✓(*)
	#2.5 add metadata to mapping	-	✓	✓	✓	-	-	-
	#2.3 move a mapping to list	-	✓	✓	✓	+	-	-
	#5.0 ontology	✓	-	✓	✓	-	-	-
inspect	#2.2;1.4 mapping suggestions	✓(*)	✓	✓	✓	+	-	✓(*)
	#2.3 mappings	✓(*)	✓	✓	✓	✓	✓	✓(*)
	#4.4 heavy-regions	✓	-	-	✓	-	-	+
	#2.4 filter/search	-/✓	-/✓	-/-	✓/✓	-/-	+/✓	-/-
	#4.1/2/3;2.1;1.1/3 ontologies	✓	✓	✓	✓	✓	+	✓
explain	#4.2/7;5.8 why/how suggested	+	+	✓	✓	+	+	+
	#4.3 why accepted	-	✓	✓	✓	-	-	-
	#4.5 starting point	+	-	-	+	✓	-	+
	#4.6 process state	✓	+	+	✓	+	-	+
large-scale	#5.1;3.3 sessions	+	✓	+	+	+	-	+
	#5.2 clustering	✓	+	-	✓	✓	✓	✓
	#5.3 reduce user interventions	-	+	+	-	-	-	-
	#5.4 collaboration	-	-	-	-	-	-	-
	#5.5 environment	-	+	+	-	-	+	+
	#5.6 recommend/rank	-	✓	+	+	✓	-	✓
	#5.7;3.4 debugging	-	✓	✓	✓	✓	✓	-
	#5.8;4.2/7 matchers configuration	-	✓	-	-	✓	✓	✓
	#5.9.1;3.2 trial execution	-	-	-	-	-	-	-
#5.9.2;1.1 temporary decisions	✓	+	+	✓	-	-	-	

**Table 2.** Requirements to support user involvement in large-scale matching tasks. (supported(✓); partly supported(+); special case, details in the text(\*); not supported(-))

the representation in the other. During the alignment process each ontology is represented as a pair of views—a tree and a small world graph—i.e., four in total. The trees provide well-known editing and exploratory functionalities. There is no clear distinction between mappings and mapping suggestions (✓(\*)). Mappings are edited, accepted and rejected in the tree views by toolbar buttons for defining the type of mappings. The small world graphs represent an ontology as a graph where the nodes (represent the entities) are clustered according to a selected level of detail. The size of the clusters corresponds to the number of nodes in them. The edges between the clusters represent the selected relation (mutual property). Intuitive exploration is achieved by the linking and brushing technique, adjustable level of details (by means of a slider) and selecting a relationship to present (from a drop-down list). The small world graphs provide an overview of the ontologies where color-coding provides an overview of the similar clusters (in the two ontologies) and the colors of the clusters are inherited from the underlying nodes according to one (out of three) strategy. Tooltips and labels can be switched on and off.

Different sessions are not directly supported, but simple interruption and resumption of the alignment process can be achieved by saving and loading the input file which

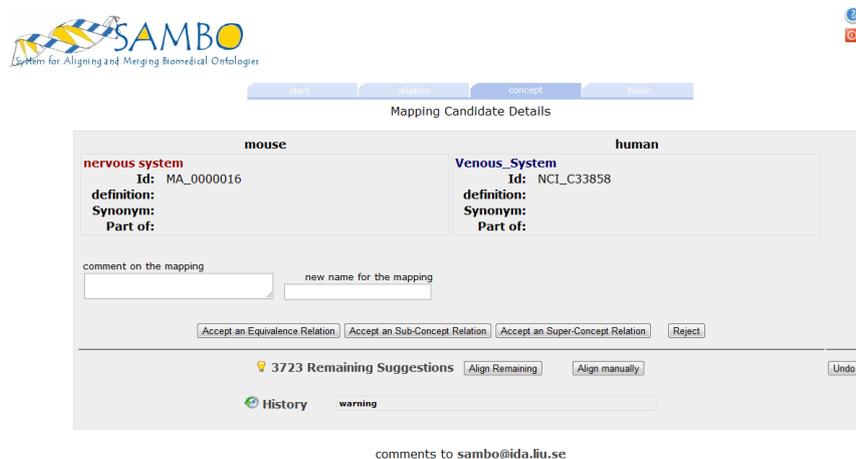


Fig. 1. SAMBO [13].

contains the mappings. Temporary decisions for questionable mappings are supported by a tracking button. Undo/redo buttons and history of activities are also provided.

### 3.2 SAMBO

SAMBO [13] (based on [14]) is an ontology alignment system that addresses the challenges related to user involvement by introducing interruptible sessions—computation, validation and recommendation sessions. The computation session computes mapping suggestions between two ontologies and can utilize results from previous validation and recommendation sessions. The user validates the mapping suggestions during the validation session. A reasoner may be used during both sessions to check the consistency of the (validated) mapping suggestions in connection with the ontologies. Both sessions can provide partial results upon interruption thus the validation session may start before the end of the computation and not all of the mapping suggestions need to be validated at once. The recommendation session matches small parts of the two ontologies offline using an oracle or previous validation decisions if available and employs different (combination of) algorithms and filtering strategies in order to recommend the best future settings for matching the two ontologies. The results of the sessions are stored in a database. The user may choose to start a new or to resume a saved session.

The user interface allows selection of matchers, their weights and strategies for combination. Two alternating modes are available during the validation—suggestion (shown in Figure 1) and manual mode. All suggestions for a concept are shown at once during the suggestion mode. The user can give a name for and annotate a mapping/concept. The user can accept/reject a suggestion by pressing a dedicated button. Both ontologies are shown as indented trees during the manual mode and the user can create a mapping by selecting a concept in each tree. A search function is implemented for locating a term of interest. Lists with the previous accepted/rejected and remaining suggestions are available. An undo button is available as well.

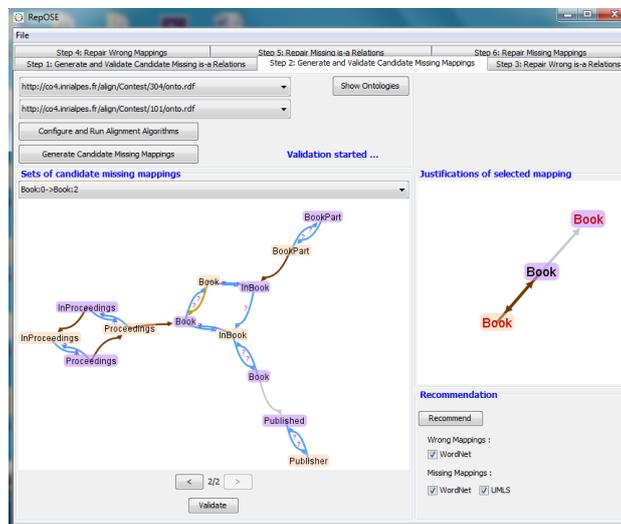


Fig. 2. RepOSE [12].

### 3.3 RepOSE

RepOSE [12], shown in Figure 2, is based on an integrated taxonomy alignment and debugging framework. The system can be seen as an ontology alignment system with a debugging component for detecting and repairing modelling defects in taxonomy networks (missing and wrong subsumption relations/mappings). The alignment process goes through three phases—generation of mapping suggestion, validation and repairing. Separate panels are provided for the validation and repairing phases to guide the user through them. Possible starting points, recommendations and ranking strategies are available during both phases. The alignment process can be configured by selecting matchers, their weights and the threshold for filtering the mapping suggestions. The algorithm for detecting defects in the debugging component can be seen as a structure-based alignment algorithm—as such it is configured separately. The suggestions computed from it are logically derivable and they are presented to the user together with their derivation paths. The rest are only presented with their confidence values.

During the validation phase the mapping suggestions are shown as graphs in groups where the last group in the list contains the most suggestions. The nodes in the graph represent concepts and the edges—relations and mappings. The nodes are color-coded according to their hosting ontology and the edges—the state of the represented relations/ mappings—mapping suggestions, asserted/added/removed relations/mappings. When the user accepts/rejects a suggestion the corresponding edge is labeled accordingly and it is moved to the list for repairing. The user can validate only a portion of the suggestions and start the repairing phase. The user can see each pair of ontologies and their current alignment and the entire ontology network upon request. During the repairing phase the system provides alternative repairing actions instead of directly adding the validated mapping. Logically derivable wrong mappings can be also repaired.

The system checks for contradictions after each group of suggestions is validated and after a repairing action and warns the user if such are found. There is no indication for the process state but it can be observed by reflecting on the validation and repairing phases. Sessions are only supported through saving/loading the ontologies and mappings, but the suggestions are not preserved and have to be computed from scratch.

### **3.4 AML**

AML has been designed based on AgreementMaker with the purpose of matching very large ontologies. Its user interface is presented in [21]. The working area in AML is divided into two panels—a Resource Panel, on the top, provides a summary of the ontologies, alignment, etc., and a Mapping Viewer where modules extracted from the ontologies and alignment are represented as graphs. Instead of showing the entire network, the visualization is focused on a single mapping where the graph depicts the mapping, up to five (default is two) levels of ascending/descending concepts of the concepts in the mapping and other mappings between the displayed concepts (if any). The nodes and edges are labeled with the names of the classes and relations (subsumptions are not labeled), respectively, and colored depending on the ontology they belong to. The mappings are labeled with their confidence values and their directions are denoted with arrows. Three options are provided for navigating through the mappings—list of mappings, previous/next buttons and search (in combination with auto-complete). The user can configure the alignment process by selecting a matcher, its threshold, cardinality for the alignment and sources of background knowledge. The final alignment can be repaired and evaluated against a reference alignment.

### **3.5 COMA++**

COMA++ is an alignment system for matching large schemata and ontologies [1]. The system consists of five components accessible through a user interface [4]. The *repository* stores the ontologies and alignments. The Workspace tab provides access to the *schema and mapping pools* which manage the ontologies and alignments in memory. Other operations involving alignments, such as merging schema and alignments, add/remove mappings in edit mode, comparing (evaluating an alignment against a reference alignment using different quality measures) and diff/intersect (determining the different/shared mappings between two alignments) are provided as well. The Match menu provides a variety of options for configuring the matching process through the *match customizer*—creating/modifying/deleting/resetting matchers and strategies, showing the dependencies between them and saving them (into the repository) for future use. The matching process is performed in the *execution engine*. Some of the strategies support iterations, where the user can modify the output prior to the execution of the next iteration. The toolbar has buttons for configuring/running/interrupting the process, step-by-step execution and editing mappings. The ontologies are shown side-by-side as unmodifiable indented trees and the mappings between them are represented as lines color-coded depending on their confidence values. There is no clear distinction between mappings and mapping suggestions (✓(\*)). The highest confidence value is assigned to the manually created mappings. The regions with many mappings can be observed by

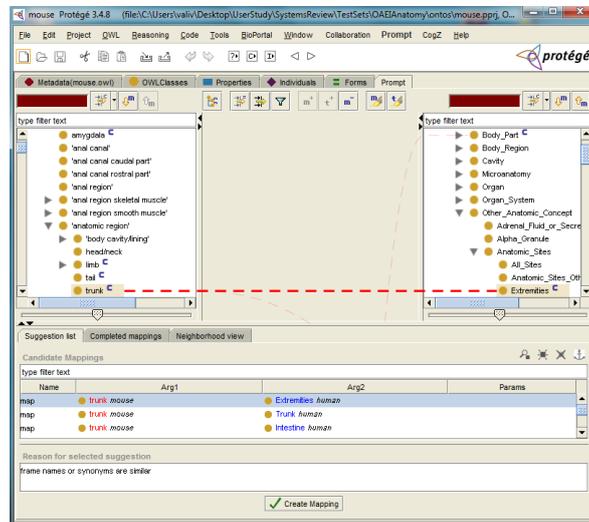


Fig. 3. The latest version of CogZ.

the high number of lines between them. The process state can be observed by the predominant color of the mappings. Sessions are supported as in RepOSE.

### 3.6 PROMPT

The PROMPT suite [18] is a set of Protégé plug-ins for managing ontologies and their versions: iPROMPT merges and aligns ontologies interactively employing the local context of the concepts; AnchorPROMPT computes additional mapping suggestions acting on a larger scale than iPROMPT; PROMPTDiff performs structural comparison between different versions of an ontology and PROMPTFactor extracts an independent modules from an ontology. These plug-ins share interface components, data structures, some algorithms and heuristics.

The first version of PROMPT, [17], shows the source and target ontologies as indented trees on both sides of the screen where the mapping suggestions are presented as a list of pairs between them. An explanation for why this pair is a mapping suggestion is provided to the user. The user can examine the suggestions from the list, save those that are correct or create new mappings. Upon user action the tool detects conflicts, if any it suggests solutions and generates new suggestions in the area the latest operation has happened. The suggestions/conflicts are resorted to first list those in the area of the latest operation. PROMPT can log operations and execute them again if needed. The process state can be observed indirectly.

### **3.7 CogZ**

CogZ, shown in Figure 3, addresses the cognitive support requirements from [6]. It is a visualization plug-in which extends the PROMPT user interface and reuses the rest of its components.

The first version of CogZ, Jambaprompt, includes a graph visualization of the neighborhood of each of the concepts in a mapping suggestion—direct super and subclasses. Each of the classes can be expanded thus providing an incremental navigation. The Jambaprompt plug-in also supports filtering of the mapping suggestions by various criteria. It is extended in [6] to provide an overview of the ontologies and mappings by employing treemaps. The user can identify potentially 'heavy regions' using the treemaps in combination with color-coding. Pie-charts provide additional details regarding already mapped concepts and mapping suggestions. Temporary mappings, different from the mapping suggestions, are introduced in CogZ to relieve the users' memory and help them to write down potential solutions. Similarly to COMA++, the mappings between the ontologies (shown as trees) are presented with lines which can be annotated to provide additional details. CogZ provides semantic zoom and interactive search.

## **4 Discussion**

Table 2 shows the systems' support for the requirements identified in section 2. The manipulation and inspection requirements are almost entirely supported by the first four systems. However to be able to draw conclusions for the level of usability of the different visualization approaches, a user study is needed. It is worth noting that COMA++ and AlViz do not distinguish between mappings and mapping suggestions, a functionality that may help the users to keep track which correspondences have been already visited. The least supported category from the requirements in [6] is the one that assists the users most in understanding the reasons for suggesting/accepting mapping suggestions—while PROMPT and CogZ provide a textual description to explain the origin of mapping suggestions, the other tools only present a confidence value (which may (not) be enough depending on how familiar the domain expert already is with the ontology alignment field). Other requirements in this category include providing a starting point and a state of the process. Even though rarely supported they can often be observed by the number/status of the verified suggestions.

Some systems limit the amount of data presented to the user by using sessions and clustering. Only one system preserves the state of the process during interruptions. The others partially address the session requirement by save/load (ontologies and alignments) functions but without preserving the already computed suggestions. Almost all of the tools support clustering of the content presented to the user (not necessary for all views/modes) to avoid cluttering of the display, clustering during the computations is also often supported. Another possibility could be to guide the user (through complex interfaces and huge input) by presenting different interfaces connected to different phases of the process, for instance, by providing a different view for each phase. The existence of different phases in general could also allow for more opportunities for fine-tuning of the process.

The session-based approach in [13] helps reducing the user interventions during the alignment process by reusing previously validated mappings. PROMPT takes into account the area of the latest user intervention while computing a new portion of suggestions to maintain the user's focus. To assist the user decision making process some systems provide recommendations in various forms—SAMBO provides a recommendation session, COMA++ default matchers configuration, RepOSE recommendations (from external sources) during the validation. Matchers' configuration is also supported to different extent—COMA++ provides advanced matchers' combinations while RepOSE only supplies a list with matchers and their weights. To support temporary decisions CogZ introduces temporary mappings and AlViz a tracking button. SAMBO partially presents such functionality by an undo button and history of actions, PROMPT by reapplying the user actions. Trial execution is not supported by any of the tools.

Looking at the table we can conclude that most of the systems provide debugging techniques, but this is not the case in reality as discussed in subsection 2.2. Although these systems consider debugging of the alignment, they address different kinds of defects—RepOSE detects/repairs modelling defects in taxonomies, SAMBO checks for inconsistencies and AML addresses disjointness assuming the ontologies are coherent. Further, RepOSE relies on manual repairing while AML repairs the alignment automatically.

The social and collaborative matching is still a challenge. SAMBO, PROMPT and CogZ provide mapping annotations but it is unlikely they have been developed to address this issue. While implementing other functionalities SAMBO and COMA++ took first steps in providing a collaborative environment by introducing permanent storages. AML, PROMPT and COMA++ have functions for evaluating an alignment against a reference alignment and for comparing two alignments.

## 5 Conclusions

This paper defines a set of requirements to address the user involvement in large-scale ontology alignment tasks. It provides a literature based overview of several systems selected due to their mature interfaces and features that address the alignment of large ontologies.

Since the papers describing the systems mostly focus on algorithms and rarely on user interfaces such assessment of the coverage of the requirements is inherently imprecise. In order to provide better understanding for how the systems support the requirements identified in section 2 we intend to conduct an observational user study as a future work. The study will consider the requirements in the manipulation, inspection and explanation categories by developing tasks that address them in a large-scale setting. It will provide detailed overview of the advantages and disadvantages of the user interfaces of several selected systems. Changes in the list with requirements may occur as a consequence of the study.

**Acknowledgments.** We thank the National Graduate School in Computer Science (CUGS) and the Swedish e-Science Research Centre (SeRC) for financial support.

## References

1. D Aumüller, H H Do, S Maßmann, and E Rahm. Schema and ontology matching with COMA++. In *SIGMOD*, pages 906–908, 2005.
2. I F Cruz, C Stroe, and M Palmonari. Interactive user feedback in ontology matching using signature vectors. In *ICDE*, pages 1321–1324, 2012.
3. B Cuenca Grau et al. Results of the ontology alignment evaluation initiative 2013. In *OM*, pages 61–100, 2013.
4. H H Do. *Schema Matching and Mapping-based Data Integration*. PhD thesis, 2005.
5. J Euzenat and P Shvaiko. User Involvement. In *Ontology Matching*, pages 353–375. 2013.
6. S M Falconer and M D Storey. A Cognitive Support Framework for Ontology Mapping. In *ISWC/ASWC*, pages 114–127, 2007.
7. M Granitzer, V Sabol, K W Onn, et al. Ontology Alignment—A Survey with Focus on Visually Supported Semi-Automatic Techniques. *Future Internet*, pages 238–258, 2010.
8. V Ivanova, J L Bergman, U Hammerling, and P Lambrix. Debugging taxonomies and their alignments: the ToxOntology-MeSH use case. In *WoDOOM*, pages 25–36, 2012.
9. E Jiménez-Ruiz, B C Grau, Y Zhou, and I Horrocks. Large-scale Interactive Ontology Matching: Algorithms and Implementation. In *ECAI*, pages 444–449, 2012.
10. E Jiménez-Ruiz, C Meilicke, B C Grau, and I Horrocks. Evaluating Mapping Repair Systems with Large Biomedical Ontologies. In *Description Logics*, pages 246–257, 2013.
11. T Kirsten, A Gross, et al. GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *Journal of Biomedical Semantics*, 2(1), 2011.
12. P Lambrix and V Ivanova. A unified approach for debugging is-a structure and mappings in networked taxonomies. *Journal of Biomedical Semantics*, 4:10, 2013.
13. P Lambrix and R Kaliyaperumal. A Session-Based Approach for Aligning Large Ontologies. In *ESWC*, pages 46–60. 2013.
14. P Lambrix and H Tan. SAMBO - a system for aligning and merging biomedical ontologies. *Journal of Web Semantics*, 4(3):196–206, 2006.
15. M Lanzenberger, J Sampson, and M Rester. Ontology visualization: Tools and techniques for visual representation of semi-structured meta-data. *J.UCS*, 16(7):1036–1054, 2010.
16. M Lanzenberger, J Sampson, M Rester, Y Naudet, and T Latour. Visual ontology alignment for knowledge sharing and reuse. *Journal of Knowledge Management*, 12(6):102–120, 2008.
17. N F Noy and M A Musen. Algorithm and Tool for Automated Ontology Merging and Alignment. In *AAAI*, pages 450–455, 2000.
18. N F Noy and M A Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
19. L Otero-Cerdeira, F J Rodríguez-Martínez, and A Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 2014.
20. H Paulheim, S Hertling, and D Ritze. Towards Evaluating Interactive Ontology Matching Tools. In *ESWC*, pages 31–45, 2013.
21. C Pesquita, D Faria, E Santos, J Neefs, and F M Couto. Towards Visualizing the Alignment of Large Biomedical Ontologies. In *DILS*, pages 104–111, 2014.
22. E Rahm. Towards large-scale schema and ontology matching. In *Schema matching and mapping*, pages 3–27. 2011.
23. F Shi, J Li, J Tang, G Xie, and H Li. Actively learning ontology matching via user interaction. In *ISWC*, pages 585–600. 2009.
24. P Shvaiko and J Euzenat. Ontology Matching: State of the Art and Future Challenges. *Knowledge and Data Engineering*, 25(1):158–176, 2013.