

AlViz - A Tool for Visual Ontology Alignment

Monika Lanzenberger*, Jennifer Sampson

Vienna University of Technology, Norwegian University of Science and Technology

lanzenberger@ifs.tuwien.ac.at, sampsonj@idi.ntnu.no

Abstract

We introduce a multiple-view tool called AlViz, which supports the alignment of ontologies visually. Ontologies play an important role for interoperability between organizations and for the semantic web because they aim at capturing domain knowledge in a generic way and provide a consensual understanding of a domain. Alignment is the process where for each entity in one ontology we try to find a corresponding entity in the second ontology with the same or the closest meaning. Existing ontology alignment tools do not adequately provide a way for users to analyse the results. While many alignment tools generate lists of mappings it is difficult to analyse these alignments without examining every pairwise correspondence in the output files and even then it is an overwhelming task. We propose the use of visualization techniques to facilitate user understanding of the ontology alignment results. AlViz is implemented as a tab plug-in for Protégé.

Keywords— Cluster Graph, Multiple View, Ontology, Alignment.

1 Introduction

An ontology is a vocabulary that can be used to describe a domain explicitly. Because ontologies represent a shared meaning of a domain, they are seen as an enabler for many applications such as data integration, e-commerce and semantic web services, which are tools exploiting internet content represented in a form that is easily machine-processable and enriched with metadata. Ontologies play an important role for interoperability between organizations and for the semantic web because they aim at capturing domain knowledge in a generic way and provide a consensual understanding of a domain. Like [1], we define an ontology as a tuple: $O := (C, H_C, R_C, H_R, I, R_I, A)$. Concepts C , which are classes of real-world objects, are organized in a subsumption hierarchy H_C . Relations R_C exist between pairs of concepts and may also be arranged in a hierarchy H_R . The instances of specific concepts I , are interconnected through relational instances R_I . A rep-

resents the axioms, which are used for inferring knowledge about the domain. With an increase in the development of ontologies done by different communities comes the need for tools and techniques for solving heterogeneity problems between different ontologies. In order to combine or map two or more ontologies we need to analyze the relationships among entities. However, establishing such relationships between entities is difficult, especially, when the ontologies have been developed by different communities with different world views. One of the main goals of ontology management research is the development of ontology mapping algorithms that automatically bring two or more ontologies into alignment. We define ontology alignment as given two ontologies O_1 and O_2 , each describing a set of discrete entities: concepts C , relations R and instances I , are examined in order to find the alignment relationships that hold between these entities. If an alignment associates a concept x in ontology O_1 to a concept y in ontology O_2 , then x and y are either equal, syntactically equal, similar, or x may be broader-than or narrower than y . There could be many concepts in O_1 or O_2 that have no equivalents nor related terms in the other ontology. Before two ontologies O_1 and O_2 can be aligned, it may be necessary to introduce new subtypes or supertypes of concepts or relations in either O_1 or O_2 in order to provide suitable targets for alignment [2]. So, ontology alignment may include changes to the source ontologies.

Mapping is the most important step in ontology alignment and is described informally as where for each entity (e.g., concept, relation, or instance) in one ontology we try to find a corresponding entity in the second ontology, with the same or the closest meaning. The resulting mappings may be as simple as a set of one-to-one correspondences between some concepts, or they can be complex declarative mappings [3]. The result of an ontology mapping can be a set of mappings between two ontologies with a corresponding similarity value. A variety of approaches have been described in literature [4] and according to the evaluations of such systems they appear to demonstrate promising results when tested on example ontologies. However,

*Part of this work was done while the author was an ERCIM research fellow at IDI, Norwegian University of Science and Technology (Trondheim, Norway) and CITI, Centre de Recherche Public Henri Tudor (Luxembourg).

evaluating the strengths and weaknesses and overall quality of the results through the use of these tools remains an open question because they create long textual lists showing pairs of mapped concepts, which are difficult for humans to understand and interpret (compare figure 1).

In order to make such mapping results accessible information visualization (InfoVis) seems to be a promising approach. InfoVis offers a bundle of techniques to represent hierarchical or semi-structured data. The use of visual metaphors supports interpretation and understanding of multi-dimensional data in order to provide the user with relevant information. Graphical primitives such as point, line, area or volume are utilized to encode information. These objects are characterized by position in space, size, connections & enclosures, shape, orientation, and visual cues like color and texture, with temporal changes, and viewpoint transformations [5]. The goal of InfoVis is to promote a more intuitive and deeper level of understanding of the data and to foster new insights into the underlying processes [6]. A great amount of work has been done in the field of InfoVis during the last years. Some prominent methods are: hyperbolic trees [7]), graph-based techniques (e.g., small world graphs [8]), maps (e.g., themescape [9]), distortion-oriented (e.g., fisheye lens [10]) and other focus+context techniques [11]. InfoVis aims at making complex data accessible for interactive investigation by the user.

The combination of several views is well-known as multiple view visualization [12]. Linking different visualization techniques and displaying distinct views of the same data objects is most commonly used and also applied in our visualization tool AlViz, although other types exist as well [13]. A multiple view visualization offers a lot of advantages, such as improved user performance, discovery of unforeseen relationships, and unification of the desktop [14]. Exploring data by using InfoVis is characterized by abstraction. Visualization reduces information or emphasizes certain aspects of the data in order to ease goal-oriented interpretation. Combining distinct visualizations yields different kinds of abstractions from the data, which allow for user-specific exploration approaches. An important challenge of multiple view visualization is its complexity for both, users and designers. Designing interaction mechanisms needs to take consistency aspects of the views and the data into account. The user switches between different views and contexts. This is only useful if switching brings about more insights than confusion. Linking and brushing of associated data is one way to support the user. In short, linking and brushing means that in one view the user selects and highlights items. The associated items in the other views are automatically highlighted so that the user gets information about the relationships among the

linked data items. Whether to use multiple view visualization is a cost/benefit trade-off between the advantages of such a multiple view system and the corresponding complexity that arises by applying it [13]. An analysis of the available tools showed that in ontology alignment the high complexity of data (even increased by the involvement of two ontologies) indicates the need for a combination of different views. Suitable interaction mechanisms and consideration of perception processes is crucial for the usefulness of such a multiple-view tool. When developing our visual ontology alignment tool we took these aspects into account thoroughly.

Recently, the semantic web community drew attention to ontology alignment and so far the main results are alignment algorithms with the intention of bringing the entities of two ontologies into concordance automatically. But not everything can be done by machine, user interaction is still essential in order to control, approve and optimize the mapping results. We developed a multiple-view tool which aims at supporting the alignment process visually. In the remaining sections of this paper, we first discuss related work in ontology alignment visualization, followed by a description of AlViz and the cluster-graph approach. Finally, we discuss some conclusions and outline future work.

2 Related work in ontology visualization

Currently, just a few ontology tools realize interactive, visual support of ontology alignment or ontology merging. In particular, we identified Prompt-Viz [15] and OLA [16] as related to our work.

Prompt-Viz is a visualization tool for Protégé's Prompt plug-in [17]. It provides one single visual representation of two (merged) ontologies using histograms within a treemap [18]. The bars in the histograms represent the percentage of descendants classified as unchanged, added, deleted, moved-from, moved-to and directly changed respectively. It is divided into four linked frames: (1) an expandable horizontal tree layout of the ontology showing the differences; (2) a treemap layout of the ontology embedded in a zoomable user interface; (3) a path window that shows the location of the currently selected concepts within the is-a hierarchy of the ontology; (4) a detailed list of the changes (if any) that have occurred to the currently selected concept. OLA (OWL Lite Alignment) [16] uses graph-based visualizations to represent ontologies. In particular, an extended JGraph API is applied. The graph structure of OLA makes relationships between language elements explicit, e.g., if a class *c* refers to another class *c'* via a `owl:allValuesFrom` restriction, a labeled path between the corresponding nodes in the OL-graph is shown such that the connection between both classes is



Figure 1: Small fragment of a text file showing pairs of alignment results produced by Foam [29].

perceived intuitively. Besides common sub-class relationship the user can activate the display of edges between objects that are reverse, symmetric or transitive.

For ontology alignment it is important to make consequences of mapping decisions visible within the individual source ontologies. We believe that a convenient ontology alignment tool needs to offer both: details and overview in an easily understandable and manipulable way. Thus, we have identified some shortcomings of OLA and Prompt-Viz when used for visual ontology alignment. Whereas OLA lacks of overview and zooming mechanisms which are necessary for the interactive visualization of ontologies, Prompt-Viz fails to show enough details about the source ontologies. Prompt-Viz enables the user to view the result of two merged ontologies but when exploring the visualization it is difficult to understand why certain actions in the merging process took place and what their consequences are. Although this tool offers an interesting visualization approach, it loses some intuitive aspects of a graph visualization (e.g., hierarchical relations among concepts). Moreover, by using one single visualization for representing both ontologies, the properties of the source ontologies lose their explicitness, which may be sufficient for merging but impedes the alignment of ontologies. OLA displays all details about the concepts, properties, and instances of the ontology at once without any overview functionality. Given more than 20 entities - which is quite a small number - the user can view only parts of the graph and the context is lost. Currently, the visualization is limited to showing only one ontology at a given time. For the alignment visualization the authors plan to display two of such graphics next to each other. Too many overlapping edges and labels are a current challenge of OLA. However, the main problem is that the graph gets cluttered because one visualization covers all the details without any zooming functionality. A growing number of visualization plug-ins for Protégé are available (e.g., OntoViz tab [19], Jambalaya [20], TGViz [21], OWLViz [22], ezOWL [23])

indicating the importance of visualization approaches for the development and management of ontologies. However, the lack of detail-and-overview functionality is an open issue for all of these existing tools and especially relevant when applied in ontology alignment. One possible solution is clustering of entities such as used in the small world visualization. In the next section we describe this cluster-graph approach in more detail and introduce our visual ontology alignment tool.

3 Alviz - a tool for visualizing ontology alignment

We developed a prototype for visual ontology alignment called AIViz. It is implemented as multiple-view plug-in for Protégé in order to support the visual alignment of two ontologies by making the type of similarity between entities explicit. The tool consists of two types of views coupled by the linking and brushing technique described in the context of our tool in more detail later on. AIViz applies J-Trees as one out of two types of views. Such trees consist of a root node, expanded or collapsed branch nodes and leaf nodes displaying the hierarchical structure by indentation. They support the access and manipulation of instances and single items within classes quite effectively and are well established within the Protégé community. But such J-Trees bear shortcomings regarding the representation of large or complex ontologies because they become cluttered and do not provide adequate overview functionality. To overcome this problem we integrate another visualization type: small world graphs [8]. Therefore, as a second view, graphs help the user to examine the structure of the ontology intuitively. This method uses clusters to group the nodes of a graph according to the selected level of detail. The nodes represent the entities (concepts or instances) connected to each other according to the selected relations, also called mutual properties, such as `ISA`, `IsPart`, `IsMember`, `locatedIn`, `IsDistinct`. So, each source ontology is visualized as a clustered graph

where the edges represent the selected mutual property (or a cumulation of properties is possible as well). An interesting issue is how to handle entities, which do not apply the selected property, so that they are not connected to any other entity. One possible solution is to show such entities by unconnected nodes in peripheral areas of the graph window. Alternatively, such entities are listed in the tree view only because they are not relevant in terms of the visualized relationship. AlViz implements both solutions with the first as default emphasizing on the selected relation as a hierarchical structure of the whole ontology. If most nodes are connected to a graph the user gets additional information about the relevance of the selected property within the ontology.

When aligning ontologies visually the users are interested in the global properties of the data such as: Are there any distinct groups of items that are strongly interconnected (i.e., graph clusters)? How do these split into separate clusters and how do these clusters relate? Small world graphs originating from [24] in social networks are graphs, which have a small average path length (average shortest path between nodes) compared to their number of nodes, but have a high degree of clustering compared to a random graph of the same size. Informal evaluation of structures of different ontologies showed the same property, which makes clustering a suitable approach. An adequate visualization for clusters should communicate the underlying structure efficiently. The number of visible elements in the graph needs to be reduced, but at the same time we shall maintain the global structure of the graph. That means to find a representative visual abstraction of the original graph. Although the small world graphs like all spring-embedded algorithms bear the problem of high computational complexity - usually $O(N^3)$, clustering the graph improves program's interactivity. The tool is fast enough to perform at interactive speeds because on average there are only $O(\log(N))$ clusters visible. Our current solution manages up to about 1000 entities per ontology. There is ongoing research [25] to reduce complexity of such graphs, which looks quite promising for visualizing graphs with some ten thousands of nodes.

Each ontology is visualized by both views, the J-Tree and the small world graph, resulting in four linked views making the ontologies available to the user on different levels of detail-and-overview. Figure 2 shows AlViz visualizing the alignment of two ontologies about tourism with a high degree of details. Clustering the nodes like depicted in figure 3 the visualization brings the hierarchical structure of the graph out. The color figures are available on the website <http://www.ifs.tuwien.ac.at/~mflanzenberger/alviz/graphics>. Both figures visualize the same source ontologies on dif-

ferent levels of detail. They show two stages of navigation. The colors of the nodes and the dots next to the entities' names represent the degree of similarity or type of association respectively. We distinguish six categories of association between entities: equal, syntactically equal, broader-than, narrower-than, similar and different. The input file representing these associations is discussed in more detail later on in this section. By default an entity of one ontology equal to an entity in the other ontology is colored red; a syntactically equal entity is colored orange; an entity broader than an entity in the other ontology appears blue; a narrower entity is colored violet, a similar entity is colored green; and finally, an entity different from all entities in the other ontology is colored yellow. In the graphs the clusters of nodes inherit the colors of the underlying nodes in accordance to the selected comparison strategy. We distinguish three different comparison strategies: the first one emphasizes similarity of entities, the second highlights differences among entities, and the third represents the entities according to the dominant type of associations. If the user focuses on similarity, the color of nodes indicating more similarity has priority against the color of nodes with less similarity. Particularly, clustering an equal and a syntactically equal node into one single node results in a red colored cluster. Of course, there are also higher levels of detail possible, maintaining two differently colored nodes as individuals but combined within one cluster. However, this case is simple. But if the cluster is represented by just one node, a priority approach is necessary. Furthermore, a cluster consisting of several different associations needs to be distinguished from a cluster representing just one single association. This is realized by introducing a saturation/brightness factor related to the amount of differing nodes. With the HSB color model, the color is identified by hue combined with values for saturation of the color and brightness. We distinguish five levels of this factor's intensity according to the amount of differing nodes. Customization of the tool allows for changing this encoding schema.

Interacting with the graph involves a number of possible tasks. Beside the zooming functionality, we implement a selection/highlighting function, several alignment functions, and tracking buttons for questionable associations. Selecting a node with a mouse click results in highlighting three other items. In particular, the entity (or group of entities) of the same ontology is highlighted in the tree view. While navigating the nodes (entities) in ontology O_1 , the associated nodes in O_2 are highlighted. This interaction technique is known as linking and brushing. If the association is equality then the equal entity (or group of entities) in the other ontology is highlighted as well. This results in linking together entities between both ontologies,

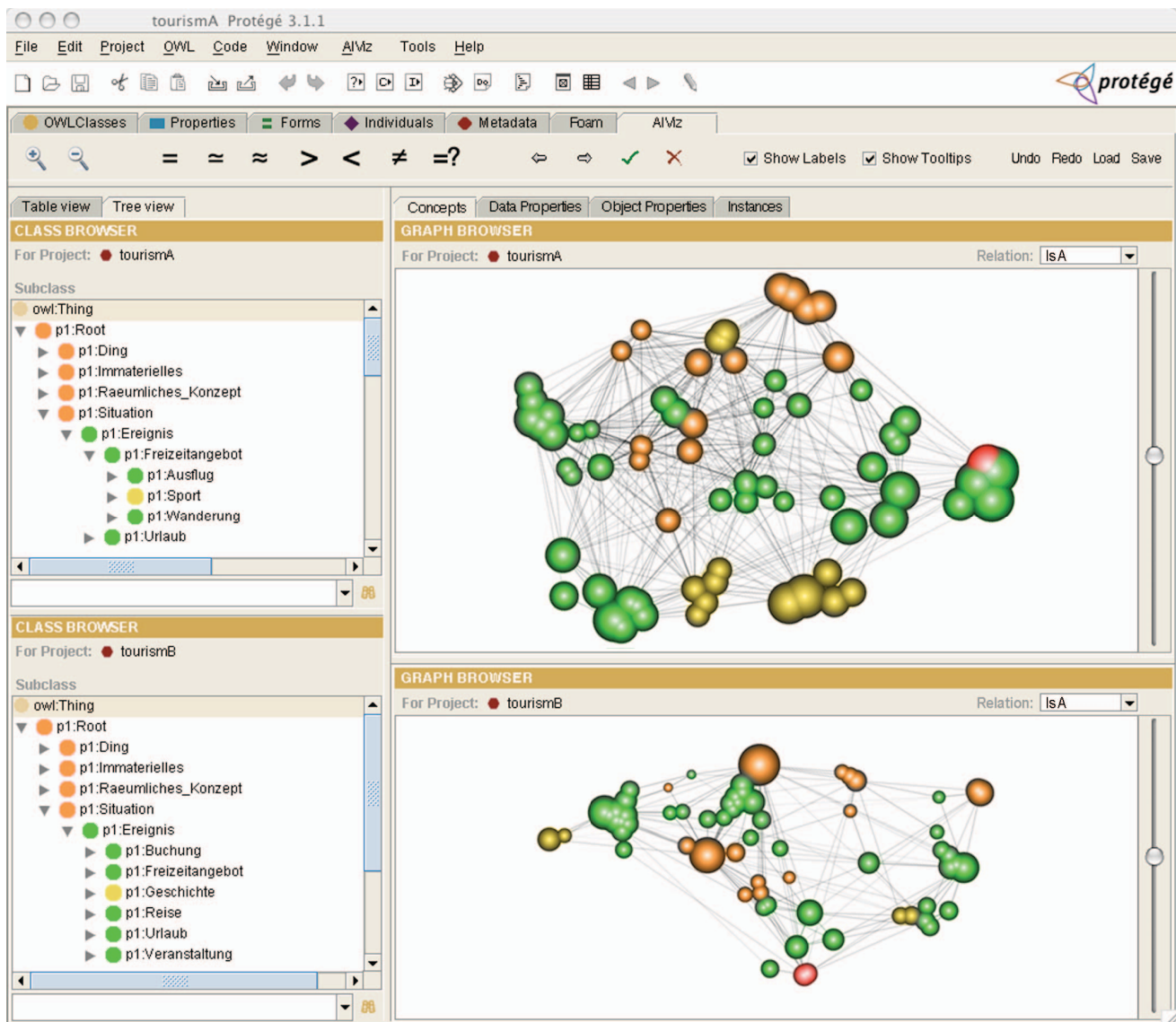


Figure 2: AIvz: the four views of the tool visualize two ontologies named tourismA and tourismB. The nodes of the graphs and dots next to the list entries represent the similarity of the ontologies by color. The size of the nodes results from the number of clustered concepts. The graphs show the *IsA* relationship among the concepts. Green indicates similar concepts available in both ontologies, whereas red nodes represent equal concepts. The sliders to the right adjust the level of clustering.

O_1 and O_2 , in both views the graphical (over)view and text-based (detailed) view. The same holds for the other types of association, i.e., syntactical equality, broader-than and narrower-than relations, and similarity. The alignment functions allow for adapting the automatically derived associations by assigning the type of association manually and thus approving or rejecting the alignment result. The user can select an entity in the graph or in the text list. Then she or he activates the required type of association, e.g., equality, followed by a selection of the associated entity (or group of entities) within the other ontology. Such changes of the alignment are rather complex. However, the interactive manner of the graph visualization makes this task easier and more manageable to the user. Undo and redo functions combined with a history of applied interactions, labeling and tooltips are included in the tool. AIViz supports the user in understanding the alignment process and manipulating its results in order to improve and maintain its quality. In terms of perception such small world graphs utilize features of the Gestalt tradition. In particular, the Gestalt Law of Organization: 'When we are presented with a number of stimuli, we do not as a rule experience a number of individual things ... Instead larger wholes ... are given in experience, their arrangement and division are concrete and definite' [26]. The Gestalt psychologists thought that the way in which parts of a figure are influenced by such factors as proximity, good continuation, and closure, reflects a natural tendency toward good forms in our experience [27]. Beside features like position, size, and shape the color is an outstanding property, which supports the perception of items belonging together. Another aspect is explained by Asch's Concept of Unit Formation: It is easier to remember two or more objects, if you see them as one unitary pattern. Gestalt psychologists believed the reason is that unitary patterns are much more coherent perceptually, and can be remembered as units. Switching among different levels of clustering the user perceives the various groups of nodes as units, which helps to remember the position of the individual entity. However, a detailed discussion of the perceptual features of AIViz exceeds the space of this paper.

AIViz is implemented as a tab plug-in for Protégé with Java based on a available solution [28], which transforms the original small world algorithm from 3 to 2 dimensions. Further optimization of the graphs is necessary in order to perform more user tests. For the calculation of the alignment values we have extended the general alignment process of [1] to include transformation, evaluation and visualization of the output results. We claim that it is necessary to transform the output from lists of entity pairs into a format that can be understood and evaluated by domain experts. Furthermore, a lot of information regarding how entities

are related between ontologies is hidden when just examining lists of entity pairs. We adapted Foam [29] an ontology alignment algorithm for generating candidate alignment relations between entities in two ontologies in order to generate the input files using XML Schema to store the alignment data. The Foam algorithm uses a large number of rules for calculating similarity between entities. Out of a possible 100 rules, we selected the 23 most important rules for indicating similarity. The reader may refer to [30] for similarity rules applied in Foam. The resulting similarities or associations between entities provide evidence that two entities are the same (or similar) and can potentially be aligned. The similarity function combines different features of the ontologies with appropriate similarity measures. As well as testing for equality, different similarity measures are used to calculate how similar entities in two ontologies are. Adapting the 22 individual rules from Foam we define six categories representing the associations between entities of two ontologies. Output from the Foam algorithm in a text file describes: `entity 1, label 1, entity 2, label 2, overall confidence, individual rule and each correct value`. The overall confidence represents the aggregated individual similarity values, the individual similarities correspond to the 23 different ontology features which are further categorized in the six groups listed above. The correct value corresponds to whether the alignment is correct or not with respect to identity. After generating the input file the user can select a property using a drop-down list, which shows all available relations (i.e., mutual properties) of the ontologies involved. Based on the selected property, the OWL source files of both ontologies, and the input file holding the categories of associations AIViz generates the visualization file. Finally, this file is used as input to draw the nodes and edges of the graphs and the entries of the J-Trees. Using XML Schema for the visualization file we aim at making the data available for other tools as well.

The development of AIViz is an ongoing process and first results of an informal evaluation indicate that the tool is suitable for ontologies up to 1000 entities. With the small world graph approach it shares its good functionality of making semi-structured data accessible on different levels of details. This features help the user when aligning ontologies visually.

Conclusion and future work

Understanding semantic relations between entities of different ontologies is a cognitively difficult task [31]. In order to align domain ontologies an ontology developer or end user is required to define the mapping between the ontologies either manually or by using a semi-automated tool. Ontology mapping is largely a human-mediated pro-

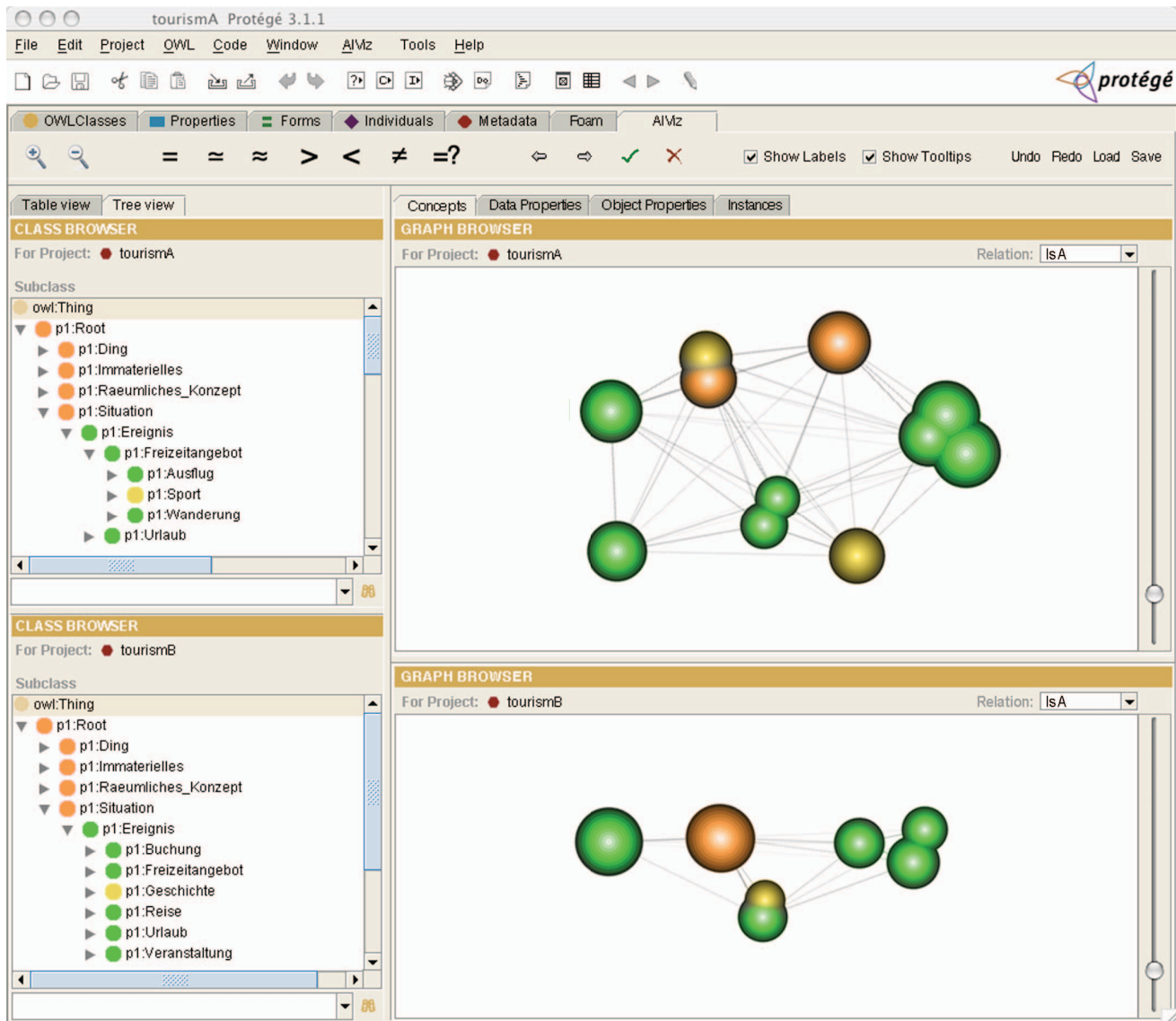


Figure 3: ALViz showing a highly clustered view of the alignment results for ontologies tourismA and tourismB. Different concepts are colored yellow representing entities which occur in only one of the two source ontologies. Orange represents syntactically equal concepts.

cess, although there are numerous tools, which can help with identifying differences between ontologies. The reason being that most conflicts or mismatches discovered by such tools require a human to recognize that different symbols have been used to represent the same concept, or that the same symbols may sometimes represent different concepts. [32] comment that in an ideal situation the easiest way to answer questions about two different ontologies is to have the designers of the two original ontologies get together and merge them. However, in practice this approach is not always feasible. Instead, during the establishment of aligned concepts, the user performing the alignment will have to make the decisions. Existing ontology alignment tools do not adequately provide a way for users to analyse the results of an ontology alignment. While many alignment tools generate mappings between entities it is difficult to analyse and validate these alignments without examining every pairwise correspondence in the output files and even then it is an overwhelming task. We propose the use of visualization techniques to graphically display data from ontology mappings to facilitate user understanding of the meaning of the ontology alignment. Our tool, Alviz, is to help the user determine the following: 1) Location: Where do most of the mappings between ontologies occur? 2) Impact: Do the mapping choices directly or indirectly affect parts of the ontology the user is concerned about? 3) Type: What kinds of alignments occur between the ontologies? 4) Extent: How different is the aligned ontology to the source ontologies? By exploring such questions in a multiple-view visualization the user may be able to understand and enhance the alignment results. Future work includes the implementation of additional selection mechanisms such as to 'lasso' a cluster with the cursor instead using a mouse-click; to improve the positioning of labels, and integrate specific zooming mechanisms like the fisheye lens [10]. Moreover, we will add export functionality using standard formats like JPEG and Scalable Vector Graphics (SVG).

Acknowledgements

We thank Markus Rester for his great support with the graphics. This work is partially supported by the Norwegian Research Foundation in the Framework of Information and Communication Technology (IKT-2010) program - the ADIS project.

References

- [1] Ehrig, M. and Sure, Y.: *Ontology Mapping - An Integrated Approach*. In: Proc. of the First European Semantic Web Symposium. C. Bussler, J. Davis, D. Fensel, R. Studer, eds, 2004.
- [2] Sowa, J.F.: *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.
- [3] Crubézy, M. and Musen, M.A.: *Ontologies in support of problem solving*. In: Handbook on Ontologies, S. Staab, and R. Studer, eds, 321–342. Springer, 2004.
- [4] Knowledgeweb Project.: *D.2.2.3: State of the art on ontology alignment*. KWEB/2004/D2.2.3/v1.2. 2004.
- [5] Card, S.K., Mackinlay, J.D. and Shneiderman, B., eds.: *Readings in information visualization*. Morgan Kaufman, 1999.
- [6] Tufte, E.R.: *The Visual Display of Quantitative Information*. 2nd Edition. Graphics Press, 2001.
- [7] Lamping, J., Rao, R. and Pirolli, P.A.: *A focus + context technique based on hyperbolic geometry for visualizing large hierarchies*. In: Proc. of ACM CHI'95, 1995.
- [8] van Ham, F. and van Wijk, J.J.: *Interactive Visualization of small world graphs*. In: Proc. of IEEE Symposium on Information Visualization 2004. CPress. 199–206, 2004.
- [9] Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., and Crow, V.: *Visualizing the non-visual: Spatial analysis and interaction with information from text documents*. In: Proc. of IEEE Information Visualization '95. IEEE Computer Press. 51–58, 1995.
- [10] Furnas, G.: *Generalized fisheye view*. In: Proc. of the ACM Conf. on Human Factors. ACM Press. 16–23, 1986.
- [11] Pirolli, P., Card, S.K. and van der Wege, M.M.: *Visual information foraging in a focus + context visualization*. In: Proc. of the ACM Conf. on Human Factors in Computing Systems. 506–513, 2001.
- [12] Tweedie, L., Spence, R., Dawkes, H. and Su, H.: *Externalising abstract mathematical models*. In: Proc. of the ACM SIGCHI 1994. ACM Press. 406–412, 1994.
- [13] Baldonado, M.Q., Woodruff, A. and Kuchinsky, A.: *Guidelines for using multiple views in information visualization*. In: Proc. of AVI 2000. ACM Press. 110–119, 2000.
- [14] North, C. and Shneiderman, B.: *A taxonomy of multiple windows coordinations*. Technical Report CS-TR-3854, 1997.

- [15] Perrin, D.: *PROMPT-Viz: Ontology Version Comparison Visualizations with Treemaps*. M.Sc. Thesis, University of Victoria, 2004.
- [16] Euzénat, J., Loup, D., Touzani, M. and Valtchev, P.: *Ontology Alignment with OLA*. In: Proc. of the 3rd EON Workshop, 3rd Intl. Semantic Web Conference, 2004.
- [17] Noy, N.F. and Musen, M.A.: *The PROMPT suite: interactive tools for ontology merging and mapping*. In: Int. Journal of Human-Computer Studies, Vol. 59, no. 6. 983–1024, 2003.
- [18] Shneiderman, B.: *Tree Visualization with Tree-maps: A 2-D Space-filling Approach*. ACM Transactions on Graphics (TOG) 11(1), 1992.
- [19] *OntoViz tab: Visualizing Protégé Ontologies*. url: <http://protege.stanford.edu/plugins/ontoviz/ontoviz.-html> last visited: Mar. 19, 2006.
- [20] Storey, M.-A. D., Musen, M.A., Silva, J., Best, C., Ernst, N., Fergerson, R., and Noy, N.: *Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé*. In: Proc. of K-CAP-2001. Victoria, B.C., Canada, 2001.
- [21] Alani H.: *TGVizTab: An Ontology Visualization Extension for Protégé*. In: Proc. of Knowledge Capture, K-Cap'03, Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, 2003.
- [22] Horridge, M.: *OWLVIz*. url: <http://www.coode.org/downloads/owlviz/OWLVIzGuide.pdf> last visited: Mar. 19, 2006.
- [23] Electronics and Telecommunications Research Institute: *ezOWL*. url: <http://iweb.etri.re.kr/ezowl> last visited: Mar. 19, 2006.
- [24] Milgram, S.: *The small world problem*. Psychology Today. 2. 60–67, 1967.
- [25] Newman, M.E.J.: *Fast algorithm for detecting community structure in networks*. In: Phys. Rev. E, 2004.
- [26] Wertheimer, M.: *Laws of organization in perceptual forms*. In: A source book of Gestalt psychology, Humanities Press, New York, 1967.
- [27] Coren, S. and Girgus, J.S.: *Principles of perceptual organization and spatial distortion: The Gestalt illusion*. In: Journal of Experimental Psychology (Volume 6), 1980.
- [28] Ingram, S.F.: *An Interactive Small World Graph Visualization*. University of British Columbia, Technical Report, 2005.
- [29] *Foam: Framework for Ontology Alignment and Mapping*. url: <http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/>, last visited: Mar. 19, 2006.
- [30] Ehrig, M., Haase, P., Stojanovic, N. and Hefke, M.: *Similarity for Ontologies - A Comprehensive Framework*. In: 13th European Conference on Information Systems, 2005.
- [31] Noy, N.F.: *Ontology management with the prompt plugin*. In: Proc. of the 7th Int. Protégé Conference, Bethesda, Maryland, 2004.
- [32] Noy, N.F. and Musen, M.A.: *Evaluating Ontology-Mapping Tools: Requirements and Experience*. In Proceedings of the Workshop on Evaluation of Ontology Tools at EKAW'02 (EON2002), Siguenza, Spain, 2002.