AN ANALYTICAL APPROACH TO ONTOLOGY MAPPING

***Meenakshi Rana, *Ashish Punya**

Department of Computer Science & Engineering,

KEC Ghaziabad

ABSTRACT

Ontology mapping is the key to data interoperability in the semantic web vision. Computing mappings is the first step to applications such as query rewriting, instance sharing, web-service integration, and ontology merging. This problem has received a lot of attention in recent years, but little is known about how users actually construct mappings. Several ontology-mapping tools have been developed, but which tools do users actually use? What processes are users following to discover, track, and compute mappings? How do teams coordinate when performing mappings? In this paper, we discuss the results from an online user survey where we gathered feedback from the community to help answer these important questions. We discuss the results from the survey and the implications they may have on the mapping research community.

Most existing ontology mapping tools do not provide exact mappings. Rather, there is usually some degree of uncertainty. We describe a framework to improve existing ontology mappings using a Bayesian Network. Omen, an Ontology Mapping ENhancer uses a set of meta-rules that capture the influence of the ontology structure and the semantics of ontology relations and matches nodes that are neighbours of already matched nodes in the two ontologies. We have implemented a prototype ontology matcher that can enhance existing matches between ontology concepts. Preliminary experiments demonstrate that Omen successfully identifies and enhances ontology mappings.

1. INTRODUCTION

Information sources, even those from the same domain, are heterogeneous in nature. The semantics of the information in one source differs from that in another. In order to enable interoperation among heterogeneous information sources or to compose information from multiple sources, we often need to establish mappings between database schemas or between ontologies. These mappings capture the semantic correspondence between concepts in schemas or ontologies. In recent years, researchers have developed a number of tools for finding these mappings in a semi-automated fashion (see Section 7 for a brief overview). In addition, there are interactive tools that enable experts to specify the mappings themselves. However, in most cases, the mappings produced are imprecise. For instance, automatic ontology-mapping tools can rank possible matches, with the ones that are more likely to be correct getting higher rankings. Most automatic ontology-mapping tools use heuristics or machine-learning techniques, which are imprecise by their very nature. Even experts sometimes could be unsure about the exact match between concepts and typically assign some certainty rating to a match. Once a particular set of mappings is established (by an expert or a tool), we can analyze the structure of ontologies in the neighbourhood of these mappings to produce additional mappings.

Our main premise in this work is the following: if we know a mapping between two concepts from the source ontologies (i.e., they match), we can use the mapping to infer mappings between related concepts. For example, if two properties and their domains match, then we can infer (with some certainty) that their ranges may be related as well. We build a Bayesian Net with the concept mappings. The Bayesian Net uses a set of meta-rules based on the semantics of the ontology relations that expresses how each mapping affects other related mappings. We can use existing automatic and semi-automatic tools to come up with initial probability distributions for mappings. Next, we use this probability distribution to infer probability distributions for other mappings. We have implemented a tool called Omen (Ontology Mapping ENhancer). Omen uses a Bayesian Net and enhances existing ontology mappings by deriving missed matches and invalidating existing false matches. Our preliminary results show that by using Omen we can enhance the quality of existing mappings between concepts across ontologies.

The primary contributions of this paper are as follows:

1. We introduce a probabilistic method of enhancing existing ontology mappings by using a Bayesian Net to represent the influences between potential concept mappings across ontologies.

2. In Omen, we provide an implemented framework where domain knowledge of mapping influences can be input easily using simple meta-rules.

3. We demonstrate the effectiveness of Omen in our preliminary experiments. To the best of our knowledge, no existing work has extensively used a probabilistic representation of ontology mapping rules and probabilistic inference to improve the quality of existing ontology mappings.

Ontology mapping is a complex and largely user-driven process that can benefit from tool support. In the past few years, researchers have developed many tools and techniques for creating ontology mappings. Tools include PROMPT, COMA++, Clio, Chimaera and OWL Lite Alignment (OLA). Much research has been spent on developing the algorithms used by these tools, and indeed the authors of cite more than 20 different algorithms that can be used to generate candidate mappings. However, in most cases, the mapping process cannot be fully automated and user input is required to accept, reject, or create new mappings. Despite the necessary role users play in the mapping process, there has been little work done to understand how and why users perform mappings. In order to design more effective tools and algorithms, we claim that a deeper understanding of the interplay between tool, user, and the process is needed. For example, who are these users that are going to use the tools? Why do they need to perform mappings and for which domains? Do they use the currently available tools and if so, how do they use them? And, do these tools meet their needs? To answer these questions, we designed a survey and gathered feedback from the ontology mapping community. To our knowledge, this survey is the first specifically designed with these goals. The information gained from this survey should be valuable to both tool and algorithm designers. For example, in part as a consequence of this survey, we believe that the biggest gains in mapping will not come from improvements in the precision and recall in matching algorithms, but rather from better tool support.

2. RELATED WORK

Specifying mappings between one or more ontologies is well recognized to be a challenging and complex process that can be made significantly easier through tool support. The typical mapping process is an iterative procedure whereby the tool presents to the user a set of candidate mappings and the user then decides to accept and reject some of those mappings. The process is repeated until the user is satisfied that the mapping is sufficiently complete.

Determining candidate mappings is a challenging algorithmic problem. Consequently much of the research to date has been expended on designing more efficient and effective algorithms for determining candidate mappings. But much of the mapping process involves a tight collaboration and coordination between the user and tool. For example, the user must decide which mappings to accept and reject, keep track of progress, and determine when enough mappings have been completed for the intended purpose. These tasks are cognitively challenging but can be made easier through an improved partnership with the tool during the mapping process. Despite the gains that can be made across the entire mapping process, little research has focused on improving the effectiveness of the user decision process. Notable exceptions include user studies with PROMPT and Chimaera, mapping experience reports and our own observational user study.

The user study experiment conducted with PROMPT concentrated on evaluating mapping suggestions provided by the tool by having several users merge two ontologies. The number of steps, suggestions followed, suggestions that were not followed, and what the resulting ontologies looked like was all recorded. Precision and recall was used to evaluate the quality of the suggestions. The experiment only involved four users, which was too small to draw generalizable conclusions. Independently, PROMPT was evaluated, along with Chimaera by Lambrix and Edberg with the purpose of merging ontologies from bioinformatics. Eight users were involved in the study, four with computer science backgrounds and four with biology backgrounds. The participants were given a number of tasks to perform a hard copy user manual, and the software's help system for support. They were also instructed to "think aloud" and an evaluator took notes during the experiment. Afterwards, the users were asked to complete a questionnaire about their experience. The tools were evaluated with the same precision and recall measurements as used in the previously described PROMPT experiment, while the user interfaces were evaluated using the REAL (Relevance, Efficiency, Attitude, and Learnability) approach. Under both criteria, PROMPT outperformed Chimaera, however, the participants found learning how to merge ontologies in either tool was equally difficult. The participants found it particularly difficult to perform non-automated procedures in PROMPT, such as creating user-defined merges.

Reed and Lenat reported on their experiences with manually mapping the CyC ontology to other ontologies over a period of 15 years. The process relied on trained ontologists collaborating with domain experts. Over time, interactive clarification-dialogbased tools were developed to help ease the mapping procedure. The authors believed that the major barrier to the adoption of ontology mapping is the heavy reliance on someone setting up the source schemas and access protocols. They also stated that better tools were needed in order to allow domain experts to perform mappings rather than relying on ontology experts.

Lomax and McCray described their experiences with mapping the Gene Ontology (GO) to the National Library's Unified Medical Language System (UMLS). The authors used a combination of methods to perform the mapping, starting with a preliminary exploration of both ontologies looking for overlap and then using an automated system to map 25% of the GO terms to UMLS terms. Following this, one of the authors visited the UMLS team for a month to work with the team in an attempt to complete the mapping. While many problems surfaced during this time, these were eventually addressed, debated and resolved. Through a combination of automated techniques, analysis and collaboration from the UMLS and GO teams, and mapping verification by humans, the GO ontology was fully mapped to UMLS.

In our previous work, we presented results from a user study where we observed teams participating in a "think-aloud" ontology mapping session with two different tools. The goal of this study was to gain a deeper understanding of the user needs and how they could be met more effectively through tool support. Although the results of the study were informative, we were left with questions that required feedback from the ontology mapping community. For example, users had trouble remembering what mappings they had created or verified while working with the tools. Also, the participants in the study were frustrated by not knowing how much of the mapping task they had already completed and what was left to be completed. The participants also had difficulty learning and working with the tools, which reinforces the findings reported in the Lambix and Edberg study. The participants in our study were not typical ontology mapping tool users and indeed were trained to use the tools before the study. Hence, we are interested in discovering if the problems encountered by our test users are also experienced by ontology mapping tool users with pragmatic and pressing needs for ontology mappings.

Despite some preliminary work in this area of understanding mapping tool users, we believe there is a lack of knowledge about the tools currently used, the users themselves, and the problems faced during the mapping process. Hence, we designed the survey that is presented in the next section of this paper to gain more insight into these questions.

Two research directions are related to our work: automatic or semi-automatic discovery of ontology mappings and the use of uncertainty in knowledge-based systems.

2.1 Automatic ontology mapping

Over the past decade, researchers have actively worked on developing methods for discovering mappings between ontologies or database schemas. These methods employ a slew of different techniques. For example, Similarity Flooding and AnchorPrompt algorithms compare graphs representing the ontologies or schemas, looking for similarities in the graph structure. GLUE is an example of a system that employs machine-learning techniques to find mappings. GLUE uses multiple learners exploiting information in concept instances and taxonomic structure of ontologies. GLUE uses a probabilistic model to combine results of different learners. Hovy describes a set of heuristics that researchers at ISI/USC used for semi-automatic alignment of domain ontologies to a large central ontology. Their techniques are based mainly onlinguistic analysis of concept names and natural-language definitions of concepts. A number of researchers propose similarity metrics between concepts in different ontologies based on their relations to other concepts. For example, a similarity metric between concepts in OWL ontologies developed by Euzenat and Volchev is a weighted combination of similarities of various features in OWL concept definitions: their labels, domains and ranges of properties, restrictions on properties (such as cardinality restrictions), types of concepts, subclasses and super classes, and so on. Finally, approaches such as ONION and Prompt use a combination of interactive specifications of mappings and heuristics to propose potential mappings. The approach that we describe in this paper is complementary to the techniques for automatic or semi-automatic ontology mapping. Many of the methods above produced pairs of matching terms with some degree of certainty. We can use these results as input to our network and run our algorithm to improve the matches produced by others or to suggest additional matches. In other words, our work complements and extends the work by other researchers in this area.

2.2 Probabilistic knowledge-base systems

Several researchers have explored the benefits of bringing together Nayes Nets an knowledge-based systems and ontologies. For instance, Koller and Pfeffer developed a "probabilistic frame-based system," which allows annotation of frames in a knowledge base with a probability model. This probability model is a Bayesian Net representing a distribution over the possible values of slots in a frame. In another example, Koller and colleagues have proposed probabilistic extensions to description logics based on Bayesean Networks. In the context of the Semantic Web, Ding and Peng have proposed probabilistic extensions for OWL. In this model, the OWL language is extended to allow probabilistic specification of class descriptions. The authors then build a Bayesean Network based on this specification, which models whether or not an individual matches a class description and hence belongs to a particular class in the ontology. Researchers in machine learning have employed probabilistic techniques to find ontology mappings. For example, the GLUE system mentioned earlier uses a Bayes classifier as part of its integrated approach. Similarly, Prasad and colleagues use a Bayesean approach to find mappings between classes based

on text documents classified as exemplars of these classes. These approaches, however, consider instances of classes in their analysis and not relations between classes, as we do. As with other approaches to ontology mapping, our work can be viewed as complementary to the work done by others.

2.3Knowledge Model

We assume a simple ontology model (similar to RDF Schema). We use the following components to express ontologies:

Classes: Classes are concepts in a domain, organized in a subclass–superclass hierarchy with multiple inheritances.

Properties: Properties describe attributes of classes and relationships between classes. Properties have one or more domains, which are classes to which the property can be applied; and one or more ranges, which restrict the classes for the values of property.

We use the following notation conventions through the rest of this paper:

– all concepts from O have no prime ('); all concepts from O0 have a prime (');

– upper-case C with or without a subscript is a class;

– lower-case q with or without a subscript is a property;

– P(C1 C2, x) indicates that the probability of the match (C1 C2) is x.

3.TOOL QUESTIONS

The first tool-related question asked which tools users had tried. Respondents could choose from seven tools: Chimaera, COMA++, FOAM, MoA Shell, OLA, PROMPT, and QOM. They could also list any other tools in the "Other" field. Each of the listed tools was used by at most one to three participants with the exception of MoA Shell, which none of the respondents used. As shown in Figure 4, no tool was particularly dominant. The bulk of the feedback came in the "Other" category, which had 17 participant responses. Other tools included Prot´eg´e, Internet Business Logic, AUTOMS, Crosi, WSMT with Ontostudio, X-SOM, OMAP, Falcon-AO, HMatch, and Snoggle. Each of these tools were used by only one participant, except X-SOM, which had been used by two.

Two participants indicated that they use a custom built solution, while one indicated that they use a completely manual process. We asked which tools and features participants found most useful and what deficiencies they found with the tools. Both Crosi and COMA++ were found to be useful because they integrate a large variety of similarity algorithms and are available online. One user indicated that they like tools to provide simple suggestions and automatic help, while another user had a contrasting view, stating that they like statistically-based tools because others require too much designer opinion. Prot´eg´e was highlighted as being good for manual creation of mappings as it makes it easy to create ontologies. Several participants pointed out that many tools are too general and are built without domainspecific mechanisms. One of the custom built solutions was indicated to be similar to PROMPT, but was built to take advantage of domain knowledge, specifically term normalization algorithms and synonyms for their domain of interest. The requirement for the tools to incorporate domain-specific analysis and features was a common theme in response to several questions in the survey. Another common theme was the lack of visual displays or easy to use tools. Specifically, one participant indicated that PROMPT's interface was too complicated to give to a domain expert to do the mapping. One user criticized specific tools for their lack of documentation, for being buggy, and for not working as described. Other interesting observations were the lack of precision and recall for mappings in real world ontologies and that the tools do not allow for expressive enough mappings (e.g. some tools only support 1-1 mappings).

In the final tool-related question, we asked the respondents to describe which features the perfect mapping tool would have. In the presentation of the survey, this question came at the end, but here we categorize it as a tool question. There were several interesting themes that came up in the responses. The first theme was that six of the 19 responses related to the desire for better and easier to use tools. Specifically, participants stated that they needed better interfaces, graphical cognitive support, improved user interactivity, and facilities for explaining manual mappings. Users highlighted a large number of desired features for the algorithms for generating candidate mappings: powerful and pluggable matching algorithms, recognition of re-occurring patterns of representing information, identification of not only simple correspondences but also of complex ones, and extending beyond mere word-pair assocations and semantics. Four of the responses stated the requirement for perfect precision of recall for the mapping algorithms. Three participants also indicated that they want better facilities for testing mappings and support for more expressive mappings. The final interesting theme was collaborative support. Most of the respondents indicated that they work on their mappings in teams (see process questions). Most available tools do not support this type of team development scenario.

3.1PROCESS QUESTIONS

The first process-related question asked whether the participants found the creation of an ontology mapping difficult. 30.8% replied "No" to this question, while 69.2% said "Yes". The follow-up question to those participants that answered "Yes", asked participants to explain why they found the process difficult. Ten of the 21 responses discussed semantic issues, such as the process being too subjective or ambiguous. One participant pointed out that the "semantics of the underlying ontologies are not usually well defined. Without a solid understanding of the semantics, it is almost impossible to perform the mapping correctly." Respondents also discussed a lack of domain expertise for performing mappings, and that "[y]ou have to get into the brains of the original developers of the ontologies being mapped." Participants also stated that tools are not flexible enough for application-specific needs, resulting in the manual creation of mappings, which is neither scalable nor efficient. One participant indicated that the OWL primitives for expressing mappings are poor and that users are faced with making difficult decisions when two related concepts "almost but not exactly match." Three participants also indicated that problems with creating mappings resulted from poorly designed and documented tools.

We next asked participants what process they followed when performing mappings. Available responses were "Tackle the easy matches first" (37.0%), "Focus on a familiar area of the ontology" (51.9%), "Rely on the tool's ordering of mappings" (14.8%), and "Other" (22.2%). Responses for the "Other" category included performing an automated matching up front and then a debugging step, while two of the responses indicated that they first applied lexical, then structural, and finally semantic methods.

In the next question, we asked how the participants remembered mappings they had created or verified. Most respondents chose from the provided answers, "The tool supports this" (37.0%) and "Paper and pencil" (55.6%), while 22.2% filled out the "Other" option. In the "Other" responses, one user indicated that the tool they use supports this, which works well when mappings are done in a single pass, but extra help is needed for multiple passes. Another respondent indicated that they use their own codes to report the mappings they create, which is similar to tracking the information by paper and pencil. Finally, one respondent indicated that they did not follow any good process.

It is interesting that the majority resort to tracking this information manually by paper and pencil. Similar types of changes exist in software development and most IDEs and source control systems handle the tracking of this data. We then asked when the participants considered the mapping to be complete. Ten of the 25 responses indicated that they used some form of testing (automated or manual) to verify that the mapping was completed to their satisfaction. For many respondents, this testing meant determining whether the mapping supports whatever application they were working on. Five responded that they knew the mapping to be complete when all concepts had been mapped. However, this implies either a perfect mapping, or that they knew when all reasonable concepts had been mapped. Interestingly, three participants responded that they never knew when the mapping was complete. Only one respondent indicated that they relied on tool support for determining whether the mapping was complete, although one participant stated that they must trust the system when mapping large ontologies because verification by hand is too slow.

We also asked participants about the types of problems they experienced while performing mappings. Similar issues outlined in previous questions came up again. Specifically, one respondent stated that “most ontology tools are difficult for business users to understand.” Testing the mapping was also a popular theme along with issues related to the problem that people model conceptualizations differently. The final two questions dealt with whether participants worked in teams and what sort of process they followed for coordinating their efforts. 53.9% indicated that only 1 or 2 people were involved in the mapping process, 42.3% worked in groups of 3 to 5 people, and finally, 3.9% worked with 6 to 10 people. Based on results from the team process question, we were able to determine that of the 53.9% working in teams of 1 to 2 people, 53.8% of these actually work in a team of 2, which means that only 24.0% work completely on their own. It is interesting that these users felt that there could be an automatic algorithm that they would trust completely.

The team-coordination processes ranged from weekly meetings to collaborating through wiki’s to coordinating through CVS. 18 of the 20 respondents relied on nonsoftware solutions for managing the team or a combination of ad hoc communication strategies like CVS, wiki’s and e-mail along with meetings. Smaller teams typically had one team lead and one implementer, and coordinated with face to face meetings. Participants also indicated that they worked as a group or partitioned the ontologies and then performed a group check to validate the mapping. Some teams used domain expertise for reviewing the composed mappings or during the mapping process for input. One respondent indicated that they use a “brainstorming” team process for coordinating the mapping effort.

(IJRST) 2015, Vol. No. 5, Issue No. III, Jul-Sep **e-ISSN: 2249-0604; p-ISSN:2454-180X**

COMPONENT REUSE BASED UPON ONTOLOGY MAPPING

Ontology mapping, which is an important part of ontology integration, can promote sharing and communication among different ontologies. The incorporation of ontology into software engineering can improve the reuse of software assets effectively . In recent years, it has become less likely to develop complete new software systems from scratch. It becomes very important to develop software by adapting or combining existing reusable components . We observe that requirement specification can provide a data source for ontology model and also the vital link for the combination of software engineering and ontology. With this insight, a software component reuse approach based on ontology mapping is formulated in Figure 1. The main idea is to process customer requirements and reusable components using ontology mapping techniques, and then construct the mapping between ontology nodes and reusable components. This approach can promote the reuse of software components. We can construct the target ontology model by analyzing requirement specification and then calculate the similarity between target ontology and source ontology using ontology mapping techniques. Therefore, we can identify the matched source ontology nodes and the corresponding sets of reusable components and then construct the mapping between the target ontology nodes and the reusable components. We assume that the mapping between source ontology nodes and reusable components has been realized, so every source ontology node has matched several reusable components. After the ontology mapping, the target ontology nodes also have matched the reusable components through the “bridge” of the source ontology nodes.

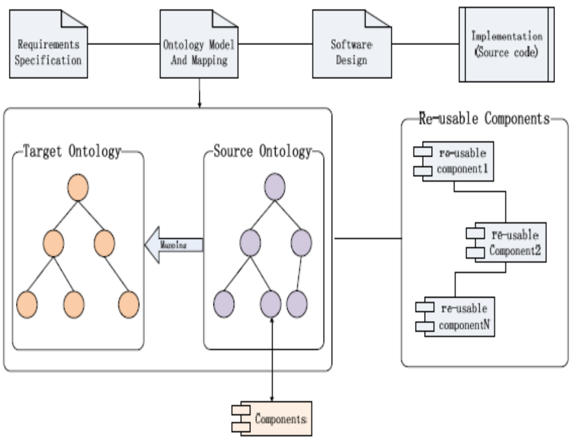

Figure.1. Software component reuse approach based on ontology mapping

CROM Algorithm: Through the above analysis, we find that the mapping between ontology nodes and reusable components is the key to realize the CROM (Component Reuse through Ontology Mapping) algorithm. We can construct the mapping by using requirement engineering. With requirement engineering, we will decompose requirement specification into several fragments and every fragment of requirement contains several functional points. Each functional point contains the input and output which are designed to match the requirement. The requirement and the corresponding input/output are the basis for the design and implementation of software components. Each node of ontology model is related to each fragment of requirement specification, and each fragment is related to several functional points so that each node of ontology model is also related to several functional points. This unique approach to construct the mapping between ontology node and reusable components by the functional point is the heart of the CROM algorithm. In general, the mapping operation is illustrated in Figure 2.

Ontology nodes and reusable components have N:N relationship. Every ontology node may correspond to several reusable components, and every reusable component may correspond to several ontology nodes.

Even though function description of customer requirement is the same, the attributes are often different, so it is difficult for each software component to completely meet different requirements of different ontology node. We need to calculate the matching degree between software component and ontology node. We consider the matching degree between a concept of the ontology node C and a reusable software component S to be a number P (C, S) between 0 and 1, with 0 representing unmatched and 1 representing completely matched. The

Formula is: P (C, S) = f (S) / f (C), with f (S) representing the matched functional points number of the current reusable software component, and f (C) representing the total functional points number of the current ontology node.

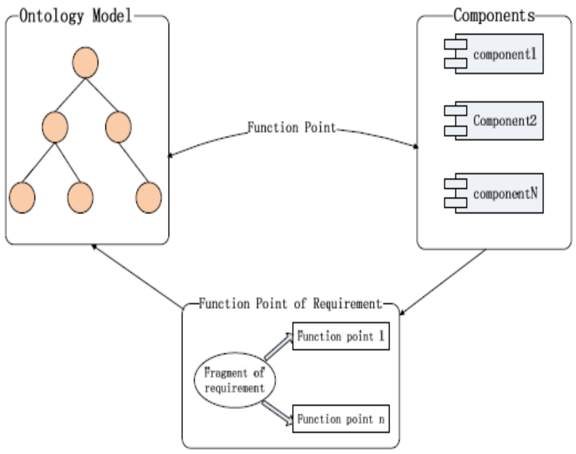

Figure.2. The mapping between ontology node and reusable components.

4. DISCUSSION OF RESULTS

We found it surprising how many tools had been tried by our respondents. There has clearly been a large effort from the research community to develop so many tools, yet there does not appear to be a dominant tool that is a benchmark for mapping tool design. This may be due in part to the variety of user needs. Some respondents highlighted that they had domain-specific needs or that existing tools do not support sophisticated enough mappings. Most of the problems, deficiencies, and issues with ontology mapping uncovered by the survey can be classified into one of two categories: fundamental issues with language and semantics, and tool-specific issues. Fundamental issues, such as different model conceptualizations and language ambiguity, are difficult, if not impossible problems to solve. It is interesting that some of the responses to the "perfect mapping tool" question were that the tool would have 100% precision and recall or full natural language understanding. While a perfect, fully-automated solution would be ideal, it is probably not realistic for any but the most simple, straight-forward mappings. As the survey also highlighted, many of the problems that people face in ontology mapping, are difficult for even a team of human experts to resolve.

It is also interesting that these users felt that there could be an automatic algorithm that they would trust completely. Issues of trust also came up in our previously mentioned user study. Although our users stated that they liked PROMPT's explanation facility, they were also often confused when it made a suggestion that was obviously wrong. Incorrectly generated candidate mappings would sometimes lead to our users ignoring the suggestions and switching to a completely manual process. Tool specific issues such as better user interfaces, graphical support, better testing facilities (data translation based on mappings), interactivity, algorithm explanation capabilities, and so forth are all problems that ontology-mapping tool developers can help with. As discussed in the results, one respondent indicated that PROMPT was too complex to give to their end-users. This sentiment was also echoed by a non-computer science participant in our user study. Mapping is a complex process, it is important that we do not further burden our users with learning a difficult to use software suite. Instead we must support them via the software. In agreement with Bernstein et al. , we believe that at this point the biggest productivity gains in mapping tasks will come from better cognitive support rather than from an improvement of precision and recall in matching algorithms.

The responses to the process-related questions brought up several interesting issues that tool designers and researchers may also need to address. First, it is noteworthy that many of the participants relied on paper and pencil to remember the mappings they perform. One individual even noted that some tools work for a first pass, but then they "forget" the steps previously performed during the second pass. Tool support should be able to address this issue. Second, it appears that most users work in small teams but tools currently lack support for team communication and collaboration, as well as for partitioning the mapping process into manageable chunks that can be tackled by individuals on the team. Many teams work together to validate the prepared mappings. Again, tool support could help with team work. Metadata annotations (perhaps visualized via color-codinng) could be used to record who composed the mapping and why they made certain decisions. In addition, the ontology mapping community could borrow ideas from the Web 2.0 social networking community 3. E.g., tools could support voting on mappings, commenting on and annotating mappings, and associating instance data with a conceptualization. There has been some experimentation with communitydriven ontology mapping , but tool support is currently limited.

5. LIMITATIONS OF THE SURVEY

There are of course limitations to this study, the first and foremost being the sampling size and population. Although we feel that 28 respondents gave us a wide variety of interesting and useful responses, with such a small sample it is possible that our responses are biased. Also, as we solicited participants from mailing lists, most of which were academically oriented, our sample may be biased towards researchers in the area rather than a balance between those working in research and industry. Finding and recruiting users from our target population was also an issue, because it is difficult to know how to best reach ontololgy mapping tool users. As with any on-line survey, the wording of some of our questions may have potentially been confusing to some participants. For example, we asked “If you were to design your perfect mapping tool, what features will it have?”. We stated that some respondents indicated full natural language understanding and 100% precision and recall. Perhaps if the question had been worded differently to solicit feedback on a “realistic ideal mapping tool” the responses would have been different.

6 . CONCLUSION AND FUTURE WORK

In this paper, we discussed the design and results from our on-line user survey. The open-ended responses gave us valuable information about the types of problems users are experiencing, what features they’d like to see improved, some insight into their mapping and team process, and which tools are being used by the community. Several issues raised by our participants indicate that their problems could be solved by improved tools and this would lead to better mapping results as well as greater adoption of the various mapping tools. This area of research has seen little activity to date. In the future, we plan to continue gathering feedback from the user community and carry out observational user studies. Our goal is to develop a comprehensive theoretical framework for cognitive support in ontology mapping. It is our hope that this will help guide the design and evaluation of future mapping tools such that the user’s role in the mapping process is fully supported.

We have outlined the design and implementation of Omen, an ontology match enhancer tool, that improves existing ontology matches based on a probabilistic inference. This tool is dependent upon a set of meta-rules which express the influences of matching nodes on the existence of other matches across concepts in source ontologies that are located in the proximity of the matching nodes. We described how we implemented a simple first version of the matching tool and discussed our preliminary results. We have also outlined several improvements that can be made to the tool and identified several open questions that if resolved can make the performance of the tool even better.

7. ACKNOWLEDGEMENTS

I would like to thank to Prof. Mayank Singh, Head of Dept. Computer Science, Krishna Engineering College, Ghaziabad for their assistance in preparing and distributing the our review paper. I would also like to thank the practitioners who have kindly responded to our survey questionnaires. I thank also our college for motivating and encouraging doing my Research work in a Successful.

REFERENCES

1. H.-H. Do. Schema Matching and Mapping-based Data Integration. PhD thesis, Department of Computer Science, Universit¨at Leipzig, 2006.

2. S. M. Falconer, N. F. Noy, and M.-A. Storey. Towards understanding the needs of cognitive support for ontology mapping. In Ontology Matching Workshop, 2006.

3. S. M. Falconer and M.-A. Storey. Cognitive support for human-guided mapping systems. Technical Report DCS-318-IR, University of Victoria, Victoria, BC, Canada, 2007.

4. P. Lambrix and A. Edberg. Evaluation of ontology merging tools in bioinformatics. In Proceedings Pacific Symposium on Biocomputing, pages 589–600, Kauai, Hawaii, USA, 2003.

5. J. Lomax and A. T. McCray. Mapping the gene ontology into the unified medical language system: Research papers. Comp. Funct. Genomics, 5(4):354–361, 2004.

6. J. L¨owgren. Human-computer Interaction. What every system developer should know. Studentlitteratur, Lund, 1993.

7. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. The chimaera ontology environment. In Proceedings of AAAI, pages 1123–1124, 2000.

8. Z. Ding and Y. Peng. A probabilistic extension to ontology language owl. In 37th Hawaii International Conference On System Sciences (HICSS-37), Big Island, Hawai, 2004.

9. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In The Eleventh International WWW Conference, Hawaii, US, 2002.

10. J. Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-Lite. In The 16th European Conference on Artificial Intelligence

(ECAI-04), Valencia, Spain, 2004.