Supporting Ontology Alignment Tasks with Edge Bundling

Muhammad Nasir
Department of Computer Science
Memorial University
St. John's, NL, Canada
muhammad.nasir@mun.ca

Orland Hoeber
Department of Computer Science
University of Regina
Regina, SK, Canada
orland.hoeber@uregina.ca

Joerg Evermann
Faculty of Business Administration
Memorial University
St. John's, NL, Canada
jevermann@mun.ca

ABSTRACT

Ontologies are commonly used for knowledge representation and to exchange information between multiple applications. When the same information is represented by different overlapping ontologies, information sharing requires a mapping between corresponding pairs of entities. While ontology alignment algorithms have been developed to support such tasks, they generally do not offer entirely complete and precise mappings. As a result, an important interactive aspect of the ontology alignment process is the validation of automatically generated mappings, as well as the addition of new mappings, by a knowledge manager. While visual ontology alignment interfaces exist to support these tasks, showing a large number of mappings can result in a significant amount of visual clutter. To address this issue, an edge bundling approach has been adapted to the constraints of an existing ontology alignment interface. A user study was designed and conducted to evaluate the value of edge bundling in this context, with positive results for both mapping validation and addition tasks.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*graphical user interfaces (GUI), evaluation*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*representation languages*

General Terms

Design, Human Factors

Keywords

Edge bundling, information visualization, ontology alignment, user studies

1. INTRODUCTION

The use of ontologies to support the sharing of knowledge and information between multiple software systems has become increasingly important in recent years [5, 14, 16]. While these benefits are easily realized when two systems use the same ontology, challenges exist when sharing is required between systems that use different ontologies to represent the same information. This may occur when systems are built in parallel, when one system is changed or updated, or when new sharing functionality is required. In these cases, information sharing requires ontology alignment, i.e. a mapping between the entities in the pair of ontologies [15]. Such a mapping can then be used to automatically translate information from one ontology to the other.

Numerous ontology alignment algorithms exist to perform this task [24], with the output being a set of candidate mappings between entities across the pair of ontologies. Fundamentally, ontology alignment is a difficult problem due to the complexities of human language. Even though significant advances have been made as a result of natural language processing and graph matching approaches, the end results are seldom certain and are often incomplete.

A common approach to addressing the uncertainty about the candidate mappings is to allow a knowledge manager to interactively validate the mappings, as well as add new mappings that were not detected by the automatic alignment algorithm. While a naïve approach might be to simply provide a list of mappings from which a knowledge manager can delete or add new mappings, doing so makes it difficult to consider the ontological structure when during this process. This in turn might lead to incorrectly validated or missed mappings. A more effective approach is to present the ontologies and their mappings in a visual manner that is specifically designed to support the validation and addition tasks, and to provide interactive tools that aid in the completion of these tasks [9, 10].

Such visual systems normally represent the two ontologies as tree structures on either side of the interface, graphically depicting the mappings as edges between the corresponding entities. When representing a relatively small set of mappings, such interfaces can be quite effective, making it easy for knowledge managers to compare the pair of ontologies. However, even with as few as 20 mappings, the usability of such interfaces quickly deteriorates due to visual clutter created by the edge crossings within the mapping region of the interface. While some studies have explored the use of curved lines to represent the mappings [11] or interactive highlighting to allow the knowledge manager to focus on a specific mapping [11, 25], the fundamental problem of trying to make sense of a visually cluttered representation remains.

The potential solution to this problem presented in this

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

i-Know '13 September 04 - 06 2013, Graz, Austria

Copyright 2013 ACM 978-1-4503-2300-0/13/09 ...$15.00.

paper is the use of edge bundling for the representation of mappings. Edge bundling is the process of distorting the shapes of the edges in a graph to provide paths that are easier for the human eye to follow [17]. By bundling the edges within an ontology mapping interface, much of the visual clutter within the mapping representations can be eliminated due to the clustering of the edges and the crossing of a few bundles rather than many individual edges.

The remainder of this paper is organized as follows. The following section outlines related work pertaining to automatic ontology alignment, ontology alignment interfaces, and edge bundling. Section 3 provides motivation for the use of edge bundling in an ontology alignment interface, explains the process of edge bundling within an existing framework, and provides a discussion on the value of the approach in the context of the primary tasks of mapping validation and mapping addition. Section 4 outlines the user evaluation methods employed, followed by the results of this study in Section 5. The paper concludes with a summary of the key findings and an outline of future work in Section 6.

2. RELATED WORK

2.1 Automatic Ontology Alignment

A number of different approaches exist for performing automatic ontology alignment. Early work focused on syntactic matching of named entities that employed simple approaches based on string matching, prefix and suffix stripping, and synonym detection [12]. More recent semantic approaches use linguistic resources such as thesauri or other external knowledge bases to try to infer similarities between entities [12]. These approaches may be used together, and further enhanced with type-based methods that consider the data type and range of values of the information stored within the entities [6, 12]. Additionally, structural matching methods use graph matching algorithms to identify groups of potentially related entities based on their organization within the ontological structures [12, 22].

Due to the complexity of human language, and variations in how the same information may be encoded in different ontologies, the output of automatic ontology alignment algorithms may include errors or omissions [24]. Hence, there is a need to incorporate human decision-making within the ontology alignment process. The human mind is better suited to understanding the nuances of language than automatic algorithms, and is very effective at making decisions based on incomplete and/or conflicting information [28].

The resulting semi-automatic approaches build upon the work of automatic methods, providing information about the candidate mappings to allow the knowledge manager to either confirm or reject each mapping. This mapping validation process is more efficient than manual alignment due to the focused decision making that is supported by the suggestion of candidate mappings produced by the automatic algorithms. Since it may be possible for the automatic algorithms to miss important mappings, these semi-automatic approaches must also allow the knowledge manager to manually add mappings.

2.2 Ontology Alignment Interfaces

A wide range of interfaces have been proposed to support the human element of mapping validation and mapping addition within semi-automatic ontology alignment systems [9, 12]. A common theme among these interfaces is the graphical representation of both the ontologies themselves as well as the mappings. Rather than simply providing a textual list of candidate mappings, these interfaces take advantage of information visualization techniques to convey both the structure of the ontologies and the mapping information to the knowledge managers.

The purpose of providing this information in a graphical format is to allow knowledge managers to take advantage of their visual processing capabilities. The human vision system and mind has a great capacity to process visual stimuli [29]. With an effective visual representation, knowledge managers are able to readily perceive, interpret, and make sense of the features and relationships among the data, resulting in an amplification of their cognitive abilities [2]. That is, by showing the ontologies and mappings in a graphical format, the knowledge managers will able to visually identify the source and target entities of a mapping, as well as make comparisons among multiple mappings.

Since the ontologies themselves are structured as a hierarchy of information, the logical method for representing them is in a tree structure [18]. This approach is used in virtually all of the prior research on ontology alignment interfaces [12, 20]. Since the ontologies themselves may be much larger than can fit on a regular computer screen, interactive features such as node collapsing/expansion, vertical scrolling, and zooming are often implemented.

With a pair of ontologies represented as trees on opposite sides of the interface, the middle region can then be used to represent mappings between the entities. While early approaches used straight lines to connect associated entities between the ontologies, there are visual difficulties when following such connections due to the sharp corners that are created by edge crossings. Since the human eye can more readily follow curved lines [29], some ontology alignment interfaces now use curves to represent the mappings [1, 11].

The status of the mapping can also be conveyed by visual parameters of the curve. In some systems, colour is used to represent the difference between candidate mappings and confirmed mappings [1, 11]. Line style (e.g., solid, dashed) may also be used to convey this information [11], allowing the knowledge manager to readily identify features of individual mappings.

An open issue is whether to visually represent mappings when either end is not visible in the ontology representations. This situation may occur when a particular portion of one ontology is collapsed, or when an entity is not visible due to the scrolling of a large ontology. Some systems continue to show these mappings, with the end pointing to a collapsed node or being directed off the bottom or top of the display [11, 25], or can be configured to dynamically filter these mappings, adding them back in as the collapsed node is expanded or as the knowledge manager scrolls the ontologies [11].

One of the fundamental problems with visually representing mappings with lines or curves is that edge crossings may occur when the order of the entities is not consistent between the ontologies. Although the use of curves to represent the mappings makes these edge crossings easier to follow, the problem is not solved with the use of curves. As such, when representing a realistic set of mappings, the edge crossings may cause a significant degree of visual clutter. This clutter reduces the usability of the interface, making it difficult for the knowledge manager to visually trace a mapping from source to destination, or to make comparisons between mappings. While one approach might be to find an optimal ordering of entities in the ontology representations, there is no guarantee that such an ordering exists. Our approach is to address this issue through the improvement of the visual representation of the edges.

2.3 Edge Bundling

Edge bundling was developed within the graph drawing community as a mechanism for simplifying the visual representations of large and complex graphs. In such graphs, edge crossings may cause a significant amount of visual clutter, making it difficult to visually follow the paths between nodes. By bundling edges that have sources and destinations in common regions of the graph, many individual edges are replace with a smaller set of bundles, "cleaning up" the clutter in the visual representation. In addition, such bundling allow the high-level structure within the graph to be made more apparent [21].

In addition to this streamlining of the visual representation, there is also a theoretical foundation for the value of edge bundling. The Gestalt Laws are commonly used in the domain of information visualization to explain human perception of relationships among graphical entities [19, 29]. The laws of *proximity* and *similarity* suggest that the human mind will judge elements to be related if they are near one another and if they look similar [19]. A group of edges bundled together will be interpreted as being related because of their proximity to one another. In addition, if the overall shape of the edges within the bundle are also visually similar, this will further enhance the impression of relatedness.

These perceptions of relationships among the edges lead to two additional theories regarding human reasoning: categorization theory [26] and cognitive load theory [3]. Categorization theory explains the cognitive process by which humans organize information, and suggests that if information is presented such that it is already organized, the cognitive work required to understand and organize the information will be reduced. Cognitive load theory explains the relationship between memory load and the ability to solve problems, indicating that if memory load can be reduced, decision-making speed and accuracy can be increased.

Based on these theoretical justifications, the proposed benefit of edge bundling is that it supports the interpretation of the relationships among the edges, allowing the analysis of clusters of edges rather than individual edges. By considering the clusters, the cognitive effort in categorizing the edges is reduced, as is the memory load associated with decision-making tasks. The final outcome is an expectation of more efficient and effective analytical reasoning, decision-making, and problem-solving.

Although there are a number of different methods for implementing edge bundling, a popular approach is to use a spring-based algorithm [17]. Spring layout algorithms have been used for many years for dynamic graph drawing [7]. As such, it is a natural approach to use within edge bundling. Spring-based edge bundling divides each edge into a number of small segments. The position of each segment is compared to the positions of all segments in the other edges, and a virtual spring is attached if the segments are sufficiently near one another. An iterative process minimizes the forces on the springs by moving the connected edge segments closer to one another. The end result is a grouping of edge segments and an associated bundling of the edges themselves.

There are a number of factors that control the tradeoff between efficiency and effectiveness in the bundling process. Chief among these are the size of the segments, the distance threshold between segments that exert spring forces upon one another, and the number of iterations taken to minimize the spring forces.

If each edge is divided into a small number of large segments, the number of springs will be relatively small, and the spring forces can be minimized quickly. However, doing so produces rather course bundling. Using a larger number of smaller segments will produce smoother bundles that are easier for the human eye to follow, but at the expense of a larger number of spring forces to minimize.Thus, the tradeoff between the quality of the visual representation and the computational costs must be considered.

The distance threshold that dictates whether a virtual spring is or is not added between a pair of edge segments has an important outcome not only on the computational cost but also the final bundling outcome. If this threshold is made too small, the computational cost will be low (due to the small number of springs), and very little bundling will occur. If this threshold is made too large, the computational cost will be great, and a small number of very dense bundles will be generated. Carefully tuning this parameter in order to produce a sufficient bundling at an acceptable computational cost is necessary.

Choosing the number of iterations taken to minimize the spring forces also requires consideration. Too few iterations and the bundles will not be well formed; too many and time is wasted on minor modifications of the segments with little visual improvement. Rather than controlling the precise number of iterations, it is more common to specify a threshold for changes between iterations that signals convergence, together with a limit for the maximum number of iterations.

While edge bundling can allow the structure of the edges within a graph to become more apparent, it does introduce a new problem. By clustering the edges within a bundle, ambiguity is introduced into the visual representation. That is, with edge bundling it is no longer possible to visually identify the precise source and destination of a given edge. Instead, all that can be done is to identify a number of possible candidate destination nodes for a given source node. Hence, whenever edge bundling is implemented in a setting that requires precise information to be extracted, interactive tools are required for the disambiguation of the bundles.

3. EDGE BUNDLING WITHIN AN ONTOLOGY ALIGNMENT INTERFACE

3.1 Framework

The prototype system developed to evaluate edge bundling support for ontology alignment tasks is based on the open and extendable framework provided by CogZ [11]. This existing framework provides the core features required by an ontology alignment interface, including integration with existing state-of-the-art ontology alignment algorithms [23], representation of the pair of ontologies as zoomable and scrollable trees on opposite sides of the interface, rendering of the mappings as curves within the middle region of the interface, and basic interaction mechanisms that support mapping validation and mapping addition tasks. Our extension integrates edge bundling into the process, and introduces new interaction mechanisms to support the disambiguation of edges within a bundle.

3.2 Edge Bundling Process

Following the edge bundling approach outlined in Section 2.3, the first step in the process is to extract the edges from the ontology alignment interface framework. This information includes the source and destination points (entities in the ontologies), the edge path, and the status of the mapping represented by the edge. The mapping edges whose source and/or target concepts are not visible (i.e., scrolled out of view or collapsed) are filtered from this set in order to ensure that the bundling is only performed on completely visible edges. The existence of the filtered mappings can be observed via a graphical mark provided by CogZ beside each mapped concept, and bringing both source and target concepts into view at the same time reveals the mapping edge allowing it to be bundled.

The next step of the process is the segmentation of each edge in the mapping set. As previously discussed, there is a tradeoff between the number of the segments and the computational costs when performing edge bundling. While 1-pixel segments would provide the smoothest curves, doing so would result in an unsatisfactory performance. Rather than statically defining this parameter, we took inspiration from Holten and van Wijk's work on edge bundling in complex graph structures [17], and dynamically adjust the number of segments (and therefore the size of the segments). By starting with a small number of edge segments, course edge bundling can be done efficiently. Iteratively increasing the number of edge segments then allows finer and finer levels of bundling to be achieved.

In our case, we start with four segments for each edge. However, these segments are not defined over the entire length of the edge. Since it is important to be able to clearly identify the source and destination entities within the ontologies, we do not want the bundles to be formed too close to the actual ontologies. As such, the segmented region of the edges is restricted to 75% of the length of the edges.

Holten and van Wijk [17] describe a number of different measures for determining whether a pair of edge segments should be bundled. Many of these measures take into account the complex nature of general graph structures, including the position of the edge segments, the lengths of the entire edges, the edge direction, and the angle of edge crossings. Given the restricted nature of the edges within an ontology alignment interface (e.g., undirected edges only between nodes on opposite sides of the interface), many of these measures do not produce noticeable effects on the edge bundles. As such, our approach only uses the position of the edge segments when determining the potential for bundling.

For each edge segment, the distance from its midpoint to the midpoint of all other segments in all other edges is calculated. If the distance between a pair of segments is below a distance threshold, a virtual spring is attached between these segments. This distance threshold is based on a ratio to the length of the edge segment, such that longer edge segments are bundled to more distant edge segments. In order to force non-connected edge segments away from one another, virtual electrostatic forces are also modelled within the system. These forces avoid the situation of over-bundling, whereby all edges are pulled together because there are no forces pushing them apart.

The next step in the process is to attempt to minimize the forces exerted upon the edge segments due to the springs and electromagnetic forces. As edges are moved closer to one another, the spring forces will be reduced, but the electromagnetic forces will be increased. We consider the sum of all forces being exerted on each edge segment, and move its location in the direction of the force by a distance that is relative to the magnitude of the force. Since the movement of one edge segment in this manner may cause the forces exerted on other edges to change, the process of minimizing the forces within the system must be done over multiple iterations. We follow a simple approach of statically limiting the number of iterations to 120.

At this point, a coarse-level of bundling will have been achieved based on only four segments for each edge. In order to smooth out the bundling, this whole process is repeated over 12 additional cycles, each time increasing the number of segments by one (and therefore reducing the length of each edge segment). This choice of the number of additional cycles (and therefore the number of additional segments) was determined through experimentation. The dividing points used when increasing the number of segments are chosen at the specified location on the straight-line edge segments in the previous cycle. This incremental process is more efficient and effective than performing the bundling at a fine level of detail straight away [17].

The final outcome is edge bundling performed over 16 segments on each edge. Since drawing the edges within the display using only these segments would not be graphically appealing, a curved line is fitted to the centre points of each edge segment. Figures 1 and 2 provide before and after examples of the outcome of edge bundling.

3.3 Bundle Ambiguity and Disambiguation

Although edge bundling produces a less-cluttered visual representation of the mappings, it has the side-effect of introducing ambiguity among the edges within the bundles. That is, for a mapping that starts at a given concept, if its edge is contained within a bundle, it is not possible to determine the exact ending concept from among those that are contained in the same bundle. This causes a significant problem during the mapping validation tasks since knowledge managers must examine and interact with mappings individually.

The solution to this problem is to provide interactive tools that support disambiguation. Since tooltips and single clicks on edges are already interactive aspects of the CogZ framework, the interaction for disambiguation must use an alternate mechanism. By holding down the alt key and clicking on any edge within a bundle, pop-up window is provided with a listing of all the mappings within the bundle. From this window, clicking on any mapping not only highlights it within the bundle resulting in the disambiguation of the edge, but also brings up a dialog box for confirming or rejecting the mapping. In addition, multiple mappings can be selected in this way from among the bundle, allowing for more efficient validation of the entire bundle.

3.4 Support for Ontology Alignment Tasks

Since the mapping validation and mapping addition tasks are already very time-consuming and require a great amount

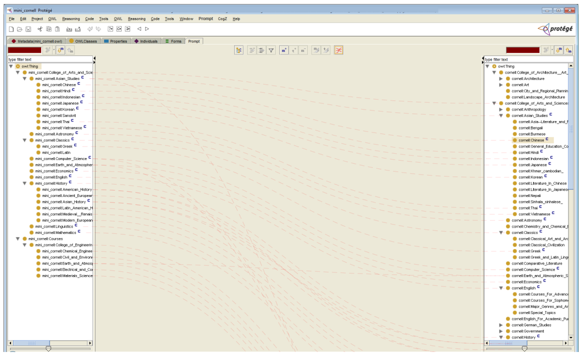

Figure 1: View of the baseline system with no bundling. The red dashed curves represent the candidate mappings generated by the system.

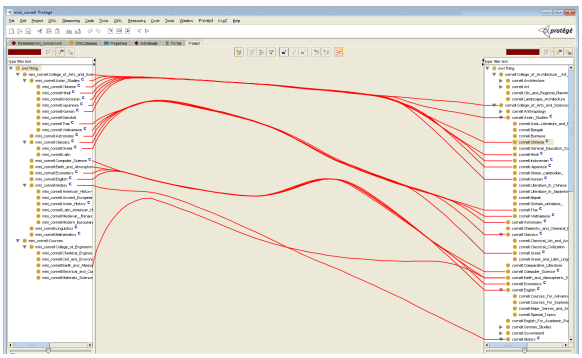

Figure 2: Application of edge bundling to the mapping edges. Note that mappings whose source and destination are not both visible have been filtered.

of cognitive effort, performing these tasks while navigating and interpreting a complex and ill-structured visual interface can make them even more challenging. Edge bundling offers the promise of presenting a less complex and more structured view of the mappings, which can help in reducing the cognitive load of the knowledge managers.

During the mapping validation tasks, edge bundling helps by dividing the larger task into multiple smaller sub-tasks of validating the mappings within a given bundle. Reviewing the mappings within a bundle provides the knowledge managers with relevant context about the set of mappings as a group, and allows the other bundles to be ignored (see Figure 3). Thus, tasks such as validating whether the set of mappings is internally consistent can be done by considering only the edges within the bundle, making efficient use of the knowledge managers' cognitive memory. As a result of thisfocus that the bundling provides to the task, the degree of searching and swapping of information in and out of cognitive memory will be greatly reduced in comparison to an unstructured examination of the candidate mappings.

The benefits of edge bundling may also be realized in the context of the mapping addition tasks. By clustering related mappings, the overall organization of how the mappings relate to the structure of the ontologies may be more easily perceived. When seeking potentially related, but as yet unmapped, entities, the structure imposed by the bundles provides the knowledge managers with a relatively small space

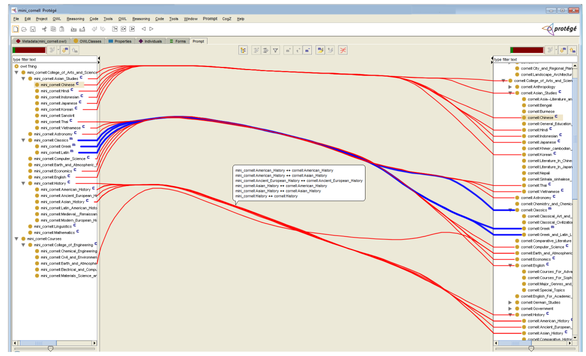

Figure 3: View of the middle stage of the alignment task. The red curves show the candidate mappings, while the blue curves show the confirmed and added mappings. The tooltip shows the information about the mappings within one of the bundles.

in which to search. Rather than examining the entire ontology for a corresponding entity, the knowledge managers can visually focus on areas around the existing bundles, thus reducing the cognitive load associated with this task.

4. EVALUATION METHODOLOGY

A user study was designed to evaluate the benefits and drawbacks of edge bundling for ontology alignment tasks. Since this enhancement was implemented within the CogZ framework, CogZ was used as the baseline comparison point. Within the study design, the interface was treated as the independent variable (with bundling vs. without bundling). The dependent variables consisted of the time-to-task completion, accuracy of the task outcomes, and the participant's perceptions of ease of use and usefulness.

In order to allow direct comparisons between the two interfaces, a within-subjects design was employed. The participants used each interface only once, conducting ontology alignment tasks with two different pairs of test ontologies. The order of interface exposure and test ontology set assignment were varied using a $2x2$ Graeco-Latin square [13] in order to counterbalance potential learning effects.

In each test condition, two specific ontology alignment tasks were assigned. Participants were first asked to verify the mappings produced by the automatic algorithm. This task included both confirmation of correct mappings, as well as rejection of incorrect mappings. After the completion of this task, participants were asked to complete a mapping addition task, adding in any new mappings they thought were missed by the automatic algorithm. This ordering of tasks remained constant in the study design since it follows the normal procedure for conducting ontology alignment.

The two sets of test ontologies chosen for this study were of similar size (number of entities, depth of the hierarchical structure, number of mappings used in the validation tasks, and number of mappings to find in the addition tasks). They represented the domain areas of a university [11] and an academic conference [8], for which we expected the participants in the study to have general knowledge.

Twenty four participants were recruited from the senior undergraduate and graduate student population of Computer Science, Engineering, and Business departments to participate in the study. Since a participant's domain knowledge and prior experience with ontologies could greatly affect the outcome of the study, questionnaires were administered to determine prior knowledge in the specific ontology domains, level of expertise with ontologies or knowledge representation, and prior experience with using ontology alignment systems. From the responses, we inferred that the participants belong to a relatively homogeneous group, with general knowledge about the test ontology domains, no significant experience with ontologies themselves, and no prior experience using ontology alignment systems.

In order to further ensure a common baseline level of knowledge and experience, all participants were provided with a training session using CogZ (both with and without edge bundling) to perform the ontology alignment tasks. This training was conducted using a third test ontology set in the domain of common things in one's daily life [8]. This training was conducted with the appropriate bundling settings (with/without) immediately preceding the exposure to each test interface. In addition, prior to conducting the tasks on the test ontologies, all participants were provided with a common explanation of the ontology domains.

While the ontology alignment tasks were being conducted, measures of time to task completion were taken; the accuracy of the results were verified post-hoc. After the completion of all test tasks, a post-study questionnaire was administered to measure perceptions of ease of use and usefulness of the interfaces for both the validation and addition tasks.

5. EVALUATION RESULTS

5.1 Time to Task Completion

The average time taken by participants to perform the mapping validation tasks is shown in Figure 4. Clearly, the participants took significantly less time when using the edge bundling interface to complete this task in both test ontology sets. The results of a statistical comparison using analysis of variance (ANOVA) [27] are reported in Table 1. The differences in completion time for the validation tasks over both ontology sets were found to be statistically significant.

In general, there are two competing aspects of edge bundling that can have an effect on the time taken to perform the mapping validation tasks. As previously discussed, by bundling the edges, the clutter within the visual interface is reduced and the task of performing mapping validation becomes more structured and focused. This has the potential to increase the speed at which the tasks can be completed. However, there is an added interactive element of disambiguating a bundle that takes additional time. From these results, we can see that the time-saving effects of providing the mappings in bundles greatly outweighs the extra work required to disambiguate a bundle.

The results for the time taken to complete the mapping addition tasks are reported in Figure 5. These results are mixed, with the statistical analysis showing no significance between the data for either test ontology set (see Table 2). As a result, we conclude that edge bundling had no effect on the time taken to perform the mapping addition tasks.

This result is contrary to our expectations. We had anticipated that the structure provided by the edge bundling would have aided the participants in finding missing mappings. One possible explanation for not realizing a time savings in these tasks is that the added overhead of the interactive disambiguation may have been more prevalent during these addition tasks. During the mapping validation tasks, this overhead was incurred once for each bundle since the disambiguation could be done in concert with the validation of all mappings within the bundle. However, for the mapping addition tasks, the participants may have incurred this time penalty of disambiguation repeatedly as they analyzed the bundles to determine an appropriate end point for a candidate entity in one of the ontologies. However, it should be noted that this added time did not result in significantly more time being taken, but instead balanced out the time savings as a result of the edge bundling itself.

5.2 Accuracy

After the participants finished the validation and addition tasks, the confirmed, rejected, and added mappings were examined by a panel of experts to ensure accuracy (correctness). The accuracy of the validation tasks was perfect (100%) for both tasks and both interfaces. This result shows that the visual encoding of mappings as edges, whether they be bundled or not bundles, is an effective means of repre-

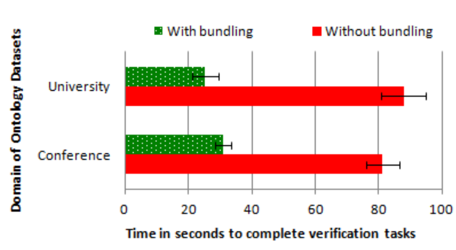

Figure 4: Average time taken to complete the validation tasks.

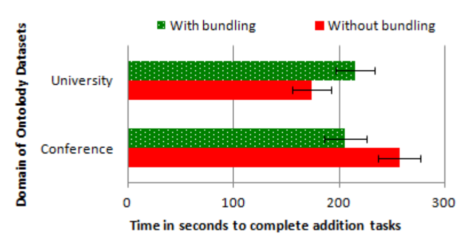

Figure 5: Average time taken to complete the addition tasks.

Table 1: Statistical analysis of the differences in the time to task completion of the validation tasks.

Ontology Domain	ANOVA
University	$F(1,22) = 71.046, p < 0.0001$
Conference	$F(1,22) = 78.791, p < 0.0001$

Table 2: Statistical analysis of the differences in the time to task completion of the addition tasks.

Ontology Domain	ANOVA
University	$F(1,22) = 2.921, p = 0.101$
Conference	$F(1,22) = 2.077, p = 0.163$

senting the mappings. Furthermore, it is an indication that the participants were careful in completing the mapping validation tasks.

For the addition tasks, the participants proved to be more accurate when using the bundling interface over both test domains, as shown in Figure 6. From the calculations of ANOVA between the interfaces (see Table 3), we found that the improvements in accuracy were statistically significant for both test ontology sets.

These results suggest that the structure provided by the edge bundling actually helped the participants to find appropriate mappings to add. That is, by using the bundles as a guide for finding missing mappings, participants were able to make better choices for such additions than using the non-bundled representation of the mappings. So, while the bundling did not improve the speed at which such bundling addition tasks could be completed (as reported in the Section 5.1), it was able to improve the accuracy of the task.

5.3 Perceived Ease of Use and Usefulness

Overall perceptions of ease of use and usefulness of the interfaces for the two task types were collected in a post study questionnaire, using an adaption of the Technology Acceptance Model (TAM) [4]. Rather than asking participants to provide answers for each interface on a Likert scale, we asked that for each question they rank one interface over the other, or indicate that they are the same. Since the TAM instrument provides six questions that relate to ease of use, and six that relate to usefulness, we aggregated the responses for all participants based on these underlying constructs in order to gauge their overall perceptions.

Figures 7 and 8 illustrate the aggregate responses to the perceived ease of use and usefulness questions, for the mapping validation and mapping addition tasks respectively. The results show that there is a strongly positive perception of both the ease of use and the usefulness of the edge bundling interface for both types of tasks. Table 4 shows the results of a Wilcoxon signed rank test [30], which confirms the statistical significance of the findings.

These results provide an indication that the participants appreciated the less visually cluttered interface produced by the edge bundling, finding the interface both easier to use and more useful for the prescribed tasks. While edge bundling introduces a degree of ambiguity in the mapping representation, this ambiguity was not perceived as being detrimental to the ease of use nor the usefulness, perhaps due to the simple yet effective approach for disambiguating the bundles.

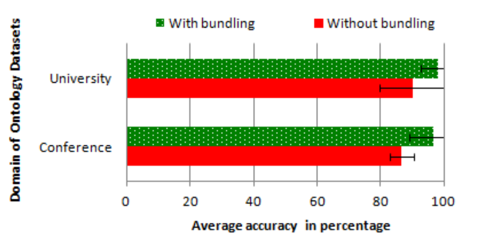

Figure 6: Average accuracy for addition tasks in percentage.

Table 3: Statistical analysis of the differences in the accuracy of the addition tasks.

Ontology Domain	ANOVA
University	$F(1, 22) = 5.851, p < 0.05$
Conference	$F(1, 22) = 5.210, p < 0.05$

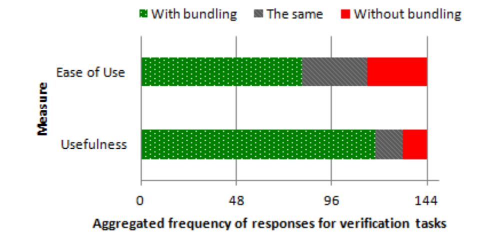

Figure 7: Aggregate perceived ease of use and usefulness for the mapping validation tasks.

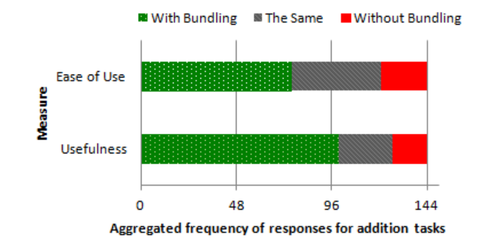

Figure 8: Aggregate perceived ease of use and usefulness for the mapping addition tasks.

Table 4: Statistical analysis of the differences in the perceived ease of use and usefulness of the two interfaces for the mapping validation and addition tasks.

Task Type	Perceived Ease of Use	Perceived Usefulness
Validation	$Z = -4.841, p < 0.001$	$Z = -9.297, p < 0.001$
Addition	$Z = -5.327, p < 0.001$	$Z = -7.673, p < 0.001$

6. CONCLUSION

The primary contributions of this work are the application of edge bundling within an ontology alignment interface and its evaluation in the context of supporting the core ontology alignment tasks. For mapping validation tasks, edge bundling allowed participants to complete their tasks faster with the same 100% accuracy as non-bundling. For mapping addition tasks, while there was no statistically significant difference in the time taken to complete the tasks, participants were more accurate when using the edge bundling interface. The participants perceptions of ease of use and usefulness were in favour of the edge bundling interface over both tasks.

These positive findings highlight the value of showing the high-level structure of the mappings via the bundles, and simultaneously reducing the visual clutter within the interface. Through the visual perception of similarity among the edges and bundles, impressions of relationships were implied. This in turn supported the cognitive activity of organizing the mappings, providing guidance to the tasks and reduced the cognitive load. Although such bundling has the side-effect of making it difficult to identify the specific source and destination concepts within a bundle, this problem was mitigated through an interactive disambiguation feature.

Our future work in this domain follows four different streams of inquiry. First, we wish to study the benefits and drawbacks of edge bundling on different sizes of ontologies and different sizes of mapping sets. Second, we plan to explore different edge compatibility measures that take into account features of the ontologies themselves such as the semantic distance between concepts within the source and destination ontologies. Third, we aim to develop a collection of measures that convey the overall visual quality of the bundling display. Fourth, using these measures, we wish to study the effects of entity ordering within the ontologies on the bundling quality, with the goal of developing a hybrid approach that combines ontology organization with edge bundling.

7. REFERENCES

[1] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with COMA++. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 906–908. ACM, 2005.

[2] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings In Information Visualization: Using Vision To Think*. Morgan Kaufmann, San Francisco, 1999.

[3] P. Chandler and J. Sweller. Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8(4):293–332, 1991.

[4] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340, 1989.

[5] J. De Bruijn. Using ontologies: enabling knowledge sharing and reuse on the semantic web. Technical Report DERI-2003-10-29, Digital Enterprise Research Institute (DERI), Austria, 2003.

[6] H.-H. Do and E. Rahm. COMA: a system for flexible combination of schema matching approaches. In *Proceedings of the International Conference on Very Large Data Bases*, pages 610–621, 2002.

[7] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[8] J. Euzenat. Ontology alignment evaluation initiative. http://oaei.ontologymatching.org, last viewed on 28 February 2013.

[9] S. M. Falconer and N. F. Noy. Interactive techniques to support ontology matching. In Z. Bellahsene, A. Bonifati, and E. Rahm, editors, *Schema Matching and Mapping*, pages 29–51. Springer, Berlin, 2011.

[10] S. M. Falconer and M.-A. Storey. Ontology mapping - a user survey. In *Proceedings of the International Workshop on Ontology Matching*, pages 49–60, 2007.

[11] S. M. Falconer and M.-A. Storey. CogZ: cognitive support and visualization for semi-automatic ontology mapping. In *Proceedings of the International Conference on Biomedical Ontology*, pages 1–2, 2009.

[12] M. Granitzer, V. Sabol, K. W. Onn, D. Lukose, and K. Tochtermann. Ontology alignment—a survey with focus on visually supported semi-automatic techniques. *Future Internet*, 2(3):238–258, 2010.

[13] D. A. Grant. The Latin square principle in the design and analysis of psychological experiments. *Psychological Bulletin*, 45(5):427–442, 1948.

[14] N. Guarino. Formal ontology and information systems. In *Proceedings of the International Conference on Formal Ontology in Information Systems*, pages 3–15, 1998.

[15] D. H. Hai. *Schema Matching and Mapping-based Data Integration: Architecture, Approaches and Evaluation.* VDM Verlag, Saarbrücken, Germany, 2007.

[16] J. Hendler. OWL Web ontology language. http://www.w3.org/TR/owl-features/, last viewed on 02 February 2011.

[17] D. Holten and J. J. van Wijk. Force-directed edge bundling for graph visualization. *Computer Graphics Forum*, 28(3):983–990, 2009.

[18] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, and E. Giannopoulou. Ontology visualization methods - a survey. *ACM Computing Surveys*, 39(4):10:1–10:43, 2007.

[19] W. Köhler. Gestalt psychology. *Psychological Research*, 31(1):18–30, 1967.

[20] M. Lanzenberger, J. Sampson, and M. Rester. Visualization in ontology tools. In *Proceedings of International Conference on Complex, Intelligent and Software Intensive Systems*, pages 705–711, 2009.

[21] F. McGee and J. Dingliana. An empirical study on the impact of edge bundling on user comprehension of graphs. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 620–627, 2012.

[22] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the International Conference on Data Engineering*, pages 117–128, 2002.

[23] N. F. Noy. Ontology management with the PROMPT plugin. In *Proceedings of International Protégé Conference*, pages 1–4, 2004.

[24] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[25] G. G. Robertson, M. P. Czerwinski, and J. E. Churchill. Visualization of mappings between schemas. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 431–439, 2005.

[26] E. Rosch and B. B. Lloyd. *Cognition and Categorization.* Lawrence Erlbaum, Hillsdale, NJ, 1978.

[27] H. Scheffé. *The Analysis of Variance.* John Wiley and Sons Inc., New York, 1959.

[28] B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley, Boston, 5th edition, 1997.

[29] C. Ware. *Information Visualization: Perception For Design.* Morgan Kaufmann, San Francisco, 2nd edition, 2004.

[30] R. F. Woolson. Wilcoxon signed-rank test. In P. Armitage and T. Colton, editors, *Encyclopedia of Biostatistics*, volume 6, pages 4739–4740. John Wiley, New York, 1998.