Usability Evaluation of Ontology Editors

Elena García-Barriocanal*, Miguel A. Sicilia**, Salvador Sánchez-Alonso***

*/**/***University of Alcalá, Polytechnic School, Computer Science Department,
Ctra. Barcelona km. 33.600, Alcalá de Henares, Madrid, Spain, 28871
*elena.garciab@uah.es **msicilia@uah.es ***salvador.sanchez@uah.es

Dr. Elena García-Barriocanal obtained a university degree in computer science from the Pontifical University of Salamanca in Madrid and a PhD from the computer science department of the University of Alcalá. From September 1998 to February 1999 she worked as a lecturer in the computer languages and information systems department of the Pontifical University, and in 1999 she joined the computer science department of University of Alcalá where she is associate professor. Her research interests mainly focus on topics related to the role of knowledge representation in fields like human-computer interaction and learning technologies.

Miguel A. Sicilia obtained a university degree in computer science from the Pontifical University of Salamanca in Madrid, Spain and a Phd from Carlos III University in Madrid, Spain. From 1997 to 1999 he worked as assistant professor at the Pontifical University, after which he joined the computer science department of the Carlos III University in Madrid as a lecturer, working simultaneously as a software architect in e-commerce consulting firms. From 2002 to October 2003, he worked as a full-time lecturer at Carlos III University. Currently, he works as an associate professor at the computer science department, University of Alcalá (Madrid). His research interests are primarily adaptive hypermedia, learning technology, semantic web and human-computer interaction.

Dr. Salvador Sanchez-Alonso obtained a university degree in computer science from the Pontifical University of Salamanca (Spain) in 1997, and a PhD in computing from the Polytechnic University of Madrid (Spain) in 2005. He worked as an assistant professor at the Pontifical University of Salamanca from 1997 to 2000 and from 2002 to 2005. He also worked as a software engineer at a software solutions company during 2000 and 2001. From 2005, he is a professor of the computer science department of the University of Alcala, in Spain. His research interests include learning objects reusability, metadata, object-oriented technologies and web engineering.

García-Barriocanal, Elena, Miguel A. Sicilia, and Salvador Sanchez-Alonso. **Usability Evaluation of Ontology Editors.** *Knowledge Organization, 32(1).* 1-9. 27 refs.

ABSTRACT: Ontology editors are software tools that allow the creation and maintenance of ontologies through a graphical user interface. As the semantic web effort grows, a larger community of users for this kind of tool is expected. New users include people not specifically skilled in the use of ontology formalisms. In consequence, the usability of ontology editors can be viewed as a key adoption precondition for semantic web technologies. In this paper, the usability evaluation of several representative ontology editors is described. This evaluation is carried out by combining a heuristic pre-assessment with a subsequent user-testing phase. The target population is comprised of people with no specific ontology-creation skills that have a general knowledge about domain modelling. For this kind of user, current editors are adequate for the creation and maintenance of simple ontologies. Also, there is room for improvement, especially in browsing mechanisms, help systems, and visualization metaphors.

1. Introduction

1.1. The Role of Ontology Editors in the Semantic Web

From an Artificial Intelligence perspective, ontologies can be described as a kind of knowledge representation (Davis, Shrobe and Szolovits 1993) for shared conceptualizations of specific domains (Decker et al. 2000), which is considered as a key enabling technology for e-commerce (Fensel 2001) and for the so-called semantic web (Ding et al. 2002). To date, widely used object-oriented modelling languages, like the UML (Unified Modelling Language) (Object Management Group 2003) have been used to represent ontologies (Cranefield and Purvis 1999; Cranefield, Haustein and Purvis 2001). However, current ontology formalisms, like KIF (Knowledge Interchange Format) (NCITS 1998), either exceed the built-in information representation capabilities of the core meta-models of those languages (Cranefield and Purvis 1999), or make necessary the introduction of a set of supplementary notational extensions (Baclawski et al. 2001), both of which result in harder-to-learn modelling languages. In consequence, it is expected that knowledge representation (KR)-specific tools, like Protégé (Noy et al. 2001), will continue to be used for ontology creation and editing in the near future, taking into account that frame-based mark-up languages – such as RDF (Resource Description Framework) and its extensions – are intended for computer interchange rather than for direct human reading and writing.

In this work, the term Ontology editor (OE) is used to refer to KR-specific software tools, explicitly based on any ontology formalism, which allow the interactive creation and updating of ontologies through a graphical user interface. The focus of this research is on the specific human-interaction characteristics of these tools, assuming that efficient and easy to use ontology creation and maintenance applications are a critical element in the semantic web infrastructure. Taking into account that a larger community of users would include a larger number of non-KR specialists, we aim at investigating whether currently available OEs are usable for people without a deep understanding of (or experience in) ontology modelling.

The rest of this paper is structured as follows. In the rest of this section the general principles and methods of the evaluation are described. In Section 2, the test procedure is explained in detail, including the findings obtained from the pre-assessment heuristic evaluation. Later on, section 3 includes the results. Finally, conclusions and future research directions are sketched in Section 4.

1.2. Overall Description of the Evaluation Method

Usability evaluation is considered an important dimension in the evaluation of systems that have some kind of knowledge acquisition interfaces (Adelman and Riedel 1997). Reports on usability evaluation of various knowledge representation systems have addressed different usability measures, e.g. the time needed to learn specific knowledge entry functionalities (Shahar et al. 1999), technical aspects that directly affect the user (e.g., explanation), error handling, system efficiency, and adequacy of programming interfaces (McGuinness and Patel-Schneider 1998).

In previous studies (Duineveld et al. 2000), a comparison between six ontology-engineering tools was made in accordance with three different dimensions: the user interface, the ontology-related issues found in the tool, and the tool's capacity to support the construction of an ontology by several people at different locations. Duineveld et al. (2000) describe their opinion about ontology engineering tools by using a checklist, but potential users did not take part in the evaluation.

In this study, a conventional usability evaluation has been carried out combining two widespread techniques: heuristic evaluation and user testing. Three groups of users/evaluators were formed, each with different backgrounds, to report on the usability of selected OEs. The main objective was not to analyse specific knowledge entry techniques, but, instead, to consider general user interaction issues. Gómez-Pérez (1994) proposed an explicit distinction between evaluation, and assessment of knowledge sharing technology (KST), including ontology editors. "Evaluation means to judge technically the features of KST, and assessment refers to the usability and utility of KST in companies." However, the term "evaluation" has been used for the sake of clarity in the application of the most common methods and techniques to measure system usability, as this is a more familiar term in the Human-Computer Interaction (HCI) community.

Preceded by an heuristic evaluation (Nielsen 1994) aimed at obtaining the present assessment of usability problems, conventional user testing techniques (Dumas and Redish 1999) have been selected as the main approach. These methods are considered complementary, as each one detects usability problems overlooked by the other (Nielsen 1994). Our evaluation is mainly formative, in the sense that it is targeted to expose usability problems in current tools. However, because of the process, some aspects have also been identified that could be used as the point of departure for a summative evaluation – i.e. an evaluation to determine which among several alternatives to ontology editing is best. In addition, because OEs are far too complex to test all of their functionalities at once, the study was purposefully limited to answering questions relevant the community of OE users that do not come from the KR field.

The general concern of our study is to determine the ease of use of OEs or, in other words, to be able to provide an answer to the question: "If users have limited or null experience in ontology creation and maintenance, or if they have an exploratory learning style, are OEs robust in terms of usability?" For this purpose, users will be considered to have an exploratory learning style if they prefer to learn about the use of the system by investigating it on their own initiative – often in pursuit of a real or artificial task – instead of working through precisely sequenced training materials. More specifically, the following two concerns have been raised: "How easy is it to create a new ontology with current OEs?" and "how easy is it to browse, search and perform updating tasks on large ontologies with current OEs?" Related activities allowed in some OEs like Protégé (Noy, Fergerson and Musen 2000), such as semantic web page annotation, collaborative ontology edition, or ontology meta-modelling, are not considered here, because they are not directly supported by the most commonly used OEs.

The following tools were initially selected for inclusion in the test:

- Protégé 2000 1.6.2 (http://protege.stanford.edu/);
- OntoEdit 2.0 (http://www.ontoprise.de);
- OILEd 2.2a (http://oiled.man.ac.uk);
- the KSL Ontology Editor (http://www-ksl-svc.stanford.edu:5915/);
- WebOde 1.1 (http://kw.dia.fi.upm.es/wpbs/);
- WebOnto (http://kmi.open.ac.uk/projects/webonto/); and,
- KADS22 (http://hcs.science.uva.nl/projects/kads22/).

Although several OEs currently exist – an exhaustive list can be found in Denny (2002) – the stability of the versions, as well as platform and licensing constraints, have served as a filter in the selection of OEs for this study. From the selected OEs, those that did not allow both edition and creation processes were discarded. Later, practitioners were asked for their opinions on the most widely used OEs, resulting in the list above. In the overall process of selection, the main criterion was that of comparing two types of interfaces: HTML-based interfaces and 'GUI-desktop' interfaces.

2. Evaluation design

2.1. Specific Concerns and Measures

The specific concerns of the evaluation were motivated by an heuristic analysis (Nielsen 1994) carried out by experts with at least one year of previous experience in ontology editing. The procedure for the evaluation consisted of three phases: a pre-selection phase, in which some tools could be discarded, the actual evaluation, and a debriefing and severity-rating phase. Although experts were free to take their own approach, it was suggested that they edit simple ontologies taken from the Internet, browse sample ontologies downloaded from the DAML library, and search in the $(KA)^2$ ontology (Benjamins et al. 1999). The latter two tasks were only performed for those OEs that included support for loading RDF ontologies. Four evaluators carried out the study, therefore, according to Nielson's curve, more than fifty percent of the usability problems are estimated to have been found (Nielsen 1992).

In the first phase, after the first three experts' pre-evaluations, KADS22 was discarded. This decision was based on the fact that it did not adhere to common platform conventions, and to its clear orientation to CML-file editing. It is important to note that KADS22 is considered to be in development. In addition, WebOde was not evaluated, because it combines HTML forms with graphical interfaces based on applets, and this would have complicated the categorization and comparison of the two established types of tools. Table 1 summarizes the most relevant results of the second phase of the analysis, structured around Nielsen's heuristics. The experts were advised to use Tognazzini's (2002) principles as a checklist. The column marked 'S', shows the *severity* rating estimated by the experts during the third phase. According to Nielsen (Nielsen 1994), a scale

Heuristic	*Problems Found*	*S(0..49)*	*A*	*B*	*C*	*D*	*F*
Visibility of system status	Lack of status bar	2	X	-	-	X	X
Match between system and the real world	Unexplained system-oriented terms	3	-	-	X	-	X
	No printing functionality	3	X	X	X	X	X
User Control and freedom	No "un-do"/"re-do" functionality	3	X	-	X	.	X
	No "replace" functionality	2	X	X	-	X	X
	No "copy & paste" functionality in hierarchies	2	X	-	X	X	X
	No "drag & drop" functionality in hierarchies	2	-	X	X	X	X
	No "cut & paste" functionality in hierarchies	2	X	-	-	X	X
	No tool tips in some elements	2	X	-	-	X	X
	Pop up menu navigation using cursors is not permitted	1	X	X	-	na	na
	No searching slots functionality	2	X	X	-	-	-
Consistency and standards	Does not follow menu platform conventions	1	-	-	-	-	X
Recognition rather than recall	Actions available only through	2	-	-	X	na	na
Flexibility and efficiency of use	Excessive time to launch	3	-	-	-	X	X
	No key accelerator	2	X	X	-	na	X

Table 1. Heuristic evaluation results.

from 0 to 4 is used, where 0 stands for no problems, 1 stands for cosmetic problems, 2 for minor problems, 3 for major problems and 4 for problems that are imperative to fix. When an expert detected a problem on an OE, the problem was recorded by marking an 'x' in the corresponding column. The acronym *n.a.* stands for *not applicable*. The main conclusion of the heuristic evaluation is that major usability problems are scarce, except for inadequate help, and user error reporting systems.

After the heuristic evaluation, WebOnto was discarded from the user test due to several behaviour problems related to the interface (buttons disappearing in the toolbar, operations that did not report errors but did nothing, and the like.) This inconsistent performance might have been due to minor issues, such as a non-compatible Web browser version or any other problem related to the common platform, but the behavior problems made it impossible to carry out a fair comparison with the other tools. Nonetheless, the graphical editing capabilities of WebOnto, that provide an appropriate and efficient way to edit hierarchies, and the unique collaborative editing capabilities, must be highlighted.

Based on the heuristic evaluation results, the general concerns of the test are detailed in specific issues, and the measures used for each of these issues are provided. First of all, the general concern relating to the question "How easy is it to create a new ontology with current OEs?" is illuminated by two questions: 1) how easy is it to create a new empty ontology and to set the initial basic properties?; and, 2) how easy is it for new users to define a new ontology construct of type X (where X stands for, respectively, a class, a property, and an instance)?

Next, the following specific issues were derived from the second general concern about the ease with which one might browse, search, and perform updating tasks on large ontologies with current OEs. These questions are: 1) how easy is it to find a specific ontology construct of type X?; 2) how easy is it to navigate through the generalization/specialization hierarchy?; and, 3) how easy is it to update a characteristic C (e.g. name, property/slot, instance) of an existing ontology construct of type X?

In all cases, the time to complete a task and the number of errors raised in completing that task have been selected as a measure for the issue. The application response time has not been included in the evaluation, as it is easy to verify that some of the current OEs require further improvements in parsing and/or caching of large ontologies. An example is the large *Universal Standard Products and Services Classification* ontology that takes about two minutes to load in Protégé 2000 on a Pentium III computer with 1GB of main memory. Loading this particular ontology makes the Protégé process grow to 150 MB of memory. In addition, Web-based OEs, in some cases, do not reach the 1 second limit necessary to keep uninterrupted the user's flow of thought (Nielsen 2000), although those OEs do not violate the 10 second response time limit that is considered necessary for keeping the user's attention focused on the dialogue.

2.2. Participants

The target population is made up of individuals who share the following characteristics: more than five years of experience in the use of computers, daily use of complex GUI-based applications, and a minimal understanding of conceptual models (but capable of understanding, at least simple UML class diagrams). A pre-test phase allowed for the rejection of users not fitting this profile, as current OEs are not considered adequate for their use. Note that the ontologies used in the test are designed for usability rather than for reusability – in the sense given in Domingue and Motta (1999), and therefore, further testing would be required for ontologies designed for reusability. Moreover, a number of features that can be considered as advanced ontology modelling, like exploiting inference engines or defining axioms through formulas, were disregarded in the analysis. An informal experiment with three users who were not familiar with KR Internet services demonstrated that including those features was simply not realistic. The experiment consisted of editing axioms with OILEd from natural language descriptions. None of the users was able to complete the task in a reasonable time, which suggests that simpler and more intuitive interface metaphors are required for those tasks to be carried out by people with no background in description logics or similar formalisms. From the basic user profile, three subgroups were considered: 1) users with experience in ontology definition; 2) users with experience in computer-based modelling (e.g. users with experience in UML modelling tools) but with no experience in ontology definition; and, 3) users with neither experience in computer-based modelling nor ontology definition, but accustomed to using computer applications. For the test, four participants from each group were selected. In addition, a participant from each of the subgroups was selected to perform a pre-test designed to detect defects in the test process itself.

2.3. Procedure and Scenarios

The final test involved three steps:

- **Learning.** Participants in subgroups 2 and 3 were given a brief introduction both to general ontology concepts and to the specifics of every OE under evaluation. For subgroup 2, the explanations were structured around concepts that are not familiar to UML users, e.g., the fact that properties are a first-class modelling element (Baclawski et al. 2001).
- **Evaluation.** This step was divided into two parts, one for each general concern.
- **Post-test**. After each part of the evaluation, the participants responded to a questionnaire aimed at measuring their subjective satisfaction.

In order to evaluate the specific issues detailed in section 2.1, scenarios in step two were set up as follows:

- **Scenario 1**. The user creates a small ontology from scratch. For this purpose, a part of the ontology described in Fensel et al. (2000) was then written in a language-neutral (from the perspective of ontology languages) textual form, and sketched as a UML diagram. A total of ten classes, and five properties were used. This scenario was the same for all the evaluations.
- **Scenario 2**. After loading a relatively large ontology, the user was requested to search a class and a property, to annotate all the relationships of the class (along with the entire generalization hierarchy), and to perform small updates on either the class, or the property, or both. Depending on the OE, the following ontologies were used: *Cyc-Transportation Ontology*, *World-Fact-Book* and *UNSPSC*.

2.4. Tools and Environment of the Test

The test team was made up of three of the experts involved in the heuristic evaluation phase. The environment in which the test was carried out was an isolated room with a personal computer running the Windows 2000 operating system. The user interaction was recorded with screen capture software, while one of the experts on the test team observed the participant's reactions. Each participant evaluated each of the OEs, but the order of evaluation was different for each participant in order to prevent biases derived from remembering previous scenarios.

3. Results

3.1. Test Results

Figure 1 summarizes the overall results obtained for each OE. The results were obtained by calculating the arithmetic mean of the time, in minutes, that each

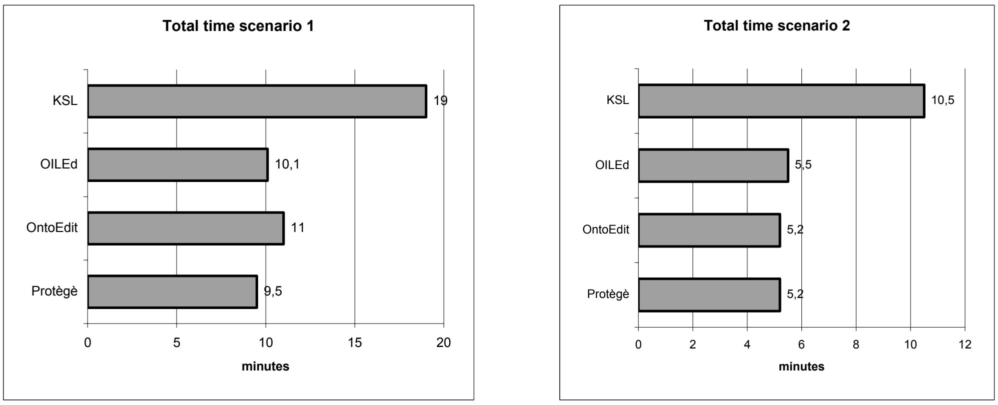
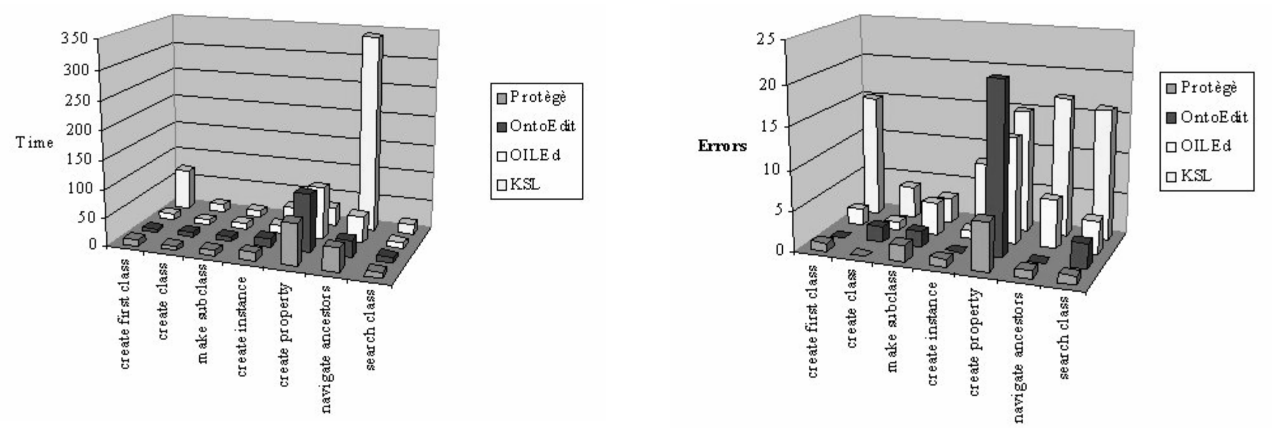


Figure 1. Overall results classified by ontology editor



	create first class	create class	make subclass	create instance	create property	navigate ancestors	search class
Protègè	10	8	10	15	74	41	9
OntoEdit	6	8	9	15	100	29	9
OILEd	11	8	10	14	90	47	9
KSL	72	15	11	24	32	340	18

	create first class	create class	make subclass	create instance	create property	navigate ancestors	search class
Protègè	1	0	2	1	6	1	1
OntoEdit	0	2	2	0	21	0	3
OILEd	2	1	4	1	13	6	4
KSL	15	4	3	8	15	17	16

Figure 2. Time (in seconds) needed to complete each task in each OE and number of errors recorded.

group (represented by the arithmetic mean of the minutes of its members) used to complete the first and second scenarios, respectively. Figure 2 shows in more detail, for each editor, the time (in seconds) spent by the groups in performing a specific task, and the number of errors each group made before the task was completed. Both measures represent the arithmetic mean of the members of the group.

In some cases, as for example the 'create class' task, the scenario involved several repetitions, and thus the time reported is the average time needed to complete a task. The measures should be considered approximate, since most of the users did not take a task-by-task approach, but instead explored the interface options, performing partial tasks that were then completed later.

The KSL ontology editor exhibited problems in both its orientation and navigability functions (e.g. frames that hid some functions, errors that did not provide links to go back, and difficulties for users in knowing what they were editing). These factors might account for the significantly higher times and error rates, which also increased significantly in the third user group recorded for the KSL ontology editor. Some participants in this third group were not able to complete the tasks in the estimated maximum time. In addition, the pages of the KSL editor do not fit the common visualization area of a browser, which results in scrolling and frame resizing, which, in turn, significantly increases the time-to-complete measure. These specific problems prevented a fair comparison between HTML-based and desktop-based interfaces.

The OntoEdit results show that only property-related operations were problematic for users, perhaps because most of them defined properties at a global level, thus preventing the users from attaching the properties to previously defined elements. This problem then caused user disorientation. Results for OILEd are of a similar magnitude, but specific problems arose in navigating the class hierarchy. Protégé results are slightly better than those of OntoEdit, but no significant conclusions can be drawn from them. The metamodel accessibility in both Protégé and KSL is perceived as a drawback that causes errors and disorientation, because non-specialists do not understand the need for such functionality. An overall analysis reveals that browsing large hierarchies in any OE environment is a time-consuming task, and that creating properties is an error-prone activity. Errors that occur during the creation of properties might perhaps be due to the duality between global and local properties.

3.2. Post-Questionnaire Results

In order to understand usability, it is important not only to measure user performance (effectiveness and efficiency), but also user satisfaction. A slightly modified version of the *System Usability Scale* (Brooke 1986) was used in a simple, five-item Likert scale (from 1-completely disagree to 5-completely agree) questionnaire. This questionnaire provided a global view of subjective assessments of usability, summarized in Figure 3. Note that the help system was not evaluated, since it was clearly identified as an area in need of improvement during the heuristic analysis phase.

The global scores clearly show that the KSL editor is perceived as complex and difficult to use. The notes of the evaluators corroborate this fact, as six of the participants complained about KSL, while four of them pointed out that the problem was that HTML-based interfaces are, in general, less usable. The high score for KSL in question 1 might correspond to the fact that most of the users found themselves lost while navigating in the KSL meta-model, because the meta-model is accessible through links in the OE. OILEd and OntoEdit obtain similar satisfaction scores, and are perceived as significantly easier to use than Protégé, with the exception of the responses to questions 4 and 7, which are directly linked to predictability. This result points out that some editing

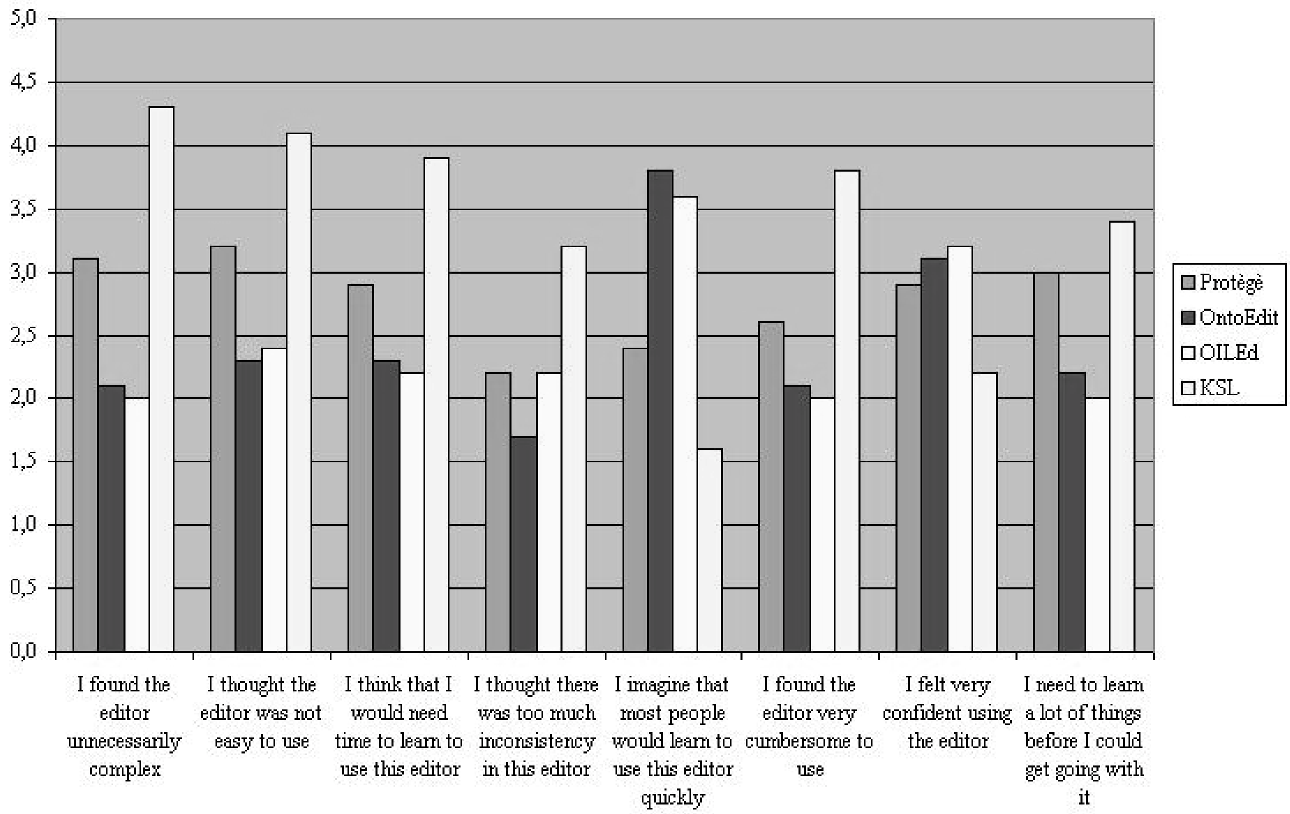


Figure 3. User satisfaction.

capabilities in Protégé that could be considered as advanced – such as the explicit edition of the meta-model – are perceived as unnecessarily complex for non-specialists. Another important conclusion is that no significant differences exist between the three user profiles, apart from a slight increase in the perception of ease of use in the third group.

The global satisfaction results for the three desktop editors show that all of them can be considered reasonably adequate for their purposes.

3.3. Summary of Major Problems

To summarize the study, a list of the most relevant areas for improvement was elaborated:

- Integrated, context-aware help systems should be developed.
- The meta-model should be considered an advanced feature, and, thus, it should be disabled by default. In addition, the use of a common meta-model terminology across OEs would be beneficial (e.g. providing a unified name for the concept of 'relation between classes,' because this is currently referred to as property, relation or slot depending on the OE), in order to hide the differences between the underlying ontology formalisms as much as possible.
- The language used in the tools should be oriented towards a non-specialized user community, thus avoiding language-specific constructs and terms.
- New interaction mechanisms to browse the generalization/specialization hierarchy should be explored. In this sense, editing should be based on a hierarchically structured view. Moreover, as in Protégé, hints should be given to recognizing multiple inheritance.
- Richer navigation and filtering mechanisms should be developed according to the user task model. For example, users should have the ability to navigate from a class to its instances, or to filter the visualization of classes by given criteria.

4. Conclusions

The overall conclusion is that current GUI-desktop-based OEs are fairly adequate for new users that prefer exploratory learning. A number of minor usability errors, which could be easily fixed, have been reported in this paper. In addition, a number of overall improvement areas have been identified, which may be the topic of future research work. As suggested by the evaluations, new visualization metaphors (e.g. 3-dimensional, filters on the class hierarchy) should be explored, because discovering the hierarchy of a specific class has been revealed to be a time-consuming task.

A more comprehensive evaluation is needed, both of the number of OEs (including *WebOnto* and *WebOde*, which posses interesting user interface characteristics) and the depth of the analysis. During this study, the authors observed (as it was previously supposed) that users usually prefer learning about how to use an ontology editor by directly using the tool, instead of by reading the documentation. For this reason, the cognitive walkthrough technique (Polson et al. 1996), which pays special attention to how well the interface supports *exploratory learning*, could be an interesting candidate for further evaluations.

References

Adelman, S.L., Riedel, L. 1997. *Handbook for evaluating knowledge-based systems: Conceptual framework and compendium of methods*. Boston, USA: Kluwer Academic Publishers.

Brooke, J. 1986. System usability scale (SUS), User information Architecture A/D Group, Digital Equipment Co. Ltd. Online at: http://www.usability.serco.com/trump/

Baclawski, K., Kokar, M.K., Kogut, P.A., Hart, L., Smith, J.E. , Holmes, W.S., Letkowski, J., Aronson, M.L. 2001. Extending UML to support ontology engineering for the Semantic web. In Martin Gogolla and Cris Kobryn (eds.) *UML 2001-The Unified Modeling Language. Modeling Languages, Concepts, and Tools. 4th International Conference, Toronto, Canada, October 2001, Proceedings*. Springer-Verlag LNCS (2185): 342-360.

Benjamins, R. Fensel, D. Decker, S. Gómez-Perez, A. 1999. $(KA)^2$: Building ontologies for the Internet: A mid-term report. *International journal of human-computer studies* 51:687-712.

Cranefield, S., and M. Purvis. 1999. UML as an ontology modelling language. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI99), Stockholm, Sweden, July 31-August 6, 1999*, ed. Thomas L. Dean. San Francisco: Morgan Kaufmann. Available online at: http://hcs.science.uva.nl/usr/richard/workshops/ijcai99/UML_Ontology_Modelling.pdf>

Cranefield, S., S. Haustein and M. Purvis. 2001. UML-based ontology modelling for software agents. In *Proceedings of the Fifth International Conference on Autonomous Agents*, *Montreal, Canada, May 28-June 1, 2001*, eds. Jörg Paul Müller, Elisabeth Andre, Sandip Sen and Claude Frasson. New York: ACM Press.

Davis, R., Shrobe, H., Szolovits, P. 1993. What is a knowledge representation? *AI magazine* 14(1): 17-33.

Decker, S.D. Fensel, F., Harmelen, I., Horrocks, S., Melnik, M., Klein, J. B. 2000. In *Knowledge representation on the Web, Proceedings of the 2000 International Workshop on Description Logics. Aachen, Germany. CEUR Workshop proceedings*. http://CEUR-WS.org/Vol-33/

Denny, M. 2004. Ontology Building: A Survey of Editing Tools. In O'Reilly, *XML.com*. http://www.xml.com/pub/a/2004/07/14/onto.html

Ding, Y., D. Fensel, Klein, M., Omelayenko, B. 2002. The semantic web: Yet another hip?. In *Data and knowledge engineering* 41: 205-227.

Domingue, J., Motta, E. 1999. A knowledge-based news server supporting ontology-driven story enrichment and knowledge retrieval. In *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling, and Management (EKAW '99)*. Lecture notes in artifical intellegence 1621. Berlin, Germany: Springer-Verlag.

Duineveld, A. J., Stoter, R., Weiden, M. R., Kenepa, B., Benjamins, V. R. 2000. Wondertools? A comparative study of ontological engineering tools. In *International journal of human-computer studies* 52:1111-1133.

Dumas, J. S., Redish, J.C. 1999. *A practical guide for usability testing*. Sheffield, UK: Intellect, Ltd.

Fensel, D., Horrocks, I., Van Harmelen, F., Decker, S., Erdmann, M., Klein, M. 2000. OIL in a nutshell. In *Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*. Lecture notes in artificial intelligence. Berlin, Germany: Springer-Verlag.

Fensel, D. 2001. *Ontologies: Silver bullet for knowledge management and electronic commerce*. Berlin, Germany: Springer-Verlag.

Gómez-Pérez, A. 1994. *Some ideas and examples to evaluate ontologies*. Tech. report # KSL-94-65. Knowledge Systems Laboratory. Stanford University. ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-94-65.ps.gz

McGuinness, D.L., Patel-Schneider, P.F. 1998. Usability issues in knowledge representation systems. In *Proceedings of the 15th National Conference on Artificial Intelligence. Madison, Wisconsin, USA: American Association on Artificial Intelligence,* pp. 608-614.

NCITS (National Committee for Information Technology Standards)-TC T2 (Information Interchange and Interpretation). 1998. American national standard for Knowledge Interchange Format Proposed Draft, NCITS.T2/98-004. http://logic.stanford.edu/kif/dpans.html

Nielsen, J. 1992. Finding usability problems through heuristic evaluation. In *CHI '92 conference proceedings: Striking a balance, Monterey, California*, eds. Penny Bauersfeld, John Bennett, and Gene Lynch. New York: Association for Computing Machinery; Reading, MA: Addison-Wesley, pp. 373-380.

Nielsen, J. 1994. Heuristic evaluation. In *Usability inspection methods*. John Wiley & Sons.

Nielsen, J. 2000. *Designing web usability: The practice of simplicity*. Indianapolis, USA: New Riders Press.

Noy, N.F., Fergerson, R.W., Musen, M.A. 2000. The knowledge model of Protege-2000: Combining interoperability and flexibility. In *Proceedings of the 2nd International Conference on Knowledge Engineering and Knowledge Management (EKAW' 2000), Juan-les-Pins, France*.

Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., Musen, M.A. 2001. Creating semantic web contents with Protege-2000. In *IEEE intelligent systems* 16:60-71.

Object Management Group. 2003. The Unified Modeling Language specification, version 1.5. http://www.uml.org

Polson, P.G., Lewis, C., Rieman, J., Wharton, C. 1992. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. In *International journal of man-machine studies* 36:741-773.

Shahar, Y., Chen, H., Stites, D. P., Basso, L., Kaizer, H., Wilson D. M., Musen, M. A. 1999. Semiautomated entry of clinical temporal-abstraction knowledge. In *Journal of the American Medical Informatics Association* 6:494-511.

Tognazzini, B. 2002. First principles and design http://www.asktog.com/basics/firstPrinciples.html