Human-Mediated Visual Ontology Alignment

Monika Lanzenberger[1] and Jennifer Sampson[2]

[1] Institute of Software Technology & Interactive Systems, Vienna Univ. of Technology, Austria
lanzenberger@ifs.tuwien.ac.at
[2] Norwegian University of Science and Technology, Trondheim, Norway
sampsonj@idi.ntnu.no

Abstract. We develop a multiple-view tool called AlViz, which aims at supporting the ontology alignment process visually. Combing views on several levels of abstraction, the tool tries to make the 'relatedness' between entities accessible. Based on a literature study we identified relevant phases emerging in ontology alignment. We extended a general alignment framework in order to reflect the adoption of visualization techniques. This framework builds the background for our user study. We evaluate visual ontology alignment with AlViz in three stages: (1) Participative software development, (2) usability evaluation, and (3) utility study. The evaluation methods proved viable even though our study design is challenging.

Keywords: Ontology Alignment, Information Visualization, Evaluation.

1 Introduction

Ontologies represent a shared meaning of domain knowledge. Being enabler for many applications such as data integration, agent communication, peer-to-peer systems, e-commerce, and semantic web services, ontologies provide content represented in a form that is easily machine-processable. With an increase in the development of ontologies comes the need for tools and techniques for solving heterogeneity problems between different ontologies. Therefore we need ontology alignment, which helps us to bring different knowledge representations into mutual agreement. But this is largely a human-mediated process, although there are existing tools which can help with identifying differences among ontologies. User interaction is still essential in order to control, approve, and optimize the alignment results. However, evaluation of human-mediated components in ontology alignment is an open question. To that end we describe the tasks involved when doing visual ontology alignment. This is our starting point when we develop a user study in order to investigate the utility of a visual ontology alignment tool called AlViz [1].

Based on a literature study we identified relevant phases emerging in ontology alignment. We extended this phases in order to reflect the adoption of visualization techniques and then described the relevant steps of our framework for visual ontology alignment. This framework builds the background for our user study. The user study enables us to find answers to the following questions: Do the users find missing alignments? Do the users identify weak alignments? Do the users understand the relatedness of both ontologies?

M.J. Smith, G. Salvendy (Eds.): Human Interface, Part II, HCII 2007, LNCS 4558, pp. 394–403, 2007.

With this paper our contribution is twofold. First we describe a core framework listing the relevant phases in visual ontology alignment. Second we develop a user study based on this framework in order to evaluate visual ontology alignment tools.

In the next section we describe the tool AlViz. Thereafter, we explain the extended alignment process and give a practical example of alignment assessment. In section 5 we discuss the evaluation methodology and the user study in more detail.

2 AlViz - A Tool for Visual Ontology Alignment

Visual metaphors encode large, multi-dimensional data sets effectively and support human interpretation and understanding. Information visualization (InfoVis) uses such visual metaphors in order to communicate the relevant information in an intuitive way and to foster new insights into the underlying processes [2] and patterns. InfoVis aims at making complex data accessible for interactive investigation by the user. 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 [3].

We develop a prototype for visual ontology alignment called AlViz [1]. It is implemented as a multiple-view plug-in for protégé-owl [4] in order to visually support alignment of two ontologies by making the type of similarity between entities explicit. The tool consists of different views coupled by the linking and brushing technique. J-Trees [5] are one type of the views applied in AlViz. 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. However, J-Trees have several 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 [6]. In a second view, such graphs help the user to examine the structure of the ontology more intuitively. This method uses clusters to group the nodes of a graph showing a specific level of detail. The nodes represent the entities (concepts or instances) connected to each other according to the selected relations, also called object properties, such as `IsA`, `IsPart`, `IsMember`, `locatedIn`, `IsDistinct`. So, each source ontology is visualized as a clustered graph, where the edges represent the selected relations among entities. A combination of object properties or relations is possible and interesting as well. In section 4 we discuss an example involving different types of relations. The Foam algorithm [7] is used to get preliminary alignment results which are visualized in AlViz.

Figure 1 depicts AlViz showing the alignment of two ontologies about tourism with a high degree of detail and one specific entity marked with a blue shadow in all four views. In particular, the concept `Ereignis` (event) is highlighted in both ontologies. Clustering the nodes would emphasize the overall structure of the graph. The colors of the nodes and the dots next to the entities' names represent the degree of similarity or type of association. We distinguish six categories of association between entities:

’equal’, ’syntactically equal’, ’similar’, ’broader than’, ’narrower than’, and ’different’. Based on the results of the mapping algorithm, we generate the input file for AlViz.

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. A variety of interaction functionality is implemented in AlViz in order to conveniently support the exploration and assessment tasks [1].

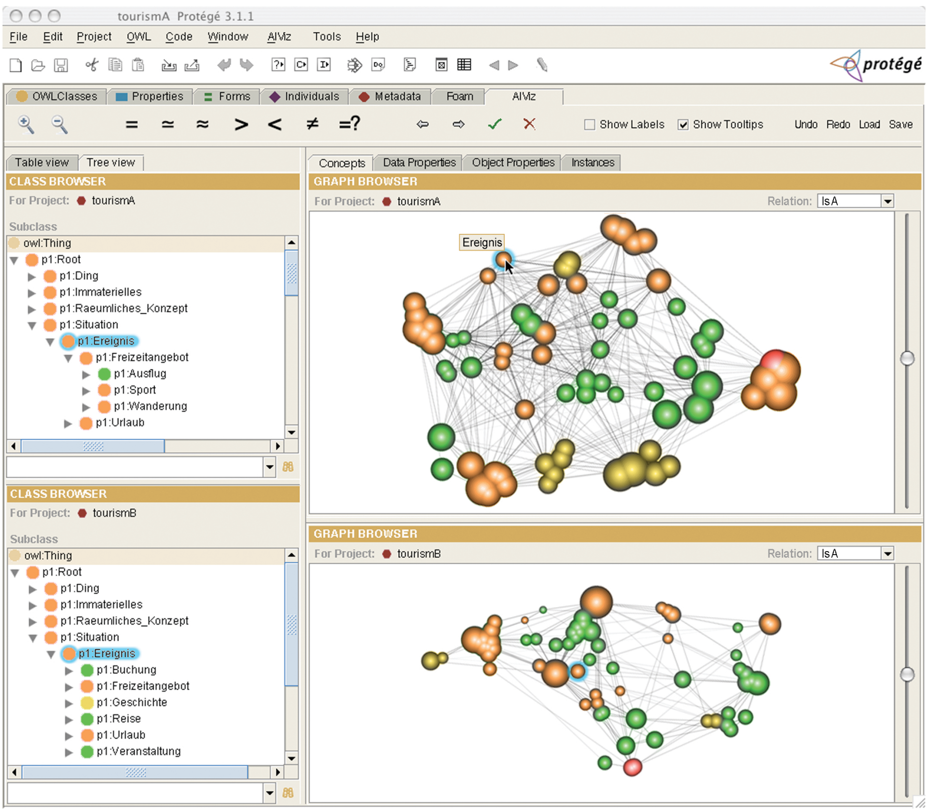


Fig. 1. AlViz [1]: the four views of the tool visualize two ontologies named tourismA and tourismB. The nodes of the graphs and dots next to the list entries represent the similarity of the ontologies by color. The size of the nodes results from the number of clustered concepts. The graphs show the `IsA` 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. One concept (called ’Ereignis’) is highlighted in all views and shown with the tooltip text in tourismA.

For the calculation of the alignment values we have extended the general alignment process of Foam [8] 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. The extended alignment process is described in more detail in the next section.

The development of AlViz is an ongoing process and first results of our 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 feature helps the user when aligning ontologies visually.

3 Ontology Alignment

The basic elements of ontologies are concepts, properties, and instances. They describe (by necessary conditions) or define (by necessary and sufficient conditions) a knowledge domain for a certain purpose. Ontology alignment means to analyze these elements or entities in order to identify overlapping parts in two ontologies.

According to Ehrig [8] entities are the same, if their features are the same or similar enough. In ontologies features represent a certain meaning of an entity. Beneath we list some examples of such features. If an alignment algorithm determines low similarity among entities of two ontologies, this may not give evidence for an alignment. But if high similarity is found we may have strong evidence. In order to estimate the similarity we need to consider the features of the neighborhood because entities may be the same if their neighborhood is the same or similar enough. That means entities are also defined by their position in the ontological structure. Structure similarity is expressed through the similarity of the other entities in the structure.

In a first round of the alignment algorithm only basic comparison methods such as string similarity are applied or even manually pre-defined alignments are used. In further rounds already computed pairs are the basis to measure structural similarity. However, not every type of similarity is of equal importance. Therefore, we need to judge and emphasize the relevant similarities. The alignment process is executed iteratively and it stops either after a fixed number of iterations, a specific time, or the number of newly identified alignments is below a certain threshold defined beforehand. Like [8] we used ten iterations which is efficient and appropriate in practice. Thereafter, in the second part human interaction is involved. Here, the results are visualized and presented to the user. She or he investigates and perhaps corrects the alignments in an iterative manner as well. This may be followed by an automatic re-calculation of the similarities going back to the first part. In this case the manually changed alignments are additional input for the algorithm such as the pre-defined entity pairs. The user decides whether or how often this re-calculation is done. In the following we describe such a general alignment process (based on the work of [8]). We extended this process to include the human-mediated alignment steps.

1. Alignment algorithm
   (a) Feature engineering and compatibility: Determining the input ontologies for alignment. When a choice of ontologies can be made, the user may assess the ontologies for compatibility with respect to the scenario and complexity of the ontologies, and also validate the ontologies prior to alignment. Additionally, in this step we extract characteristics of both ontologies, i.e., the features of their ontological entities. Such features are for instance ontology language primitives (e.g., range and domain definitions for properties), identifiers (e.g., unified resource identifiers - URIs), derived features exploiting the inheritance structure, and external features (e.g., additional text files describing the instance data). So, when measuring similarity this features will allow for comparison of entities.
   (b) Search step selection: Two initial entities from the two ontologies are chosen for comparison. Every entity pair may be compared or only an efficient subset may be chosen using heuristics to lower the number of mappings [8]. We can reduce the number of candidates by comparing entities of the same type only.
   (c) Similarity assessment: Based on the features of the selected candidate pairs we do the similarity computation. As stated in [8] comparing two ontology entities goes far beyond the syntax level. It considers their relation to the real world entities they are representing (their meaning), as well as their purpose in the real world (their usage). Therefore, a semiotic view on ontologies is used and the similarity measures are classified into three layers: data-, ontology-, and context layer. On the first layer data types such as integers, strings etc. are compared by operations like relative distance and edit distance. Second, on the ontology layer we measure the similarity of ontological objects such as comparing the domains and ranges of two properties. Finally, we consider the context: for example two books may be similar if their authors have many coauthored publications. For all three layers we take domain knowledge into account, e.g. domain specific vocabularies like Dublin Core in the bibliographic domain.
   (d) Similarity aggregation: The individual measures from the previous step are input to the similarity aggregation. They are weighted and combined using a sigmoid function. This emphasizes high individual similarities and de-emphasizes low individual similarities.
   (e) Determination: The aggregated numbers, a threshold and an interpretation strategy are used to generate the actual alignments. Borderline alignments are marked as questionable. The threshold is used to determine whether similarity is high enough to align the candidate pair. Everything above is aligned, lower values indicate different entities.
2. Human-mediated alignment
   (a) Visualization: A visualization tool gets the input from the algorithm and renders the preliminary alignments. Each ontology is shown as a detail and overview visualization capable of displaying a high number of items.
   (b) Clustering and navigation: The user changes the level of detail according to her or his needs. For validation the user selects individual items which are connected in the different views by linking and brushing.

(c) Competency questions: To check whether specific alignments make sense the user collects competency questions which should be answered by the aligned ontologies. Moreover, results of earlier competency questions are reviewed again.
(d) Validation: Appropriate changes may be made to the alignments based on the visualization analysis. The user may verify questionable alignments, either by accepting or rejecting the suggestions. Moreover, all alignments can be subject to changes if the user decides to assign other types of alignments.
(e) Ontology update: Ontology alignment may involve changes to one or both source ontologies. In order to get better targets for the similarity assessment the user can create additional entities. Before such ontology updates are done the user needs to consider the usage of the source ontologies carefully.
(f) Set similarity threshold: The algorithm's threshold used for identifying alignments is a crucial value. Sometimes it is necessary to adapt the threshold in order to get better alignment results. If the user finds many missing alignments and the preliminary results derived from the algorithm show a high amount of different entities this could indicate that the threshold is too high.

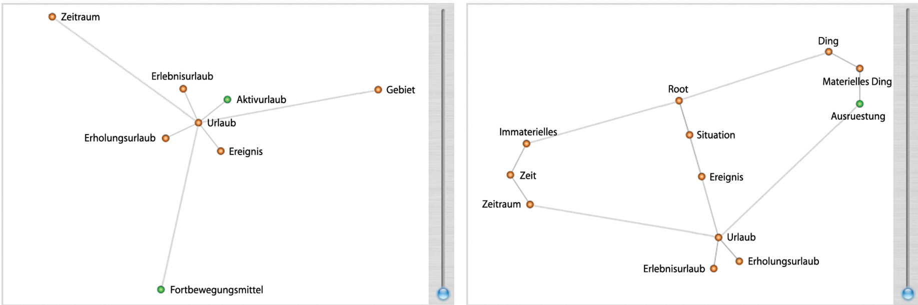
These phases are not applied in a linear process but rather the alignment results are re-fined when users iteratively work through the various thematic or hierarchical subareas of the ontologies on different levels of detail. The next section gives an example of the human-mediated alignment process.

4 Interpretation of Alignments

To ease the understanding of the process only parts of the two source ontologies are visualized. AlViz (compare Fig. 1) looks a bit different because it arranges the two ontologies horizontally and provides interaction functionality. However, to describe an example of a verification task practically, it is appropriate to reduce the complexity of the user interface. So, besides the data, only the clustering sliders are depicted. Figure 2(a) shows a subgraph of the first ontology called tourismA.

Sometimes it is not necessary to see the whole ontology - clustered or not. Moreover, too many visual objects not relevant for a specific tasks may impede exploration. Therefore, the user can focus on a certain entity, so that the graph only represents the related entities, emphasizing the context of this specific entity. This view includes all sub-entities (transitive relation) and directly related entities (non-transitive relation), supplemented with all relations and entities among them within a beforehand defined number of hops (relations). The second source ontology (tourismB) is visualized in Fig. 2(b).

In this particular example, the labeled nodes represent concepts, the edges represent three different types of relations, and the entity 'Urlaub' (vacation) is focussed. In tourismA the depicted relations are: `IsA`, `HasPeriod` (hatReisedauer), `GoesTo-Region` (hatZiel), `UsesVehicle` (hatReisemittel). The `IsA` paths are shorter than the other because we gave these edges a higher weight. In order to distinguish different types of relations such as functional, transitive, or non-transitive we apply different weights, which can be modified by the user according to the exploration needs.



(a) Subgraph of ontology tourismA

(b) Subgraph of ontology tourismB

Fig. 2. Small world graph visualizations of two ontologies in the tourism domain: the focus of the graph is on the entity 'Urlaub' (vacation) showing all related concepts for both ontologies. Labeling is activated, the colors represent the type of alignment based on the results of the alignment algorithm: orange means, there is at least one syntactically equal entity in the other ontology; green indicates similarity with at least one entity of the other ontology based on the neighborhood. In this view the nodes are not clustered meaning each node of the graph represents one entity.

In a simpler version of graph visualization the user may select one type of relation only. However, showing all relations of the central entity at once gives an impression of its context. On the right hand side the subgraph of tourismB consists of the relations: `IsA`, `HasAPeriod` (hatEineDauer), `EquipmentNeeded` (manBenoetigtAusruestung), `HasAGoal` (hatEinZiel). Whereas tourismA has three subclasses of the central concept, tourismB has only two.

Although the entities `Erholungsurlaub` (recreation) and `Erlebnisurlaub` (adventure) are identified as syntactically equal in both ontologies, the third called `Aktivurlaub` (active vacation) in tourismA with no directly corresponding entity in tourismB, was found to be similar to another entity by the alignment algorithm. By highlighting this concept we perceive the associated concept in the other ontology. It is `Erlebnisurlaub`. Based on the similar object properties which are inherited from the superclass (`Urlaub`), on the one hand, and a smaller distance (in terms of characters to be changed) between `Aktivurlaub` and `Erlebnisurlaub` than between `Aktivurlaub` and `Erholungsurlaub`, on the other hand, the alignment algorithm calculates the association which says `Aktivurlaub` and `Erlebnisurlaub` are similar. Similar concepts are colored green, whereas syntactically equal concepts are of orange color.

Obviously, in this subgraph the algorithm found many syntactically equal concepts but just a few similar concepts. For the moment we ignore the other green nodes and investigate the concept of `Aktivurlaub` only. The graphs show that the neighborhood of `Aktivurlaub` in tourismA and of `Erlebnisurlaub` in tourismB are similar which strengthens the claim that both concepts are closely associated. But `Erlebnisurlaub` in tourismA is also related to `Erlebnisurlaub` in tourismB: both concepts are syntactically equal. Now the user needs to decide whether changes of one or both ontologies are required and confirm or reject the alignment of type 'similar' between `Aktivurlaub` in tourismA and `Erlebnisurlaub` in tourismB.

Although our application example is static, the process of exploration is a highly interactive process. In terms of our alignment framework this example involves steps of clustering/navigation and validation. In practice the exploration of alignments is more complex as described in the previous section. However, to give an idea how interpretation of ontology alignments could be done, this simplified example is complex enough.

5 Evaluation Methodology and Study Design

The evaluation of AlViz should on the one hand yield information on its quality. But the proposed methods of evaluation should on the other hand give indications on the question which specific features of the visualization supports the users in the ontology alignment process. The use of diverse evaluation methods enables different views on the technology under investigation. We will use heuristic evaluation, thinking aloud, observation, software logs, interviews, focus groups, and benchmarks.

We evaluate visual ontology alignment with AlViz in three stages: (1) Participative software development, (2) usability evaluation, and (3) utility study.

The first stage is a formative evaluation done together with two ontology engineers who are involved in the development of the software. In general, the development of new methods and their implementation is a time-consuming task and involves different users having different types of models of understanding. Norman [9] pointed out that the mental models programmers have about software systems can differ fundamentally from the models the users have. Making these different models explicit can help to solve problems in software development projects at an early stage. In the AlViz project potential users (two ontology engineers) are integrated into the development process to a high degree. This enabled us to get informal feedback about usability and utility in an early stage.

Second, usability evaluation is done in order to identify problems and enhance the user interface of the software. We use Heuristic Evaluation [10]. It is well known usability evaluation method. A small group of evaluators (usually experts in the field of usability) checks the conformity of a given software with 10 usability principles. These principles are of a very general nature and therefore called "heuristics".

In the third stage we adopt a bundle of methods capable of evaluating the utility of visual ontology alignment software. Two ontology engineers (domain experts) and ten computer science students (semi-experts who attended two ontology engineering classes with excellent grades) do the evaluation. First, about one week before we conduct the evaluation, the subjects receive an introduction to the software tool using two simple ontologies to get a first impression. They get information about the functionality of the visualization tool, the data, and the alignment results derived from the algorithm. This is done with all subjects at the same time and takes around 30 minutes.

For the evaluation itself we include log functionality in AlViz in order to monitor the user activities. Only simple activities are logged: time stamps, view numbers, entity names, clustering levels, and validation actions. The advantage is that software logging does not interrupt the user's activities because it is not visible for the user. But it has shortcomings and therefore we need additional methods. For the individual evaluation these are: thinking-aloud, observation, and interviews.

One part of the utility evaluation is individual testing of AlViz for 60 minutes. Subject's overall goal in ontology alignment is bringing the source ontologies into mutual agreement and getting a maximum of correct alignments. Additionally, subjects are asked to report competency questions which arise in the process of aligning ontologies. A tutor and an observer attend the evaluation of each subject. The task of the tutor is to explain the evaluation procedure and to help if severe problems occur, which would make further evaluation impossible. The observer takes notes about important actions of the user, such as change of alignments or threshold, movements to other tabs, usage of linking and brushing and labeling, selection of views, execution of the alignment algorithm, etc. In addition, the subject is asked to describe her or his thoughts while using the tool. This method is called thinking-aloud (cf. [11]). The subject's voice is recorded and this gives more details about the intention of the subject. And finally, subjects report their competency questions in written form. This is necessary because subjects need to see and review their competency questions while doing the evaluation.

Immediately after the individual evaluation the observer changes the role and interviews the subject for around 10 minutes. The questions are: Did the subject have enough time to align the ontologies? Did she or he think the tool is appropriate for ontology alignment? How confident does the subject feel about her or his performance? Did the subject thinks she or he understood the knowledge represented by both source ontologies and the alignments?

After completing all individual evaluation procedures, the second part of the utility evaluation is done with the group of all subjects using the focus group method. Focus groups [12] often give reasons and additional subjective opinions. This also ensure correct interpretation of the results of other the evaluation methods. One major strength of this method lies in the interaction of participants. This can lead to valuable information on the discussed subject none of the participants would have stated in individual interviews. Once again a moderator and an observer are present. The moderator monitors the time and topic constraints and ensures equal chances of participation for all subjects. In a focus group setting the subjects discuss the strengths and shortcomings of the alignment tool. Afterwards, they get a list of all competency questions identified previously. They are asked to discuss and classify the questions in respect to their significance for representing the aligned ontologies. The following categories are given by the moderator: Crucial- , relevant- and further competency questions. Then the moderator asks each subject what helped and what hindered the finding of their competency questions during the individual evaluation. Finally, the subjects get a short questionnaire to give feedback about the evaluation. In the analysis of the evaluation we use the crucial competency questions as benchmarks for evaluating the alignment results of the subjects.

6 Conclusion

The tool implementation and the user study is an ongoing process. So far we have collected the data of 12 subjects and are currently analyzing the data. The evaluation methods proved viable even though our study design is challenging.

In particular, the competency questions reflect the users' understanding of the knowledge domains of both ontologies and the alignments. They allow for adequate handling of an open knowledge space. The benchmark evaluation of the alignment process results with the outcomes of the focus groups (rated competency questions) will give information on the quality of different problem-solving strategies. A combined interpretation of all the results will enable for a profound assessment of AlViz and furthermore provide knowledge on the ontology alignment process in general.

Acknowledgments. We are grateful to Markus Rester for his interesting inputs to our evaluation discussion. Furthermore, we thank Slaven Banovic for his help with the implementation of the software.

References

1. Lanzenberger, M., Sampson, J.: Alviz - a tool for visual ontology alignment. In: Society, I.C.S. (ed.) Proceedings of the IV06, 10th International Conference on Information Visualization, London, UK (July 2006)
2. Tufte, E.: The Visual Display of Quantitative Information. 2 edn. Graphics Press (2001)
3. Card, S., Mackinlay, J., Shneiderman, B. (eds.): Readings in information visualization. Morgan Kauffman, San Francisco (1999)
4. Stanford Medical Informatics: protégé-owl. Online `http://protege.stanford.edu/overview/protege-owl.html`
5. Sun Microsystems: Class J-Tree. Online `http://java.sun.com/j2se/1.5.0/ docs/api/javax/swing/JTree.html`
6. Van Ham, F., van Wijk, J.: Interactive visualization of small world graphs. In: Proceedings of the IEEE Symposium on Information Visualization 2004, pp. 199–206, CSPress, (2004)
7. Ehrig, M., Sure, Y.: Ontology mapping - an integrated approach. In: Bussler, C., Davis, J., Fensel, D., Studer, R. (eds.) Proceedings of the First European Semantic Web Symposium (2004)
8. Ehrig, M.: Ontology Alignment - Bridging the Semantic Gap. Springer, New York (2007)
9. Norman, D.: Cognitive engineering. In: Norman, D., Draper, S. (eds.) User Centered System Design, Lawrence Erlbaum, Hillsdale, N.J (1986)
10. Nielsen, J.: 2. In: Heuristic Evaluation, pp. 25–62. John Wiley & Sons, New York (1994)
11. Boren, T.M., Ramey, J.: Thinking aloud: Reconciling theory and practice. IEEE Transactions on Professional Communication 43(3), 261–278 (2000)
12. Kuniavsky, M.: User Experience: A Practitioner's Guide for User Research. Morgan Kaufmann, San Francisco (2003)