

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

Valentina Ivanova<sup>1,2</sup>, Patrick Lambrix<sup>1,2(✉)</sup>, and Johan Åberg<sup>1</sup>

<sup>1</sup> Department of Computer and Information Science,  
Linköping University, 581 83 Linköping, Sweden  
`patrick.lambrix@liu.se`

<sup>2</sup> 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 [29] 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 [25]. The results from the first two editions of the track 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 [13] 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 [29] 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. 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 [13]. According to [27] 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.

This paper provides requirements for ontology alignment tools that encourage user involvement for large-scale ontology alignment tasks (Sect. 2). We also present the results from a literature study (Sect. 3) and two user interface evaluations (Sect. 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. We extend the requirements in [8] 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. 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. They are extracted from existing works and systems and from the authors' personal experience from developing ontology alignment and debugging systems [16–18]. 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. We note that the requirements in [10] and [7] may also be seen as subsets of the requirements in this paper.

The requirements identified in [8] are based on research in the area of cognitive theories. They are grouped in four conceptual dimensions (Table 1). The *Analysis*

**Table 1.** Cognitive support requirements adapted from [8].

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

*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. 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. The requirements in this dimension can be considered to utilize the functionalities represented by the requirements in the *Interaction* dimension—during which 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.

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 needed for the ontologies (#5.0), as well, since the user may need to, for instance, introduce a concept in order to provide more accurate mappings, as described in [20] as well. Those in the second category cover a broad set of actions for inspecting the ontologies and alignments—exploring the ontologies, mappings and mapping suggestions, search and filter by various criteria, zoom, overview, etc. The third category includes services for presenting information to the user, for instance, reasons to suggest/accept a mapping suggestion, how the tool has calculated it, hinting at possible starting points and showing the current state of the process.

Various requirements arise from the tendency of increasing the size and complexity of the ontologies, alignments and alignment problems. They are grouped in the infrastructure and algorithms category in Table 2. We do not discuss techniques for large-scale matching [27] or matching with background knowledge despite they 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 [17] implements this by introducing interruptible computation, validation and recommendation sessions.

**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	✓	+	+	✓	-	-	-	

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. The authors of AIViz [19] highlight that clustering the graph improves the interactivity of the program. Clustering of the ontologies and alignments will allow reusing visualization techniques that work for smaller problems. A fragment-based strategy is implemented in [6] 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 [25] 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 [13] only requires user validation for problematic suggestions. In [17] the authors demonstrate that the session-based approach can reduce the unnecessary user interventions by utilizing the knowledge from previously validated suggestions. GOMMA [15] can reuse mappings between older ontology versions in order to match their newer versions. PROMPT [23] 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 [24]. The authors in [5] and [28] discuss criteria for selecting mapping suggestions that are shown to the user and strategies for user feedback propagation in order to reduce the user-system interactions. The same issues in a multi-user context are presented in [4]. A dialectical approach reusing partial alignment to map portions of two ontologies without exposing them is evaluated in [26].

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 [29].

Another challenge insufficiently addressed [29] 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 [7]. The environment should support communication services between its members as well—discussion lists, wikis, messages, annotations, etc.

Providing **recommendations (#5.6)** is another approach to support the user during the decision making process. They can be based on external resources, previous user actions, etc. and can be present at each point user intervention is needed—choosing an initial matcher configuration [1, 17], validating mapping suggestions [16] etc.

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 [11] 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 are incoherent which questions the quality of the results of the semantically-enabled applications utilizing them. According to [13] *reasoning-based error diagnosis* is one of the three essential features for alignment systems. Almost half of the quality aspects for ontology alignment in [20] address lack of correctness in the alignment in terms of *syntactic*, *semantic* and *taxonomic* aspects. The increasing size and complexity of the alignment problem demands debugging techniques thus a debugging module should be present in every alignment system. The authors in [14] 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 is a necessity.

In the field of ontology debugging there is already ongoing work that addresses explanation of defects. 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 [24] specify generating human understandable explanations for the mappings as a challenge as well. The authors in [1] implement advanced interfaces for **configuring the matching process** (#5.8) which provide the users with insights of the process and contribute to the understanding of the matching results.

**Trial execution of mappings** (#5.9.1) (what-if) will be of even greater 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), 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 (presented in [12]). The systems were selected because they have mature interfaces, often appear in user interface evaluations and accommodate features addressing the alignment of large ontologies. Table 2 shows the systems support for the requirements identified in Sect. 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 ( $\checkmark(*)$ ), a functionality that may help the users to keep track which correspondences have been already visited. The least supported category from the requirements in [8] 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.

### 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: efficiency, effectiveness, satisfaction. 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. Details for their configurations are available in [12]. We evaluated the user interfaces using a heuristic evaluation (effectiveness) by an expert user as well as through an observational study (efficiency, effectiveness) using novice users. The satisfaction aspect is addressed by the SUS questionnaire [2].

#### 4.1 Heuristic Evaluation

Our first evaluation is a heuristic evaluation. We present its most important findings here, the entire evaluation is available in [12]. 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 [22] and presented briefly below. 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;
- 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 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* and *Align Manually*—to validate and create mappings. 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}. The browser window in the *Suggestion Align* mode is divided into two parts by a thick gray



line but the buttons above and below are very close to it, {a}, and thus the components may be perceived as one 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}. All mappings for a concept are presented as a list. 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 that provides information for them. This label is actually a link which lists all remaining suggestions but it does not look clickable {d}. Below is the history label with the same issues and a warning box next to it shows a message relevant to the previous action. The window is divided similarly in the *Align Manually* mode. The top and central parts contain both ontologies represented as unmodifiable indented trees, the comment box is below them together with a search field. 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 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. 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.

**COMA 3.0** is a desktop system which provides one view during the alignment process [21]. Most of the screen space is occupied by the two ontologies which are placed side-by-side. 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 the buttons in the resizable left side of the screen {c}. The ontologies are represented as unmodifiable indented trees where explicit guiding lines connect a concept with its parent. The mappings are depicted with color-coded lines in the red-green spectrum depending on their confidence values. There is no explicit distinction between validated and potential mappings as there is in the other two systems {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 a unfolded

branch  $\{g\}$ . The user cannot select a mapping. All actions for a mapping are available through its concepts' context menus  $\{d\}$ . To achieve more intuitive interaction the mappings should be selectable and the corresponding actions should be available in the mappings context menu  $\{d\}$ . 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  $\{b, d\}$ . The search function has several issues—the scroll bar shows pink markers where the results appear but there is no jump to hit. Only concepts under expanded branches are considered during the search.

**CogZ** has a more complex interface than those of the other two systems. The screen is divided into two major resizable 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. The mappings can be selected and have a tooltip but do not have a context menu  $\{d\}$ . Several buttons above the mappings are used to apply different functions to them. 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 are aligned with these above and apply different filters on the mappings. They have different icons but two of them have the same tooltip. 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 each ontology are validated. Next to the progress bar a toggle button filters the ontologies according to the different mappings.

The first tab in the bottom part contains a table with all potential mappings. When a potential mapping is selected it is also highlighted in the upper view (if it is not filtered)  $\{g\}$ . A search strip on top of the table is activated on a key press  $\{g\}$ . 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)  $\{a\}$ . A double click on a potential mapping opens the same dialog as the view/create operation buttons. 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; it does not show the same dialog as them. The three operation buttons could be moved down to the *Create Mapping* button or in a potential mapping context menu (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 with different appearance and tooltips which look like they are meant for the same two actions, i.e., validate and create a mapping  $\{c, d\}$ . The reason why the system has calculated the mapping is shown at the bottom  $\{c, e\}$ . The second tab shows the completed mappings and is synchronized with the

upper view {c, d, g}. The third tab contains two parts, each showing (as a graph) the neighborhood of the selected concept in each ontology.

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 those 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 manipulation, inspection and explanation categories. We describe the study design, the participants and show its results.

**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 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 h (with a break after 1 h) and 1 h, respectively. It was expected that the user would work with each system for approximately 1 h. To prevent carry-over effects (learning) between the systems we changed the order in which they were presented to the users. We also used a different order of the input ontologies.

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. After that the participants started solving the 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 and their comments after each task and regarding the systems. The observer provided the right answer if a participant gave a wrong one.

**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

**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

twice since we were interested in their subsequent execution times. Task success and task times were collected for each task. The participants filled in the System Usability Scale (SUS) [2] 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.

**Results.** Table 4 shows the number of participants that successfully completed each of the tasks per system. 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. COMA 3.0 does not provide explicit explanation why it has suggested a potential mapping. It provides the similarity value on top of the link however it was not directly perceived as an explanation by the users.

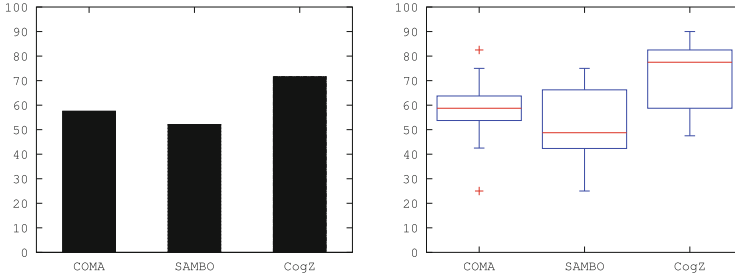
**Table 4.** Number of participants (max 8) successfully completed a task. / Average task time per system in seconds. (details in the text (\*); not applicable (-))

System/Task	A	B	C	D	E	F	G	H	I	J	K	L	M	D	A	N	O	E	P
SAMBO	8/30*	1/254	5/191	6/76	7/149	4/265	-	-	7/118	5/28	8/6	-	8/1	6/21	8/9*	-	6/47	7/67	-
COMA 3.0	2/174	2/183	7/99	2*/93	8/41	4/243	*	-	6/64	-	-	-	-	8/34	8/31	-	-	8/25	-
CogZ	7/167*	4/87	8/97	8/44	3/128	4/108	7/40	5/37	8/64	8/6	-	4/105	8/102	8/29	7/17	8/11	-	7/38	8/5

Several 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 first 6 tasks were solved with varying success by the participants. Most (4 out of 6) who did not complete task A in COMA 3.0 chose a wrong option to reject the mapping—instead of deleting a mapping using *Delete Correspondence* 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 it 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 4 also 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. Figure 1 shows the results of the SUS questionnaire.



**Fig. 1.** SUS questionnaire scores. (average (left) and boxplot of the dataset (right))

### 4.3 Discussion

In [12] we discuss the results of the user study and heuristic evaluation in connection with the requirements from Table 2. Here we present additional important findings.

Several issues became 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. Although COMA 3.0 provides guiding lines 3 users counted a sibling concept as a parent, while for others the guiding lines were very helpful. One participant counted one parent twice in CogZ. In SAMBO two had problems aligning parents and children in the trees, two others 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 of them managed to solve all F tasks correctly. 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 others did it in the B and after that in the F tasks with the same system.

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. 5 participants complained about the search functionality in COMA 3.0 as well. Although CogZ provided the best search functionality among the three systems two users did not solve task E due to search problems as well.

Another issue that constantly appeared across all systems was the terminology (also covered by the second Nielsen heuristic in Subsect. 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. SAMBO uses the term *mapping suggestions* for potential mappings, *Suggestion Align* for the mode where potential mappings

are validated and *Remaining Suggestions* for the mappings left for validation. Two users were unsure what *Suggestion Align* means/does and two others complained about the other two terms. The term *correspondence* denotes a mapping in COMA 3.0 and *Match Result*—the alignment. Two users were unsure for the meanings of the options in the context menu. It was observed that the users hesitated to press *Set Highest Similarity Value*. The terminology issue in CogZ had another aspect—the users were not confident in choosing actions. 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. One user was not sure what *Remove operation* does and three others said they were wondering which button to use.

We list briefly comments that appeared in the like/dislike section and other observations 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. Apart from the search and terminology the users also disliked that the potential mappings were not shown in the *Align Manually* mode. In COMA 3.0 the users liked the mapping representation—color-coded lines between the trees. 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 {d} in Subsect. 4.1 which suggests that common conventions should be followed. The *Neighborhood View* appeared as one of CogZ advantages. 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. One user stated that the button *Create Mapping* draws attention and the two views are not well separated. Comparing the three systems CogZ was most unstable in the sense that it was not clear if an action took place.

It comes at no surprise that most of the tasks are supported in CogZ since they are based on the requirements in the manipulation, inspection and explanation categories which are based on [8]. As shown in Tables 2, 3 and 4 SAMBO and COMA 3.0 cover fewer requirements. The explanation category is the least supported. 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. 1) which falls at the border between *OK* and *GOOD* in the adjective rating scale in [3]. 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” [3]. As take away issues from this study we would pinpoint the search and filter functionality especially in 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 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 to provide additional critique to the systems interfaces and cover aspects slightly or not mentioned in the user study (e.g., 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 first three 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 [9] 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. Aumüller, D., Do, H.H., Maßmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: SIGMOD, pp. 906–908 (2005)
2. Brooke, J.: SUS: a quick and dirty usability scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A., McClelland, I.L. (eds.) Usability Evaluation in Industry, pp. 189–194. Taylor and Francis, London (1996)
3. Brooke, J.: SUS: a retrospective. *J. Usability Stud.* **8**(2), 29–40 (2013)
4. Cruz, I.F., Loprete, F., Palmonari, M., Stroe, C., Taheri, A.: Pay-As-You-Go multi-user feedback model for ontology matching. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAU 2014. LNCS, vol. 8876, pp. 80–96. Springer, Heidelberg (2014)
5. Cruz, I.F., Stroe, C., Palmonari, M.: Interactive user feedback in ontology matching using signature vectors. In: ICDE, pp. 1321–1324 (2012)
6. Do, H.H.: Schema matching and mapping-based data integration. Ph.D. Thesis (2005)
7. Euzenat, J., Shvaiko, P.: User involvement. In: *Ontology Matching*, pp. 353–375. Springer, Heidelberg (2013)



8. Falconer, S.M., Storey, M.-A.: A cognitive support framework for ontology mapping. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 114–127. Springer, Heidelberg (2007)
9. Fu, B., Noy, N.F., Storey, M.-A.: Indented tree or graph? A usability study of ontology visualization techniques in the context of class mapping evaluation. In: Alani, H., et al. (eds.) ISWC 2013, Part I. LNCS, vol. 8218, pp. 117–134. Springer, Heidelberg (2013)
10. Granitzer, M., Sabol, V., Onn, K.W., et al.: Ontology alignment-a survey with focus on visually supported semi-automatic techniques. *Future Internet* **2**, 238–258 (2010)
11. Ivanova, V., Bergman, J.L., Hammerling, U., Lambrix, P.: Debugging taxonomies and their alignments: the ToxOntology-MeSH use case. In: WoDOOM, pp. 25–36 (2012)
12. Ivanova, V., Lambrix, P., Åberg, J.: Extended version of this paper. <http://www.ida.liu.se/~patla/publications/ESWC15/>
13. Jiménez-Ruiz, E., Grau, B.C., Zhou, Y., Horrocks, I.: Large-scale interactive ontology matching: algorithms and implementation. In: ECAI, pp. 444–449 (2012)
14. Jiménez-Ruiz, E., Meilicke, C., Grau, B.C., Horrocks, I.: Evaluating mapping repair systems with large biomedical ontologies. In: Description Logics, pp. 246–257 (2013)
15. Kirsten, T., Gross, A., et al.: GOMMA: a component-based infrastructure for managing and analyzing life science ontologies and their evolution. *J. Biomed. Semant.* **2**, 6 (2011)
16. Lambrix, P., Ivanova, V.: A unified approach for debugging is-a structure and mappings in networked taxonomies. *J. Biomed. Semant.* **4**, 10 (2013)
17. Lambrix, P., Kaliyaperumal, R.: A session-based approach for aligning large ontologies. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 46–60. Springer, Heidelberg (2013)
18. Lambrix, P., Tan, H.: SAMBO - a system for aligning and merging biomedical ontologies. *J. Web Semant.* **4**(3), 196–206 (2006)
19. Lanzenberger, M., Sampson, J., Rester, M.: Ontology visualization: Tools and techniques for visual representation of semi-structured meta-data. *J. UCS* **16**(7), 1036–1054 (2010)
20. Lanzenberger, M., Sampson, J., Rester, M., Naudet, Y., Latour, T.: Visual ontology alignment for knowledge sharing and reuse. *J. Knowl. Manag.* **12**(6), 102–120 (2008)
21. Massmann, S., Raunich, S., Aumüller, D., Arnold, P., Rahm, E.: Evolution of the COMA match system. In: OM, pp. 49–60 (2011)
22. Nielsen, J.: Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco (1993)
23. Noy, N.F., Musen, M.A.: Algorithm and tool for automated ontology merging and alignment. In: AAAI, pp. 450–455 (2000)
24. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: a literature review. *Expert Syst. Appl.* **42**(2), 949–971 (2015)
25. Paulheim, H., Hertling, S., Ritze, D.: Towards evaluating interactive ontology matching tools. In: Cimiano, P., Corcho, O., Presutti, V., Hollink, L., Rudolph, S. (eds.) ESWC 2013. LNCS, vol. 7882, pp. 31–45. Springer, Heidelberg (2013)
26. Payne, T.R., Tamma, V.: A dialectical approach to selectively reusing ontological correspondences. In: Janowicz, K., Schlobach, S., Lambrix, P., Hyvönen, E. (eds.) EKAW 2014. LNCS, vol. 8876, pp. 397–412. Springer, Heidelberg (2014)

27. Rahm, E.: Towards large-scale schema and ontology matching. In: Bellahsene, Z., et al. (eds.) *Schema Matching and Mapping*, pp. 3–27. Springer, Heidelberg (2011)
28. Shi, F., Li, J., Tang, J., Xie, G., Li, H.: Actively learning ontology matching via user interaction. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) *ISWC 2009. LNCS*, vol. 5823, pp. 585–600. Springer, Heidelberg (2009)
29. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges. *Knowl. Data Eng.* **25**(1), 158–176 (2013)