Requirements for and Evaluation of User Support for Large-Scale Ontology Alignment - Extended version

Valentina Ivanova[1,2] and Patrick Lambrix[1,2] and Johan Åberg[1]

(1) Dept of Computer and Information Science and (2) 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. Further, we present and discuss the results of a literature study for 7 ontology alignments systems as well as a heuristic evaluation and an observational user study for 3 ontology alignment systems to reveal the coverage of the requirements in the systems and the support for the requirements in the user interfaces.

1 Motivation

The growth of the ontology alignment area in the past ten years has led to the development of many ontology alignment tools. The progress in the field has been accelerated by the 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 participants in the OAEI 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 [43] 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 [37]. The results from the first two editions of the track, [10] and [12], show the benefits from introducing user interactions (in comparison with the systems' non-interactive modes). In the first edition the precision for all (five) participants and the recall for three was raised. For the second edition three (out of four) systems increased their precision and two their recall. The test cases presented in [21] 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 [43] 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 as stated in [21]. According to [40] 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 many 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. For instance, the authors in [15] 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 is going to need a suitable environment.

This paper provides requirements for ontology alignment tools that encourage user involvement for large-scale ontology alignment tasks (section 2). We also present the results from a literature study (section 3) and two user interface evaluations (section 4) to reveal the coverage of the requirements in current ontology alignment systems and the support for the requirements in their user interfaces. 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 [15] 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, their influence becomes more pressing with increasing ontology size and complexity. Further, the focus in the community has shifted towards large-scale matching since the time they have been developed. For instance, DBpedia and the Gene Ontology are among the five most commonly used ontologies according to a survey conducted in 2013 and presented in [44]. Thus other requirements (not necessary related to the user interface) to assist the user in managing larger and more complex ontologies and alignments are in demand (subsection 2.2). They are extracted from existing works and systems and from the authors' personal experience from developing ontology alignment and debugging systems [27, 26, 25]. These requirements contribute to the development of a complete infrastructure that supports the users during large-scale alignment tasks (and may pose additional visualization and interface requirements). 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 [17] identify requirements for supporting user interactions in alignment systems which can be seen as a subset of the requirements in this paper. The same applies for those in [13] which lists requirements for alignment editors and visualizers relevant for individual and collaborative matching and explanation of alignments.

2.1 User Interface Requirements

The requirements identified in [15] 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 [17] and the first and the third requirements in [13] 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 [17] cover similar interactions. Such requirements are also identified in [13]. 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 subcategories of the user interface category (shown in table 2)—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 also needed for the ontologies (#5.0) since the user may need to, for instance, introduce a concept in order to provide more accurate mappings, as described

Table 1. Cognitive support requirements adapted from [15].

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;
Repre-sentation 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)

in [29] 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 (name, already mapped concepts, rejected suggestions, etc.), zoom, overview, etc. Those tasks can be aligned with the first five tasks identified in [23] and based on [42] which discusses the tasks that should be supported by information visualization applications. 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 Infrastructure and Algorithms Requirements

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

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

Requirements				AlViz	SAMBO	PROMPT	CogZ	RepOSE	AML	COMA
large-scale	user interface	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	✓	+	+	✓	+	-	+
	infrastructure & algorithms		#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	✓	+	+	✓	-	-	-

the techniques for large-scale matching identified in [40] 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. SAMBO [26] implements this by introducing interruptible computation, validation and recommendation sessions. Requirement 9 in [17] 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 [17] proposes distributing parts of the task among several users. The authors of AlViz [28] 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 [11] 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., tool developers should aim at **reducing unnecessary user interventions (#5.3)**. The authors in [37] 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 [21] only requires user validation for problematic suggestions. In [26] the authors demonstrate that the session-based approach can reduce the unnecessary user interventions by utilizing the knowledge from previously validated suggestions. GOMMA [24] can reuse mappings between older ontology versions in order to match their newer versions. PROMPT [34] 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 [36]. The authors in [9] and [41] discuss criteria for selecting mapping suggestions to show and strategies for user feedback propagation in order to reduce the user-system interactions. The same issues in a multi-user context are presented in [8]. A dialectical approach reusing partial alignment to map portions of two ontologies without exposing them is evaluated in [38].

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 [43]. Requirement 7 in [17] 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 [13]. One of the quality aspects for ontology alignment discussed in [29] is the social aspect—it can be achieved by means of collaboration and information visualization techniques.

Another challenge insufficiently addressed [43] 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 as explained in [13]. 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 communication services between its members—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 [3], validating mapping suggestions [25], choosing a starting point, etc. The authors in [26] 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 the introduction of **a debugging step during the alignment process (#5.7)**. It was shown in [18] 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 [10] are incoherent which questions the quality of the results of the semantically-enabled applications utilizing them. According to [21] *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 [29] 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 [22] 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 [4], [32]. These techniques could be borrowed and applied to ontology alignment to address the challenge for **explaining the matching results** to the users (#4.2, #4.7). The authors in [36] specify generating human understandable explanations for the mappings as a challenge as well. The authors in [3] 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), supports the user by confirming his/her expectations (#3.2), will be of great help during the debugging and alignment by aiding the user understanding the consequences of his/her actions. Additionally **support for temporary decisions (#5.9.2)**, including temporary mappings (#1.1), a 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 Literature Study

A literature study was performed on a number of systems. It does not aim to provide a complete overview of the existing alignment systems. The systems in the study were selected because they have mature interfaces, often appear in user interface evaluations and accommodate features addressing alignment of large ontologies. Some systems were omitted since their user interfaces are not described in papers (YAM++ [31]) or provide only limited functionalities (LogMap [20]). Table 2 shows the systems' support for the requirements identified in section 2.

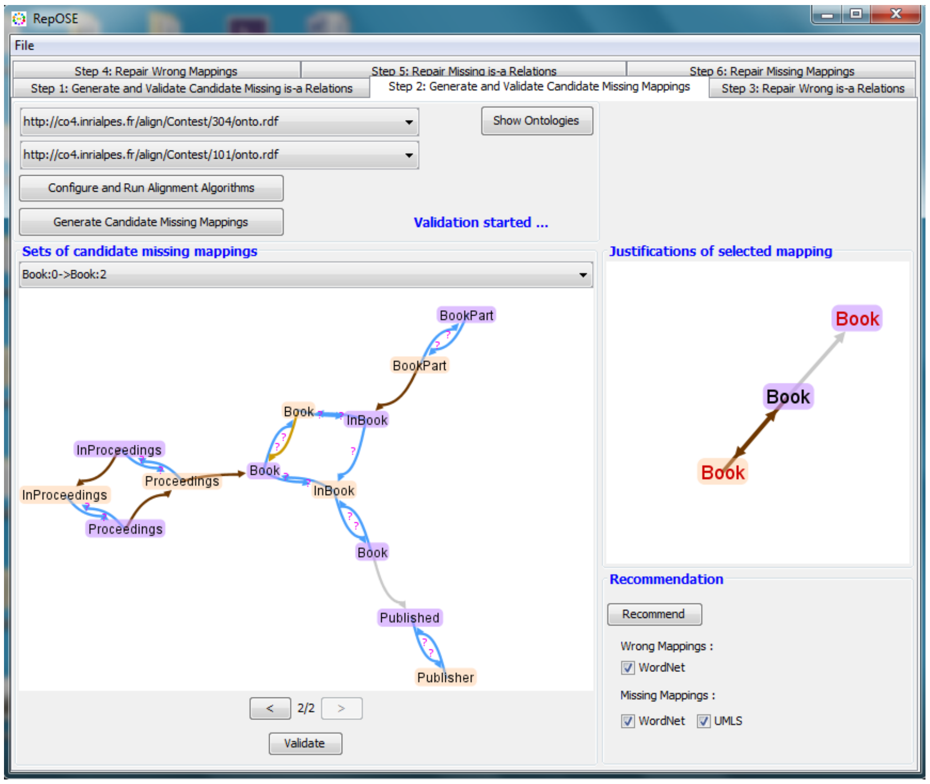
3.1 Systems

AlViz [28] 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 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 views in total for the two ontologies. 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). 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 contains the mappings. Temporary decisions for questionable mappings are supported by a tracking button. Undo/redo buttons and history of activities are also provided.

SAMBO [26] (based on [27]) 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. It employs different (combination of) algorithms and filtering strategies in order to recommend the best future settings for matching the two ontologies. The different strategies and mapping suggestions are stored in a database. The user may choose to start a new or to resume a saved session. A detailed description of the current user interface is availabe in subsection 4.1.

RepOSE [25], shown in Figure 1, is based on an integrated taxonomy alignment and debugging framework [19]. The system can be seen as an ontology alignment system


RepOSE
File
Step 4: Repair Wrong Mappings
Step 5: Repair Missing is-a Relations
Step 6: Repair Missing Mappings
Step 1: Generate and Validate Candidate Missing is-a Relations
Step 2: Generate and Validate Candidate Missing Mappings
Step 3: Repair Wrong is-a Relations
Show Ontologies
Configure and Run Alignment Algorithms
Generate Candidate Missing Mappings
Validation started ...
Sets of candidate missing mappings
Justifications of selected mapping
BookPart
BookPart
Book
InBook
InProceedings
Proceedings
Book
InProceedings
InBook
Proceedings
Book
Published
Publisher
Book
Book
Book
Recommendation
Recommend
<
2/2
>
Validate

http://co4.inrialpes.fr/align/Contest/304/onto.rdf

http://co4.inrialpes.fr/align/Contest/101/onto.rdf

Book:0->Book:2

Wrong Mappings : [✓ WordNet]

Missing Mappings : [✓ WordNet] [✓ UMLS]

Fig. 1. RepOSE [19].

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 to guide the user through the validation and repairing phases are provided. 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 it computes 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.

AML has been designed based on AgreementMaker [7] with the purpose of matching very large ontologies thus we consider only AML in our study. Its user interface is presented in [39]. 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 focuses 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.

COMA++ is a system for matching large schemata and ontologies [3]. It consists of five components accessible through a user interface [11]. The *repository* stores the ontologies and alignments (sessions support). 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. A detailed description of the current user interface is availabe in subsection 4.1.

PROMPT suite [35] 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 independent modules from an ontology. These plug-ins share interface components, data structures, some algorithms and heuristics. The first version of PROMPT, [34], 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 a pair of concepts 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 list first 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.

CogZ (Figure 5) addresses the cognitive support requirements from [15]. 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 [14], 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 [15] 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. A detailed description of the current user interface is available in subsection 4.1.

3.2 Results

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 user interface requirements 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 two systems preserve 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. Such approach is implemented in RepOSE where the validation and repairing phases are presented in different views. The existence of different phases in general could also allow for more opportunities for fine-tuning of the process.

The session-based approach in [26] 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.

4 User Interface Evaluations

As a further step in our study, we conducted a usability evaluation to reveal to what level the requirements are supported. We applied a multiple method approach by conducting an observational study and a heuristic evaluation to address the three aspects of the ISO 9241-11 standard for usability, [1]: efficiency, effectiveness, satisfaction. The standard defines these terms as follows:

- *usability*—extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.
- *effectiveness*—accuracy and completeness with which users achieve specified goals.
- *efficiency*—resources expended in relation to the accuracy and completeness with which users achieve goals.
- *satisfaction*—freedom from discomfort, and positive attitudes towards the use of the product.

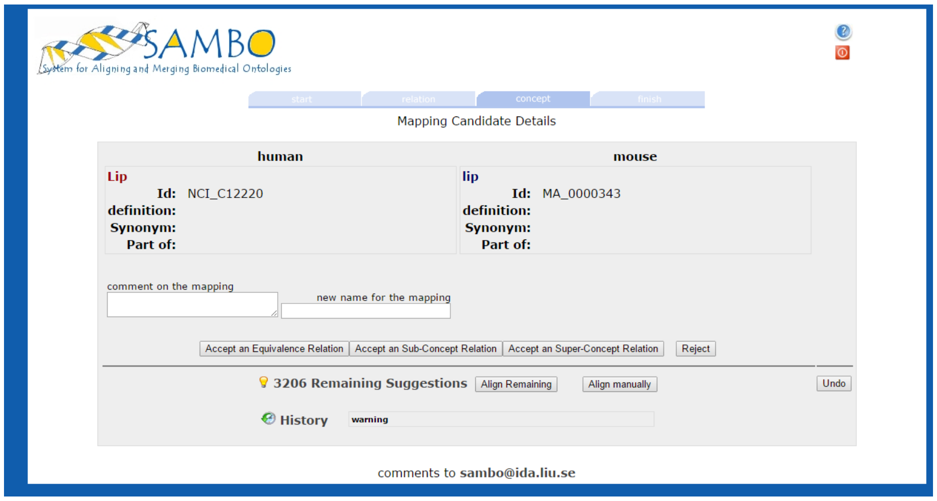
We selected three ontology alignment systems (CogZ, COMA 3.0 and SAMBO), from those in the literature study, that support as many as possible of the requirements in the user interface category; were freely available to us and that could be used without the installation of additional software packages. We evaluated the user interfaces using a heuristic evaluation (the effectiveness aspect) by an expert user as well as through an observational study using novice users. From the observational study we collected task completion times (the efficiency aspect) and task success (the effectiveness aspect). The satisfaction aspect is addressed by the SUS questionnaire [5].

CogZ [15] and COMA 3.0 [30] were downloaded on October 6, 2014. COMA 3.0 CE V3 [30] was downloaded from http://sourceforge.net/projects/coma-ce/. To use CogZ [15] we downloaded Protégé 3.4.8 (build 269) since it is bundled with PROMPT and there is no need of additional software packages installation. We used the default matcher $NodesNameW in COMA 3.0 which generated 1361 mappings. We used lexical matching with synonyms in CogZ which generated 1447 mappings. In SAMBO [26] (our own system) we used two configurations of matchers depending on the order of the input ontologies—Ngram and TermWN (3809 suggestions) and Ngram (single threshold 0.7) (3206 suggestions) when the first ontology was AMA, respectively, NCI-A.

4.1 Heuristic Evaluation

Our first evaluation is a heuristic evaluation. It aims to reveal usability issues by comparing the systems' interfaces to a set of accepted usability heuristics. This evaluation considers Nielsen's ten heuristics defined in [33] and presented briefly below. According to Nielsen [33] *"The principles are fairly broad and apply to practically any type of user interface, including both character-based and graphical interfaces."*. We note that these heuristics are not related in any way to the requirements in table 2.

a. *Simple and Natural Dialog*—provide only absolutely necessary information, any extra information competes for the users' attention; group relevant information together and follow gestalt principles;


SAMBO
System for Aligning and Merging Biomedical Ontologies
start
relation
concept
finish
Accept an Equivalence Relation
Accept an Sub-Concept Relation
Accept an Super-Concept Relation
Reject
Align Remaining
Align manually
Undo
History

Mapping Candidate Details

human	mouse
Lip	lip
Id: NCI_C12220	Id: MA_0000343
definition:	definition:
Synonym:	Synonym:
Part of:	Part of:

comment on the mapping

new name for the mapping

3206 Remaining Suggestions

warning

comments to sambo@ida.liu.se

Fig. 2. SAMBO [26], Suggestion Align mode.

b. *Speak the Users' Language*—use users' familiar terminology and follow the natural information workflow; use metaphors with caution;
c. *Minimize the Users' Memory Load*—pick from a list rather than recall from the memory; use commonly recognizable graphic elements;
d. *Consistency*—the same things are at the same place and perform the same function; follow accepted graphical/platform/etc. conventions;
e. *Feedback*—provide timely and accurate feedback for all actions and task progress information;
f. *Clearly Marked Exits*—provide components to revoke or reverse actions;
g. *Shortcuts*—design the system proactively rather than reactively, provide accelerators for (experienced) users or default configurations for novice users;
h. *Good Error Messages*—meaningful error messages showing the problem in users' language and possible recovery actions instead of system codes;
i. *Prevent Errors*—provide confirmation dialogs for irreversible actions;
j. *Help and Documentation*—provide documentation for different type of users.

SAMBO provides two separate modes—*Suggestion Align*, figure 2, and *Align Manually*, figure 3,—where a single high level task is performed in each. All potential mappings for a single concept (calculated by the system) are displayed for validation in the *Suggestion Align* mode. The user can create mappings manually in the *Align Manually* mode where the ontologies are displayed side-by-side. The system is web-based and the navigation between the modes is performed with a button, however, a link would be a more intuitive choice {d}. Both modes provide minimalistic design but they also contain elements that are not necessary for the tasks and take vertical space on the screen—the logo and the email address at the bottom {a}. While the layout in the *Suggestion Align* mode fits nicely in the browser dimensions this could be an issue in the

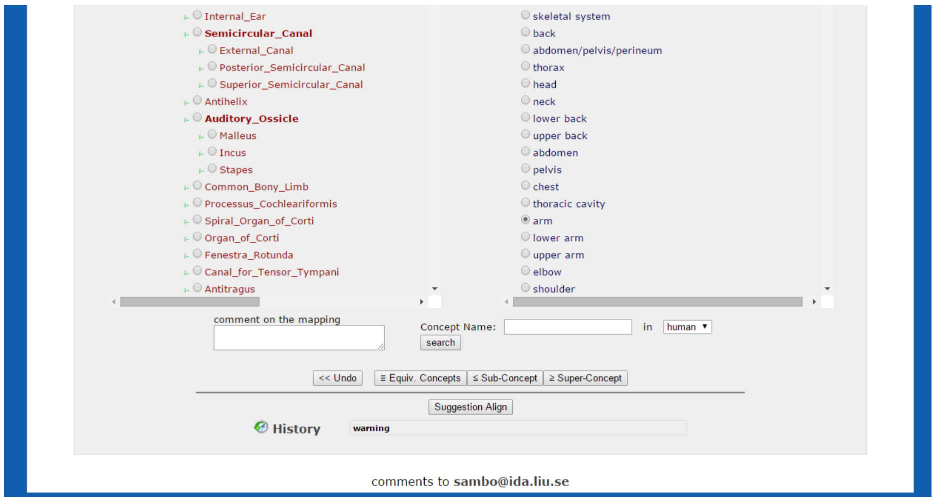


Fig. 3. SAMBO [26], part-of Align Manually mode.

Align Manually mode (and some screen dimensions) since part of the information may not be presented on the screen (which will require scrolling) {a}. The tab set takes vertical space but provides limited information for the state of the process {a}.

The browser window in the *Suggestion Align* mode (figure 2) is divided into two parts by a thick gray line but the buttons above and below are very close to it (taking into account the distances between the other elements as well) {a}. Thus the line and the buttons may be perceived as one unit instead of different units. The information belonging to a concept is grouped together and enclosed in a box in the upper and central parts {a}. However it does not show the concept context in the ontology hierarchy. All mappings for a concept are presented as a list where the user chooses the current mapping by a radio button. The user can annotate and rename a mapping using the text fields below. Each mapping can be accepted as equivalence or subsumption mapping or rejected by the corresponding buttons. Their labels clearly explain their function, however, the buttons' color matches the background color, they are glued together and slightly change their appearance on hover. Since they perform the most important function in this mode they can be designed such that they stand out among the other elements {a}. The bottom part of the screen encloses several elements with various functions {a}—the button for navigation between the modes is aligned together with the undo button, a button that automatically aligns the remaining potential mappings and a label (*Remaining Suggestions*) that provides information for them. This label is actually a link which lists all remaining suggestions but it does not look clickable {d}. The same issue appears with the *History* label which is a link as well. It presents the sequence of user accepting/rejecting/creating actions {d}. A warning box next to the *History* label shows a message relevant to the previous action. While the message is explanatory {h} it appears after the action took place.

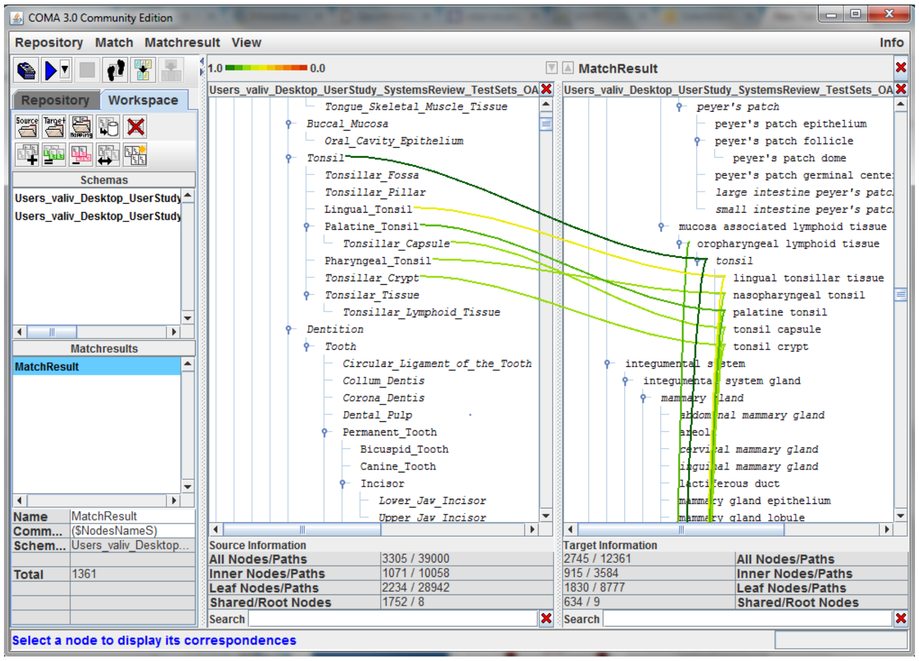
The window is divided similarly in the *Align Manually* mode (figure 3). The top and central parts contain both ontologies represented as unmodifiable indented trees, the comment box is below them together with a search field (discussed below). The concepts that participate in validated mappings are marked with 'M' in both trees. The buttons for creating mappings are aligned with the undo button (placed on the other side of the screen) and their labels look differently than in the other mode {d}. The bottom part is the same as in the other mode without the elements related to the mapping suggestions.

As mentioned already the system provides an undo button but it does not show a confirmation dialog for possibly slowly reversible actions (such as those connected to the *Align Remaining* or *Lock Session* buttons) {f, i}. It lists the sequence of accepted, rejected and created mappings in the history and provides comments even for the rejected mappings {c}. It helps the user to set up the alignment process by providing a choice between several matchers and default strategies {g}. Some of them are explained in the help which is a bit outdated {j}. The system does not show the alignment process configuration after the process has been run or the similarity values for the mapping suggestions {c}.

The search function has several issues—it is case sensitive, accepts only exact input (no autocomplete or correction) and it should be activated by the search button next to the text field (pressing Enter on the keyboard does not work) {g}. The search reloads both trees and loses the current selection. It does not jump to hit and highlights only the first match in the hierarchy {g}.

COMA 3.0 is a desktop system which provides one view during the alignment process [30]. Its user interface is depicted in figure 4. A standard toolbar contains four drop down menus for importing/exporting information to/from the repository (*Repository*), configuring and executing the matching process (*Match*), set operations over alignments (*Matchresult*) and additional information for the current alignment (*View*). More details regarding their functions are presented in section 3. Most of the screen space is occupied by the two ontologies which are placed side-by-side in separate scrollable windows. Several labels below each ontology show statistical information regarding its structure which is not directly related to the ontology alignment task {a}. As a concept is selected the labels are updated to show the concept name and path to it in the hierarchy. The labels for both ontologies are connected through small colored squares. Their colors resemble mappings color-coding but no explanation what they represent is given {a}. Search boxes are available for each of the ontologies. Selected functions from the toolbar menus are available through buttons in the resizable left side of the screen {c}. Some basic information for the current alignment is constantly presented to the user although it is not constantly needed {a}.

There is no explicit distinction between validated and potential mappings as there is in the other two systems in the user interface evaluations {c}. In our opinion the list with calculated mappings in COMA 3.0 is closer to (and thus considered as) mapping suggestions, since the users go through it and choose which of them (not) to keep in the final alignment. If a concept in a mapping is selected the system automatically shows the other concept in the mapping if it is under an unfolded branch {g}. The user



Name	MatchResult
Comm...	($NodesNameS)
Schem...	Users_valiv_Desktop...
Total	1361

Source Information

All Nodes/Paths	3305 / 39000
Inner Nodes/Paths	1071 / 10058
Leaf Nodes/Paths	2234 / 28942
Shared/Root Nodes	1752 / 8

Target Information

2745 / 12361	All Nodes/Paths
915 / 3584	Inner Nodes/Paths
1830 / 8777	Leaf Nodes/Paths
634 / 9	Shared/Root Nodes

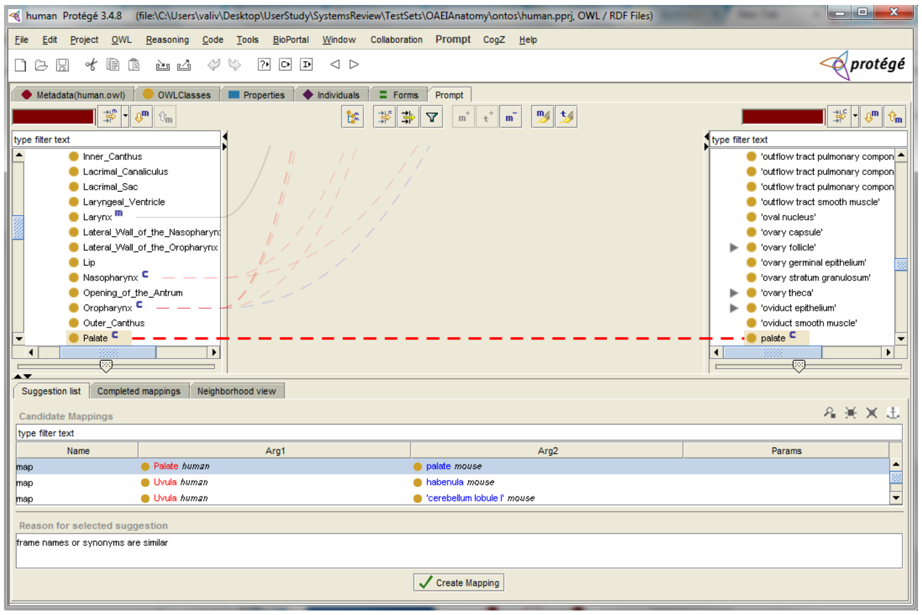
Select a node to display its correspondences

Fig. 4. COMA 3.0. [30].

cannot select a mapping. All actions for a mapping are available through its concepts' context menus {d}. The concept context menu contains (un)folding options, a show instances option and all mappings related actions. To achieve more intuitive interaction the mappings should be selectable and the corresponding actions should be available in the mapping context menu (currently not existing) {d}. Actions available for a single mapping include *Create Correspondence*, *Delete Correspondence* and *Set Highest Similarity Value*. The last action is only available for mappings computed by the system and carries the 'validate mapping' semantics, i.e., the user wants to preserve this correspondence in the final alignment. However its phrasing significantly differs from the phrasing of the other two {b, d}.

Since there is no explicit difference between validated and potential mappings the user needs to remember the mappings he had already visited {c}. There is a list with mappings but it only shows their current confidence values and not if they have been changed or existed before. The system provides default matchers' configuration {c, g}.

Messages at the bottom left corner constantly inform the user what it is going on in the system {e}. Messages to explain/hint for actions also appear there. A progress bar at the bottom right corner is provided for time-consuming actions. The system provides neither undo nor cancel buttons or confirmation messages {i, f}. Such could be very useful in connection with the 'X' button (*Delete selected Schema or Matchresult from Workspace*). Tooltips are supported for many of the buttons, explicit help is missing {j}.


human Protégé 3.4.8 (file:\C:\Users\valiv\Desktop\UserStudy\SystemsReview\TestSets\OAEIAnatomy\ontos\human.pprj, OWL / RDF Files)
File Edit Project OWL Reasoning Code Tools BioPortal Window Collaboration Prompt CogZ Help
protégé
Metadata(human.owl) OWLClasses Properties Individuals Forms Prompt
type filter text
Inner_Canthus
Lacrimal_Canaliculus
Lacrimal_Sac
Laryngeal_Ventricle
Larynx
Lateral_Wall_of_the_Nasopharynx
Lateral_Wall_of_the_Oropharynx
Lip
Nasopharynx
Opening_of_the_Antrum
Oropharynx
Outer_Canthus
Palate
type filter text
'outflow tract pulmonary compon
'outflow tract pulmonary compon
'outflow tract pulmonary compon
'outflow tract smooth muscle'
'oval nucleus'
'ovary capsule'
'ovary follicle'
'ovary germinal epithelium'
'ovary stratum granulosum'
'ovary theca'
'oviduct epithelium'
'oviduct smooth muscle'
palate
Suggestion list
Completed mappings
Neighborhood view
Candidate Mappings
type filter text
Reason for selected suggestion
Create Mapping

Name	Arg1	Arg2	Params
map	Palate *human*	palate *mouse*	
map	Uvula *human*	habenula *mouse*	
map	Uvula *human*	'cerebellum lobule I' *mouse*	

frame names or synonyms are similar

Fig. 5. The latest version of CogZ.

The search function has several issues—the scroll bar shows pink markers where the search results appear but there is no jump to hit {g}. Only concepts under expanded branches are considered during the search. It is not case sensitive and works with part of the input.

CogZ is a Protégé plug-in which extends the PROMPT user interface and reuses the rest of its components. Its interface, presented in figure 5, is more complex than those in the previous two systems. The screen is divided into two major views—each side of the upper part contains an ontology represented as a unmodifiable indented tree; the space between them is occupied by their mappings; the bottom part contains three tabs. Both views are resizable and can be completely removed from the screen. The mappings are presented as lines which run through the middle part of the screen and connect concepts in both ontologies. The lines can be selected and have a tooltip but do not have a context menu {d}. The tooltip has different content depending on the type of the mapping {c}—for a potential mapping it shows the reason the system has suggested it; for temporary and validated mappings it shows the mapped concepts. Several buttons above the mappings are used to apply different functions to them—create(m^+)/remove (m^-) (temporary, t^+) mapping, mark as mapped/temporary. The mark as mapped/temporary and m^- buttons apply actions on potential mappings while m^+, m^- and t^+ are used to add, delete and add temporary mappings. m^- is placed in group with m^+ and t^+ and at a distance from mark as mapped/temporary (it also looks differently from them) {d}. Four buttons that apply different filters on the mappings are aligned with these above. They have different icons but two of them have the same tooltip. The potential, validated and temporary mappings are depicted in dashed red, thick blue and dashed blue lines respectively. They appear thinner when they are not selected. The concepts in these mappings are marked with 'C', 'M' and 'T' after the concept name in the ontology trees. There is a search box above each ontology and a red-green progress bar which shows the state of the process {e}, i.e., what portion of the mappings for this ontology are validated. Next to the progress bar a toggle button filters the ontology according to the different mappings which are selected from a drop-down list next to the filtering button. Next to it two buttons navigate back and forth through the mappings.

The first tab in the bottom part of the screen contains a table with all potential mappings. When a potential mapping is selected in the table it is also highlighted in the upper view (if filtering is not applied there) {g}. There is a search field on top of the table. It continues through the entire screen and it is activated on a key press {g}. The user can sort the mappings by a concept name by clicking at the column headers (however nothing hints that the column headers are clickable) {c}. Four buttons on top of the search strip and at the far right corner apply actions on a single potential mapping. They are almost unnoticeable due to their distance, color, unfamiliar icons and tooltips (view/create/remove operation) {a}. Remove operation removes a potential mapping from the list, view and create operation open the same dialog box which is prefilled with the concepts in a potential mapping or empty depending on the choice of operation. At the same time there is a *Create Mapping* button at the very bottom of the window which is much more visible than these four (the fourth runs a mapping algorithm) but it does not show the same dialog. Moreover a double click on a potential mapping opens the view operation dialog. So it seems to us that this part of the screen contains three ways to validate a mapping (view operation, *Create Mapping* button and double click) {d}. The three operation buttons could be moved down to the *Create Mapping* button or in a context menu for a potential mapping (currently not existing) {d}. The *Create Mapping* button attracts attention even when the user is working at the upper part of the screen. This is due to its size, the size of the buttons (smaller) at the top of the upper view and probably because of the unclear separation of both views. In short the system provides several buttons looking differently with different tooltips which look like they are meant for the same two actions, i.e., validate and create a mapping {c, d}. At the bottom a resizable window shows the reason why the system has calculated the mapping {c, e}. The second tab shows the completed mappings and is also synchronized with the upper view as the potential mappings tab {c, d, g}. The third tab contains two parts, each of them shows the neighborhood (parents and children) and the selected concept (and the different relationships between them) in each of the ontologies above. They are visualized as graphs where the layout can be changed with buttons above the view or by dragging. The different types of nodes and relationships can be switched on and off {a, g}.

While running the alignment algorithms the system shows a progress bar and the red-green progress bar shows the proportion between the validated and potential mappings {e}. Several times while working with the system it became unresponsive or froze for a while without any clear sign that the action in question is (successfully) performed {e}. Sometimes the actions had to be repeated in order to appear in the interface. Mean-while exceptions are thrown in the console but nothing shows on the screen that there is a problem or how it could be resolved {h}.

The system does not provide help but there are tooltips helping the user to understand the purpose of the buttons {j}. Most of the actions assigned to buttons can be canceled but there is no undo button {i, f}.

The system provides carefully designed search functionality—it filters away the concepts which do not match the search criteria and jumps to the first hit {g}. The concept names consisting of more than a word and including space are enclosed in a single quote ('). When searching for such concepts the users have to use the same character at the beginning of the input or '*' which replaces an arbitrary number of characters.

4.2 Observational User Study

We conducted an observational user study in order to achieve better understanding of how the systems support the requirements in the user interface category. According to Nielsen [33] *"User testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested."*. We describe the study design, the participants and show its results. The tasks together with their answers, participants' remarks, task time and success can be found in [2].

Procedure and Participants 8 participants took part in the study—3 master and 5 PhD students (7 male, 1 female). All had Computer Science background and none of them has shown a particular interest in the Semantic Web area. All participants have acquired basic ontology engineering knowledge as part of ongoing or past university courses. Each participant performed between 11 and 17 tasks with the systems (since not all of the systems supported all of the requirements). The study was scheduled for 2 sessions, which lasted for 2 hours (with a break after 1 hour) and 1 hour, respectively. It was expected that a user would work with each system for approximately 1 hour. To prevent carry-over effects (learning) between the systems we counterbalanced the order in which they were presented to the users. We also used a different order of the input ontologies.

We were particularly interested in how the requirements in section 2 are supported in a large-scale setting. Thus we used the two ontologies from the Anatomy track from the OAEI 2014—AMA (2,737 concepts, 1,807 asserted is-a relations) and NCI-A (3,298 concepts, 3,761 asserted is-a relations) as representatives of the smallest use cases in a large-scale setting.

The study was conducted as follows. Each participant was presented with a project introduction and a tutorial during the first session. The tutorial provided basic knowledge about ontologies and ontology alignment and ended with several small tasks in order to ensure that all participants possessed the same level of understanding of the area. This took around 10 minutes per participant. The tutorial was available to the participant at the beginning of all sessions. After that the participants started solving the

Table 3. User study tasks.

Task	Requirement
A. Discard following potential mapping.	#2.2, 1.2
B. Count mapping suggestions for X in A and Y in B.	#2.2
C. Find ONE parent and child for X in A and Y in B.	#2.1/4, 1.1/4, 4.1/2/3
D. Keep following potential mapping.	#2.2, 1.2
E. Create following mapping.	#2.5, 1.1
F. Count ALL parents and children of X in A and Y in B.	#2.1/4, 1.1/4, 4.1/2/3
G. Find in the system why/how it has suggested potential mapping between X in A and Y in B.	#4.2, 4.7
H. Set up the system to display ALL concepts in potential mappings.	#2.4
I. Find a concept that has nearby children and/or parents with more than 10 potential mappings.	#4.4
J. Give estimation of the validated mappings.	#4.6
K. Write in the system your arguments to decide there is a mapping between X in A and Y in B.	#2.5
L. Record in the system the mapping between X in A and Y in B is correct, such that you can change your current decision.	#1.1, 5.9.2
M. Give estimation of the potential mappings for validation.	#4.6
N. Set up the system to display ALL concepts in verified mappings.	#2.4
O. Find in the system why the mapping X in A and Y in B was created/accepted.	#4.3
P. Show in the system ALL concepts for which you may change your decision.	#2.4

tasks with a particular system. Before the first task with each system the participants received the same hints on how to use search (since there are issues in all three systems). They were observed by one of the authors who took notes regarding their actions on the screen and their comments after each task and regarding the systems. When a participant gave a wrong answer, the observer provided the right answer. In three cases the first system took more time than expected, thus an additional third session was scheduled.

Tasks Design The tasks in the study were developed to include as many of the requirements in the user interface category as possible. Most of the requirements in the infrastructure and algorithms category were not covered due to their limited support in the systems and since they would require significantly longer sessions and domain knowledge. A brief description of the tasks and the corresponding requirements are listed in table 3. Some tasks were performed twice since we were interested in their subsequent execution times. Since the systems do not support all requirements not all tasks were possible with every system. Task success and task times were collected for each task. The participants filled in the System Usability Scale (SUS) [5] questionnaire after all tasks with one system were completed. They were asked to provide at most three things that they like and dislike after working with each system as well.

Table 4. Number of participants (max 8) successfully completed a task (details in the text (*)).

System/Task	A	B	C	D	E	F	G	H	I	J	K	L	M	D	A	N	O	E	P
SAMBO	8	1	5	6	7	4	n/a	n/a	7	5	8	n/a	8	6	8	n/a	6	7	n/a
COMA 3.0	2	2	7	2*	8	4	*	n/a	6	n/a	n/a	n/a	n/a	8	8	n/a	n/a	8	n/a
CogZ	7	4	8	8	3	4	7	5	8	8	n/a	4	8	8	7	8	n/a	7	8

Results Table 4 shows the number of participants that successfully completed each of the tasks per system. The tasks not applicable to a system are denoted with n/a. Although we collected time per task and task success for task G (*) in COMA 3.0 we use this to understand how the users perceive the similarity value rather than to provide comparison with CogZ.

The first 6 tasks were solved with varying success by the participants. Most participants (4 out of 6) who did not complete task A in COMA 3.0 chose a wrong option to reject the mapping—instead of selecting both concepts in a mapping and choosing *Delete Correspondence* from one of the concepts context menu they used the 'X' button which deletes the entire alignment. The participant who did not solve task A in CogZ could not find the mapping but after help from the observer he was able to solve it. The success in task B varied due to different reasons. For SAMBO most users (4 out of 7) could not find where the mapping suggestions are listed. They had to open a separate link, however the link looks like a label. For COMA 3.0 the users provided wrong numbers due to not realizing that a concept may exist in several places in the tree and as a consequence several lines represent a mapping between the same two concepts. For CogZ 2 participants did not understand the task and 2 gave wrong numbers since they were counting the suggestions between the two ontologies while one of the ontologies was filtered because of previous search. Most of the users that did not solve task F (all systems) did not realize that a concept may appear several times in the hierarchy although this was hinted in the task description and a similar situation appeared in task B. Task E in CogZ was not solved since 2 participants had problems finding one of the concepts, 1 participant did not realize that this is not a mapping suggestion and looked at the mapping suggestions list (after help from the observer he still had problems finding it). As mentioned earlier there is no explicit separation between mappings and mapping suggestions in COMA 3.0. Thus the way task D (*) is interpreted is that the user keeps the mapping if he chooses *Sets Highest Similarity Value*. In 3 out of 6 cases the participants selected *Retain only Fragment Correspondences*.

Table 5 shows the average time per task per system. The task times for task A (*) in SAMBO are not directly comparable with the other systems due to the system's design and study scenario. While the user has to search for a mapping suggestion in COMA 3.0 and CogZ and then delete/remove it in SAMBO the suggestion was presented to the user (due to the system design). Task A (*) in CogZ took much longer for one of the participants. The average time for this task is 1:35 min if we exclude his time from the results. The task success and time improved significantly for the subsequent execution of tasks A, D and E.

Table 5. Average task time per system in minutes (details in the text (*)).

System/Task	A	B	C	D	E	F	G	H	I	J	K	L	M	D	A	N	O	E	P
SAMBO	0:30*	4:14	3:11	1:16	2:29	4:25	-	-	1:58	0:28	0:06	-	0:01	0:21	0:09*	-	0:47	1:07	-
COMA 3.0	2:54	3:03	1:39	1:33	0:41	4:03	*	-	1:04	-	-	-	-	0:34	0:31	-	-	0:25	-
CogZ	2:47*	1:27	1:37	0:44	2:08	1:48	0:40	0:37	1:04	0:06	-	1:45	1:42	0:29	0:17	0:11	-	0:38	0:05

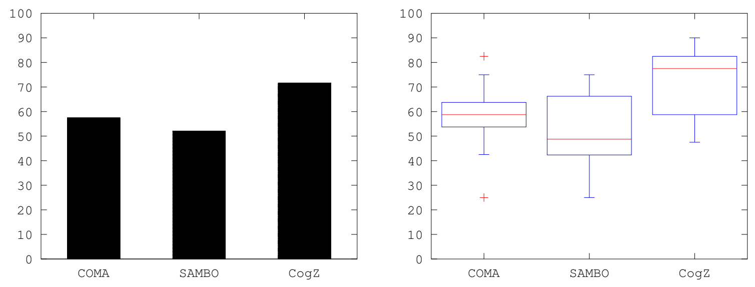


Fig. 6. SUS questionnaire scores. (average (left) and boxplot of the dataset (right))

Figure 6 shows the results of the SUS questionnaire.

4.3 Discussion

We discuss the results of the user study and heuristic evaluation in connection with the requirements from table 2 in Section 3. Slight differences in the supported requirements could be in place since table 2 is based on a literature review and table 3 is based on the most recent available versions of the systems (not necessarily the same as in the literature review). Further we present additional important findings.

Requirements Coverage Tables 3 and 4 show that most of the requirements in the manipulation, inspection and explanation categories from table 2 are covered by at least one of the systems and most of them are covered by all three. Mappings can be created manually in all three systems {#2.5, 1.1}. As already discussed COMA 3.0 does not explicitly support mapping suggestions. In our opinion the list with calculated mappings in COMA 3.0 is closer to (and thus considered as) mapping suggestions, since the users go through it and choose which of them to keep in the final alignment and which not. It supports modifying the similarity value of a mapping which can be seen as accepting/keeping the mapping {#2.2, 1.1}. CogZ and SAMBO support accepting/rejecting mapping suggestions. Only SAMBO among the three supports adding metadata to a mapping {#2.5}. None of the systems support editing of the ontologies.

Although there was not a dedicated task related to the inspection of the verified mappings {#2.3} task J required browsing the table with completed mappings in CogZ and the history list in SAMBO. The navigation through the table in CogZ is synchronized with the tree visualization of the ontologies. The history list in SAMBO is static and verified mappings cannot be removed. In COMA 3.0 the user can obtain a list of the current mappings but without any information if they had been changed earlier or existed before. The mapping suggestions can be inspected in both CogZ and SAMBO with some differences {#2.2, 1.4}. SAMBO provides an unordered list and a view (*Suggestion Align*) and CogZ provides a sortable list synchronized with the tree representations of the ontologies. Regions with many mappings {#4.4} can be identified easily in COMA 3.0 and CogZ through the number of lines connecting the ontology trees and more difficult in SAMBO where the user should go through the list with mapping (suggestions) and separately browse the ontologies hierarchy. CogZ provides several filters {#2.4} for potential, temporary and validated mappings and filters the ontology trees during search. Filters are not supported by COMA 3.0 or SAMBO. As discussed earlier CogZ also provides carefully designed search {#2.4} which *jumps to hit*, while the search in SAMBO and COMA 3.0 is not functioning that well. All systems covered in the study represent the ontologies as unmodifiable indented trees. The hierarchy of the ontologies can be explored in all three systems however in SAMBO this can not be done in the *Suggestion Align* view {#4.1, 4.2, 4.3, 2.1, 1.1}. Only CogZ provides access to the full definitions of the terms in an ontology {#1.3}.

SAMBO is the only system that provides the possibility for an explanation why a mapping has been accepted and even rejected since it is the only one that provides annotations for a mapping {#4.3}. Only CogZ provides textual human readable explanation why/how a mapping has been suggested {#4.2, 4.7, 5.8}. COMA 3.0 provides the calculated similarity value on top of the link however it was not directly perceived as an explanation by the users (task G), several of them were looking for an explicit explanation. Three of the participants stated that if they would know the matching algorithm they would know the reason. The user can observe the state of the matching process {#4.6} by the number of mapping suggestions left and validated mappings in CogZ and SAMBO. A progress bar is provided in CogZ for this purpose as well. It was not used by the participants since the number of mapping suggestions they had been working on was too small to be marked on it. The filtering of the ontologies according to potential/validated/temporary mappings can be used for the same purpose. As mentioned several times COMA 3.0 users can only see the current similarity values but not how they have changed. Temporary decisions are supported only by CogZ {#1.1, 5.9.2}.

It comes at no surprise that most of the tasks are supported in the CogZ system since they are based on the requirements in the user interface category which are based on [15]. There were several tasks for some of the requirements and they were not performed in the systems that do not support them. For instance tasks H, N and P are related to the filtering part of #2.4. Since both SAMBO and COMA 3.0 do not support filtering these tasks were not performed with them.

Additional Observations and Discussions In what follows P followed by a number indicates the participant who made the remark. The other letters indicate the task.

Several issues became clearly noticeable while observing the users performing the tasks and in their comments after each task. In several cases the users could not complete a task or gave a wrong answer because they could not find the concept they were looking for because of the visualization. In task F performed with COMA 3.0 P2 counted only the concepts with children as children of a concept. In three cases in COMA 3.0 with three different users (C.P1, F.P3, F.P4) the participants counted a sibling concept as a parent concept although the tree in COMA 3.0 is visualized with guiding lines on all levels. One participant (P5) stated that the guiding lines helped him a lot during counting (task F). Another participant (P2) commented after task C that the tree representation in COMA 3.0 is better than CogZ because of the guiding lines. P6 also explained after task C that he followed the guiding lines to see which is parent and which is child. P1 counted one parent twice in CogZ and commented (after task F) that it is harder to see if the parents are the same because of the distance. P4 stated (after task F) that it was easier to see the parents in CogZ in comparison with COMA 3.0. Although the trees are collapsible in all systems P7.C said that in CogZ it was easy to find parents because the nodes can be contracted (CogZ was the last system for P7). P6.C said regarding the trees in the SAMBO system that he had problems aligning which is a parent and which is a child. The same was true for P8.F but he had expanded the entire ontologies and did not collapse any branch. P4.C and P2.F used a pen to align the concepts.

Another issue appeared around the tree representation of the ontologies. The participants had to consider multiple inheritance, i.e., the same concept appears several times under different parents (and thus places) in an ontology, for task B (COMA 3.0) and F (all three). An example of multiple inheritance was given in the tutorial as well. Two participants did not experience difficulties with that but only one (P7) of them managed to solve all F tasks correctly. The other one (P4) did not succeed because he counted a sibling concept as a parent in one of the systems and due to a search problem in another. P7 managed to solve all F tasks and explicitly noted that he had an ontology engineering course and due to that he looked for another appearance of a concept. He had solved task B as well and commented that representing a mapping in the tree several times (when a concept has several parents) is how this representation works, but it was confusing for him. All other participants did not think of searching for more than a single occurrence of a concept. While some of them did not make the same mistake again (P6, P8) others did it in the B and after that in the F tasks with the same system. P6 commented twice he remembered he had done a mistake regarding multiple inheritance.

As commented in the heuristic evaluation the search functionality was tricky and due to it several tasks were not successfully completed. In three cases that happened with SAMBO (C.P2, E.P7, F.P4). All participants complained about different aspects of it in the like/dislike section. 5 participants complained about the search functionality in COMA 3.0 as well. 3 liked the pink markers although 2 of them complained about other aspects of the search. Only P5 was satisfied with the search functionality and said that it worked well. Although CogZ provided the best search functionality among the three tested systems P1 and P4 did not solve task E because of search problems as well. The same task was solved by P6 but it took him long time. To solve this task the users had to find the *upper leg* concept in the AMA ontology. When searching for this concept the user has to input * or ' at the beginning. It appeared last in the search results while *upper* appeared many times before. Despite this issue CogZ search and filtering functions were greatly appreciated by the participants.

Another issue that constantly appeared across all systems was the terminology. It is also covered by the second Nielsen heuristic [33] in section 4.1. While it should be noted that the participants are not regular ontology alignment systems users all of them have had an ontology engineering course. Thus the terminology is not completely new for them. In the tutorial and task descriptions we tried to avoid any of the terms that appeared in the systems. *Mapping* was an exception since it is general and we could not find a more suitable word to represent its meaning. We also avoided terms that can be considered hints by the participants (accept/reject/remove/delete etc.). SAMBO uses the term *mapping suggestions* for potential mappings, *Suggestion Align* for the mode where potential mappings are accepted/rejected and *Remaining Suggestions* for the list with mappings left for validation. Two users were unsure what *Suggestion Align* means and does[1]. Two other participants complained about the other two terms while performing different tasks. The term *correspondence* is used in COMA 3.0 to denote a mapping and *Match Result* to denote the alignment. P1.A and P6.A (COMA 3.0 was their second system) complained that *correspondence* is not something they expect and P1 explicitly said that he got used to the terminology in the system before (CogZ). P6 mentioned that he does not know what *Set Highest Similarity Value* and *Retain only Fragment Correspondences* mean and P8 also added that the words in the concept context menu did not sound familiar. It was observed that the users hesitated to press *Set Highest Similarity Value*. While the terminology issues in COMA 3.0 and SAMBO were caused by (we would generalize) little familiar words, the terminology issue in CogZ had another aspect. In the former case the search took significant part of the task times but once it was done the users were quick to choose what to press (here we mainly consider the A, D and E tasks). In CogZ the search was quicker but it was observed that the users were not confident in choosing actions (this could, however, be equally attributed to the fact that in CogZ create/accept/reject could be performed in different ways). As said earlier CogZ has *Mark as mapped*, m^{+}, *Create Mapping*, *View operation*, *Create operation* and a *View operation* dialog which opens on double click on a potential mapping. The users were unsure of using *Mark as mapped* in at least four cases. P6 was not sure what *Remove operation* does and P8.D, P4 and P1.D said they were wondering which button to use. P4 even stressed he did not like that there are many buttons for the same thing in CogZ.

The like/dislike section gave the users possibility to express at most three thing they like and dislike regarding the systems. There were also other observations which did not appear with the same frequency as these above. We list them briefly here for each system. One of the most appreciated features in SAMBO was the *Suggestion Align* view. *Remaining suggestions* and *History* were also explicitly mentioned although lists with potential/completed mappings are presented in CogZ as well. SAMBO was the only system that provided help documentation (although outdated) but it was used by three participants. One of them mentioned that there he saw how the things look like. Apart

[1] The button opens the *Suggestion Align* mode in SAMBO. However it looked like that it shows the suggestions for a concept selected in a tree in the *Align Manually* mode (because of the button's place on the screen).

from the search and terminology the users also disliked that the potential mappings were not shown in the *Align Manually* mode. Several users commented that it was not obvious that the *Remaining suggestions* and *History* labels are actually links.

In COMA 3.0 the users liked the mapping representation—color-coded lines between the trees placed side-by-side. Many of the users tried to select a mapping by clicking on it and were also looking for a context menu. One disliked that the mapping context menu actually appeared for a concept. This comment can be juxtaposed to heuristic $\{\mathrm{d}\}$ in subsection 4.1 which suggests that common conventions should be followed.

The *Neighborhood View* in CogZ appeared as one of its advantages. Many participants found it (while solving tasks not related to it) but did not use it at all. It could have been very useful for task F discussed in detail earlier. The users expected a context menu in the table with potential mappings as well. During the first task several users were confused because it was not clear which ontology is presented on which side of the screen. In order to figure out one of them used the difference in the concept names and another the potential mappings list. One user stated that the button *Create Mapping* draws attention and that the two views are not well separated inline with one of the points from the heuristic evaluation. Comparing the three systems CogZ was most unstable in the sense that it was not clear if an action took place or the user has to repeat an action in order to see it in the interface. This was the only thing ('bugs') one of the participants mentioned in his dislike section.

It comes at no surprise that most of the tasks are supported in CogZ since they are based on the requirements in the user interface category which are based on [15]. As shown in tables 2, 3 and 4 SAMBO and COMA 3.0 cover fewer requirements. The explanation category is the least supported user interface category inline with the literature study. As it can be seen from the task success and time the users showed varying performance at the beginning which improved in terms of success and decreased in time to the last tasks. CogZ achieved the highest SUS score from the three (Fig. 6) which falls at the border between *OK* and *GOOD* in the adjective rating scale in [6]. COMA 3.0 scored a bit higher at SUS than SAMBO, both at the beginning of the *OK* interval. *OK* should not be perceived as satisfactory but rather that improvements are needed. SUS provides a good assessment of the perceived usability of a system with a small sample as in our case and SUS scores have "modest correlation with task performance" [6]. As take away issues from this study we would pinpoint the search and filter functionality especially in a large-scale context, explicit explanation of the matching results (reduces the users cognitive load) and the *Suggestion Align* mode which was appreciated by the users.

5 Conclusions and Future Work

We have developed and presented requirements to foster user involvement in large-scale ontology alignment problems and have conducted a user study to reveal to what extent the requirements in the user interface category are supported in three selected systems. A heuristic evaluation was conducted by one of the authors as well. This provided additional critique to the systems interfaces and covered aspects slightly or not mentioned in the user study (such as positioning of the elements on the screen). We also showed that the heuristic evaluation can provide quick yet valuable feedback for the user interface design.

The literature study showed that the requirements in the infrastructure and algorithms category are supported to a varying degree and more research and support is needed in, e.g., sessions, reducing user intervention, collaboration and trial execution. The explanation category, which assists the users most in understanding the reasons for suggesting/accepting mapping suggestions, is the least supported from the user interface categories. The user interface evaluations show that state-of-the-art ontology alignment systems still have many weaknesses from a usability point of view. The study highlighted the importance of seemingly trivial issues like search and issues like ontology visualization which become crucial in a large-scale setting. Regarding our study, one limitation, that needs to be addressed in future work, is that all systems in the interface evaluations represent ontologies as trees. It was shown in [16] that a graph representation may be more suitable when dealing with multiple inheritance.

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

References

1. ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. `http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883`. Accessed: 2015-03-20.
2. Materials from the observational user study. Available at `http://www.ida.liu.se/~patla/publications/ESWC15/`.
3. D Aumüller, H H Do, S Maßmann, and E Rahm. Schema and ontology matching with COMA++. In *SIGMOD*, pages 906–908, 2005.
4. S Bail, B Parsia, and U Sattler. Declutter Your Justifications: Determining Similarity Between OWL Explanations. In *Proceedings of the 1st International Workshop on Debugging Ontologies and Ontology Mappings (WoDOOM 2012)*, volume 79 of *LECP*, pages 13–24, 2012.
5. J Brooke. SUS: A quick and dirty usability scale. In *Usability Evaluation in Industry*. 1996.
6. J Brooke. SUS: A Retrospective. *J. of Usability Studies*, 8(2):29–40, 2013.
7. I F Cruz, F P Antonelli, and C Stroe. Agreementmaker: Efficient matching for large real-world schemas and ontologies. *Proc. VLDB Endow.*, 2(2):1586–1589, 2009.
8. I F Cruz, F Loprete, M Palmonari, C Stroe, and A Taheri. Pay-As-You-Go Multi-user Feedback Model for Ontology Matching. In K Janowicz et al., editor, *EKAW*, volume 8876 of *LNCS*, pages 80–96. 2014.
9. I F Cruz, C Stroe, and M Palmonari. Interactive user feedback in ontology matching using signature vectors. In *ICDE*, pages 1321–1324, 2012.
10. B Cuenca Grau et al. Results of the ontology alignment evaluation initiative 2013. In *OM*, pages 61–100, 2013.
11. H H Do. *Schema Matching and Mapping-based Data Integration*. PhD thesis, 2005.
12. Z Dragisic et al. Results of the ontology alignment evaluation initiative 2014. In *OM*, pages 61–104, 2014.
13. J Euzenat and P Shvaiko. User Involvement. In *Ontology Matching*, pages 353–375. 2013.

14. S M Falconer, N F Noy, and M A Storey. Towards Understanding the Needs of Cognitive Support for Ontology Mapping. *Ontology Matching*, 2006.
15. S M Falconer and M A Storey. A Cognitive Support Framework for Ontology Mapping. In K Aberer et al., editor, *ISWC/ASWC*, volume 4825 of *LNCS*, pages 114–127. 2007.
16. B Fu, N F Noy, and M A Storey. Indented Tree or Graph? A Usability Study of Ontology Visualization Techniques in the Context of Class Mapping Evaluation. In H Alani et al., editor, *ISWC*, volume 8218 of *LNCS*, pages 117–134. 2013.
17. 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.
18. 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.
19. V Ivanova and P Lambrix. A unified approach for aligning taxonomies and debugging taxonomies and their alignments. In *The Semantic Web: Semantics and Big Data*, volume 7882 of *LNCS*, pages 1–15. 2013.
20. E Jiménez-Ruiz and B C Grau. Logmap: Logic-based and scalable ontology matching. In *The Semantic Web ISWC 2011*, volume 7031 of *LNCS*, pages 273–288. 2011.
21. 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.
22. 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.
23. A Katifori, C Halatsis, G Lepouras, C Vassilakis, and E G. Giannopoulou. Ontology visualization methods - a survey. *ACM Computing Surveys*, 39(4), 2007.
24. T Kirsten, A Gross, et al. GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *J. of Biomedical Semantics*, 2:6, 2011.
25. P Lambrix and V Ivanova. A unified approach for debugging is-a structure and mappings in networked taxonomies. *J. of Biomedical Semantics*, 4:10, 2013.
26. P Lambrix and R Kaliyaperumal. A Session-Based Approach for Aligning Large Ontologies. In P Cimiano et al., editor, *ESWC*, volume 7882 of *LNCS*, pages 46–60. 2013.
27. P Lambrix and H Tan. SAMBO - a system for aligning and merging biomedical ontologies. *J. of Web Semantics*, 4(3):196–206, 2006.
28. 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.
29. M Lanzenberger, J Sampson, M Rester, Y Naudet, and T Latour. Visual ontology alignment for knowledge sharing and reuse. *J. of Knowledge Management*, 12(6):102–120, 2008.
30. S Massmann, S Raunich, D Aumüller, P Arnold, and E Rahm. Evolution of the COMA match system. In *OM*, pages 49–60, 2011.
31. D Ngo and Z Bellahsene. Yam++ : A multi-strategy based approach for ontology matching task. In *EKAW*, volume 7603 of *LNCS*, pages 421–425. 2012.
32. T A T Nguyen, R Power, P Piwek, and S Williams. Predicting the Understandability of OWL Inferences. In *Proceedings of the 10th European Semantic Web Conference (ESWC 2013)*, volume 7882 of *LNCS*, pages 109–123. 2013.
33. J Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1993.
34. N F Noy and M A Musen. Algorithm and Tool for Automated Ontology Merging and Alignment. In *AAAI*, pages 450–455, 2000.
35. N F Noy and M A Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *J. of Human-Computer Studies*, 59(6):983–1024, 2003.
36. L Otero-Cerdeira, F J Rodríguez-Martínez, and A Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971, 2015.
37. H Paulheim, S Hertling, and D Ritze. Towards Evaluating Interactive Ontology Matching Tools. In P Cimiano et al., editor, *ESWC*, volume 7882 of *LNCS*, pages 31–45. 2013.

38. T R Payne and V Tamma. A Dialectical Approach to Selectively Reusing Ontological Correspondences. In K Janowicz et al., editor, *EKAW*, volume 8876 of *LNCS*, pages 397–412. 2014.
39. 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.
40. E Rahm. Towards Large-Scale Schema and Ontology Matching. In Z Bellahsene et al., editor, *Schema Matching and Mapping*, pages 3–27. 2011.
41. F Shi, J Li, J Tang, G Xie, and H Li. Actively Learning Ontology Matching via User Interaction. In A Bernstein et al., editor, *ISWC*, volume 5823 of *LNCS*, pages 585–600. 2009.
42. B Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343, 1996.
43. P Shvaiko and J Euzenat. Ontology Matching: State of the Art and Future Challenges. *Knowledge and Data Engineering*, 25(1):158–176, 2013.
44. P Warren, P Mulholland, T Collins, and E Motta. Using ontologies. volume 8876 of *LNCS*, pages 579–590. 2014.