

# Chapter 2

## Interactive Techniques to Support Ontology Matching

Sean M. Falconer and Natalya F. Noy

**Abstract** There are many automatic approaches for generating matches between ontologies and schemas. However, these techniques are far from perfect and when the use case requires an accurate matching, humans must be involved in the process. Yet, involving the users in creating matchings presents its own problems. Users have trouble understanding the relationships between large ontologies and schemas and their concepts, remembering what they have looked at and executed, understanding output from the automatic algorithm, remembering why they performed an operation, reversing their decisions, and gathering evidence to support their decisions. Recently, researchers have been investigating these issues and developing tools to help users overcome these difficulties. In this chapter, we present some of the latest work related to human-guided ontology matching. Specifically, we discuss the cognitive difficulties users face with creating ontology matchings, the latest visual tools for assisting users with matching tasks, Web 2.0 approaches, common themes, challenges, and the next steps.

### 1 Introduction

As ontologies become more commonplace and their number grows, so does their diversity and heterogeneity. As a result, research on ontology matching has become a prominent topic in the Semantic Web and ontology communities. There are rigorous evaluations that compare the effectiveness of different algorithms [Euzénat et al. 2009], and researchers have proposed a standard matching language [Euzénat 2006]. As the results of the evaluations show, ontology matching is far from being a fully automated task. In most cases where high precision is required, manual intervention will be necessary to verify or fine-tune the matchings produced by the automatic algorithms.

---

S.M. Falconer (✉) and N.F. Noy  
Stanford University, Stanford, CA 94305, USA  
e-mail: [sean.falconer@stanford.edu](mailto:sean.falconer@stanford.edu), [noy@stanford.edu](mailto:noy@stanford.edu)

In many areas of science, researchers are investigating how best to pair human input with automated procedures. For example, in the area of intelligent robot design, some researchers believe that the future of the field lies not in the development of fully automated robots, but in the development of partially automated ones [Coradeschi and Saffiotti 2006]. Some tasks, such as classification and pattern recognition, are very difficult and robots need help from humans in performing these tasks. At the same time, robots can help humans with tedious and repetitive tasks. Similarly, in ontology matching, humans have access to vast amounts of background knowledge, which they can use to help make inductive judgments about potential correspondences.

In general, potential matching correspondences produced by a matching tool must be examined by a domain or ontology expert. The expert must determine the correspondences that are correct, remove false positives, and create additional correspondences missed by the automated procedure. This process is both time consuming and cognitively demanding. It requires understanding of both ontologies that are being mapped and how they relate to each other. Furthermore, both the ontologies and the number of candidate matching correspondences that the tools produce can be very large. Researchers have largely focused on improving the performance of the algorithms themselves. However, recently there has been a growing trend toward a more human-centered approach to ontology matching.

Examining and supporting the symbiosis between tool and user has been gaining more prominence and more tools that support a *semiautomatic* process are becoming available. Shvaiko et al. discuss ten challenges for ontology matching, three of which directly relate to the user: *user involvement*, *explanation of matching results*, and *social and collaborative ontology matching* [Shvaiko and Euzénat 2008]. One approach researchers have been exploring to help support user involvement is information visualization techniques, such as those used by AlViz [Lanzenberger and Sampson 2006] and COGZ [Falconer and Storey 2007b]. The *International Workshop on Ontology Alignment and Visualization*<sup>1</sup> was created as a platform for researchers to share and explore new visual techniques to support the matching process. Another growing trend is the use of Web 2.0 approaches to help support the social and collaborative matching process. Researchers are exploring the utility of *crowdsourcing* to help facilitate the process of generating many matching correspondences [Noy et al. 2008; Zhdanova 2005].

These new trends in ontology matching research offer an exciting and interesting alternative to completely manual or completely automated processes. The research emphasis is shifting. New research is investigating how to gain a better understanding of the cognitive demands placed on the user during a matching procedure, how communities of users can work together to create more comprehensive and precise matchings, and how to make the most effective use of automation. Research on these topic areas is still in its infancy, but the future of the field lies in a joint effort between human and machine.

---

<sup>1</sup> <http://www.ifs.tuwien.ac.at/?mlanzenberger/OnAV10/>.

In this chapter, we explore research and tools that support the visual and interactive ontology matching process. We begin by discussing the cognitive difficulties with creating an ontology matching (Sect. 2). In Sects. 3–5, we discuss interactive tools for ontology matching, schema matching, and Web 2.0 approaches. In Sect. 6, we present several user-oriented evaluations and experiments that researchers in this area have carried out. We discuss common themes in Sect. 7, challenges and future directions for this field in Sect. 8. We conclude the chapter in Sect. 9.

## 2 Why is Ontology Matching Difficult?

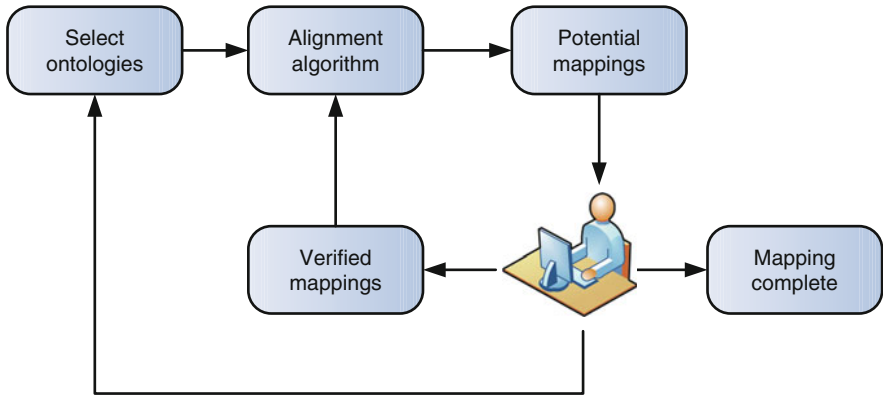
Reconciling different ontologies and finding correspondences between their concepts is likely to be a problem for the foreseeable future. In fact, every self-assessment of database research has listed interoperability of heterogeneous data as one of the main research problems [Bernstein and Melnik 2007]. Despite years of research on this topic, ontology and schema matching is far from being a fully automated task. In general, a user must interact with an ontology-matching tool to examine *candidate matchings* produced by the tool and to indicate which ones are correct, which ones are not, and to create additional correspondences that the tool has missed. However, this validation process is a difficult cognitive task. It requires tremendous patience and an expert understanding of the ontology domain, terminology, and semantics.

Obtaining this understanding is very difficult. Languages are known to be *locally ambiguous*, meaning that a sentence may contain an ambiguous portion that is no longer ambiguous once the whole sentence is considered [PPP 2006]. Humans use detailed knowledge about the world to infer unspoken meaning [NLP 2002]. However, an ontology often lacks sufficient information to infer the intended meaning. The concepts are largely characterized by a term or a small set of terms, which may be ambiguous.

The underlying data format that is used for specifying the ontology also introduces potential problems. The language used (e.g., OWL, RDF, XSD) constrains the expressiveness of the data representation. For example, many formats lack information relating to units of measure or intended usage [Bernstein and Melnik 2007].

Ontologies are also developed for different purposes and by users with potentially opposing world views or different requirements. As a result, two ontologies may describe the same concept with different levels of granularity or the same concept with different intended application or meaning. All of these issues make discovering and defining matchings a very challenging problem for both man and machine.

As a consequence, to create accurate matchings in a reasonable amount of time, users and tools must be paired together. This process, usually referred to as *semi-automatic ontology matching*, typically follows an iterative process that is similar to the one that we describe in Fig. 2.1. Recently, this approach has received greater attention and an increasingly larger number of semiautomatic tools are becoming



**Fig. 2.1** Example of semiautomatic matching process. A user is involved in iteration with the tool. As the user evaluates potential matching correspondences, their decisions are used by the tool to make other suggestions about correspondences. This iteration continues until the user determines the matching is complete

available (more discussion in Sect. 3). Beyond tool design, some researchers have started to carry out behavioral studies in an attempt to identify the cognitive difficulties with validating matching correspondences.

Falconer and Storey have used results from several studies to propose a “cognitive support framework” [Falconer and Storey 2007b; Falconer 2009] that helps guide the design of ontology matching tools. They also used their experiments to uncover several themes that describe human and tool limitations: human memory limitations, decision-making difficulty, problems searching and filtering for information, issues with navigating the ontologies, understanding the progress of the task, and trusting the results from the automated procedure [Falconer and Storey 2007a].

In another study, Yamauchi demonstrated that humans tend to bias their inductive judgments based on class-inclusion labels [Yamauchi 2007]. In this work, Yamauchi carried out several studies examining how human subjects classify properties and class-labels for randomly generated cartoon images. Using the results from these experiments, he drew several interesting conclusions directly relating to ontology construction and matching. For example, because people tend to overuse class-labels for comparison, even when other information is available, the impact between the similarity of concept labels between two ontological concepts may bias the decision made by user of an ontology matching tool.

Research exploring the cognitive challenges of resolving data heterogeneity is still very new. Such research provides theoretical foundations for the design and evaluation of ontology matching tools. In the next three sections, we provide a short survey of different tools and approaches for ontology and schema matching.

### 3 Existing Tools

Researchers have developed a number of tools that enable users to find matching correspondences between ontologies. For example, Euzenat et al. discuss more than 20 different algorithms and tools [Euzénat et al. 2004b]. In this section, we focus our discussion on semiautomatic tools that follow an iterative process that is similar to the one shown in Fig. 2.1. The user selects the ontologies to be mapped, an algorithm runs to compute an initial set of correspondences, the user interacts with the tool to validate the matching correspondences, and the tool uses this information to provide other possible matches. Some of the projects that we discuss in this chapter are no longer under active development; and some of the projects are still in the early research prototype phase and are not available for public use. However, each system provides an interesting example of the variety of approaches available for supporting semiautomatic ontology matching.

COMA++ [Do 2006] automatically generates matchings between source and target schemas (XML or OWL), and draws lines between potentially matching terms (see Fig. 2.2). Users can define their own term matches by interacting with the schema trees. Hovering over a potential correspondence displays a confidence level about the match as a numerical value between zero and one. COMA++

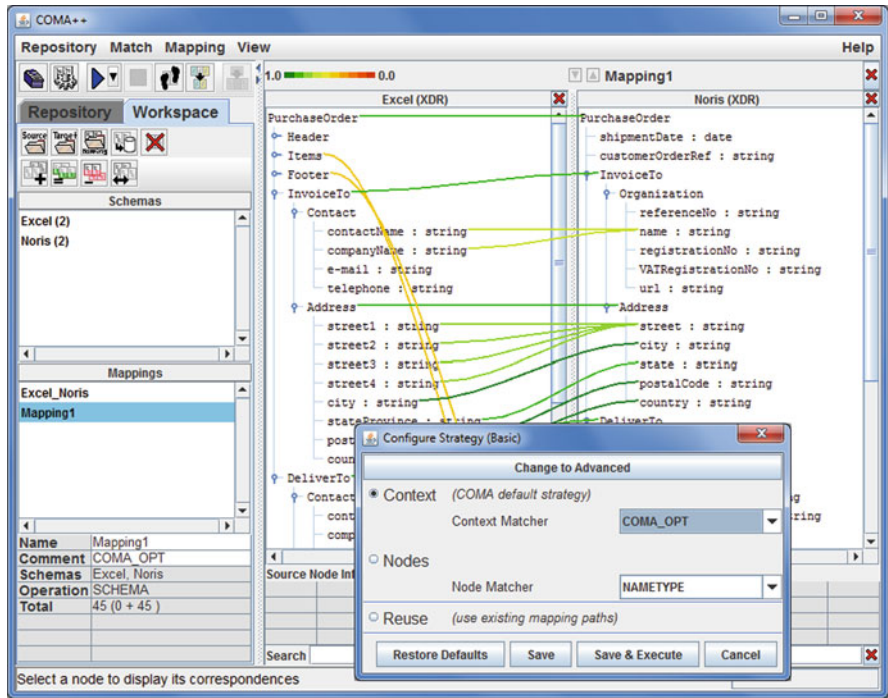


Fig. 2.2 Screenshot of COMA++ interface

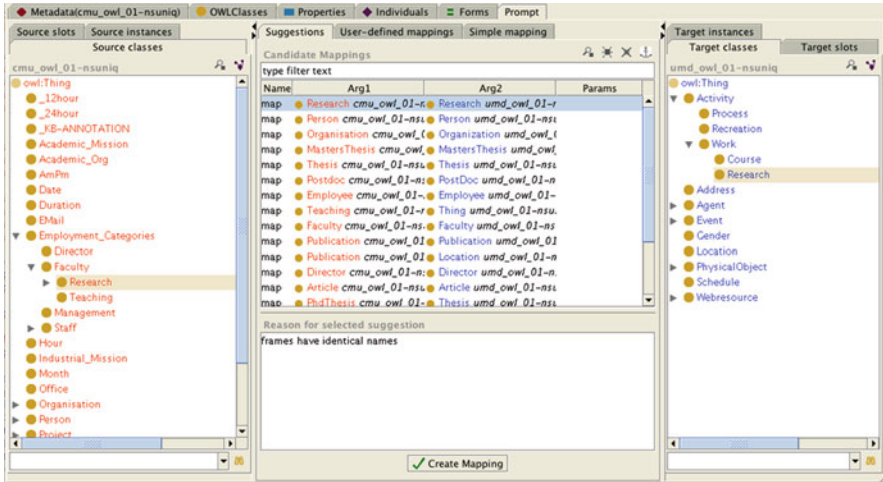


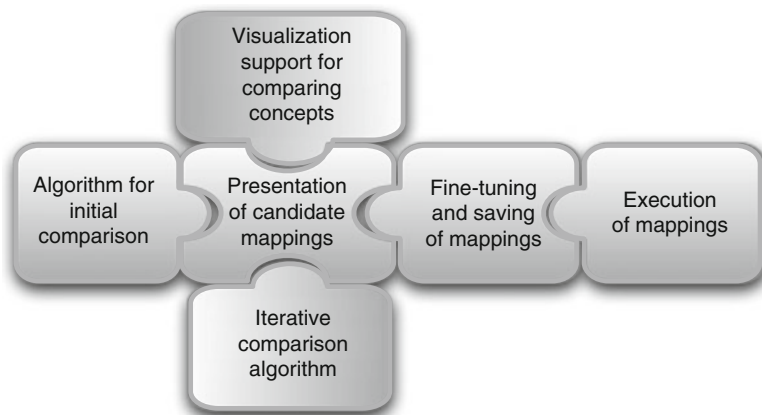
Fig. 2.3 Screenshot of PROMPT plugin while matching two university ontologies

contains several different matching algorithms in its library and the library is extensible. It also assumes interaction with a user: as a user approves of certain matches, COMA++ uses this information to make further suggestions.

PROMPT [Noy and Musen 2003] (see Fig. 2.3) is a plugin for the popular ontology editor Protégé.<sup>2</sup> The plugin supports tasks for managing multiple ontologies including ontology differencing, extraction, merging, and matching. PROMPT begins the matching procedure by allowing the user to specify a source and target ontology. It then computes an initial set of candidate correspondences based largely on lexical similarity between the ontologies. The user then works with this list of correspondences to verify the recommendations or to create correspondences that the algorithm missed. Once a user has verified a correspondence, PROMPT's algorithm uses this information to perform structural analysis based on the graph structure of the ontologies. This analysis usually results in further correspondence suggestions. This process is repeated until the user determines that the matching is complete. PROMPT saves verified correspondences as instances in a *matching ontology* [Crubézy and Musen 2003]. The matching ontology provides a framework for expressing transformation rules for ontology matchings. The transformation rule support depends on the matching plugin and ontology used. In the default matching plugin, the matching ontology simply describes the source and target correspondence components and metadata, such as the date, who created the correspondence, and a user-defined comment.

Like COMA++, PROMPT is extensible via its own plugin framework [Falconer et al. 2006]. However, while COMA++ supported extensibility only at the

<sup>2</sup> <http://protege.stanford.edu>.



**Fig. 2.4** Configurable steps in the PROMPT framework. Developers can replace any component in the figure with their own implementation

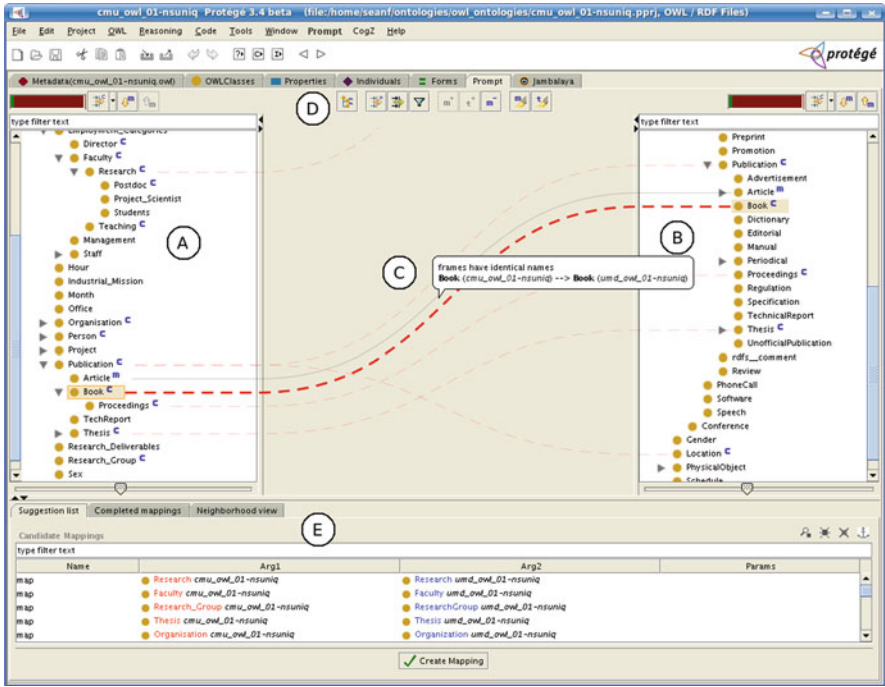
algorithm level, PROMPT supports a much more comprehensive set of extensions. It decomposes the matching process into several steps: an algorithm for comparison, the presentation of matching correspondences, fine-tuning and saving of correspondences, and execution of a matching (see Fig. 2.4). These steps represent plugin extension points in PROMPT: a new plugin can replace or augment any of these steps.

COGZ is an interactive visual plugin for PROMPT. Figure 2.5 presents the main COGZ interface. Like COMA++, COGZ uses a visual metaphor for the representation of matching correspondences. Candidate correspondences are represented by dotted, red arcs, while validated correspondences are represented by solid, black arcs. The tool supports incremental search and filtering of both source and target ontologies and generated correspondences. For example, as a user types in a search term for the source ontology, after each keystroke, the tree representation of the ontology is filtered to show only terms and hierarchy that matches the search criteria. Other filtering is available that allow a user to focus on certain parts of the hierarchy or help hide unwanted information from the display.

COGZ uses highlight propagation to assist users with understanding and navigating matching correspondences. When a user selects an ontology term, all matchings except those relevant to the selected term are semitransparent, while the relevant matchings are highlighted. To support navigation of large ontologies, a fish-eye zoom is available. The fish-eye zoom creates a distortion effect on the source and target trees such that selected terms are shown in a normal font size while other terms are shown progressively smaller depending on their relevance to the selected values (see Fig. 2.6).

Similar to PROMPT, AlViz [Lanzenberger and Sampson 2006] is a plugin for Protégé, however the tool is primarily in an early research phase. AlViz was developed specifically for visualizing ontology alignments. It applies multiple-views



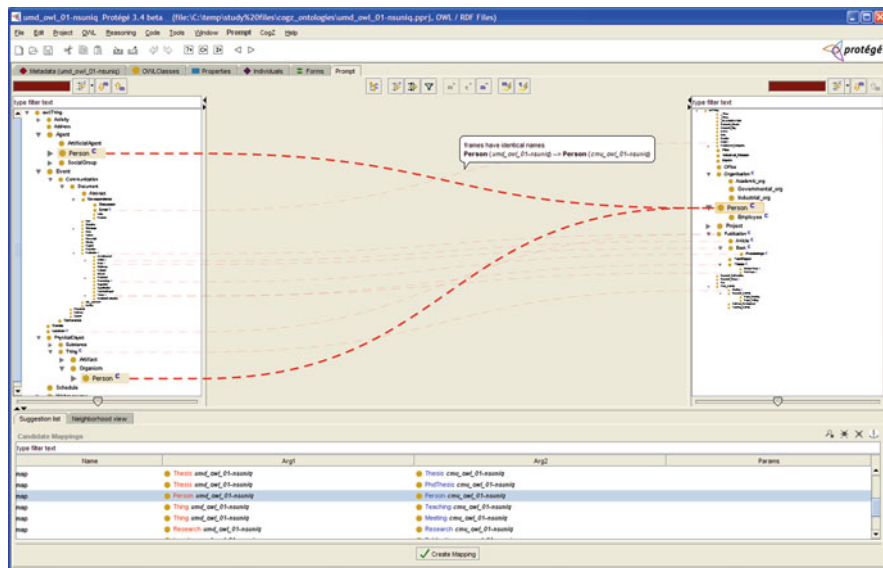


**Fig. 2.5** The COGZ perspective in PROMPT. (A) and (B) show the source and target ontologies. Concepts with “C” icons represent terms with candidate correspondences that were discovered automatically, while concepts with “M” icons (e.g., Article) are terms that have been validated and mapped. (C) shows a visual representation of correspondences. (D) shows the main toolbar. Each ontology has a set of buttons for applying filters, moving through the correspondences, and representing the overall progress. Finally, (E) shows three tabs. The first tab displays all the candidate or suggested correspondences found automatically. The second tab displays only the correspondences validated by the user. The final tab displays a side by side visual comparison between the concepts selected in the source and target ontologies

through a cluster graph visualization along with synchronized navigation within standard tree controls (see Fig. 2.7). The tool attempts to facilitate user understanding of the ontology matching results [Lanzenberger and Sampson 2006] by providing an overview of the ontologies in the form of clusters. The clusters represent an abstraction of the original ontology graph; moreover, clusters are colored based on their potential concept similarity with the other ontology.

OWL Lite Alignment (OLA) is a tool for automated matching construction as well as an environment for manipulating matching correspondences [Euzénat et al. 2004a]. The tool supports parsing and visualization of ontologies, automated computing of similarities between ontology entities, manual construction, visualization, and comparison of matching correspondences (see Fig. 2.8). OLA supports only OWL Lite ontologies and uses the Alignment API specified in Euzénat [2006] to describe a matching. The matching algorithm finds correspondences by analyzing





the structural similarity between the ontologies using graph-based similarity techniques. This information is combined with label similarity measures (e.g., Euclidean distance, Hamming distance, substring distance) to produce a list of matching correspondences.

The NeOn toolkit [Le Duc et al. 2008], developed as an Eclipse plugin,<sup>3</sup> is an environment for managing ontologies within the NeOn project.<sup>4</sup> NeOn supports run time and design time ontology matching support and can be extended via plugins. The toolkit includes a matching editor called OntoMap, which allows a user to create and edit matchings (see Fig. 2.9). Similar to the previously mentioned tools, NeOn supports OWL ontologies; however it also supports RDF and F-Logic. The toolkit can convert a variety of sources (e.g., databases, file systems, UML diagrams) into an ontology to be used for matching.

<sup>4</sup> <http://www.neon-project.org>.

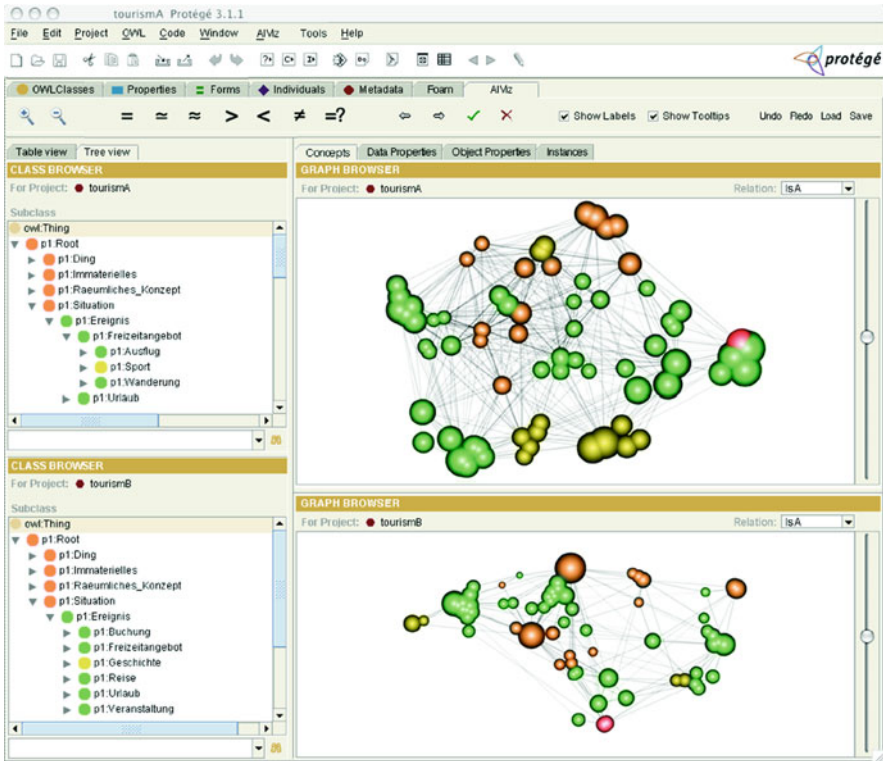


Fig. 2.7 Screenshot of AIViz plugin while matching two tourism ontologies [Lanzenberger and Sampson 2006]

These are just a few of the visual and interactive tools available for ontology matching. In the next section, we discuss similar tools that have been developed to support the related problem of schema matching.

## 4 Schema Matching

Typically in schema matching the goal is to map entities from relational database schemas, XML schemas, Web catalogs, or directories rather than entities of an ontology. While the process of schema matching is very similar to the process of ontology matching, there are some significant differences. For example, there are fundamental differences in terms of the representational semantics of a database schema versus an ontology. An ontology is a representation of the concepts for a domain of discourse, which is often developed independent from an application. A database or XML schema is usually modeled to represent data with a particular application in mind. Moreover, ontologies are often constructed and published

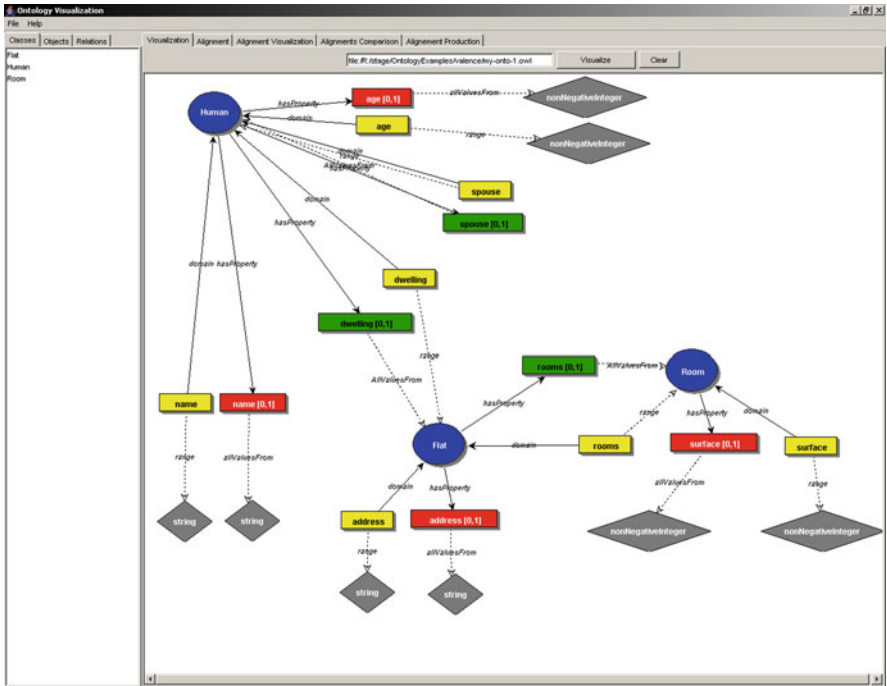


Fig. 2.8 Screenshot of OLA visualization of an OWL ontology

publicly with sharing in mind. In contrast, schemas are often internal to a particular application and are not available for others to consume or re-use. In terms of matching, the focus in ontology matching is usually to create semantic links between two independent ontologies that can later be used for various applications. With data-specific schemas, data translation or integration is often the focus. Thus, a lot of schema-matching tools support sophisticated methods for constructing transformation rules to translate data from a source schema to a target. Finally, while ontology matching has primarily been confined to research laboratories, there is a number of commercial tools available for schema matching. Microsoft, IBM, and Oracle are just a few of the companies that have commercial tools available.

Many of these tools have been developed through successful collaborations between researchers and industry. Clio, one of the first and most sophisticated schema matching tools, was a research prototype developed through a collaboration at IBM's Almaden Research Center and the University of Toronto [Miller et al. 2001]. Clio can automatically generate a view to reformulate queries from one schema to another or transform data from one representation to another to facilitate data exchange.

Like the previously discussed ontology matching tools, Clio proposes a semi-automatic approach and supports a visual matching representation similar to COMA++, CogZ, and OntoMapper (see Fig. 2.10). Users can draw arrows

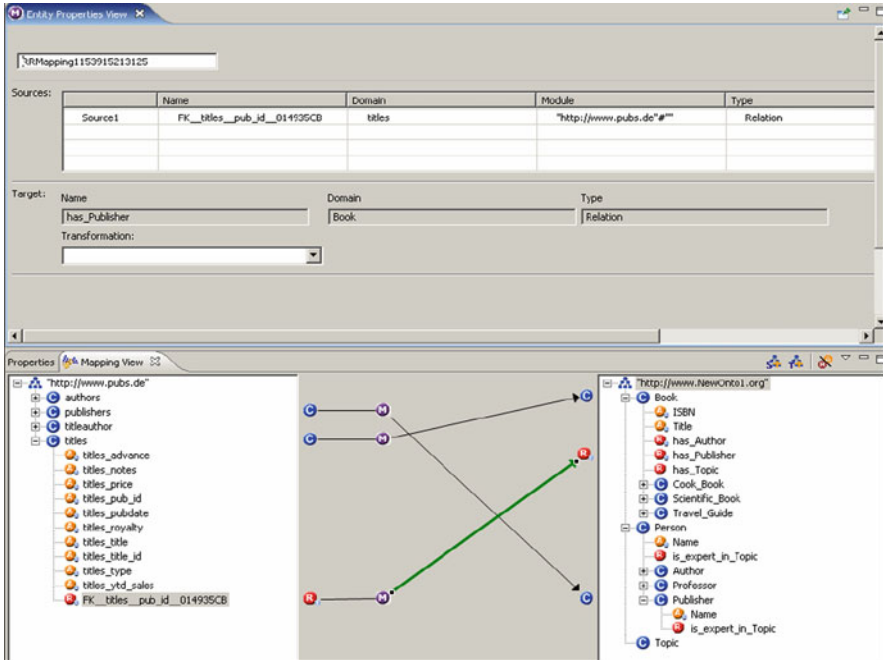


Fig. 2.9 Screenshot of NeOn toolkit matching editor [NE08 2008]

between the source and target schema elements and these arrows are interpreted as matchings and translated into a query. The heart of Clío is its incremental matching engine, which uses information about the matching that is input from a user to infer and re-rank correspondences. The Clío project has been in development since 1999, and a product version is now available as part of the Rational Data Architect.<sup>5</sup>

MapForce is part of Altova's XML suite of tools.<sup>6</sup> Similar to Clío, users can draw matching correspondences between the source and target schemas and these are used to automatically generate queries to support data integration. For XML and database matchings, the matching can be saved as XSLT, XQuery, or generated as programming language code (e.g., Java). MapForce supports a feature to "auto connect matching children." When a parent element is manually connected, children with the same name are automatically mapped.

Similar to MapForce, the Stylus Studio contains an XML matching tool that supports visual matching between XML, relational databases, and web service data.<sup>7</sup> Users can drag and drop lines between source and target elements and matching

<sup>5</sup> <http://www-01.ibm.com/software/data/optim/data-architect/>.

<sup>6</sup> <http://www.altova.com/mapforce.html>.

<sup>7</sup> [http://www.stylusstudio.com/xml\\_to\\_xml\\_mapper.html](http://www.stylusstudio.com/xml_to_xml_mapper.html).

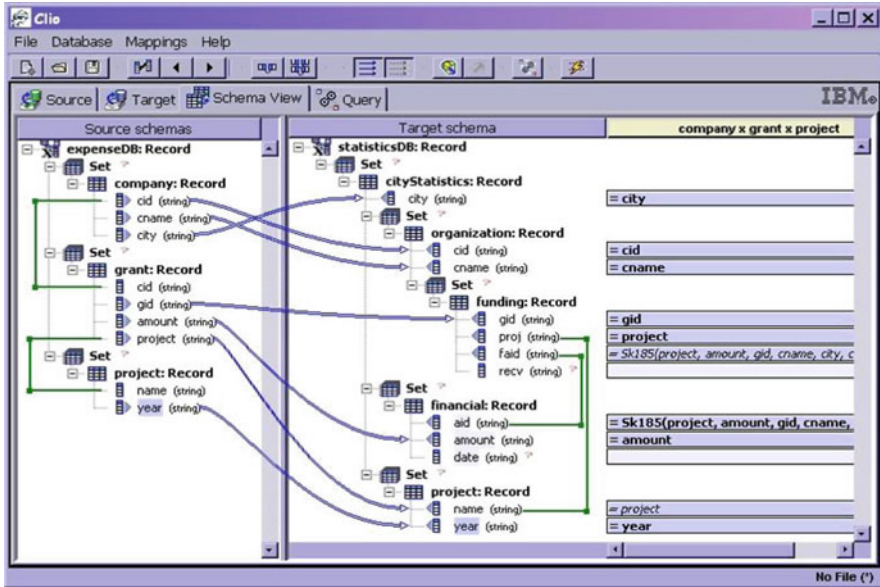


Fig. 2.10 Screenshot of the Schema Viewer from [http://www.almaden.ibm.com/cs/projects/criollo/\(2009\)](http://www.almaden.ibm.com/cs/projects/criollo/(2009))

correspondences can be interpreted as XSLT or XQuery code. This tool also only supports manual creation of matching correspondences.

Finally, like the Clio project, Microsoft’s BizTalk mapper<sup>8</sup> has had both a research and commercial focus. BizTalk mapper provides similar functionality as MapForce and the matching tools in the Stylus Studio, however, work from Microsoft’s Research has been incorporated to allow the matching tool to work more effectively for large schemas.

Robertson et al. discuss visual enhancements that were made to BizTalk mapper as well as a user evaluation [Robertson et al. 2005]. The tool uses the same visual metaphor for matching as many of the previously mentioned tools (see Fig. 2.11) and many of the visual enhancements are similar to features of the COGZ tool.

One of the problems with such a visual metaphor is that the interface can quickly become unmanageable as the number of matchings increases. To help alleviate this issue, Robertson et al. made several small enhancements to the interface that led to great impact in terms of usability. First, like COGZ, highlight propagation was incorporated to make it easier to follow the correspondences for a particular schema entity. This feature simply highlights all the relevant correspondences for a selected entity, while all other correspondences are made semitransparent. Moreover, auto-scrolling was incorporated so that when a user selects a source entity, the target

<sup>8</sup> <http://www.microsoft.com/biztalk/en/us/product-documentation.aspx>.

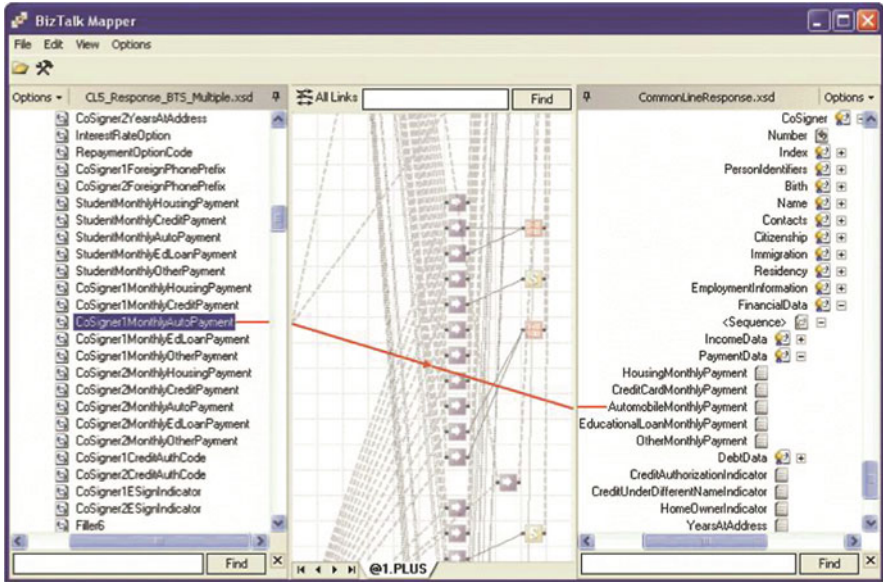


Fig. 2.11 Screenshot of the BizTalk mapper [Bernstein et al. 2006]

schema tree is automatically scrolled to display the area of the target most likely to have a correspondence of interest. As with the COGZ tool, features were introduced to help users deal with a large number of entities in the schema. Instead of zooming or distortion effects, tree coalescing is used to automatically collapse and hide entities deemed to be nonrelevant to the current selected and highlighted elements. Finally, search enhancements were incorporated to support incremental search. Unlike COGZ's incremental search that filters to display results, BizTalk mapper uses scrollbar highlighting. The scrollbar highlighting is used to mark areas of the tree that have search hits.

Besides visualization research, the BizTalk mapper developers have incorporated research for workflow enhancements [Bernstein et al. 2006]. In this research, Bernstein et al. argued that presenting all schema matching correspondences to a user at once is too overwhelming and in fact annoys the user as they become frustrated sifting through all the false positives. Alternatively, the authors suggest that an incremental approach is necessary, where a user can select an entity of interest and then be presented with just the candidate correspondences for that entity. The correspondences are ranked based on their match likelihood, and the user can easily navigate between the candidates. Once a decision is reached and committed by the user, this information can be incorporated into further incremental suggestions.

Each of these tools uses similar visual interaction techniques as the ontology matching tools that we discussed in Sect. 3. However, there is more focus on data translation rule construction than with the ontology-related tools. In the next section, we discuss a different interaction approach, one based on creating matchings by harnessing the power of a community of users.



## 5 Web 2.0 Approaches

Besides interactive desktop tools, researchers have started to explore how to use communities of users to develop ontology matchings collaboratively and to share them. *Crowdsourcing* – outsourcing of a task to a community of motivated individuals – has had huge success in projects such as Wikipedia and social bookmarking sites such as Digg. Similar wisdom of the crowd approaches are beginning to gain traction in the matching community.

Zhdanova and Shvaiko developed an online application to support and collect community-driven matchings [Zhdanova 2005]. The web application allowed users to upload ontologies and to use online tools to perform an automatic matching between the ontologies. Once the users generated the matching, they could save and share it with other members of the community. Realizing that matchings can often be subjective, the authors designed their application to collect information about the users of the community in terms of their expertise, experience levels with particular ontologies, and their goals for a particular matching. Other members of the community could therefore make informed decisions about whether or not to rely on an uploaded matching. The application also stored information about the relationship between users of the community.

Similarly, the OntoMediate Project, as part of their research initiative, has been exploring to what extent collaborative online environments can help to facilitate the specification of ontology matchings [Correndo et al. 2008b]. The prototype system supports the matching of local ontologies to already uploaded ontologies and matchings. Furthermore, the automated procedures make use of the existing matchings to improve the quality of suggested matchings. The tools exploit social interaction to help improve matching quality. Users of the community that work with similar data can socially interact with each other to help validate matchings, spot errors, provide feedback, and propose alternatives [Correndo et al. 2008a].

McCann et al. have also been exploring Web 2.0 approaches. They have proposed an interesting approach to engage the user community [Robert McCann et al. 2008]. In their research, they have been investigating how to gather feedback from users in the form of simple questions in which the answers are used to improve the accuracy of the underlying algorithms. The goal is to pose questions to users that will have a significant impact on the tool's accuracy, as well as be questions that are easy for a human to answer but difficult for a machine. For example, an automated procedure may guess that a particular attribute is of type date, but may not be completely confident about the choice. User-expertise can be exploited in this circumstance to clarify whether the particular attribute is a date or not, leading to significant improvement in the algorithm choices.

In BioPortal,<sup>9</sup> an online tool for accessing and sharing biomedical ontologies, researchers have been exploring the impact of supporting matchings as a form of ontology metadata. Users can upload matchings that are generated offline as well as

---

<sup>9</sup> <http://bioportal.bioontology.org/>.



create matchings interactively through the web application. The online community can comment on the matchings, discuss and refine them. There is currently more than 30,000 such matchings available [Noy et al. 2008].

One important aspect of BioPortal's matching support is that both the ontologies and the matchings are available via web services. This is an important distinction from the early work of Zhdanova and Shvaiko. By making the consumption of these resources readily available to anyone that wishes to make use of this information, it greatly lowers the barrier of entry for applications that need matchings. The consuming applications do not need to be concerned with updates to the ontologies or matchings, as those are handled by BioPortal and immediately available via the services. The services also potentially help promote feedback and improvement about the matchings in BioPortal as it is in consuming application's best interest to improve the matchings. However, without the services, if the matchings were simply downloaded, consumers could make local changes without contributing those back to the community.

There is great potential with a community web-based approach for collecting and sharing matchings. However, this area of study is still very new. To the best of our knowledge, researchers have not yet performed any evaluation to determine whether users can be motivated to contribute to such projects and whether such an approach is feasible. In the next section, we survey existing user-based evaluations and experiments that have been carried out in the ontology matching community. These experiments have mostly been focused on the differences between two tools or how users interpret the automatic suggestions computed by the underlying algorithms.

## 6 Experiments and Evaluation

As our survey of tools in this chapter demonstrates, the development of semi-automatic tools for ontology matching has been gaining momentum. However, evaluation of such tool is still very much in its infancy. There has been only a handful of user-based evaluations carried out in this area. All of these experiments have involved the PROMPT system.

The first experiment was led by the authors of the PROMPT tool. The experiment concentrated on evaluating the correspondence suggestions provided by the tool by having several users merge two ontologies. The researchers recorded the number of steps, suggestions followed, suggestions that were not followed, and what the resulting ontologies looked like. Precision and recall were used to evaluate the quality of the suggestions: precision was the fraction of the tool's suggestions that the users followed and recall was the fraction of the operations performed by the users that were suggested by the tool. The experiment only involved four users, which was too small to draw any meaningful conclusions. The authors stated that, "[w]hat we really need is a larger-scale experiment that compares tools with similar sets of pragmatic criteria [Noy and Musen 2002, p. 12]."

Lambrix and Edberg [Lambrix and Edberg 2003] performed a user evaluation of the matching tools PROMPT and Chimaera [McGuinness et al. 2000] for the specific use case of merging ontologies in bioinformatics. The user experiment involved eight users, four with computer science backgrounds and four with biology backgrounds. The participants were given a number of tasks to perform, a user manual on paper, and the software’s help system for support. They were also instructed to “think aloud” and an evaluator took notes during the experiment. Afterward, 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 [Noy and Musen 2002], while the user interfaces were evaluated using the REAL (Relevance, Efficiency, Attitude, and Learnability) [Löwgren 1994] 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.

The third experiment evaluated PROMPT and the alternative user-interface COGZ. The experiment focused on evaluating the cognitive support provided by the tools in terms of their effectiveness, efficiency, and satisfaction [Falconer 2009]. Researchers assigned eighteen matching and comprehension tasks to participants that they had to perform using each tool (nine per tool). The evaluators then measured the time that it took a participant to complete the task and accuracy with which they performed the task. They measured the participant satisfaction via exit interviews and the System Usability Scale [Brooke 1996].

This last experiment was significantly more comprehensive than the previous studies. Researchers used quantitative analysis to analyze the differences in participant performance across the tasks. They used qualitative approaches to help explain the differences in participant task performance. Furthermore, the design of the experiment was guided by an underlying theory that the authors previously proposed [Falconer and Storey 2007b].

## 7 Discussion

In this section, we return to the ontology tools discussed in our survey. We provide a brief summary of these tools in terms of their visual paradigms, plugins, and algorithm support (see Table 2.1).

Table 2.1 provides a high-level comparison between the surveyed tools. However, more details of comparison and evaluation are needed. In the next section, we discuss this need more deeply as well as other challenges facing the area of interactive techniques for ontology matching.

**Table 2.1** Tool comparison

Tool	Visual and interaction paradigm	Pluggable	Algorithm support
COMA++	<i>Line-based</i> representation of matchings. <i>Tree-based</i> representation of ontologies. <i>Strength</i> of correspondence (number between zero and one). Line color indicates similarity strength	Plugin support for matching algorithms	A variety of automatic matchers
PROMPT	<i>List</i> representation of matchings. <i>Tree-based</i> representation of ontologies. Interaction is <i>synchronized</i> with the source and target ontology trees. <i>Strength</i> of correspondence (description of the “reason for suggestion”)	Extensive plugin architecture	Default algorithm is lexical based. Verification of a correspondence is used to infer new suggestions
COGZ	<i>Line-based</i> representation of matchings. <i>Tree-based</i> representation of ontologies. Interaction is <i>synchronized</i> between search, ontology browsing, and correspondence browsing. <i>Strength</i> of correspondence (description of the “reason for suggestion”)		
AlViz	<i>Tree-based</i> representation of ontologies. <i>Small world graphs</i> representation of matchings. Interaction <i>synchronized</i> with Protégé class browser. Color is used to represent the types of correspondences (e.g., equal, similar, broader than). The cluster display can be filtered by selecting particular entities in the source	No pluggable architecture	FOAM algorithm
OLA	<i>Graph-based</i> visualization of ontologies. The source and target ontologies can be compared side by side. Interaction <i>synchronized</i> between the two ontology displays	No pluggable architecture	A custom algorithm that combines similarity metrics based on lexical similarity, neighbor node similarity, and descriptive features
Muse	Interaction based on <i>wizards</i> that help a user disambiguate matching correspondences	No pluggable architecture	A custom algorithm that incorporates user feedback and automatically generates questions and examples for the user
OntoMap	<i>Drag and drop, line-based</i> representation for matchings. <i>Filters for data transformation</i> can be created interactively based on a particular matching correspondence	No pluggable architecture	Does not support automatic generation of matchings

## 8 Challenges and Next Steps

As our survey in this chapter demonstrates, workers are developing more and more interactive approaches for supporting semiautomatic ontology matching. Many desktop tools for both ontology and schema matching make use of a similar visual representation of matchings – the line-based metaphor for representing a correspondence. This approach is attractive because it is easy to understand what the visualization is attempting to convey. However, previous studies have indicated large variation in the usability of such an approach [Falconer and Storey 2007a; Falconer 2009]. It appears that visual support for matching is not as simple as copying this particular interface style. It is a combination of features and support techniques that assist with a user's workflow that is ultimately needed to help matching users make efficient and effective matching decisions.

Most of the tools in this research area have not been based on theoretical findings from behavioral user studies. They have instead often evolved from a need for some level of interaction with the underlying algorithm. However, without tool evaluations or underlying theories, it is impossible to pinpoint the exact features that lead to a more usable tool. Researchers must address this lack of evaluation and theoretical foundations.

In 2005, a group of researchers started the Ontology Alignment Evaluation Initiative (OAEI)<sup>10</sup> to help provide a standard platform for developers to compare and evaluate their ontology matching approaches. OAEI provides benchmark matching datasets that enable developers of different matching systems to compare their results. At the moment, OAEI evaluates only automatic approaches. We must extend this evaluation framework to compare and contrast interactive tools as well.

Such evaluation will require the development of a standardized comparison framework and evaluation protocols. Comparing interactive tools is more challenging than comparing automatic tools for several reasons: First, the evaluation of interactive tools is more expensive because it requires participation of domain experts in creating the matchings. Second, participation of humans in the evaluation introduces the inevitable bias and differences in the level of expertise and interests of those users who perform the matchings. Familiarity with some tools might bias users toward particular approaches and paradigms. Third, as our survey shows, the tools vary significantly in the types of input that they take and the types of analysis that they perform during the interactive stages. To compare the tools, we must not only characterize these differences but also develop protocols that would allow us to evaluate unique aspects of the tools, while keeping the comparison meaningful. There will need to be common interfaces that would enable evaluators to provide similar initial conditions for the tools, such as the initial set of matchings and to compare the results, such as the matchings produced by the users.

This evaluation would also face some of the same challenges that OAEI faces. For example, there are many strong tools from both industry and research, yet many are not publicly available, making even informal comparisons challenging.

---

<sup>10</sup> <http://oei.ontologymatching.org>.

One of the contributions of OAEI was the development of a framework that identified various features of the tools, and enabled researchers to understand which tools works best under which circumstances. We hope that a similar framework can be developed for interactive tools, where there is an even greater variability in capabilities and workflows supported by the tools. Some interaction and visual paradigms only work well for small-scale ontologies, however, depending on a particular use case, these approaches may be appropriate. It would be useful to evaluate this criteria and make such information publicly available.

The criteria for evaluation of matching tools needs to be specified. This should include usability features, technical details about what ontologies are supported, as well as criteria for evaluating the scalability of the approach.

Besides desktop tools, researchers are exploring web applications that make use of crowdsourcing techniques. This paradigm introduces new directions in interaction, such as social interactions between users, interactions to upload and share ontologies, and services for consuming the matchings. This is a growing research direction and it will take time to determine how to motivate users to contribute to such projects. Also, evaluation will be important to help determine the quality of matchings that are contributed in this way, compared to more closed settings.

Such an approach is very attractive given the success of many existing crowdsourcing applications. This technique is one possible approach for helping deal with the scalability issue of generating a matching. It is a difficult and time-consuming process for a single individual to create the entire matching between two large ontologies. Crowdsourcing potentially alleviates some of this burden by allowing any Web user to contribute.

Researchers who work on the tools for interactive ontology matching, must focus more attention on the issues of scalability of the tools. As the sizes of the ontologies grows (e.g., some biomedical ontologies have tens of thousands of classes), so do the computational demands on the tools: they must be able to work with ontologies that may not load into memory or may take huge computational resources to process. Scalability of visualization techniques is another issue that must be addressed by the tools. As the ontologies become larger, some of the visualization paradigms that worked very well for small ontologies, with all the classes fitting on a single computer screen, simply may not work for ontologies where only a small fraction of the classes will fit on the screen. Both incremental matching [Bernstein et al. 2006] and ontology modularization [Stuckenschmidt et al. 2009] are approaches that potentially address this problem. They have the potential to help reduce cognitive overload during the matching process by restricting the focus of the user to particular areas of the ontology.

Finally, we still must explore new questions in interactive ontology matching, such as how to match the expertise of the user with particular areas of the ontology, where the best location to begin a matching process is, and how to best locate candidate-heavy areas of two ontologies.

## 9 Conclusion

There are many exciting questions to address in the growing research field of interactive techniques for matching. Industry and research has been attempting to address problems of data heterogeneity for many years, yet this problem is ever more prevalent. When precision is necessary, we must rely on human reasoning and domain expertise to help contribute to the matching process. Yet, it is important that we assist users with the process by designing tools that give them access to the information they require to make good decisions, by not hindering the process with overwhelming information, and by automating parts of the procedure when possible. From a research perspective, it is important that we address the lack of tool evaluation by carrying out more user-based evaluations. Heuristic evaluation procedures could also be useful for comparing feature sets of matching tools. There also needs to be more effort to make such findings and tools publicly available to help with evaluation.

We need evaluation to help distinguish what features and approaches are useful for particular use cases. We need theories to help explain these differences. Tools encode a workflow process and this process must align with the user's own internal process. By aligning these processes, we will be able to assist rather than hinder the user. We must incorporate a "human in the loop," where the human is an essential component in the matching process. Helping to establish and harness this symbiotic relationship between human processes and the tool's automated process will allow people to work more efficiently and effectively, and afford them the time to concentrate on difficult tasks that are not easily automated.

## References

- Alexe et al. (2008) Muse: Mapping understanding and design by example. In: international conference on data engineering, Cancun, 7–12 April 2008, pp 10–19
- Bernstein PA, Melnik S (2007) Model management 2.0: Manipulating richer mappings. In: ACM special interest group on management of data (SIGMOD), Beijing, China, September 2007. ACM, NY, pp 1–12
- Bernstein et al. (2006) Incremental schema matching. In: VLDB '06: Proceedings of the 32nd international conference on very large databases, Seoul, Korea, September 2006. VLDB Endowment, pp 1167–1170
- Brooke J (1996) Usability evaluation in industry. In: Jordan PW, Thomas B, Weerdmeester BA, McClelland IL (eds) SUS: A quick and dirty usability scale. Taylor & Francis, London, pp 184–194
- Coradeschi S, Saffiotti A (2006) Symbiotic robotic systems: Humans, robots, and smart environments. IEEE Intell Syst 21(3):82–84. doi:<http://doi.ieeecomputersociety.org/10.1109/MIS.2006.59>
- Correndo et al. (2008a) Collaborative support for community data sharing. In: 2nd workshop on collective intelligence in semantic web and social networks, Sydney, Australia, 12 December 2008
- Correndo et al. (2008b) A community based approach for managing ontology alignments. In: Ontology matching workshop, Karlsruhe, Germany, 26–30 October 2008

- Crub  zy M, Musen MA (2003) Ontologies in support of problem solving. In: Staab S, Studer R (eds) Handbook on ontologies. Springer, Heidelberg, pp 321–342
- Do HH (2006) Schema matching and mapping-based data integration. PhD thesis, Department of Computer Science, Universit  t Leipzig
- Euz  nat J (2006) An API for ontology alignment (version 2.1). <http://gforge.inria.fr/docman/view.php/117/251/align.pdf>
- Euz  nat et al. (2004a) Ontology alignment with OLA. In: Proceedings of the 3rd evaluation of ontologies for the web (EON) workshop. CEUR-WS, pp 59–68
- Euz  nat et al. (2004b) State of the art on ontology alignment. deliverable d2.2.3. Tech. Rep. IST Knowledge Web NoE
- Euz  nat et al. (2009) Results of the ontology alignment evaluation initiative 2009. In: Proceedings of the 4th international workshop on Ontology Matching (OM-2009)
- Falconer SM (2009) Cognitive support for semi-automatic ontology mapping. PhD thesis, University of Victoria
- Falconer SM, Noy NF, Storey MA (2006) Towards understanding the needs of cognitive support for ontology mapping. In: International workshop on ontology matching at ISWC-2006, Athens, GA
- Falconer SM, Storey MA (2007a) Cognitive support for human-guided mapping systems. Tech. Rep. DCS-318-IR, University of Victoria, Victoria, BC, Canada
- Falconer SM, Storey MA (2007b) A cognitive support framework for ontology mapping. In: Proceedings of international semantic web conference, Busan, Korea, November 2007. Springer, Heidelberg, pp 114–127
- Lambrix P, Edberg A (2003) Evaluation of ontology merging tools in bioinformatics. In: Proceedings pacific symposium on biocomputing, pp 589–600
- Lanzenberger M, Sampson J (2006) Alviz – a tool for visual ontology alignment. In: Proceedings of the conference on information visualization (IV), London, July 2006. IEEE Computer Society, Washington, DC, pp 430–440. doi:<http://dx.doi.org/10.1109/IV.2006.18>
- Le Duc et al. (2008) Matching ontologies for context: The neon alignment plug-in. Tech. Rep. Deliverable 3.3.2, IST NeOn IP, NeOn. <ftp://ftp.inrialpes.fr/pub/exmo/reports/neon-332.pdf>
- L  wgren J (1994) Human-computer interaction. What every system developer should know. Chartwell-Bratt, England
- McGuinness et al. (2000) The chimaera ontology environment. In: Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence, Austin, TX, 30 July–3 August 2000. AAAI, CA, pp 1123–1124
- Miller et al. (2001) The clio project: Managing heterogeneity. SIGMOD Record 30(1):78–83
- NE08 (2008) Neon wiki. <http://www.neon-toolkit.org/wiki/index.php>
- NLP (2002) An introduction to NLP. <http://www.cs.bham.ac.uk/~pxc/nlpa/2002/AI-HO-IntroNLP.html>, <http://www.cs.bham.ac.uk/~pxc/nlpa/2002/AI-HO-IntroNLP.html>
- Noy NF, Musen MA (2002) Evaluating ontology-mapping tools: Requirements and experience. In: Proceedings of OntoWeb-SIG3 workshop, Siguenza, Spain, October 2002, pp 1–14
- Noy NF, Musen MA (2003) The PROMPT suite: Interactive tools for ontology merging and mapping. Int J Hum Comput Stud 59(6):983–1024
- Noy et al. (2008) Collecting community-based mappings in an ontology repository. Tech. Rep. BMIR-2008-1310, Stanford Center For Biomedical Informatics Research
- PPP (2006) Performance, Parsing and Pragmatics. <http://www.phon.ucl.ac.uk/home/marco/Principles of Linguistics Handout2006-2007.htm>
- Robert McCann et al. (2008) Matching schemas in online communities: A web 2.0 approach. In: Proceedings of the 2008 IEEE 24th international conference on data engineering. IEEE Computer Society, Washington, DC, pp 110–119
- Robertson et al. (2005) Visualization of mappings between schemas. In: CHI '05: Proceedings of the SIGCHI conference on human factors in computing systems, Portland, OR, April 2005. ACM, NY, pp 431–439. doi:<http://doi.acm.org/10.1145/1054972.1055032>
- Shvaiko P, Euz  nat J (2008) Ten challenges for ontology matching. In: ODBASE, Monterrey, Mexico, November 2008. Springer, Heidelberg, pp 1164–1182



- Stuckenschmidt et al. (2009) *Modular ontologies – Concepts, theories, and techniques for knowledge modularization*. Springer, Heidelberg
- Yamauchi T (2007) The semantic web and human inference: A lesson from cognitive science. In: *Proceedings of ISWC/ASWC*, Busan, Korea, 13 November 2007. Springer, Heidelberg, pp 609–622
- Zhdanova AV (2005) Towards a community-driven ontology matching. In: *K-CAP '05*, Banff, AB, Canada, October 2005. ACM, NY, pp 221–222. doi:<http://doi.acm.org/10.1145/1088622.1088678>
- <http://www.almaden.ibm.com/cs/projects/criollo/> (2009). Website on Clio and schema management

Schema Matching and Mapping

Bellahsene, Z.; Bonifati, A.; Rahm, E. (Eds.)

2011, XII, 320 p., Hardcover

ISBN: 978-3-642-16517-7