BIOINFORMATICS

Vol. 19 no. 12 2003, pages 1564–1571
DOI: 10.1093/bioinformatics/btg194

Evaluation of ontology development tools for bioinformatics

Patrick Lambrix*, Manal Habbouche and Marta Pérez

Department of Computer and Information Science, Linköpings universitet, 581 83 Linköping, Sweden

Received on October 7, 2002; revised on November 22, 2002; accepted on February 26, 2003

ABSTRACT

Ontologies are being used nowadays in many areas, including bioinformatics. To assist users in developing and maintaining ontologies a number of tools have been developed. In this paper we compare four such tools, Protégé-2000, Chimaera, DAG-Edit and OilEd. As test ontologies we have used ontologies from the Gene Ontology Consortium. No system is preferred in all situations, but each system has its own strengths and weaknesses.

Contact: patla@ida.liu.se

1 INTRODUCTION

Ontologies define the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary. They are being used nowadays in many areas, including bioinformatics, for several reasons. Ontologies are used for communication between people and organizations by providing a common terminology over a domain. They provide the basis for interoperability between systems. They can be used for making the content in information sources explicit and serve as an index to a repository of information. Further, they can also be used as a basis for integration of information sources and as a query model for information sources. To assist users in developing and maintaining ontologies a number of tools have been developed and a few comparative studies of ontology tools have been performed. In this paper we use Gene Ontology (GO) ontologies to compare the basic functionality of the ontology tools Protégé-2000, Chimaera, DAG-Edit and OilEd.

In Section 3 we briefly describe the test ontology. Section 4 describes the ontology development tools used in the evaluation. In Section 5 we discuss the evaluation criteria, the evaluation method and the set-up of the evaluation. Because of space limitations, Section 6 discusses only a high-level view of the results of the evaluation. More details are found in (Habbouche and Pérez, 2002). We continue, however, with an overview of related work.

*To whom correspondence should be addressed.

2 RELATED WORK

In Duineveld *et al.* (1999) Ontolingua, WebOnto, ProtégéWin, OntoSaurus, ODE and KADS22 were compared. The authors used ontologies concerning academia and university studies for testing. In Das *et al.* (2001) requirements were defined for industrial strength ontology management. Scalability, reliability, security, internationalization and versioning were considered to be the most important requirements. Chimaera, Protégé, WebOnto and OntoSaurus were evaluated with respect to these criteria. No single tool met the requirements. Some of the tools did meet or surpass other requirements such as ease of use, knowledge representation and merging, but these were not considered to be the most important for industrial strength ontology tools. Two studies have focused on ontology mapping or merging. In Lambrix and Edberg (2003) Protégé-2000 with PROMPT and Chimaera were evaluated as ontology merging tools with GO ontologies and Signal-Ontology as test ontologies. In Noy and Musen (2002) Protégé-2000 with PROMPT was evaluated against a number of proposed criteria. Recently, a survey of ontology tools became available from the OntoWeb Consortium (OntoWeb, 2002). Regarding ontology development tools the following evaluation criteria are proposed: interoperability with other tools, knowledge representation, inference services and usability. A number of tools are compared according to these criteria and test experiments are planned for OilEd, OntoEdit, Protégé-2000, WebODE and WebOnto. In Yeh *et al.* (2003) Protégé-2000 is assessed as a tool for maintaining and developing the GO ontologies. It is shown, for instance, how to enforce strict inheritance and how to track changes in different versions of the ontologies.

3 GENE ONTOLOGY

The Gene Ontology (http://www.geneontology.org/) Consortium is a joint project. The project's goal is to produce a structured, precisely defined, common and dynamic controlled vocabulary that describes the roles of genes and proteins in all organisms (Gene Ontology Consortium, 2000). Currently, there are three independent ontologies publicly available over the Internet: biological process, molecular function and cellular component. The biological process ontology deals with biological objectives to which the gene or gene product contribute. A process is accomplished via one or more ordered assemblies of molecular functions. The molecular function ontology deals with the biochemical activities of a gene product. It only describes what is done without specifying where or when the event takes place. The cellular component ontology describes the places where a gene product can be active. The GO ontologies are becoming a de facto standard and many different bio-databases are today annotated with GO terms. The ontologies grow continuously. The terms in GO are arranged as nodes in a directed acyclic graph, where multiple inheritance is allowed. The two most important relations that are modeled are the is-a relation and the part-of relation.

4 ONTOLOGY DEVELOPMENT TOOLS

After doing a search on currently used ontology tools, we decided to use Protégé-2000, Chimaera, DAG-Edit and OilEd for our evaluation. Protégé-2000 and Chimaera are systems that have been around for a few years and they, or their predecessors, have been used as ontology tools by quite a few projects and in different domains. We chose OilEd as a new ontology tool that also supports the DAML+OIL (http://DAML.SemanticWeb.org/, http://www.ontoknowledge.org/oil/) language that was recommended by the BioOntology Consortium as its choice of ontology representation language in bioinformatics. We note that the other tools in our evaluation announced after our evaluation that they provide support or will in the near future provide support for DAML+OIL. Finally, DAG-Edit was selected as a well-known representative for the GO tools. This tool should give us a feeling for the current state of the art in ontology tools in bioinformatics. It must be noted that the different systems are in different phases of development and that further development is ongoing for most of them.

Protégé-2000

(Noy *et al.*, 2001; http://protege.stanford.edu/index.html) is software for creating, editing and browsing ontologies developed by Stanford Medical Informatics. The design and development of Protégé-2000 has been driven by two goals: to be compatible with other systems for knowledge representation and to be an easy to use and configurable tool for knowledge extraction. Protégé-2000 is available as free software and should be installed locally. It also has a number of plug-ins, among others PROMPT, which is an algorithm for merging and aligning ontologies (Noy and Musen, 2000).

The Protégé knowledge model is frame based. An ontology consists of classes, slots, facets and axioms (Noy *et al.*, 2000). Classes are concepts in the domain. Slots describe properties or attributes of classes. Facets describe properties of slots and axioms define additional constraints. Both individuals and classes can be instances of classes. In the latter case we have a metaclass.

Chimaera

(McGuinness *et al.*, 2000a,b; http://www.ksl.stanford.edu/software/chimaera/) is developed by the Knowledge Systems Laboratory at Stanford University and aims to provide assistance to users for browsing, editing, merging and diagnosing of ontologies. It was built on top of the Ontolingua Distributed Collaborative Ontology Environment. The initial goal was to develop a tool that could give substantial assistance for the task of merging knowledge bases produced by different users for different purposes with different assumptions and different vocabularies. Later the goals of supporting testing and diagnosing ontologies arose as well. The user interacts with Chimaera through a browser such as Netscape or Microsoft Internet Explorer (McGuinness *et al.*, 2000a).

Chimaera's knowledge model is frame based. An ontology consists of classes and slots. As the current user interface is not a general-purpose editing environment, non-slot individuals and facets are not displayed (McGuinness *et al.*, 2000a).

OilEd

(Bechhofer *et al.*, 2001; Stevens *et al.*, 2001; http://oiled.man.ac.uk) is a graphical tool for creating and editing OIL ontologies developed at the University of Manchester. The tool is installed locally. One of the goals of OilEd is to show the use of the DAML+OIL language. OilEd uses the FaCT system (Horrocks, 1999; http://www.cs.man.ac.uk/~horrocks/FaCT/), a description logic system, for checking the consequences of the statements in the ontology.

The knowledge model for OilEd is based on description logics. In contrast to frame systems, OilEd allows for arbitrary boolean combinations of classes. OilEd also allows several types of constraints such as value-type, has-value and cardinality restrictions.

DAG-Edit

(http://www.geneontology.org/doc/dagedit_userguide/dagedit.html) is open source software implemented in Java and is installed locally. The tool offers a graphical user interface to browse, search and edit Gene Ontology files or other ontologies based on the directed acyclic graph (DAG) data structure. The relationships that are supported are is-a and part-of.

5 EVALUATION SET-UP

In this section we present the evaluation set-up including the evaluation method, test ontology and the evaluation criteria.

5.1 Method

The evaluation of the tools has been divided into two parts. In the first part we evaluated the systems based on a number of predefined criteria. The evaluation was done through a literature study and tests using GO. Although we are not common end users of ontologies, we do have experience of knowledge-based information retrieval where ontologies are an important component (e.g. Lambrix, 2000; Lambrix and Shahmehri, 2000). The criteria that were evaluated are presented in Section 5.3.

In the second part of the evaluation the user interface of the systems was evaluated. For this we used an empirical approach. In this technique end users test the systems. This approach is usually time consuming and requires the use of test persons, but has the advantage that the problems that are discovered are real problems. An important aspect of this method is to formulate good instructions for the test persons. As evaluation method we used the REAL method (Löwgren, 1993). This method was chosen for its ease of use, because it requires few test persons and it usually gives good results. In our evaluation eight test persons participated, four with a computer science background and four with a biology background. As we wanted to investigate whether a test person's knowledge of biology would influence the results in the user interface evaluation, we took test persons that work daily with biology and persons that have only high school knowledge of biology. The biologists were used to work with computers. The test persons had no experience of working with ontologies. Before the actual evaluation the test persons were given information on ontologies, what they are, how they can be represented, what they are used for etc. Then the test persons familiarized themselves with the tools. The actual evaluation for each system was divided into two parts. During the first part the test persons were given a number of tasks to perform. These tasks included loading an ontology, search within the ontology, creation, removal and update of classes, attributes and instances, addition and removal of is-a relations and the use of the help system when available. We were concerned about the fact that doing a task in one tool may make it easier to perform a similar task in the next tool. Therefore, to minimize such effects on the results of the evaluation we varied the order in which the different test persons evaluated the tools. For support the test persons had a manual on paper and the system help (when available). They were asked to think loud and an evaluator took notes during the process. Afterwards, the test persons were asked to fill in a questionnaire. The questionnaire consisted of questions with coded values. The test persons were also asked to write comments on their grading. In total this took about 1.5 to 2.5 hours per person.

5.2 Description of test ontology

In the first part of the evaluation we used a small part of the GO molecular function ontology. The test ontology that was used in most of the interface evaluation is a part of GO that contains approximately 60 terms and has ‘behavior’ as its most general term. A complete listing of the terms that are included can be found in Habbouche and Pérez (2002). The terms in GO include concepts, attributes and instances. The concepts are arranged in a hierarchy. Concepts may have attributes. The attributes are not hierarchically organized. Each concept may also have instances.

5.3 Evaluation criteria

The criteria we decided to use in the first part of the evaluation are the following.

Availability: How is the tool used: local installation or via the web?

Functionality: What functionality does the tool provide?

Multiple inheritance: Is multiple inheritance supported? How is it visualized in the tools?

Data model: What is the underlying data model for the ontologies in the tools?

Reasoning: Do the tools verify newly added data and check for consistency when the ontology changes?

Example ontologies: Are example ontologies available? Are they helpful in understanding the tools?

Reuse: Can previously created ontologies be reused?

Formats: Which data formats are compatible with the tool? What formats can be imported and exported?

Visualization: Do the users get a good overview over the ontology and its components (concepts, attributes etc.)?

Help: Is help available? Is it easy to use?

Shortcuts: Are shortcuts for expert users provided?

Stability: Did the tool crash during the evaluation period?

Customization: Can the user customize the tool and in what way?

Extendibility: Is it possible to extend the tools?

Multiple users: Can several users work with the same tool at the same time?

The user interface was evaluated using the REAL approach. The aspects studied in this evaluation are Relevance, Efficiency, Attitude and Learnability. Relevance measures how well a user's needs are satisfied by the tool. Efficiency measures how fast users can perform their tasks using the tool. The subjective feelings towards the tools are measured by attitude. Finally, learnability measures how easy or difficult it is to learn the tool for initial use as well as how much a user can remember the workings of the tool.

Our criteria cover a large part of the criteria that were proposed by the other evaluations. Probably the most important criterion that we did not cover is the issue of scalability. As our tests were performed on not so large parts of the GO ontologies, we have no results on how the tools perform on large ontologies. We note, however, that scalability was already considered a weak point for ontology tools in Das *et al.* (2001) where two of the tools considered here were among the evaluated tools.

6 RESULTS AND DISCUSSION

6.1 General

We discuss the tools according to the criteria presented in Section 5.3.

Availability Protégé-2000 is available as free Java software and is installed locally. DAG-Edit is an OpenSource tool implemented in Java that is installed locally. Also OilEd is Java-based and installed locally. The underlying reasoning system, FaCT, can be installed locally or can be run remotely. Chimaera is available over the Internet and requires a relatively fast connection to be able to work efficiently. With high use of the network operations can take a long time to perform.

Functionality Protégé-2000 provides ontology editing functionality on different levels. Classes, attributes and instances can be created, added, deleted, viewed and searched for. Super-classes can be added, deleted and replaced. Further, it is possible to query the ontology. Plug-ins are provided for querying based on F-Logic, merging and annotation of the ontologies with WordNet. DAG-Edit has its main focus on browsing and editing GO. Functionality is provided for the creation and deletion of concepts, adding synonyms, adding database references, and merging concepts. OilEd provides creation and editing functionality for classes, attributes, instances and axioms. Further, reasoning functionality is provided via the FaCT system. Chimaera provides about seventy commands in the user interface, thereby providing a taxonomy and slot editor. The applicable commands at each point in time are made available by the interface. In addition to editing some of the commands are related to ontology merging. There are also commands related to diagnosis that, among others, check for incompleteness, cycles and value-type mismatches.

Multiple inheritance and visualization of the ontologies Multiple inheritance is supported in all tools. The ontologies are shown using a tree structure. Protégé-2000 uses special symbols and colors for properties of concepts and attributes including the fact that a concept has multiple parents. Also Chimaera uses colors and symbols to indicate properties of concepts. In DAG-Edit a concept can be marked and then all its parents are shown. OilEd has a list over all concepts in the ontology in alphabetical order. The tree structure in all systems can be expanded. One can choose whether the sub-concepts of a concept should be shown or not. For smaller ontologies the tree structure gives a good overview, but for larger ontologies obtaining a good overview can be difficult.

Data model The Protégé-2000 and Chimaera conceptual models are frame based and are expressive enough to represent most of the current bio-ontologies. The information exchange mechanisms in Protégé-2000 and Chimaera are based on the Open Knowledge Base Connectivity (OKBC). This is an application programming interface for frame-based knowledge representation systems (Chaudhri *et al.*, 1998). Protégé-2000 has restricted OKBC in a few ways (Noy *et al.*, 2000). The representation in OilEd is based on DAML+OIL. The conceptual model behind DAG-Edit is the directed acyclic graph. Flat files are used for storage, although adapters for Postgres and MySQL databases exist.

Reasoning OilEd uses FaCT, a description logic system, for reasoning about the ontology. This includes services such as classification and consistency checking. FaCT provides reasoning for a part of the language. For instance, reasoning with primitive data types is not supported. Protégé-2000 checks the input of new data. The tool does not allow two classes or attributes with the same name. In DAG-Edit newly created concepts receive a unique identifier. This allows for different concepts having the same name. It is also possible to allow ontologies with cycles. As mentioned before, Chimaera provides support for diagnosis.

Example ontologies Protégé-2000, OilEd and Chimaera include example ontologies. To some extent, GO could be seen as the example ontology for DAG-Edit. They are very useful in explaining the workings of the tools to new users.

Reuse All tools allow the reuse of previously defined ontologies. In OilEd it is also possible to work on different projects where each project has its own window.

Formats Both Protégé-2000 and Chimaera support the import of ontologies in different formats. Protégé-2000 supports the import of text files, database tables and RDF files. The user can save her work in three different formats: as standard text files, in a JDBC database and in RDF format. Plug-ins can be used to support other formats such as XML. Chimaera accepts as input 15 different formats among which Protégé files and OKBC. The user can create knowledge bases of types ATP, CLOS, Ontolingua and Tuple KB. OilEd supports export to Simple RDFS, DAML+OIL, SHIQ, HTML, Dotty and DIG. The save formats in DAG-Edit are GO Flat File, GO MySQL database, GO RDF Flat File and GO Postgres database. Import and export of flat files is supported.

Visualization DAG-Edit and Chimaera show the components of the ontology in one window. As editing ontologies was not the main focus of Chimaera, instances are not shown. The Protégé-2000 user interface and the OilEd user interface, which has been strongly influenced by Protégé-2000, contain several tabbed panes where in each pane relevant information about a current ontology component is shown. This approach gives the user a good overview and a feeling of control.

Help DAG-Edit and OilEd do not provide a help function in the user interface. There is help on the web. Chimaera provides a large and good help system. The Protégé-2000 help in the tool is divided in help on icons, tutorial, user's guide, shortcuts and FAQ.

Shortcuts Only Chimaera has more extensive support for shortcuts.

Stability Protégé-2000 and OilEd were stable during the evaluation. For Chimaera the server has been down on some occasions. DAG-Edit crashed a few times.

Customization Chimaera has many different alternatives with respect to customization. The focus is on customization of the behavior of the tool. The Protégé-2000 and DAG-Edit customization focuses on layout. The user can customize Protégé-2000 in the sense that she can choose between different layouts and she can choose which tabbed panes are shown in the user interface and in which order. DAG-Edit supports configuration via the Configuration Manager Plugin. The user can choose formats and what plug-ins should be shown. OilEd allows a limited amount of customizations pertaining partly to the look of the tool and partly to the reasoning capabilities.

Extendibility DAG-Edit supports user-defined plug-ins. They can be used for reading, saving, importing and exporting. Also in Protégé-2000 plug-ins are supported.

Multiple users Only Chimaera provides real support for multiple users. Users can join session groups. The group members receive change notifications when something in the ontology has changed. In Protégé-2000 multiple users can read and make changes in ontologies as long as this does not result in conflicts. There is no support for multiple changes on the same component. Users are not notified about changes made by others.

6.2 User interface

The results from the questionnaire are found in Table 1. In this section we summarize these results as well as the observations reported by the test persons during the evaluation. For each of the REAL aspects we give general comments as well as comments for each of the systems. We also note that there was no significant difference between the results of the test persons with biology background and the test persons with computer science background.

Relevance All systems were considered to provide adequate functionality for solving the given tasks. All systems had implemented support for the basic functionality that the tasks required. Chimaera and OilEd seemed to have more functionality than the other systems. It was commented that some functionality was not available in the context menu of OilEd, but it was possible to solve the tasks via the buttons.

Efficiency With respect to navigation between the different windows in the tools and with respect to the overview over the terms in the ontologies that the tools provide, Protégé-2000 was considered to give the best support, while Chimaera was considered to give the least support for the former and OilEd the least support for the latter. It was reported that Protégé-2000 does not always give feedback on user actions such as name changes, changes in attribute values, creation of new classes and attributes. Also, the test persons did not always feel in control. Basic tasks, such as the creation and deletion of components in different windows, are done in the same way. It was very easy to navigate and switch between the windows in the tool. The reasons that were given were that the number of windows was relatively small and only the information relevant to the current task was shown. As mentioned before the overview over the terms in the ontology was considered good. For DAG-Edit it was mentioned that there was a clear visual updating of the windows in connection to performed actions. It was also easy for the test persons to perform the given tasks in the tool. Sometimes the users experienced that there were 'hidden actions'. In this case certain functionality was implemented (e.g. moving of classes), but not made available via the user interface. Further, it was considered positive that the user received feedback in relation to disallowed actions. However, in some cases the test persons expected to receive feedback, but received none. An example of this was when non-saved files were closed. In this case the test persons expected the system to ask whether saving was necessary. Further, the overview over the components of the ontology was considered good and visible at all times. The test persons felt that they were in control of the system. In OilEd the effects of actions were

Table 1. Evaluation of graphical user interface

	P avg	P min	P max	D avg	D min	D max	O avg	O min	O max	C avg	C min	C max
Relevance												
Does the tool solve the tasks? (never = 1, sometimes = 5, always = 10)	9	7	10	9	7	10	9	8	10	8.9	6	10
The functions in the tool to create and manage ontologies are (too few = 1, right = 5, too many = 10)	5.8	5	8	5	2	8	6.5	3	10	6.4	5	10
Efficiency												
Do you get feedback on your operations? (never = 1, sometimes = 5, always = 10)	5	2	10	5.6	1	10	6.5	4	9	5.8	2	10
To perform the different operations feels (difficult = 1, easy = 10)	6	4	8	6.3	3	9	5.4	1	8	5	1	10
To navigate between the different windows in the tool feels (difficult = 1, easy = 10)	9.4	8	10	8.4	6	10	7.8	4	10	6.4	3	10
The overview over the concepts, attributes and instances feels (bad = 1, good = 10)	8.5	6	10	7	3	10	5.6	2	8	6.3	2	10
Attitude												
The user interface designed in a uniform way. (do not agree = 1, agree = 10)	8.5	7	10	7.5	4	10	7.3	3	10	7.4	5	10
Using the interface is (boring = 1, fun = 10)	5.6	5	8	5	1	9	6	3	8	3	1	8
The available help feels (not enough = 1, adequate = 10)	7.4	3	10	-	-	-	-	-	-	7	2	10
Learnability												
Is it easy to understand the meaning of the icons and menus? (difficult = 1, easy = 10)	6	1	8	6	1	8	5	2	8	4	1	8
Is the terminology easy to understand and adapted to the user? (never = 1, sometimes = 5, always = 10)	7.8	5	10	6.8	4	9	6.6	5	8	6.6	3	10
To use the help feels (difficult = 1, easy = 10)	8	5	10	-	-	-	-	-	-	6.6	2	10
To understand the purpose of the user interface feels (unclear = 1, clear = 10)	7	2	10	6.4	4	9	7	3	9	5.5	2	10
To learn the interface for ontology management feels (difficult = 1, easy = 10)	8	5	10	6.4	1	9	6	3	9	5.3	1	10
To remember how to use the interface feels (difficult = 1, easy = 10)	9	7	10	8.4	6	10	7.6	4	10	5	1	10

usually clearly visible. In some cases indirect feedback was presented. In some cases no feedback was given. For instance, no feedback was given upon removal of ontology components. Some people found it easy to work with the tool, while others thought it was difficult. The latter group thought it was difficult to find the right buttons and understand their functionality. The navigation between windows was considered to be easy. The overview over the components of the ontology was considered good. It was also considered positive that information about different components is managed in different windows. Chimaera was considered to be difficult to use. It was easy to make mistakes. Good feedback was given for some functionality (e.g. while creating sub-classes). The updating of the data was considered slow, which may be explained by the fact that Chimaera was the only tool that had to access the internet. The overview over the components of the ontology was considered good. It was also considered positive that the user had the possibility to choose which information that should be shown in the interface.

Attitude Protégé-2000 received the best results with respect to uniformity of the user interface. The basic parts of the Protégé-2000 user interface were designed in a uniform way, but certain apparent inconsistencies still appeared. For instance, when a class was marked in a window, then the right mouse button could be used, but this was not the case for instances. The help that was provided was considered adequate for the tasks at hand. Several test persons felt a positive attitude towards the DAG-Edit tool. The interface was rather uniformly designed. The negative comments were about the lack of a help function. OilEd's interface was considered by most test persons to be uniformly designed with a consistent use and feel of the icons. Some test persons commented that it was fun to use the tool. The rest thought it was too difficult to learn and therefore boring during the learning phase. A negative aspect was the fact that input fields were not clearly marked. Chimaera's layout was considered ugly and boring by most test persons. The help was considered good, but also necessary.

Learnability The meaning of icons and menu items in the Protégé-2000 user interface was not always evident. The test persons found that some symbols were not chosen so well. For instance, the symbol for removal is very similar to the windows symbol for closing a window. Sometimes, the same symbol was used for two similar, but still different kinds of functionality. It was difficult to find the search functionality in the interface. The terminology that is used in the interface is standard within ontologies. The tool was considered easy to learn and to remember. The meaning of icons and menu items in the DAG-Edit user interface was not always evident. The used terminology was easy. The tool had a clear and simple design and structure and the test persons found that the information is grouped in a logical way. They commented the low learning threshold for the system. Also the meaning of icons and menu items in the OilEd user interface was not always evident. OilEd's use of terminology was easy to understand, although it was commented that some terminology is OIL specific. The test persons had different feelings on how easy it is to learn the tool. Some thought it was easy to remember the use of the tool while the rest thought it was difficult to take in that much information. It was considered to be difficult to learn the meaning and use of certain icons and menu items in Chimaera. It took time to learn the order in which actions needed to be performed. The used terminology was easy to understand. It was considered positive to receive an explanation of why certain menu choices are inactive. In general though, Chimaera was considered difficult to learn and much help was needed.

6.3 Method critique

Although we have some experience with ontologies, we are not the common end users of ontology tools. This may have an influence on the results of the first part of the evaluation. However, we have tried to minimize this influence by studying other evaluations, learning to work with the tools and clearly defining the criteria before the actual evaluation.

The test persons in the user interface evaluation were novices in using ontologies. Therefore, there may have been an influence on the way the test persons performed the given tasks. In particular, there may have been a larger influence for the results of Chimaera as the choice of operations in Chimaera is significantly larger than in the other tools. Also, Chimaera is the only system that provides shortcuts for expert users.

In the evaluation we have used GO. This may influence the results of the evaluation. For instance, as DAG-Edit was designed for editing DAG ontologies, it can be expected to have a highly relevant interface. Further, the underlying structure of GO may not be so useful to evaluate some capabilities of the tools such as the reasoning capabilities of OilEd and the knowledge acquisition capabilities of Protégé-2000.

7 CONCLUSION

In this paper we have evaluated Protégé-2000, Chimaera, OilEd and DAG-Edit for their use as ontology development tools in bioinformatics. We used Gene Ontology as test case.

No system is preferred in all situations. All systems have their strengths and weaknesses. The main strengths of Protégé-2000 compared to the other systems are its user interface, the extendibility using plug-ins, the functionality that the plug-ins provide (such as merging) as well as the different formats that can be imported and exported. Chimaera's main strengths are its functionality, including merging and diagnosis, the different formats that can be imported and exported, its help functionality, the shortcuts for expert users and the fact that multiple users can work with the same ontology. Its user interface is its main weakness. The main advantage of OilEd is the fact that its model is description logic-based and that the underlying FaCT system can perform reasoning tasks such as classification and consistency checking. DAG-Edit was specifically built for GO ontologies and has an interface that is easy to use and learn.

ACKNOWLEDGEMENTS

We thank Vaida Jakonienė and the anonymous referees for valuable comments on the paper, Anna Edberg for discussions and the test persons in the user interface evaluation for their participation in the project. We also acknowledge the financial support of the Center for Industrial Information Technology.

REFERENCES

Bechhofer,S., Horrocks,I., Goble,C. and Stevens,R. (2001) OilEd: a reason-able ontology editor for the semantic web. In *Proceedings of the Joint German Austrian Conference on Artificial Intelligence*, LNAI 2174, pp. 396–408.

Chaudhri,V., Farquhar,A., Fikes,R., Karp,P. and Rice,J. (1998) OKBC: a programmatic foundation for knowledge base interoperability. *Proceedings of the Fifth National Conference on Artificial Intelligence–AAAI98*. Madison, WI, USA, pp. 600–607.

Das,A., Wu,W. and McGuinness,D. (2001) Industrial strength ontology management. In *Proceedings of the First Semantic Semantic Web Working Symposium*. Stanford, USA.

Duineveld,A.J., Stoter,R., Weiden,M.R., Kenepa,B. and Benjamins,V.R. (1999) Wondertools? A comparative study of ontological engineering tools. *Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management*. Banff, Canada, pp. 4.6.1–4.6.20.

Gene Ontology Consortium (2000) Gene ontology: tool for the unification of biology. *Nature Genet.*, **25**, 25–29.

Habbouche,M. and Pérez,M. (2002) *Evaluation of ontology tools for bioinformatics*, B.Sc. thesis LiTH-IDA-Ex-Ing-02/10, Department of computer and information science, Linköpings universitet. In Swedish.

Horrocks,I. (1999) *FaCT and iFaCT*, Proceedings of the International Workshop on Description Logics, Linköping, Sweden, pp. 133–135.

Lambrix,P. (2000) *Part-whole reasoning in object-centered framework*, LNAI 1771, Springer.

Lambrix,P. and Edberg,A. (2003) Evaluation of ontology merging tools in bioinformatics. In *Proceedings of the Pacific Symposium on Biocomputing–PSB03*. Kauai, Hawaii, pp. 589–600.

Lambrix,P. and Shahmehri,N. (2000) Querying documents using content, structure and properties. *J. Intelligent Infor. Sys.*, **15**, 287–307.

Löwgren,J. (1993) Human-Computer Interaction. What every system developer should know, Studentlitteratur, Lund.

McGuinness,D., Fikes,R., Rice,J. and Wilder,S. (2000a) An environment for merging and testing large ontologies. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning–KR00*. Breckenridge, CO, pp. 483–493.

McGuinness,D., Fikes,R., Rice,J. and Wilder,S. (2000b) The chimaera ontology environment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence–AAAI00*. Austin, TX, pp. 1123–1124.

Noy,N.F., Fergerson,R. and Musen,M. (2000) The knowledge model of Protégé–2000: combining interoperability and flexibility. In *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management*, LNAI 1937, Juan-les-pins, France, pp. 97–112.

Noy,N.F. and Musen,M. (2000) PROMPT: algorithm and tool for automated ontology merging and alignment. In *Proceedings of Seventeenth National Conference on Artificial Intelligence–AAAI00*. Austin, Texas, pp. 450–455.

Noy,N.F. and Musen,M. (2002) Evaluating ontology-mapping tools: requirements and experience. In *Proceedings of the EKAW Workshop on Evaluation of Ontology Tools*. Siguenza, Spain.

Noy,N., Sintek,M., Decker,S., Crubézy,M., Fergerson,R. and Musen,M. (2001) Creating semantic web contents with Protégé–2000. In *IEEE Intelligent Systems*. pp. 60–71.

OntoWeb Consortium (2002) A survey on ontology tools, Deliverable 1.3.

Stevens,R., Horrocks,I., Goble,C. and Bechhofer,S. (2001) Building a reasonable bioinformatics ontology using OIL. In *Proceedings of the IJCAI Workshop on Ontologies and Information Sharing*.

Yeh,I., Karp,P., Noy,N. and Altman,R. (2003) Knowledge acquisition, consistency checking and concurrency control for Gene Ontology (GO). *Bioinformatics*, **19**, 241–248.