NeOn-project.org

NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 – "Semantic-based knowledge and content systems"

D 4.1.1 Analysis of user needs, behaviours & requirements wrt user interfaces for ontology engineering

---

Deliverable Co-ordinator: **Martin Dzbor (OU)**

Task Co-ordinating Institution: **Intelligent Software Components (ISOCO)**

Contributors: **Enrico Motta (OU), Jose Manuel Gomez (ISOCO), Carlos Buil Aranda (ISOCO), Klaas Dellschaft (UKO-LD), Olaf Görlitz (UKO-LD), Holger Lewen (UKARL)**

In this deliverable we summarize the rationale, motivation, methodology and findings related to the observational user study that has been conducted on 31 participants using existing tools supporting ontology engineering. The study looks, in particular, at various aspects influencing effectiveness, efficiency and user satisfaction while carrying out an engineering task with three public ontologies

Document Identifier:	NEON/2006/4.1.1/v1.0	Date due:	August 31, 2006
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	August 31, 2006
Project start date:	March 1, 2006	Version:	v1.0
Project duration:	4 years	State:	Final
		Distribution:	Public

NeOn Consortium

This document is part of a research project funded by the IST Programme of the Commission of the European Communities' grant IST-2005-027595. These partners are involved in the project:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta} @open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 28 D-76128 Karlsruhe, Germany Contact person: Peter Haase E-mail address: pha@aifb.uni-karlsruhe.de
Universidad Politécnica de Madrid (UPM) Campus de Montegancedo 28660 Boadilla del Monte Spain Contact person: Asunción Gómez Pérez E-mail address: asun@fi.upm.es	**Software AG (SAG)** Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
Intelligent Software Components S.A. (ISOCO) Calle de Pedro de Valdivia 10 28006 Madrid Spain Contact person: Richard Benjamins E-mail address: rbenjamins@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
Institut National de Recherche en Informatique et en Automatique (INRIA) ZIRST – 655 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat E-mail address: jerome.euzenat@inrialpes.fr	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham E-mail address: hamish@dcs.shef.ac.uk
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	**Consiglio Nazionale delle Ricerche (CNR)** Institute of cognitive sciences and technologies Via S. Martino della Battaglia, 44 - 00185 Roma-Lazio, Italy Contact person: Aldo Gangemi E-mail address: aldo.gangemi@istc.cnr.it
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	**Asociación Española de Comercio Electrónico (AECE)** C/Alcalde Barnils, Avenida Diagonal 437 08036 Barcelona Spain Contact person: Gloria Tort E-mail address: gtort@fecemd.org
Food and Agriculture Organization of the United Nations (FAO) Viale delle Terme di Caracalla 1 00100 Rome, Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@fao.org	**Atos Origin S.A. (ATOS)** Calle de Albarracín, 25 28037 Madrid, Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document: The Open University (OU); iSOCO (ISOCO); Universität Koblenz-Landau (UKO-LD); Universität Karlsruhe (UKARL)

Change Log

Version	Date	Amended by	Changes
0.1	20-07-2006	Martin Dzbor	Theory, motivation, setup, variables
0.2	01-08-2006	Martin Dzbor	Summary of findings, analysis
0.3	08-08-2006	Martin Dzbor	Further analysis, discussion
0.4	16-08-2006	Martin Dzbor	Feedback inclusion, conclusions, implications
0.5	18-08-2006	Martin Dzbor	Executive summary, abstract, references
0.6	05-09-2006	Martin Dzbor, Diana Maynard	Comments from review by Diana Maynard
1.0	11-09-2006	Martin Dzbor	Addition of appendices, PDF creation

Executive Summary

In this deliverable we summarize the rationale, motivation, methodology and findings related to the observational user study that has been conducted on 31 participants using existing tools supporting ontology engineering. The study looks, in particular, at various aspects influencing effectiveness, efficiency and user satisfaction while carrying out an engineering task with three publicly available ontologies. This study differs from other previous studies in aiming at more general developer audience (not solely 'power users') and in using a task-based approach. A 'power user' in this context roughly corresponds to a knowledge engineer or logician skilled in formal representation formalisms such as OWL. The rationale, background and setup of the study is summarized in sections 1 through 3.

The key findings from the study are that ontology engineering tools have made substantial progress over the past decade, yet some issues with their accessibility and flexibility remain. In particular, there were obvious deficiencies in the gap between the languages of representation and languages of conceptualization, lack of usable visual and navigational support, lack of support for managing, visualizing and navigating through the networks of imported and otherwise dependent ontologies, and also a significant amount of confusion due to non-standard user interface metaphors, visual elements and ambiguous components. Various findings are summarized and briefly commented on in section 4, which is sub-divided into four sub-sections each looking at one of the four areas of focus: effectiveness, efficiency, user experiences and attitudes to functionalities.

The study has identified several most frequently carried out operations that are involved in integrating networked ontologies – with some minor surprises. For example, the users had to perform such operations as search or import more frequently than they thought which in the case of searching was partially due to substituting this operation for an impromptu 'spell-check'. Furthermore, we found that people had a tendency to make an overall neutral judgement about a tool, despite expressing fairly strong positions against specific features or shortcomings. This and various other findings, together with correlations between some findings, are discussed more broadly in sections 5 and 6.

Table of Contents

List of tables

List of figures

1. Introduction

Ontologies provide a key technology to support interoperability on the Web and for enabling semantic integration of both data and processes. Since the time the notion of ontology was first proposed by Tom Gruber, ontologies matured, and we are now entering a phase, in which ontologies are produced in larger numbers and exhibit greater complexity than before. In recent years we saw substantial effort on ontologies in the medical, genetic engineering and bio-medical domains, as well as on generic, top-level ontologies. The authors of these ontologies faced many challenges, some were domain-related, others were social. However, a substantial challenge comes from lack of appropriate tools. For example, the most popular tool available today for ontology development, Protégé, supports building an ontology (usually from scratch and usually in the low-level representation formalism), but offers limited support for managing this ontology, integrating it with other ontologies or re-using it elsewhere.

Most tools provide some editing facilities, navigation support, reasonable language-level interoperability, and usually, a database support. Some provide limited support for collaboration and ontology versioning. Unlike the current situation, NeOn adopts the view that ontologies will be *networked*, *dynamically changing*, *shared* by many applications and strongly dependent on the *context* in which they were developed or are used. The aim of this document is to investigate the shortcomings of current support mechanisms for working with such relatively advanced ontologies. Additionally, we aim to gather material about the needs of users that would inform designers of NeOn tools, techniques and methods.

Many past projects on semantic technologies paid limited attention to the user with the result that much ontology engineering technology is tried out and discarded by the user after a brief trial. At The Open University we recently collaborated on creating an ontology with a well-known international publisher, and found out that their tool of choice was simply a *word processor*. Apparently, they tried and rapidly discarded the available ontology engineering tools – as these were simply too difficult to use. While this is an extreme reaction, it is undeniable that little attention has been paid to the needs of ontology authors and domain experts so far.

As a first step in recognizing the importance of acceptability of a tool by the users, we aim to actively investigate what we tentatively label as *human-ontology interaction*. A direct consequence of this focus is the task intending to conduct an observational user study in order to further our understanding of *what difficulties* the users of ontology engineering tools face – now, at the beginning of the project. The corollary of an early needs observation is the creation of a baseline for assessing any future NeOn toolkit developments.

In this document we report on the motivation, setup and analysis of an observational study that focused on the integration of ontological definitions from several, non-trivial ontologies. The study expected the users to carry out a series of tasks and then express their attitudes, opinions and suggestions with regard to the effectiveness, efficiency and overall user experience with the tool support for these tasks. In addition, users expressed their attitudes to a range of existing functions related to networked ontologies and to a range of envisaged functionalities of tools supporting the work with such ontologies. The details of the study setup appear in section 3, the account of the key findings (including the distributions of participants' attitudes and statistical significances in their variances, if applicable) is given in section 4. Further analysis and correlations among findings are discussed in section 5, and section 6 contains discussion and implications from this study. However, before presenting the actual study, we start with a broader motivation for considering the interaction between users and ontologies as an integral part of the development of new tools. This motivation forms section 2 and also contains brief summary of previous tool evaluations.

2. Interactions between people and ontologies

As long argued in the human-computer interaction (HCI) domain, interactions involve three parts: the user, the technology, and the ways they work together [15]. We expand these notions to human-ontology interaction with the aim of securing a place for the human users, the networked ontologies and their mutual interaction within a realistic ontology lifecycle. This aspect of NeOn focuses on the means of customizing and adapting an ontology for a particular purpose, for a stage of ontology evolution, and for a range of contexts.

It is clear that technologies (and hence, applications) that fulfil at least some needs of ordinary users carrying out their daily tasks have a much better chance of becoming broadly adopted. The use of a certain technology, no matter how good it is, does not guarantee that the application supports users in the right tasks or that the users have a good user experience when performing the tasks. According to [15], at a certain development stage successful applications are required to balance technology with user experience and functional features.

Good user experience for non-technical users is often best achieved when the technology, such as ontology engineering in general, is hidden from the users, or at least, the systems and tools subscribe to the same or similar models as the user. Each user engaged in any interaction has a *task model*; this reflects the user's subjective understanding and expectations about the activities that need to be performed to reach a goal. On the other hand, the user's *model of the system* reflects the user's understanding and expectations from the tool, and how this tool can be used to perform the tasks implied by the task model. The major gap between the two models is due to their implicit nature. It means that model of a system is often unknown to the end user and its working is established from the interaction with the system's user interface. Sometimes the user interface reflects the view of system designers, not of the users, which leaves the user to guess what the tool capabilities and functions actually mean for their tasks and how they correspond with their user models of the activities carried out.

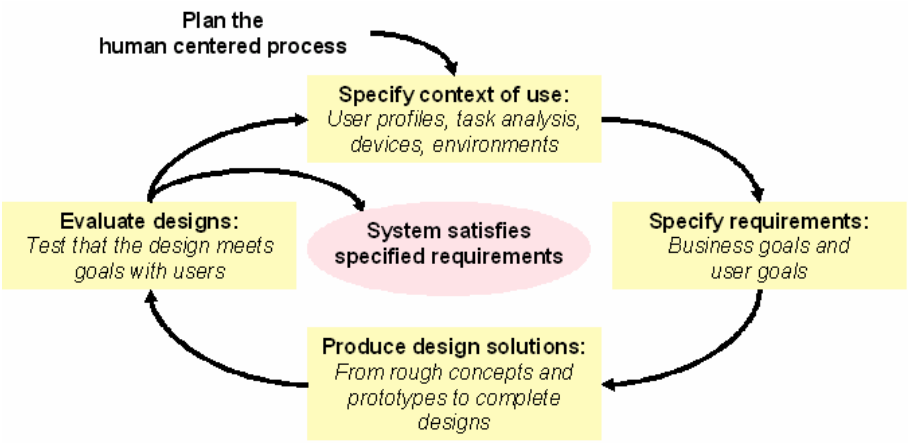

Figure 1. A typical user-centred development spiral [1]

Successful tools are typically built thanks to a substantial understanding of the users, their tasks, their goals, and their environments. Development of such application usually includes iteration and user evaluation of the intermediate designs. A general process for including human-centred activities throughout a development lifecycle has been standardized in ISO 13407[1]. One benefit of the design spiral, as shown in Figure 1, is that it helps to bring in different aspects of user experience and needs early in the application lifecycle increasing the chance to develop successful applications. According to Figure 1, NeOn and its toolkit are currently at the stage of planning and

[1] For an overview of this standard see e.g. http://www.ucc.ie/hfrg/emmus/methods/iso.html.

specifying the user profiles, understanding users' tasks, preferences, etc. – at least, with respect to this particular work package and user study.

The correct definition of a problem and its context is only half the success. Technology-driven models for problem solving, such as computational models [14, 23] often neglect the need for a problem interpretation from the user's viewpoint. Knowing the users, their tasks and the context helps designers to understand the effects of their design choices. This is particularly acute in domains like ontology engineering, where the product is represented in a formal language that is often alien to the ordinary users. *Adaptation* of the tool to the user's language or problem understanding is more difficult than the (often forced) *adoption* of the tool by the user, but, as we mentioned earlier, could make a difference between continuing to use and discarding a particular tool.

2.1 Past findings

In this section we summarize the findings from some evaluation work done in the past. It is, however, difficult to use these findings as any kind of baseline for our work. This is for several reasons: (i) in many cases only a specific facet of the tool was assessed (e.g. visualization), (ii) the scope of evaluation was orthogonal to our study (e.g. collaboration support), or (iii) significantly different tools were assessed. Nevertheless, it is useful to be aware of past findings, and perhaps attempt, at a sufficiently high level of abstraction, to observe the evolution in user needs and/or experiences.

The authors in [5] conclude that in general, tools at the time of their survey existed as research prototypes with limited intuitiveness of user interfaces, which made the use of tools hard for inexperienced users. This is in line with the anecdotal experience from The Open University's work with several domain experts, who often switched to more generic tools, such as word processors or UML modellers. One of their interesting observations concerned a large selection of different options that provided rich opportunities to tweak the developed ontology, but universally led to the user's confusion. Facets like graphs, visual elements, warnings or messages were generally found to be poor, which only added to the already steep learning curve for a non-expert user. Another generic observation is that tools often expect the user to be fairly proficient in low-level representational formalisms – so, rather than an exercise in conceptual modelling, ontology engineering often resembles an exercise in coding using a particular language.

Another team of authors [19], on the other hand, evaluated Protégé from a power user's perspective. They found the tool intuitive for experienced knowledge engineers, as long as the operations were triggered by the user (e.g. knowledge re-arrangement). However, difficulties arose when help or assistance from the tool was expected; e.g. in inference or consistency checks. Weak performance was also noted on the level of language interoperability; i.e. although representational standards were met during imports and exports, this did not guarantee that the ontologies would be indeed reusable.

In their survey of ontology engineering tools, authors in [7] found that there was a wide range of tools, each developed with for a slightly different purpose or audience, but all exhibiting much shared functionality. A typical example are ontology editors or annotating tools – some editors were more graphics-centred, others were predominantly text-based, but all shared methods for simple transformation of serialized formal representations into graphs or trees. Similarly, some annotation tools were solely for manual annotation, others supported some pattern learning, but they typically shared a notion of embedding annotations into the original text by means of some graphical widgets (e.g. coloured highlighting).

Nonetheless, the authors note significant issues with tool interoperability and also with support for more advanced operations on ontologies beyond mere editing (e.g. integration or re-use). In

particular, they point to the lack of ontology interoperability; namely, that this increases chances of from-scratch design instead of re-use. Also, such an attitude towards re-use may lead to what they call “a one-tool market” where ontologies designed by tool X would be imperfect but at least readable by others using the same tool X that dominates the market. In terms of other general findings, the survey notes that there is limited ‘intelligence’ in the engineering tools – e.g. no support for repetitive operations and no possibility to re-use previously used processes, versions or cases in current design. In their view, most tools expect the user to drive the interaction and the task, with the tool imposing rather severe constraints rather than adapting itself to the users’ needs. Nonetheless, as with previous studies, [7] also evaluated the tools from the expert’s and researcher’s perspective rather than observing the users who cannot be classified as power users or experienced knowledge engineers.

A range of function-specific studies has been conducted. For instance, authors in [27] surveyed Protégé users about their ontology visualization experiences with extensible visualization widgets. They observed that such tools and their customization models are too complex and do not reflect users’ models of what they would normally want to see. Furthermore, lack of tool support for interpreting and explaining the visualized fragments only added to the complexity and reduced user acceptance. In the context of ontology (and specifically, OWL) debugging, cleaning and maintenance, authors in [10] mention that users have great difficulties with description logic based formalisms in general. Tools often expect detailed knowledge of intricate language and logic details, and this often leads to modelling errors. They particularly point to syntactic issues caused by the unintended migration between different OWL species, but also highlight the lack of tool support for alerting users about potential clashes in larger and imported ontologies.

Other studies on generic user interaction with description logic based formalisms apply (to some extent) to ontology engineering tools, which are based on logical representations. For instance, authors of [20] discuss issues with ontology reusability and complexity of this activity largely due to grounding in abstract logical syntax. They observe the need to customize reusable, generic formal representations to a form or intermediate representation that is more accessible to the user working in a specific domain. They also observe high complexity of user interaction, and ascribe it to the rich choices in underlying representations, similarly to findings mentioned by others above.

2.2 Conclusions from the past evaluations

To summarize, existing empirical work highlighted several problems with ontology engineering tools. However, at the beginning of the NeOn project we felt that there was a need to conduct a novel study, as none of the studies mentioned above provided the kind of data we wanted to obtain as a baseline to inform the development of the next generation ontology engineering tools envisaged by NeOn. Specifically, the studies did not satisfactorily address the following key concerns of our project:

- **Emphasis on ‘normal users’.** As ontologies become an established technology, it makes less sense to focus only on highly skilled knowledge engineers. There are so many organizations currently developing ontologies that it seems safe to assert that indeed most ontologies are currently built by people with no formal training in knowledge representation and ontology engineering. Therefore, it is essential to conduct studies which focus on ‘normal users’, i.e., people with some knowledge of ontologies, but who cannot be classified as ‘power users’.
- **Emphasis on ontology reuse.** NeOn adopts the view that ontologies will be *networked*, *dynamically changing*, *shared* by many applications and strongly dependent on the *context* in which they were developed or are used. In such a scenario it would be too expensive to develop ontologies ‘from scratch’, and the re-use of existing, possibly imperfect, ontologies becomes the key engineering task. Thus, it makes sense to study the broad re-use task for OWL ontologies,

rather than focusing only on a narrow activity (e.g. ontology visualization or consistency checking).

- **Formal definition of ontology engineering tasks.** Studies reported earlier focused on generic tool functionalities, rather than specifically assessing performance on concrete ontology engineering tasks. This creates two problems: (i) the results are tool-centric, i.e., it is difficult to go beyond a specific tool and draw generic lessons on how people do ontology engineering tasks; (ii) by assessing the performance of our users on concrete tasks using OWL ontologies, we acquire robust, benchmark-like data, which (for example) can be used as a baseline to assess the support provided by other tools (including those we plan to develop in NeOn).

For these reasons we decided to conduct our own user study, addressing the three motivational criteria described above. We describe the actual study, its methodology in detail in the next section, and then complement this with some findings and their discussion.

3. Observational Study of Users

As has been mentioned earlier, there are many different modes of interaction between the user and the computer. Even with restricting ourselves to one particular type of tools – ontology engineering environments – there are potentially many types of human-tool or human-computer interaction (HCI). To make the process of gathering requirements and observing user needs manageable, we have formulated a few assumptions to frame the study. The most critical ones are as follows:

- Focus on ontology *engineering* – as opposed to e.g. annotation with or population of ontologies;
- The phases of ontology *re-use*, *integration* and *adaptation/customization* – i.e. activities like knowledge acquisition, conceptual modelling, semi-automated ontology mapping or validation were not considered in this study;
- Working with *knowledgeable users* – i.e. participants to the study are people knowing something about building ontologies and ontology languages, absolute novices were not considered;
- Running the study within *generally comprehensible (but not trivial) domains* – as opposed to working with ontologies that are highly domain specific and require considerable domain expertise to understand and to conceptualize;
- Conduct an *observational study* rather than experiment – we are interested in capturing information on user needs, requirements and gaps in the tool support, rather than e.g. solely testing the performance of different tools or different levels of expertise

Next, we briefly justify these assumptions. Networked, potentially large and intertwined ontologies raise many challenges in terms of human-ontology interaction. Nevertheless, the need is determined by the aims of the project. NeOn is investigating different aspects, such as dynamics of networked ontologies, collaborative development and re-use of networked ontologies, contextualization of networked ontologies, and even more sub-topics. Many of these aspects are currently supported to a very limited extent. This lack of tools and techniques makes it rather difficult to assess the actual user performance in any of these tasks. If there are any practical procedures, e.g. for contextualizing a third-party ontology, these procedures tend to be domain specific and often serendipitous.

In other words, had we decided to conduct merely a mock-up user study without any understanding of what toolkit we may offer to the users, there was a substantial danger of making

observations biased to a specific organization, a specific problem domain or indeed, a particular school of thought. There is no doubt that such concrete observations are valuable; however, in the context of NeOn, such requirements and needs are gathered from the primary sources – the use case partners in the respective work packages. Hence, it is possible to make a high-level distinction between studying the users in NeOn *application cases* and studying *more generic, engineering* needs. As the distinction suggests, it is likely that use cases provide requirements and insights on a particular, ontology-driven *application*; whereas if we discount the application element, we are more likely to observe requirements and insights on a broader ontology engineering *task* or *process*.

To satisfy this broad assumption, we opted for studying a broad feature that to some extent appears in all the different phases of the ontology lifecycle. One such shared feature, and indeed one core focus of the project, is the notion of *ontology networks*. While other NeOn work packages are formalizing the model of networked ontologies, ontology engineers are already familiar with the notion of networking – or *integrating* ontologies. By its definition [8], an ontology is an artefact that is shared and that serves to integrate the views of different parties. One obvious way of sharing and integration is so-called multi-agent, when two or more agents/users choose an ontology as a common vocabulary to describe their own problems. Another, less obvious way of sharing and integration may be called temporal, where one or more agents re-use previously agreed ontologies, perhaps originating in different domains.

Hence we ground the study in *integrating and re-using* ontologies that are available on the public Web. As we show further in the description of study tasks, ontology integration is sufficiently broad so that it gives us some insight into aspects of contextualization and network dynamics, without confounding the study with an improper view of how context should/could be supported, for instance.

As summarized earlier, other evaluation studies have been conducted; for instance, [5] or [19]. In particular, the former provides useful advice about some parameters for analyzing an ontology engineering tool. While some of their questions are too low-level, and since the time of publication many questioned *functions* are in the engineering tools by default, questions from the 'general' category provide a valuable input to parameters we chose for analyzing tool effectiveness and efficiency. Another difference of our approach from the reported ones concerns the evaluators – in both cases, the evaluation was done by knowledge engineers and expert modellers. In our case, we opted for a clearer split between the modellers/engineers and evaluators – hence, we talk about observational study rather than mere tool evaluation. Finally, although authors in [5] evaluated as many as six tools, since then any substantial development and market penetration was achieved solely by the Protégé team – especially in terms of coping with emerging standard representational languages (such as OWL). Therefore, our decision of which tools would participate in the study was (i) different from theirs and (ii) biased towards OWL-based integrated environments rather than mere ontology editors.

3.1 Study setup and methodology

The observational study (further only 'the study') focused on the integration of ontological definitions from several, non-trivial ontologies. All three studied ontologies were publicly available on the Web, and all were results of principled engineering processes and knowledge acquisition. The ontologies chosen for the study aim to model domains that are comprehensible to essentially any knowledge engineer. The following list summarizes the re-usable ontologies and lists their characteristics:

- *Copyright* ontology[2] … a medium-sized ontology capturing the concepts of different kinds of rights and being able to perform different actions with those rights (e.g. a right to perform or sell, economic or moral rights, etc.)
- *AKT Support* ontology[3] … a small upper-level ontology defining concepts like time, temporality, tangibility, etc.
- *AKT Portal* ontology[4] … a medium- to large sized domain ontology focusing on various aspects of organizational life, such as different kinds of agency, organization, etc., and also models of various academia-specific activities (e.g. publications, research projects, students, etc.) – all for the purposes of driving an organizational knowledge portal

Table 1. Features of the ontologies used in the study

Ontology	Lang	Classes	Properties	Restrictions	Notes
Copyright	OWL	85	49	128	Mostly cardinality & value type restrictions, some properties untyped
AKT Support	OWL	14	15	n/a	All properties fully typed
AKT Portal	OWL	162	122	130	10 classes defined by enumeration, most properties untyped

The *Copyright* ontology was chosen as a base that was to be adapted by re-using, integrating and/or committing to the terms from the other two ontologies. The study comprised three tasks in increasing order of complexity, which were presented to the participants one at a time. The study was carried out on each participant working individually, but it was *facilitated* by a member of the study team. The facilitator's role was giving answers to participant's enquiries, asking questions why a particular decision has been made, etc. Facilitators were also given a typical solution to the tasks and instructed to note their observations of what was happening during the study, what queries were raised and what responses were given.

Participants were not asked to follow any particular ontology engineering methodology, but the facilitator could ask them how and why a specific decision was made. The aim of the study was not to assess domain expertise of the participants nor their proficiency in OWL. When invited to the study, we confirmed participants' level of expertise, familiarity with ontology engineering tools and representation languages. They were all expected to have knowledge of basic OWL features (e.g. sub-classing, restriction,…) but not of the advanced ones (e.g. SPARQL, C-OWL, e-connections,…).

At any time during the session, the participant could ask a clarifying question, and was provided with tools like pen and paper to formulate ideas, proposals or approaches to the task. During the study, they were recorded using Techsmith's screen capturing software Camtasia[5], and at the end they were asked to fill in a questionnaire centred on their experiences of different aspects of the interaction.

Altogether we worked with 31 participants from four different institutions, which included academic and industrial ones. Participants were mixed in terms of different level of experience with designing ontologies and with using ontology engineering environments. Three such environments were considered for the study – Protégé from Stanford University[6], OntoStudio from Ontoprise GmbH[7],

[2] Copyright ontology has been accessed from http://rhizomik.net/2006/01/copyrightontology.owl
[3] AKT Support ontology has been accessed from http://www.aktors.org/ontology/support
[4] AKT Portal ontology has been accessed from http://www.aktors.org/ontology/portal
[5] Information on Camtasia and download are available from http://www.techsmith.com/camtasia.asp.

[6] More information on Protégé and the download of the latest version is available from http://protege.stanford.edu.

[7] Information about OntoStudio is available through http://www.ontoprise.de/content/e1171/e1249/index_eng.html.

and TopBraid Composer from TopQuandrant[8]. However, during the pilots it turned out that the version of OntoStudio we intended to use did not handle the OWL-Full fragment – mostly due to such features as untyped properties. Nevertheless, as our intention was to use third-party ontologies *as found* we decided to exclude OntoStudio from this round, rather than start amending the ontologies. That left the study with two tools – Protégé 3.2.6 and TopBraid. Since the former is the market leader and the latter was only just released, we asked participants to assess their experience with using Protégé in the past. Participant cohorts were composed so that we had 15 people considered highly experienced or expert and 16 considered less experienced. This roughly balanced the levels of expertise among our participants. The classification criterion for considering someone "experienced" was the number of ontologies designed, their type and complexity, and also we took into account the frequency of using ontology engineering environments (due to market dominance, Protégé version 3.2.6 or any earlier was considered).

Another criterion we applied to our group of participants was to which tool we expose them during the user study. Due to licence restrictions on the TopBraid, we could not achieve a balance between the subsets, but this was later taken into account in the analysis. At the end, out of 31 participants, there were 18 assigned to carry out the study with Protégé and 13 worked with TopBraid Composer.

3.2 Tasks given to users

As mentioned earlier, participants were given three tasks – all considering different means of integrating the individual ontologies into a network. Tasks were of increasing complexity and open-endedness with Task 1 being simplest and most precisely set, and Task 3 being most complicated and requiring the participants to recognize the core of the problem and break the overall goal into operational steps.

Each task presented to the participants had the following structure: high-level *motivation* that gave the background and rationale for the problem, then *task specification* setting task-specific objectives and finally, any *additional information* or examples (if applicable). Each task was on a separate sheet of paper so as not to distract their attention from the task at hand, and with enough space for participants' notes.

3.2.1 Task 1: Simple ontology import and sub-classing

In Task 1, participants were pointed to concept *CreationProcess*, which informally refers to temporal aspects of the creation. Then participants were alerted that the *Copyright* ontology did not contain any formalization of temporal aspects, and thus had to be augmented with the relevant definitions from other ontologies, which already contain conceptual definitions of time and temporal aspects (e.g. *AKT Support*).

The objective was to review the three given ontologies, locate the classes in question (i.e. *CreationProcess* and *Temporal-Thing*), and assert that *subClassOf (CreationProcess, Temporal-Thing)*. Participants were expected to import *AKT-Support* into *Copyright*, and were prompted to ensure that it was indeed possible to express various temporal aspects – e.g. by pointing to the list of properties and sub-classes. The objective of the task was to observe the approaches to search, browse and import ontologies in a procedurally correct way. This task was also considered as an aid to familiarize participants with different parts of the user interface.

[8] TopQuandrant download and further details are available from http://www.topquadrant.com/tq_topbraid_composer.htm.

3.2.2 Task 2: Import of two ontologies and axiom amendments

In Task 2 we motivated the problem by pointing to a western-centric notion of any right being associated solely with a person. This formally excluded a community or organization having a collective right (rather than a set of individual rights). Participants were asked to review the concept of *copyright: Person*, and replace (or otherwise adapt) it with more formal and deeper conceptualizations from the *AKT Portal* and *AKT Support* ontologies, in particular, considering different levels of individual and communal agents. Four sub-tasks were given, each expecting the participants to amend an existing axiom in the *Copyright* ontology. Specifically, participants were asked to do the following steps:

- Change the range of property '*agent*' (i.e. owner) for concept *Economic-Rights* to class *akt:Legal-Agent*;
  - Sample solution: *rangeOf* (*agent* , *Legal-Agent*)
- Change the range of the same property to something broader for concept *Moral-Rights*;
  - Sample solution: *rangeOf* (*agent* , *Generic-Agent*)
- In the case of the concept *Distribute*, properties '*agent*' (owner) and '*recipient*' should formally be different levels of an agent (not necessarily *Legal-Agent*);
  - Sample solution: *rangeOf* (*recipient* , *Generic-Agent*)
- Express for the concept *Distribute* the fact that "recipients can be any generic agents except geopolitical entities"
  - Sample solution: *rangeOf* (*recipient* , (*Generic-Agent* $\wedge$ ($\neg$ *Geo-Political*)))

The objective in this task was to observe how people formulate restrictions of increasing logical complexity, and how they go about finding and comparing relevant parts of the ontology.

3.2.3 Task 3: Use of imported concepts to formally (re)define an existing one

The motivation for this task was similar to Task 2 – but this time motivated in terms of formally refining the definition of the concept *Collective* inside the *Copyright* ontology so that formal definitions reflect the informal description as closely as possible. Participants were told to make amendments only to their base – *Copyright* ontology, rather than to the other two. We expected them to propose a strategy that would enable them to make *akt:Organization* a sub-class to *copyright:Collective* without making modifications to any AKT ontology. The task was first, to recognize that this could only be achieved by creating new local sub-classes to the concept *Collective*, and making them equivalent to the actual AKT classes, and second, to realize this as, e.g.:

1. Creating new concept *subClassOf (copyOrganization, Collective)*

2. Making this new concept equivalent to an existing one *copyOrganization* $\equiv$ *akt:Organization*

3. Proving that indeed the tool 'knows' about intended meaning *subClassOf (akt:Organization, Collective)* by running an inference engine to classify the classes/individuals
4. Attempts as e.g. *Collective* $\equiv$ *(akt:Organization* $\sqcup$ *akt:Organization-Unit*) were considered incorrect
5. Participants were prompted to explain their formalizations.

The second half of Task 3 tackled the second half of informal description – i.e. making sure that any individual belonging to the concept *Collective* comprises at least two objects classed as

akt:Persons. In other words, a group has to contain minimum two people to be considered an instance of a *Collective*. This sub-task expected participants to browse the list of properties, introduce a new property (e.g. *hasMember*) and appropriately restrict this property for concept *Collective*. For example, the outcome was expected to contain the following statements:

- Introducing domainOf (*hasMember* , *Generic-Agent*)
- Introducing rangeOf (*hasMember* , *Generic-Agent*)
- Formulating intersection *hasMember* $\in$ { X: *Person*(X) } $\sqcap$ cardinalityOf (*hasMember*) $\geq$ 2

3.3 Evaluation and analysis instruments

In the group discussions in work package 4, the partners agreed that the primary goal of the study is to identify capabilities and gaps in the existing support tools and empirically justify why and where various aspects of the NeOn vision could be useful. To that extent it was decided to base our analysis on *observing users* carrying out a specific task using existing tools, and express their interaction in a set of measures looking at different facets. In other words, we aim to conduct an *observation* of user experiences that would inform design of new tools in the context of NeOn. In selecting the measures, two key constraints were observed: first, data to be gathered should not be too tool-specific (e.g. particular version of Protégé), as it was not our objective to prove that one environment is better than other. Second, while generic tool usability was considered important, measures were expected not to be solely usability-centric.

Nevertheless, although the objective may not be to prove that one tool is better than the other, parts of the analysis in this report are concerned with comparing the two tools – since this is a good way to identify gaps or problems with one tool that are not present in the other tool, and hence to take account of for future design. Moreover, asking users to perform tasks using two specific tools and then evaluating their performance and opinions about the tools is not necessarily equivalent to an evaluation of those tools. Rather, we are using this evaluation as a means to an end; i.e. to improve our understanding of how existing tools support the users in carrying out tasks involving ontology networks.

Tool and expertise level would make classic independent parameters – *should we want* to carry out an experiment. However, we don't have our own tool or support techniques ready to be hypothesized about, and we are not evaluating third-party tools. Similarly, it is premature to hypothesize what experts would do when tackling a specific task. Therefore, we opted for a more formative, *observational* study [24], the key elements of which included: (i) the *needs* (where are the gaps and how great they are), (ii) the *effectiveness* (where potential impact could be made), and (iii) the *evaluability* (what makes sense to evaluate in the future).

The assessment of the three categories was carried out using participants' responses to a range of questions that acted as 'measures' and included: task completion and difficulty, attitude expressions, facilitator's observations and participants' commentaries. Three basic instruments for data analysis were available to us: (i) participants' questionnaires, (ii) facilitators' notes, and (iii) audio/video records of the sessions using Camtasia. In terms of what has been observed and analyzed, we can refer to [11] and in particular, to the following of his proposed levels of analysis. He argues that one can assess: (i) user's general satisfaction with (or reaction to) the tool, (ii) effectiveness of the tool in achieving goals, (iii) behavioural efficiency. In our study, these broad categories take the form of questions and measures exploring *usability*, *effectiveness*, and *efficiency* categories, to which we add generic *functional assessment* category.

A questionnaire has been designed with 53 questions covering several categories of data. Questionnaire items were compiled so as to reflect issues that typically appear in the literature as correlated with enhancing or reducing usability, user satisfaction, etc. [25] Four questions inquired

about the participant (expertise, tool used,…) and eight about broad study settings, overall views on the tasks, most obvious gaps in tools, etc. Five questions focused on effectiveness of the tool, four on support efficiency, five inquired about specific engineering and design experiences, and seven considered broad usability aspects (e.g. help, interface clarity,…) Three questions asked participants to provide comments on support tools and other generic suggestions (incl. blue sky wishes of functionalities). The remaining (17) questions inquired about various functional requirements that were considered relevant to the NeOn vision by experts; incl. ontology re-use, visualization, contextualization, mapping, reasoning, etc.

The other two instruments (facilitators' notes and recordings) were similarly assigned analytic parameters from similar categories. We decided to use these additional instruments to gain more insight into findings that would originally come from the questionnaire. The rationale of this was to start with the participants' experiences and identify where major gaps appear or major differences in participants' opinions. This initial approach should allow us to narrow the scope of analysis, and essentially answer *what* is difficult with the existing tools and *how* difficult it is. As a follow up, facilitators' notes and recordings expressed in measures from the same categories may be used afterwards to attempt answering *why* the difficulties occurred. Another purpose of the recordings was to create a base of observations that could be later used in finer-grained or more narrowly focused studies and analyses (e.g. of a particular operation, such concept mapping). However, this latter opportunity has not formed part of the current study, and is not reported here.

Table 2 (overleaf) presents several measures that were selected as relevant to analyzing the observations. The measures are presented in the same categories as used in the questionnaire, together with their rationale and source.

The questions included some *closed* (e.g. evaluative) and some *open* (i.e. those asking participants to write down their opinions, suggestions or perceptions). The evaluative questions used a simplified Likert-style format [13] ranging from very useful, satisfactory, etc. (+1) to not very useful, missing, etc. (–1). The items in the questionnaire were both positively oriented and negatively oriented and they were interspersed so as to avoid the tendency of people to agree with statements rather than disagree [3]. Nevertheless, we will return to this tendency towards agreeing later – e.g. in the context of giving an overall neutral or positive mark to the tool's user friendliness and design quality, in spite of complaining consistently about sub-features.

For the responses to each question we calculate frequencies, their distribution (expressed in percentages from a population of those users who responded to a particular question), and symbolically, also a mean value, which is listed below to three decimal places together with a sign – 'minus' denoting a negative attitude. The purpose of the 'mean' is to act as an abbreviated way of describing prevailing attitudes, rather than having any statistical interpretation. In case of assessing the effect of the tool and/or effect of the expertise, we conduct a series of χ^2 tests with 2 degrees of freedom (2 groups of participants responding with 3 possible categories) on nominal data. Finally, each study session included a one-on-one debrief where participants could comment on their observations, opinions or suggestions – incl. comments on the design of the study, timing, tools selected, etc.

4. Observational study – findings

In this section we summarize key findings from the user study conducted as described in the previous section. The report on analysis is structured similarly to the way we introduced the measures. For each category of measure we first offer a general summary of observations across the whole studied population. This is followed by commenting on differences (if any) between two most common denominators of user performance in knowledge-intensive tasks – the choice of a tool and the expertise in the domain.

Table 2. Summary of categorized measures for observation analysis

Measure	Source	Notes
Category: effectiveness of tool support		
Degree of using different elements of the tool	A/V and facilitator	How many different facets / functionalities did participant invoke? Counting features like 'search', 'browse hierarchy', 'import file', 'import ontology', 'diagram summary', 'detailed graph', …
Degree of understanding tool messages, guidance	A/V and facilitator	How interactive and comprehensible the tools are; e.g. cases where people lost some work or 'got stuck' because they haven't noticed a message/notice
Depth of an operation within the tool	facilitator	How easy it was for the user to locate a particular menu option, button, widget; in qualitative terms 'immediate', 'quick', 'impossible without hint', 'facilitator intervened'
Consistency of user model and labels in the tool	facilitator	Does the operation reflect what user intended to do; intuitiveness of labels; qualitatively 'good match', 'good once got used to', 'confusing'
Consistency of an operation across the tool	facilitator	Could people transfer knowledge learned in one context to another (e.g. search classes → search properties)
Additional tools used	facilitator	Did the user miss something in his interaction and need additional tools?
Category: efficiency of carrying out task		
Time to achieve the task	A/V	How long did it take participant to conceptualize the task and carry it out; also expressed qualitatively (‘quick’,…)
Repetitions of an operation	A/V	How many times was a given operation used by the participant and was there any support from the tool to simplify subsequent uses
Operation realization timel	A/V	How long did it take to realize an operation once conceptualization has been concluded → could be addressed by training
Navigational interventions	A/V	How many interventions from the facilitator were needed to keep the task on the track → could be addressed by training
Category: quality of results		
Understanding vs. ‘coding’ ratio	A/V	Estimated ratio of conceptualization/understanding vs. coding/debugging times
Reaching correct result	facilitator	How close did people get to an ideal integrated ontology for each task; qualitative % estimate of solution quality
Number of mistakes/errors	A/V and facilitator	How many engineering problems did the participant generate regardless of rectifying them later after being pointed out by facilitator
Manner of error discovery	facilitator	How did the participant discover / recover from the error ('facilitator solved', 'facilitator intervened', 'self-reflection', 'tool hint/message')
Category: usability, design experience, other follow-ups from questionnaires		
Surprise vs. expectancy w.r.t. implicit features	quest	How did people use 'hidden' features such as drag&drop, double click to edit, right click to select restriction types,…
‘Being in control’	quest	Did the participant feel in control of the experiment or was s/he merely driven by the facilitator

4.1 Effectiveness-related findings

Here we are concerned with the measures looking at how effectively the tools were used in carrying out the tasks of the user study. Namely, we take in account such measures as complexity of getting acquainted with the tool, support for repetitive activities, overall behaviour of the tool, identification of and suggestions for removing major (subjectively perceived) obstacles, and ideas about improving the tool. We start with Table 3 and give some general observations.

4.1.1 Effectiveness-related findings – general

On average, our participants considered the process of getting around the tools they used somewhere between *not very easy* and *reasonable*. If we express 'not very easy' as '–1', 'reasonable' as '0' and 'easy' as '+1', the 'mean' attitude of the total population of **31 participants** was inclined more towards the negative end of attitudes, with 'mean' being **–0.065** (see Table 3). In other words, while the perception is not clearly negative, participants were not convinced by the tool's capability to reduce the complexity by introducing its functionality simply and effectively.

A stronger negative attitude emerged when participants were asked to rank how they felt about the support for frequently repeated operations (such as repeated definition modifications). In this case, most participants perceived the existing support more towards the 'not very good' value; the 'mean' is **–0.367**. In other words, the tool does not effectively make use of operations that could be potentially 'bundled' into macros or pre-defined sequences of steps. Taking into account participants' comments, this forced them to repeat such operations as searching not for the sake of actually finding a class satisfying their criterion, but as *an impromptu 'spell check'* – to get a correct name of a concept or property. This can be illustrated, for instance, by the following rationale for operation of "Auto-completing" appearing among the most frequent operations: "*searching of concept- and relation labels in order to […] avoid typos*".

Table 3. Selection of a few general observations across population

Measure/question	Avg. response	–1	0	+1	Total	Mean
process of getting around the tool and understand it	not very good / reasonable	23%	61%	16%	31	–0.065
support for frequently repeated operations	not very good / reasonable	43%	50%	7%	31	–0.367
overall behaviour of the tool	reasonable	7%	90%	3%	31	–0.032
effectiveness of dealing with task 1 (e.g. difficulties)	very easy	61%	26%	12%	31	–0.452
effectiveness of dealing with task 2 (e.g. difficulties)	very easy / moderate	27%	54%	19%	31	–0.161
effectiveness of dealing with task 3 (e.g. difficulties)	moderate / not very easy	11%	67%	22%	31	+0.194
help from the facilitator	reasonable / very useful	3%	45%	52%	31	+0.483

When asked about the overall effectiveness and behaviour of the tool, the opinion inclined towards reasonable (with 'mean' attitude **–0.032**). This means that participants judged the tools acceptable; however, an interesting pattern has emerged for those participants who considered that the tool provided them with a rather low support. While they would like to see more support and the existing support for their task is lower than needed, they tend to take a neutral view towards the tool. These two judgments seem to be inconsistent, because people see their experiences with such functionalities as documentation, customization, visualization, etc. clearly on the negative side; yet their overall judgment tends to be more moderate.

The operations or activities that were most frequently found among complaints about the effectiveness are led by the "import function". The "import function" covers the operation of opening and loading an ontology into the tool, and making it available in another ontology. We started with a belief that this operation would be among the most frequent ones in the networked ontology context, and this has been confirmed. Out of 31 participants, 12 mentioned this among most common operations. At the same time, 8 out those who reported this operation considered it unclear or as one source of their difficulties. Hence, the reason why this operation probably made it among the frequently used ones is likely to do with people's perception of how difficult it was. In the brief afterwards, most people concurred that opening and importing was so far almost a subconscious click rather than a fully-fledged meaningful operation.

In terms of effective support to remedy the difficulties with the tasks they were given, the emerging pattern across the whole population is that the perception of tool confusing the user seems to grow with the task complexity and open-endedness. For instance, on average participants judged task 1 more as 'very easy', but task 3 was weighed more to the 'difficult' side; as also shown by **'mean' attitudes –0.452 vs. +0.194**. Taking in account the attribution of support to either tool or facilitator, most of the help that reduced the task complexity was provided by the facilitator rather than tool. 'Mean' attitude towards **facilitator's assistance with difficulties** was inclined positively, towards very useful (with 'mean' **+0.483**), which contrasts with the aforementioned more negative inclination for the tool (–0.065 or not very useful/reasonable).

4.1.2 Effectiveness-related findings – effect of tool

The effect of the tool on the 'mean' attitudes from the previous section was **not very obvious**; it was barely observed in some questions. Table 4 shows some measures where different attitudes were observed. In cases of perceived task complexity, TopBraid users considered generally task 1 and task 3 **less complex than Protégé** users. However, the significance of the variance dependent on the tool has **not been confirmed** by χ^2 test at p=0.05. Only in the case of the third task did the χ^2 value approach significant levels. If we combine these (not significant) variances with what functionality different people reported, at least a part of the reduction in complexity seems to be due to the capability of TopBraid to work with several ontologies and to perform more intuitive search. In fact, intuitiveness of the user interface was a common comment with respect to improving Protégé and its user interactions. The other possible explanation of why one tool was perceived as complicated, which also appeared in people's comments, may be attributed to too many features available to the user at any one time – all being visualized on screen permanently. This was particularly frequent comment from the participants using Protégé.

Table 4. Comparison of attitudes between tools and expertise groups (TB: TopBraid, Pr: Protégé, Le: less experienced, Ex: expert) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
overall behaviour of the tool	tools	TB (0.0) vs. Pr (-0.143)	3.14	no
subjective complexity of task 1	tools	TB (-0.70) vs. Pr (-0.33)	2.17	no
subjective complexity of task 3	tools	TB (-0.10) vs. Pr (+0.33)	4.09	no
process of getting around the tool and understand it	experience	Le (+0.12) vs. Ex (-0.27)	3.02	no
role of tool in reducing complexity of task 2	experience	Le (-0.44) vs. Ex (+0.13)	9.71	yes
role of tool in reducing complexity of task 3	experience	Le (-0.06) vs. Ex (+0.47)	5.63	no

A similar minor variance is observable in participants' perception about the tool behaviours. Most TopBraid users considered the tool neutral ('mean' attitude 0.000), whereas Protégé users tended to opt for more negative attitudes ('mean' attitude –0.143). Yet, the χ^2 test at p=0.05 **did not confirm** the significance of this variance. Interestingly, similar 'means' appeared in response to the tool support for getting acquainted with its functionality. Despite TopBraid being a new offer on the market and totally unknown to any participants, people judged it easier to understand compared with Protégé (although many Protégé participants had used Protégé before). This finding seems to correlate with the slightly simpler interface of TopBraid and also the aforementioned 'multi-dimensionality' of the interface (e.g. working with multiple ontologies, clearer structuring of inference, etc.)

On other fronts, the **two tools were judged very similarly**; mostly participants tended towards negative attitudes in supporting often-repeated operations. Also the problems raised by users were similar – import function, confusion due to non-standard icons, dialogs or mouse click interactions.

4.1.3 Effectiveness-related findings – effect of expertise

If we take another common source of difference among tool users – their expertise with the procedures, languages and methodologies – the variances are also observable, and in one case even significant. Table 4 shows some measures where different attitudes were observed. In both more complex tasks – i.e. task 2 and task 3 – the participants with less expertise suffered more from the lack of tool support. The difference in their 'mean' attitudes towards the perceived role of the tool's user interface in reinforcing or reducing the impression of overall complexity of the operation was confirmed by the χ^2 test to be **statistically significant** at p=0.05 **for task 2**, but did not quite reach the significance threshold for task 3. In the following paragraph we present one possible explanation of the variance; however, we do not claim this is the only cause for our observations. Whether our hypothesis is correct or not might be subject of further research.

Perhaps as can be expected, less experienced participants seem to have considered tasks as more difficult compared to experts' easy or neutral judgment. One possible explanation of why task 2 differed from task 3 could be in its nature. Task 2 expected participants to amend restrictions in one ontology by choosing an appropriate concept from another ontology. In TopBraid, two ontologies could be browsed side-by-side; hence the user had a 'working ontology' (in our case Copyright) and a 'reference ontology' (in our case AKT Portal), where s/he could locate and explore various potential alternative conceptualization without losing the concept currently modified in the working ontology. This simple functionality of switching between two or more ontologies was more visible with more experienced users who quickly developed a routine of 'ontology flipping'

Otherwise, both groups gave similar marks with respect to the overall tool behaviour; mostly neutral to slightly negative. However, there were slightly more extreme reactions from the expert group in terms of expecting some support for frequent operations or in terms of acquainting themselves with the tool's functionality. However, in this case the χ^2 test **did not confirm** that this variance of the perception of how easy it was to acquaint oneself with the tool was significant at p=0.05). This observation is interesting because more experienced users reported they had used Protégé in the past on numerous occasions, yet they did not feel it was easy to get acquainted with the tool. In our opinion, the answers from the experienced group to this question could have been somewhat diluted by their previous experiences with possibly other versions of Protégé.

Taking qualitative comments into account, most expert participants were suggesting improvements on the level of using standard features and behaviours (e.g. delete or move functions), and also on the level of interaction modality. In other words, there was a frequent point about the tool's capability to support keyboard only rather than having to use mouse all the time. An extreme reaction quoted from one user was: "*Too much mouse interaction. […] Feels like programming with the mouse*". This particular quote comes from a participant using Protégé, but similar, although less strongly worded statements were noted also in the TopBraid group.

4.2 Efficiency-related findings

Here we look at such measures as how efficient people felt in different tasks, how they were assisted by the help system or tool tips, how the tools helped to navigate the ontologies or how easy it was to follow formalisms used in definitions. We start with Table 5 and give some general observations.

4.2.1 Efficiency-related findings – general

In terms of efficiency of tool support, most responses were about frequencies and qualitative suggestions; however, participants felt the tools did not provide them with enough useful information about ontologies. While there was a perception of tools giving a lot of information in

general, participants were more inclined towards the more negative end of the spectrum ('mean' attitude **–0.172**). Also, as could perhaps be expected, the *subjectively perceived* efficiency of the tool support was correlated with the *subjectively perceived* time participants felt they spent on the individual tasks. For instance, task 3 was slightly more complex than the previous two, but it essentially combined the lessons learned from these previous tasks. Nevertheless, people perceived themselves as less efficient in carrying out this task. This seems to be correlated with such issues as lack of documentation ('mean' attitude **–0.500**) and limited usefulness of the tool tips, hints etc. ('mean' attitude **–0.423**). Further correlations can be found when looking at usability aspects and presence or absence of several functional elements.

On the level of how often people perceived that they carried out repeated operations, clear winners were the modification of definitions (as could be expected from the tasks) and the repeated search for concepts – both reported by **17 out of 31 participants**. Both operations were closely followed by writing logical expressions for axioms (**13 out of 31**) and the import function (**12 out 31**).

Table 5. Selection of a few general observations across population

Measure/question	Avg. response	–1	0	+1	Total	Mean
providing sufficient information about ontologies	not very good	32%	55%	13%	29	–0.172
support provided by documentation, help	not very good	60%	40%	0%	16	–0.500
usefulness of the tool tips, hints, …	not very good	50%	46%	4%	27	–0.423
subjective time taken for task 1	very low	61%	26%	13%	31	–0.484
subjective time taken for task 2	moderate	25%	55%	20%	31	–0.065
subjective time taken for task 3	moderate/too long	6%	56%	38%	31	+0.300

From people who reported search as a frequent operation, 9 complained about having to repeat the operation due to losing the class they were working on or due to an unclear distinction between single and double mouse clicks. Also, from the same subset, 8 participants complained about inconsistencies in the dialogs, hints, user interfaces and responses – particularly related to searching in the full-text mode.

4.2.2 Efficiency-related findings – effect of tool

On the tool level, the efficiency of the two groups was approximately the same. Table 6 shows some measures where different attitudes were observed. There was a slightly **faster performance of TopBraid** users compared to Protégé, but only in a fairly trivial task 1; however, the χ^2 test **did not confirm** the variance was significant at p=0.05 (hence, TopBraid users were *possibly* faster than Protégé users). This might be attributed to a slightly simpler import function in TopBraid, which was basically the only major functional block expected to be used in task 1. Obviously, on the technical level, the import function was realized equally in both tools; it was the user interface and its presentation of the import function in a more visible manner that made the users perceive they were more efficient in this particular operation.

Other factors that might affect the efficiency of the tool support emerge e.g. in the perception of how the tool helped to handle ontology dependencies. Here Protégé users expected much more help than they got from the tool – the χ^2 test proved that the variance between the two tools **was significant at p=0.05**. Similarly, higher inefficiencies in Protégé were partly attributed to the limited visualization and ontology navigation facilities – the χ^2 test confirmed that the **variance between the two tools was significant at p=0.05**). Another variance, not significant this time however according to the χ^2 test, was observed with restricting ontology formats. If we look into qualitative remarks, many users were not happy with the abstract syntax of their axiom formulae, and in the

case of Protégé this was slightly exacerbated by the participants' inability to edit more complex restrictions (i.e. intersections or unions) in the same dialog as popped up for a simple axiom (i.e. cardinality or range).

One qualitative feature that was observed in both tools was related to the so-called depth of an operation in the user interface. Subjectively, **7 participants out of 31** felt that they had an explicit problem with finding an operation within a menu or workspace. Most frequent 'offenders' were the import function (which people expected to be behind the File → Import… menu option, but it wasn't) and in some cases search within one particular ontology (which was not the same as using the search dialog from Edit → Find… menu option). Other usability-related points that will be elaborated later include the positioning of frequent operations at the level of toolbars – from the five frequent operations (new concept, modify concept, search, import and assist with logic), only two were clearly available at the top – new (sub-)concept and partially search.

Table 6. Comparison of attitudes between tools and expertise groups (TB: TopBraid, Pr: Protégé, Le: less experienced, Ex: expert) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
subjective speed of completing task 1	tools	TB (-0.80) vs. Pr (-0.33)	2.94	no
help with handling ontology dependencies	tools	TB (0.0) vs. Pr (-0.37)	7.65	yes
useful visualization & ontology navigation facilities	tools	TB (-0.33) vs. Pr (-0.63)	6.00	yes
handling ontology syntax / abstract syntax	tools	TB (+0.40) vs. Pr (-0.07)	2.33	no
ease/speed of carrying out mappings	experience	Le (-0.21) vs. Ex (+0.27)	9.75	yes
level of visualization and navigation support	experience	Le (-0.69) vs. Ex (-0.40)	2.40	no
specifics of ontology representation languages, abstract syntax, etc.	experience	Le (-0.22) vs. Ex (+0.23)	3.64	no
management of versions for engineered ontologies	experience	Le (+1.0) vs. Ex (–0.50)	3.37	no

4.2.3 Efficiency-related findings – effect of expertise

The level of expertise seemed to have minimal effect on the differences in perception of efficiency. Table 6 shows some measures where different attitudes were observed. Both groups concurred that while a lot of information was made available to them about ontologies or concepts, these were not very useful. On the contrary, what was missing was a clearer access to such 'hidden' functions as defining equivalencies or importing an ontology. In terms of functionalities, it is not very surprising that non-experts found themselves less efficient without some simple visualization and navigation support compared with experts (however, the χ^2 test **did not confirm** the variance between the two groups was statistically significant at p=0.05).

Similarly, as mentioned in the previous section, non-experts found themselves more handicapped by having to deal with a very specific notation of the ontology representation language and the underlying logical formalism. While experts were more at ease with the existing formats, non-experts considered support for this aspect not very useful. While the confidence of this variance was higher, the χ^2 test **did not confirm** the significance at p=0.05 in this case either.

The only question where **significantly different opinions were confirmed** by the χ^2 test at p=0.05, concerned the ease with which the users subjectively carried out the mappings between concepts. In this case, less experienced users found it (surprisingly) very easy and experts reasonable or very easy. At this stage we may only hypothesize that this observation may be partially explained by more experienced users expecting some kind of automated support for

mapping discovery, whereas less experienced users simply considered mappings as associations of the concept labels, which after initial learning quickly developed into a routine comprising: search for concept using partial string match → confirm the selection → finalize the definition. Similarly, with the expectations of version management, the experts thought this feature was missing; there was a difference in opinions as mentioned in Table 6, but the χ^2 test **did not confirm** any statistical significance of this variance.

Qualitatively, a frequent complaint from experts related to the need to repeat many operations using mouse rather than standard keyboard keys (incl. such basic ones as 'delete'). The overwhelming demand was for complying with common and established metaphors of user interaction. A quote from one participant sums this potential source contributing to inefficiency as follows: "*More standard compliance and consistency. The search works differently at different places of the editor. And the usual keyboard commands as expected from the OS don't always work, like clicking on a class and pressing the del key.*" This observation came predominantly from expert users who were using the Protégé environment.

4.3 Design and user experience related findings

With respect to user experiences with the tool's particular functionalities and also in general, several sets of questions were asked. Since the number of the experiential findings is rather large, we opted to report on the users' positions towards certain functional requirements that were identified by a group of NeOn researchers earlier in the project in a separate section. This split between experiential responses related to the user study and the attitudes towards new and proposed functionalities enables us to analyze the *experiences with existing tools* separately from more visionary *attitudes towards future developments*. We start with Table 7 and give some general observations regarding the user experiences with the existing functionalities.

4.3.1 User experiences findings – general

Two key aspects were evaluated with respect to user experiences – (i) *usability* of the tool (which included accessibility, usefulness and so on) and (ii) more general *satisfaction* with the tool. The latter included comments regarding user interface intuitiveness, acceptability, customization, and so on. Compared with categories measuring effectiveness and efficiency of the tool support, the responses in this category were more negative; i.e. participants considered the existing support as "very low" or "not very useful", rather than "adequate", "very good" or "very useful".

For instance, almost invariably, participants were rather dissatisfied with the role that tool documentation, tool tips and various other tool-initiated hints played in easing them into the tool. A 'mean' attitude towards the usability of the tool's help system was inclined towards negative ('mean' **–0.500**); usefulness of tool tips and hints was marginally better, but still negative, with 'mean' **–0.423**. Similarly, participants considered the support for customization of the tool – i.e. either its user interface or functionality – mostly negatively. This measure received a 'mean' attitude **–0.400**. A common justification for this low score among participants was (among others) the lack of opportunity to automate some actions, lack of support for keyboard-centric interaction, lack of support for more visual interactions. As can be seen from these examples, the reasons were quite diverse, and to some extent depended on the user's expertise with ontological engineering. We shall return to these tool- and experience-specific differences later.

One emerging general feedback on the tools' usability was the fact that too many actions and options were typically available at any given point during the integration tasks. This comment was actually two-pronged. On the one hand it referred to the amount of information displayed and consequently, the number of widgets and window segments needed to accommodate this information. A typical representative of this type of usability shortcoming is the (permanent)

presence of all properties on screen. While users were willing to accept the constant presence of this part of the developed ontology, they considered it too rigid. For instance, one could not easily see only those properties relevant to a given concept – i.e. in a slot-like manner. This confused quite a few participants; in fact 32% claimed that unclear indication of inheritance and selection was a major drawback of the tool, and a further 14% reported confusion with being unable to find all uses of a term (e.g. property or concept label) in a particular ontology.

An interesting clash between functionalities that people thought would be useful and actually reported as problematic was observed e.g. with respect to such features as drag&drop or multiple ontologies. In the former case, drag&drop functionality was highlighted as a generic requirement – both by experts involved in the pilot and initial discussion of the project team. However, only 11% of participants actually reported that they missed drag&drop features. Even if these features were pointed out to them by the facilitators, participants usually 'ignored' them. On the contrary, the capabilities to work with multiple ontologies simultaneously were largely expected, yet this feature went largely unnoticed in the case of TopBraid and generated only a small number of comments in the case of Protégé. However, the tab-based interaction with multiple ontologies was subconsciously used by all TopBraid users, almost without noticing it.

Table 7. Selection of a few general observations across population

Measure/question	Avg. response	–1	0	+1	Total	Mean
usability of tool's help system	not very good / reasonable	60%	40%	0%	16	–0.500
usefulness of the tooltips, hints, …	not very good / reasonable	50%	46%	4%	27	–0.423
support for customization of the tool, its GUI or functionality	not very good / reasonable	48%	44%	8%	25	–0.400
usefulness of handling ontology dependencies	not very good / reasonable	31%	66%	3%	27	–0.259
visualization of imports, constraints & dependencies	not very good	58%	39%	3%	28	–0.536
support for [partial] ontology import	not very good	62%	14%	4%	29	–0.739
useful tool interventions in establishing mapping	not very good / reasonable	48%	52%	0%	26	–0.480

Another general point affecting the usability of the user interfaces concerned the non-standard and variable use of icons, buttons and hints. As the following quote suggests, in places there were too many choices how to perform an operation without any hint of a difference between them: "*Too much actions possible at any time. Too much possibilities of doing one and the same thing. An editor should be more intuitive to use. It is not always clear why thing had to be done in the way they were done*". What was even more confusing was the fact that these different paths of carrying out a particular action (e.g. search) exhibited slightly different behaviours, used slightly different and inconsistent dialogs, and produced different results.

Other comments related to usability are summarized below (frequencies of individual observations are reported later in section 5.1 to 5.3):

- Clearer error messages and hints (e.g. red boundary around an incorrect axiom was mostly missed);
- Possibility to add axioms without having to map it to the [tools'] interface (while most participants had little trouble formulating and conceptualizing a complex axiom such as "intersectionOf", translation of the conceptual statement into the tool's half-natural and half-description logic – DL – formalism was a frequent obstacle that in most cases required direct intervention of the facilitator – for experts it was too different from standard DL, for less experienced it was still too much DL);

- Common or established user interface conventions (e.g. icons for browsing looked different from in most other GUIs, the search icon was not obvious, menu labels were misleading – already mentioned earlier in case of importing, usual shortcuts from an OS non-functional); also 32% of (Protégé) participants considered the capability to load multiple ontologies into one instance of the tool as core for working on ontology networking and integration;
- Intuitiveness of the tool (esp. actions like finding an operation in the menu or on screen, finding the concept in the ontology and flagging it so that it does not disappear, full-text vs. random text search capabilities);
- Support for editing, defining or amending terms (incl. concepts, instances, properties and axioms) … a frequent source of confusion was a different treatment of apparently similar logical notions – e.g. while "subClassOf" was visible at the top level of the editor, "equivalentTo" was hidden, or a similar issue with adding a new (simple) restriction and adding a statement (i.e. simple or complex restriction).

Furthermore, in general, responses with respect to the usability of the existing tools in many key functional areas were all inclined towards the negative attitude. For instance, handling interdependencies among ontologies received 'mean' attitude **–0.259**; visualization of imports, constraints and other dependencies received 'mean' **–0.536**; as expected, support for partial import – e.g. of one concept and its branch received only **–0.739**, and tool's intervention in establishing mapping was also considered of limited use, with a 'mean' **–0.480**. These negative marks can be contrasted with participants' responses to some possible functional extensions, which are going to be discussed in section 4.4.

Another interesting clash between participants' reports concerns the fact that there was too much information available on screen at any one time, in opposition with the fact that participants thought the tool did not provide them with sufficient information about the edited ontology. The reason for this seems to be largely twofold: (i) the lack of capability to customize the GUI and thus see the right thing at the right time and at the right level of detail, and (ii) the lack of support for more interactive querying, exploration and filtering (e.g. in showing only properties relevant to a selected concept).

4.3.2 User experiences findings – effect of tool

As mentioned above, the lack of satisfaction with the help system, tool tips and messages was observed across the whole population – the tool made only a minor difference. However, a minor difference of opinion was observed on the level of overall satisfaction with the tools, their overall design and intuitiveness, where it was more likely that people complained about Protégé than TopBraid (but the χ^2 test **did not confirm significance** at p=0.05). In spite of TopBraid being a recent product with participants having no prior experience using it, overall satisfaction with this environment was also slightly more positive (yet **not significantly** according to χ^2 test). However, it is visible from the results that users of both tools had a somewhat more lenient approach to judging their subjective satisfaction with the tools and user interfaces. In other words, responses to specific queries were rather negative (between –0.500 and -0.100), yet overall experiences oscillate between –0.111 and +0.100. Does this mean people water down their expectations based on past experiences? Table 8 shows some measures where different attitudes were observed.

Table 8. Comparison of attitudes between tools and expertise groups (TB: TopBraid, Pr: Protégé, Be: less experienced, Ex: expert) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
level of overall satisfaction with the tools	tools	TB (+0.10) vs. Pr (-0.19)	2.67	no
overall satisfaction with tool's environment	tools	TB (+0.10) vs. Pr (-0.24)	3.14	no

useful support for handling dependencies among ontologies	tools	TB (0.0) vs. Pr (-0.37)	7.65	yes
level of visualization and navigation support	tools	TB (-0.33) vs. Pr (-0.63)	6.00	yes
ease of carrying out concept mapping	tools	TB (+0.50) vs. Pr (+0.10)	5.85	no
usefulness of the tooltips, hints, …	experience	Be (-0.25) vs. Ex (-0.57)	2.45	no
availability of customization of the tool, its GUI or functionality	experience	Be (-0.64) vs. Ex (-0.21)	7.90	yes
effort to get acquainted with the tool	experience	Be (-0.27) vs. Ex (+0.12)	3.02	no
overall satisfaction with tool functionality	experience	Be (-0.33) vs. Ex (0.0)	3.10	no
support for reasoning and inferences	experience	Be (0.0) vs. Ex (+0.07)	3.19	no
support for multiple ontology representation formats	experience	Be (-0.22) vs. Ex (+0.23)	3.64	no

Another feature that was perceived as subjectively better supported in TopBraid was the support for handling interdependencies among ontologies, which included e.g. automated loading of an imported ontology if this was known in the workspace. This was confirmed by χ^2 test at p=0.05; the variance between the usefulness of the two tools in this aspect was statistically significant. From other responses, we wish to highlight slightly better performance of TopBraid in terms of visualizing interdependencies among ontologies and mapping the concepts. In case of visualization, the TopBraid interface was perceived less crowded and clearer, so it was slightly easier to see what is imported, what is a defined class, etc. (χ^2 test **confirmed that this variance was significant** at p=0.05). Possibly due to the same factor of a less crowded interface and easier navigation among multiple ontologies, also the concept mapping was seen as better supported in TopBraid (but the χ^2 test **did not fully reach threshold of significance** at p=0.05).

Other differences of opinion were perceived differently by the users of the two tools, but the variances were not proven to be significantly tool-specific.

4.3.3 User experiences findings – effect of expertise

On the level of experience, we highlight several responses – some of which are to be expected. Table 8 shows some measures where different attitudes were observed. For instance, while help and documentation were considered inadequate by both types of user, less experienced participants were more likely to miss the tool tips and the interactive tool messages, but the χ^2 test **did not confirm significance** of this variance. On the other hand, less experienced users felt that the tools were too rigid for them and expected more possibilities to customize them. The χ^2 test found **this variance was statistically significant** at p=0.05. This observation has been frequently coupled with the aforementioned opinion about the complexity of GUIs and presence of vastly greater amount of information on screen than what they could have processed. Not surprisingly, this led to a reduction of the overall satisfaction among the less experienced users; but according to the χ^2 test the variance in the overall satisfaction was not significantly dependent on the expertise.

Another observation, where experience seems to weigh strongly on less experienced users, related to intuitiveness of the tool GUI. While there was little difference between the two tools in their support for easing participants into their use interfaces, the less experienced participants felt it required much more effort from them to get acquainted with the tool (but according to χ^2 test this **was not a significant variance** at p=0.05). Probably the key factor affecting this question were the aforementioned non-standard icons, lack of standard keyboard shortcuts, ambiguous operation labels, and also an overall depth of certain operations in the tool.

From other functionalities, more experienced users expected to find some advanced functionalities, such as version management but these were not implemented in the tested setups of the tools (again, the variance between the user groups was not statistically significant). Here less

experienced users were probably not clear in how versioning might actually work in this particular case. Interestingly, less experienced users were more likely to expect some support for multiple ontology representation and storage formats but this has not been provided in the tested tools (but the χ^2 test **did not confirm significance** of this variance). This rather surprising finding might be partially due to the inability of the tools to move from an OWL- and DL-based syntax to the frame syntax, which might be more easily comprehensible in some specific circumstances (such as modification of ranges in task 2 of our user study).

4.4 Functionality-related findings

In this section we summarize participants' expectations with respect to various functionalities, which were associated with the individual measures reported on in section 4.3. The motivation of this needs gathering was to assess how satisfied the users are with the status quo of the existing tools, and how would they relate to a selection of potential amendments or improvements. The suggested improvements were compiled from the opinions of domain experts on the possible innovations in NeOn. Due to the early stage of the project, most of these opinions were presented to the participants only on a conceptual level rather than any mock-up or prototype. This may possibly lead to different participants imagining completely different things under the same statement – such as "*How useful would you consider techniques for visualizing and navigating multiple branches of ontologies simultaneously?*" Different associations may obviously lead to ascribing different values to a particular functionality.

For these reasons, we believe a fairer picture could be obtained by using a two-pronged approach to analysing these future-centric attitudes towards supporting a range of user needs. First, in this section we look at different functionalities and techniques, and merely highlight the attitudes of the participants towards them. Then, in the next section, we start exploring linkages between the subjectively perceived satisfaction (or lack of it), objectively observed problems and attitudes to certain modifications. Table 9 shows a selection of perceived issues and possible remedies.

As this user study has been conducted with an aim to contribute towards better understanding of user needs in the scenarios where they work with networked ontologies over their lifecycle, we selected a few phases of this lifecycle, where the improvements might have the greatest impact – according to the experts. These phases include: ontology re-use, mapping, contextual representation and versioning. Three functional areas were also considered: visualization of, reasoning with and storage of ontologies.

For example, in the category of *ontology re-use* it was observed that participants perceived the support they received from the tool as insufficiently adequate. In general, the 'mean' value of the attitude towards the existing support for re-use of whole ontologies was **–0.097**, but in the case of supporting partial re-use of ontology sub-sets, the mark fell to **–0.739**. These negative attitudes could be contrasted with a fairly positive response towards suggesting minor 'cosmetic' changes making the re-use process look simpler. For instance, on average participants considered the facility of merely flagging the chunks of ontologies they are working with as "not present but very useful" – in total **56%** supported this idea. Similarly, another minor modification was about hiding relevant/irrelevant chunks of ontologies during re-use also elicited the same 'mean' value of the attitude – but this time only **36%** of participants supported this functionality unconditionally.

In terms of mapping support, participants were of opinion that mappings were fairly easy (with 'mean' attitude **+0.233**), but when it came to more complicated mappings that were formulated in terms of a particular context or contextual boundaries, these were considered less positively ('mean' was found around **–0.065**). Nevertheless, the support from the tools to manage any of these mappings – whether simple or complex – received a response towards a negative attitude, with 'mean response' being **–0.480**. In connection with the mappings people most positively related

to the facility of querying ontology for a particular set of items instead of having to browse, navigate or merely search the labels. This correlates with the tendency of people to formulate a correct statement, issue or sub-task, but failing to express it directly in DL-like formalism of restriction definition. People wanted to see what would be an effect of a particular conceptualization before they committed to its formal representation – the most often cited example was from task 2: "*recipients are generic agents but not geographic entities*", which some people conceptualized as disjunction of sub-classes to "generic agent", whereas others opted for conjunction of "generic agent" with the negation. The difference became apparent when they tried to explain their choice – a simple query mechanism to test possible interpretations might simplify the mapping process.

Table 9. Summary of gaps vs. support for partial fixes

Question (existing feature or 'proposed fix')	Avg. marks	–1	0	+1	Total
Existing support for ontology re-use	–0.097 (not very good / reasonable)	26%	58%	16%	31
Support for partial re-use of ontologies	–0.739 (not very good)	62%	14%	4%	29
→ flag chunks of ontologies or concept worked with	+0.674 (*would be* useful)	20%	24%	56%	25
→ hide selected (irrelevant?) parts of ontologies	+0.465 (*would be* reasonable / useful)	25%	38%	38%	24
Ease of mappings in tasks	+0.233 (reasonable / easy)	19%	25%	42%	31
Existing support for mappings, esp. with contextual boundaries	–0.065 (not very good / reasonable)	19%	68%	13%	31
Management and assistance with any mappings	–0.480 (not very good / reasonable)	48%	52%	0%	26
→ propose mappings & ensure their consistency	+0.433 (*would be* reasonable/useful)	3%	50%	47%	30
→ compose testing queries to try out consequences of mappings	+0.045 (*would be* reasonable)	9%	77%	14%	23
Existing support for versioning, parallel versions/alternatives	–0.200 (not very good)	50%	20%	30%	11
Existing visualizing capabilities & their adaptation	–0.536 (not very good)	57%	39%	4%	28
→ mechanism to propagate changes between alternative versions	+0.519 (*would be* reasonable / useful)	7%	33%	60%	28
→ compare/visualize different interpretations/versions	+0.700 (*would be* useful)	6%	17%	77%	30
→ performing operations in graphical/textual mode	+0.414 (*would be* reasonable / useful)	7%	45%	48%	29
→ visualize also on the level of ontologies (not just concepts)	+0.357 (*would be* reasonable / useful)	7%	50%	43%	28

Next, we consider the *ontology versioning* phase. As could be expected, the existing support was on average perceived as "not very useful" ('mean' attitude **–0.200**). Yet, the participants concurred that they would benefit from a mechanism that would propagate a change throughout an ontology to maintain the working version as close to consistency as possible ('mean' attitude was inclined towards usefulness, i.e. **+0.519**). Even more users considered a mechanism comparing two versions of an ontology possibly showing two different interpretations of a particular conceptual issue as useful, i.e. the 'mean' attitude was **+0.700**.

This finding seems to lend more support to the functionality mentioned in the previous paragraph where the users tentatively welcomed functionality supporting the exploration of possible consequences of a particular commitment. Only, in this case, we went a bit further and proposed a way to operationalize such exploration of consequences. Perhaps unsurprisingly, the functionality enabling the users to compare the ontologies or the consequences of certain conceptual commitments was welcomed more strongly by less experienced participants. But at χ^2=3.93 the variance was not over the significance threshold. A similar distribution of responses was found in the context of techniques supporting propagation of mappings and changes through ontology

versions, but as before, χ^2=3.77 meant the preference of the less experienced users was not significant.

Even more specifically, when we used the experts' idea of using CVS[9] as a possible implementation metaphor, the users considered this would definitely offer some benefits; on average, this idea was met with a "not present but very useful" attitude ('mean' attitude **+0.357**). CVS as a very specific metaphor grounded in the software development circles attracted stronger support from expert users, but the variance was not significant.

We already mentioned above some potential functionalities that fall into the *visualization* category – e.g. hiding of certain concepts or 'diffing' two versions. In general, the visualization capabilities of the two used tools were considered as "not very useful" (with 'mean' attitude **–0.536**). This was probably one of the areas where participants had the strongest feelings and reactions in the session debriefs. As mentioned in the earlier section under different categories, the re-emerging theme was non-standard user interaction metaphor and confusing icons, messages and menu options. Ontology hierarchies in particular received their above-average share of negative comments (quote from one user "*These hierarchies are killing me*" sums up the attitude of at least 32% of participants who admitted an issue with visual interaction).

From the functional amendments proposed by experts, people had positive attitudes to both – the capability to perform operations in the graphical mode (e.g. creation of a sub-class) and the capability to see several relevant branches with terms and concepts worked on next to each other, in parallel. The former idea of (also) a graphical execution of certain tasks received on average a more positive attitude (**+0.414**), and, interestingly, there was not a significant variance between the experts and less experienced users on this *optional* functionality. The latter proposal of not only opening and working with multiple ontologies but being able to see them next to each other received on average a more positive attitude (**+0.357**), again with expertise not affecting the attitude to such an option. In order to avoid misunderstanding with an earlier observation that experts wanted more keyboard-based interaction (i.e. less mouse-based and visual), we shall emphasize that the capability to carry out an operation visually was considered a useful *option* but not a *default* modality.

5. Qualitative analysis and exploration of findings

The purpose of this section is to consider relationships and correlations between different observations made during the user study. While the content of this section cannot be considered *explanatory* in terms of clearly identifying cause of marks mentioned in the previous sections, it can be *exploratory*. In other words, more research would be needed to prove direct causality between the issues and observation discussed below; nevertheless the emerging trends provide a useful input to developing tools and techniques addressing various shortcomings as perceived by the users.

5.1 User interaction and navigational issues

We mentioned earlier that two of the most frequent operations as perceived by the participants were *search for a term* and the actual *modification of a definition* (both operations were explicitly reported by 61% of users). These numbers reported by the participants themselves are supported by the facilitators' observations that show that 80% of users needed at least one hint in relation to

[9] CVS – concurrent versioning system, an infrastructure and protocol enabling groups of developers to work on their code in a collaborative and concurrent manner. A typical feature of CVS is the capability to "diff"; i.e. to see the changes between two nodes/files in the CVS tree.

search dialogs, icons or search format. From the recordings it was also visible that almost every user (90%) encountered at some point in their interaction an issue with inconsistent operations or confusing labels, icons or suchlike. The reason these observations are mentioned in the context of *searching* is that the search widgets were the most likely offender. For instance, the case of participants working with a wrong concept was positively correlated with the observations of being confused in the menu labels or in the (results of) search dialogs. The correlation coefficient for this relationship was **ρ=0.962**, which meant that **ρ^2=92.5%** of variation is shared.

The next functionality that caused a lot of trouble related to *importing an ontology*, which was explicitly highlighted by 43% of participants as one of the most frequently used. A major source of this high visibility of an otherwise routine operation was the absence of the actual menu label launching an import dialog from the menu (in both tools). This confusion on the level of menu labels may have led to a subjective perception that one needs to go through this operation many times. As in the case of searching, the majority of users had at least one problem with either differentiating between “open ontology” and “import ontology”, or between “import file into workspace” and “import ontology” (80% of users according to facilitators’ notes). Only a part of the problem can be attributed to the icons and menus – the correlation between problems with import and menu/label inconsistency was positive (**ρ=0.612**) but not very high. In other words, only **ρ^2=37.5%** of change in import troubles seems to be due to labels.

We may hypothesize that another comparable part could be *due to semantic closeness* of the three operations: “open ontology” vs. “import ontology” vs. “import ontology file into workspace”. The rationale for this hypothesis comes from the requests of some participants for standard interaction metaphors, phrases, icons and keyboard shortcuts. It is not unreasonable to assume that people bring certain routines from other software development or office production environments into ontological engineering. Hence, in line with an established usability guideline, it would be advisable to adapt the tools to the users’ routines and habits rather than bring in new labels, icons and/or operations.

Another usability and user-tool interaction concern we want to explore more in depth relates to the initiative of the tools in our study. Typically, tool may take initiative in the interaction in numerous ways – the most usual being error messages, tool tips, hints and auto-corrections. In our case, 60% of users were observed at least once as not noticing an incompatibility between their formal definition and conceptualization. If such an error occurred, it was almost invariably within the axiom definitions. Two sub-issues contributed to this lack of awareness: (i) auto-correction after too shallow a syntactic check, and (ii) translation or composition of a restriction in semi-logical language.

In the former case, the observation of not noticing some incompatibility or a clear error in a definition and the failure of the tool to alert the user about carrying out syntactic checks (e.g. on brackets) was positively correlated at **ρ=0.632**; i.e. **ρ^2=40%** of variation in the lack of user’s awareness of the mistakes could be attributed to the tools. Once users realized or were notified by the facilitator to re-check their input, the auto-corrections had to be changed manually. This not only leads to possible frustration of the users, but also significantly affects the efficiency of the tool in the engineering process.

In Table 10 (below) we summarize the most frequently observed mistakes, problems and confusing situations from the user interaction perspective. In other words, these observations were made by facilitators in relation to navigation through a specific GUI – which contrasts them with issues related to either preparatory activities (e.g. opening or importing files) or the actual structural amendments (e.g. axiom definitions). The data in Table 10 shows a category of observations together with frequency, number of affected users and examples. The difference between frequencies and affected users is in the fact that sometimes problems of the same kind were noted multiple times for the same user; whereas each user was only counted once in “% affected”.

Table 10. Observations of issues related to navigation

Observation	Frequency	% affected	Examples
Dialogs, buttons,… (confusion, inconsistency)	43x	89.1%	Buttons/icons after axioms misleading; Single/double clicks to select, edit, etc.
Searching for the class (partial text search on labels)	25x	64.3%	Expected vs. real results – e.g. label starts with X different from label contains X; namespaces included in search/match
Functionality not noticed or ignored (drag&drop, full-text search, alphabetic view,…)	26x	60.7%	Am I in the edit mode?; Where is it alerting me about error?;
Other issues with labels, menus,…	27x	53.6%	Classify classes, hierarchy or individuals?; checkboxes in import dialog; red highlight of axioms
Locating import, search, edit, etc. buttons, widgets, options	27x	39.2%	Where is “import” option?; Where is “equivalence” definition?; How to add “intersectionOf” axiom?
Locating the class in a hierarchy	20x	39.2%	Native classes 'hidden' under imported ones.; Losing working class while previewing another one.
Working with incorrect concept (concept edited without explicitly selecting anything)	13x	32.1%	Are you changing/editing concept in the right ontology? Is it the right concept? (see also single vs. double clicks)
Visualization issues	9x	17.8%	RDF graphs only, not really ontologies.; Seeing equivalences ‘next to each other’
Incomplete dialog parameters → unexpected behaviour	3x	7.1%	Load into unspecified workspace

In addition to the aforementioned purely navigational issues, the translation of a conceptual model of a restriction into DL-style formalism was a separate source of problems. While only 18% of users explicitly demanded an easier definition of axioms among their necessary customizations, 70% were observed to stumble during the axiom definition. Much has been written about the tradeoffs between simpler visual notations and more expressive textual notations, particularly in the field of programming and logic education. For a survey of various studies of the effects of notation see [2]. These authors highlight the fact that use of logical vs. visual notation is often expertise-dependent, problem and context dependent and also dependent on individual preferences. Nevertheless, they note that “every notation makes some information accessible at the expense of obscuring other information”. The implications from this general knowledge in the visual programming community, combined with the responses of our participants, would suggest the need to consider *multiple ways* for defining and editing axioms. While DL and text-based interaction may be a preferred choice of our experts, this is unlikely to be the case with experts and modellers in other domains.

To put the 70% of users observed to have a problem with axiom definition in perspective, consider that only in 10% of sessions did we observe a problem with adding or re-defining a sub-class. However, there is one potentially common theme here. In the observed cases where people made a mistake with class definition (e.g. they attempted what we labelled ‘double definition’), an issue was with translating the verbal statement into a visual and logical notation. While all users formulated the operation in terms of “making concept X a sub-class of Y” (e.g. in task 1), some attempted to implement it by *opening concept Y* and editing it so as to add an existing concept X to the existing class Y. This was not possible for existing concepts, only new sub-classes could be created for any class Y. If one wanted to say something concerning concept X, it was necessary to edit X and state “subClassOf (X, Y)” there. So, in general this is a similar problem as with the visual vs. textual languages, only this time the clash is between intuitive verbal statement and its explicit, formal translation.

5.2 Level of prior expertise issues

Much has been written and said about making ontology engineering tools usable and accessible to the users beyond the description logic or knowledge modelling communities. However, far less has been done than discussed, as some examples from the user study show. Obviously, it is important to acknowledge the work done with respect to making the tools such as OWL editors more robust and more standardized in terms of their interoperability. Nevertheless, both tools that participated in our study are simply too deeply embedded in and dependent on one specific knowledge representation paradigm – namely, description logic (DL). While formal properties make DL a highly suitable language to reason with and share encoded knowledge, even among ontology engineers, DL is not necessarily the preferred "medium for thinking".

This was apparent fairly frequently in the situations where our participants wanted to make some amendments to follow the instructions given to them in the task (e.g. amend definition of a particular property). As we mentioned several times above, the gap between conceptual formulation of the activity and formal realization of the same in the tool was often too big. There seems to be an inverse correlation between the users' request for functionalities that address the problem from the conceptual viewpoint and the frequency of observed difficulties when carrying out translations or compositions of logical formulae (ρ**= –0.793**). The coefficient of determination corresponding to the fraction of shared variance between the two measures was ρ^2**=62.9%**, which is fairly high. Almost two thirds of problems with logical translations might be due to using an unnatural way of expressing things.

Another pointer to the issues with gaps between the language of users and language of tools could be a fairly high number of users affected by a surprise in the form of a syntactically incorrect statement. In 64.3% of sessions we observed at least one issue due to syntax (e.g. of a complex axiom) that people needed to be alerted to by the facilitators (also shown in Table 11). It is true that typically after one such warning the users learned to re-check each of their modifications, but this has an impact on the trust given to the tool. For instance, people had a tendency to doubt results of other operations (e.g. search or classification) if this was slightly different from what they expected. Such lack of trust is then rather problematic because it puts the tool solely in the position of a plain editor, which leads to further reduction in the tool's initiative and assistance.

Table 11. Observations of issues related to structural differences between the user and tool

Observation	Frequency	% affected	Examples
Syntactic check (brackets, logic,…) → user not alerted or not noticing	21x	64.3%	Buttons/icons after axioms misleading; Single/double clicks to select, edit, etc.
Testing (inference, meaning, interpretation,…)	26x	64.3%	Which inference is the right one?; How to check if ontology says the same as verbal task?
Carry out (translate, formulate, compose) logical operation (e.g. equivalence, inheritance)	37x	60.7%	How to start complex axiom?; Stepwise definition?; Is "same" equal to "equivalentTo"?
Incompatibilities between task and formal definition, errors in definitions	36x	53.6%	Property type/range elsewhere than cardinality; Wrong translation of "X but not Z"
(Re-)definition of a class A in different branch ("double definition")	18x	53.6%	Defining new sub-class for Y and labelling it X vs. re-defining existing X as "subClassOf" Y
Issues with namespaces	8x	28.6%	Role of namespace in general; role in search

5.3 Desired vs. nice vs. actually used features

Software engineering is familiar with the issue of developing a software product incrementally; i.e. by adding to its original blueprint. While such an evolutionary development process is inevitable in the current dynamic business environment, much of the effort in improving a product actually results into so-called "feature creep". As described by formal books (e.g. [9]) or informal bloggers (e.g. [28]), with each new feature, the requirements change, complexity increases, and as far as the users are concerned, they are now being exposed to something new, which is not surprising. However, tool designers sometimes overlook the point whether the user actually might or might not need or want those new features.

Feature creep is a very common feature of many software engineering projects, whereby the developed systems are almost constantly updated by introducing new and new features. First, this often leads to delays in the development process, but that is not our primary concern in our user study. Second, the new features are often introduced in response to a request from a specific group of users-specialists, and included into a general functionality of the tool. While those features are useful for a sub-set of users, it is often the case that the majority of 'ordinary' users (i.e. those not belonging to the specialist sub-group) find those additions unnecessary, confusing and increasing the learning curve. It is this second issue with the feature creep that is particularly evident in our user experience study.

While users perceived the tools as "reasonable" and "adequate", they were simultaneously complaining about too high a complexity of the user interaction. At any time during the interaction, there were usually several features available to the user – all displayed at the same, top level. Both tested tools had a tendency to show most of the possibly relevant chunks of information on screen at all times. A typical example was a generic list of *all* properties defined in the ontology or imported from another ontology. A slightly less typical, but also problematic example was the presence of different types of inference or reasoning. For instance the appropriate menu distinguished between checking consistency, classifying classes and individuals – despite the fact that the ontologies contained largely structures, so it made little sense to classify individuals.

Yet another category of choices that seems to hint at "feature creep" is the aforementioned open/import ontology function with up to ten choices. These choices included things like open "RDF or OWL from DB" as opposed to "…from a local file system" or "…from a remote location". While it is true that *some* users surely benefit from such a wide choice of widgets, window panes, menu options or loading mechanisms, there is no evidence that *all* such features are used with the same frequency, which would justify their equally prominent position in the user interface. Obviously, a large portion of this "feature creep" is due to effort to maintain some backward compatibility and to cater for as wide a user audience as possible.

The extensive use of features becomes perhaps more difficult to understand if we realize that both logicians and product designers (essentially the two groups of users interacting with tools like this) claim to subscribe to the centuries-old rule known as Ockham's razor[10]. In plain language the core principle of this rule is to resist using more (tools, concept, objects, features, etc.) if one suffices with less. That this maxim was indeed followed (at least to some extent) by the designers of the two tested tools can be seen in an extensive use of plug-in components, which enable users to add custom tabs (e.g. for visualization or knowledge acquisition). Nevertheless, it seems to be the case that the level of granularity at which the tool is customizable has been set fairly high. In other words, one can indeed add new visualization techniques into Protégé or use different (DIG-compliant) reasoning tools, but one cannot *modify the components of user interaction*.

Table 12. Tool customization to address "feature creep"

[10] William of Ockham (1280-1349) – a philosopher, logician, and one of the first 'modern' representatives of a conceptual stance towards knowledge.

Customization need	% affected	Examples
Different representation levels	30%	Natural language vs. semi-natural translation vs. DL formalism ; also relates to request for naive vs. expert mode
Control the number of panes open	20%	Seeing all facets of developed ontology at all times seems to lead to 'blindness' (people don't realize that e.g. all properties are already displayed)
Different views on the ontology	70%	Seeing all related properties to the selected class in a slot-like manner, all uses of a term in ontology, all inherited restrictions

A classic example of this gap came out in our study particularly from those users who were more familiar with knowledge modelling paradigms other than DL-based OWL. If we add together the number of users who indicated in the debriefing that they would like to see the following customization opportunities, we achieve almost perfect coverage of the entire user population we worked with (i.e. beginners, moderately experienced users and experts).

Furthermore, from the initial discussions with the use case partners there is another potentially dangerous misunderstanding that may well make the whole issue of feature creep even bigger. Where many users talk about *using* context and *adapting* ontologies to a particular context, in many case they mean *viewing* different parts of ontologies in a manner that is suitable to their knowledge, experience, etc. The reason why this particular case is mentioned alongside the feature creep is that it nicely illustrates the gap between the features the users *think may be* useful or needed, and features they *actually need* to carry out their tasks.

Thus drag&drop appears frequently among the desirable (high-level) features, yet its implementation on the level of classes and restrictions does not seem to be particularly often used by the users. Similarly, rich means for ontology visualization are equally desirable, yet not on the level of triples and triple graphs. Where we are aiming with this discussion is the recognition that it is insufficient to ask users to label an abstract feature they desire, and then to express some priority or importance of it. One needs more knowledge of when a particular feature is desired, for what purposes and how it is conceptualized by the user. For example, in our study we observed participants attempting drag&drop feature but on the level they were familiar from standard operating systems – i.e. to open a new file in an application. In other words, the desire can still be labelled as "drag&drop" but it seems more naturally used at the level of *external* interfaces (from OS to the tool) than on the level of *internal* interaction modalities (within the tool or between its panes).

For these reasons we tried to stay clear of making too strong statements about user desires and wishes in a form like "X% of users definitely wants to see functionality Y in their tool." Without at least a mock-up interface implementing any particular functionality, such statements would have rather restricted validity as well as re-usability. It seems to be more valuable to highlight the difficulties people are currently facing in the ontology integration tasks together with their attitudes to various abstract improvements or modifications. However, to tell whether the actual customization technique (for instance) is indeed preferred by the real users, we suggest conducting targeted studies working with mock-up prototypes investigating different means of implementation. In other words, in this study we are focusing on *what* are the problems, explore their potential *sources*, but leave statements about *how best* to tackle them or *how useful* they really are for the subsequent studies.

To conclude this section, we shall note that in order to address the functional shortcomings as summarized in section 4.4, it seems to be a more promising approach to go for a more *modular and flexible* solution. That is – a tool or toolkit that can be quickly and easily customized to accommodate to the user's preferences or experiences. An important side-effect of the discussion in this section seems to be the recognition that we need to decouple *requirements on the actual ontologies* from the ontology *design requirements*, and also from the requirements on tools and

user interactions. To illustrate this we refer to a trivial requirement of using OWL as a coding language for ontologies – this, however, does not mean that the user necessarily needs to express all restrictions or relations in the native OWL formalism.

For example, as illustrated earlier in the sub-class vs. super-class definition causing a conceptual confusion, we saw that people could easily draw two nodes and connect them to express inheritance. Formally, this inheritance is always stored with the 'child' node, which needs to be linked to its 'parent' by means of using the 'subClassOf' relation. However, this formal outcome can be achieved by (at least) two distinct user interactions: (i) selecting the 'child' and defining/choosing a 'parent' for it – i.e. isomorphic to the formalism; or (ii) selecting the 'parent' object and choosing an existing class for it to act as a 'child' – i.e. interacting in an opposite way to what formal representation demands.

5.4 Visually rich vs. textual interaction

Since ontology visualization is such a prominent topic in the research agendas we dedicate a separate section to this particular user need. In our user study we observed two rather contradictory requirements. On one hand side there were users who wanted to have more opportunities to interact visually (e.g. to define inheritances or restrictions in a graphical mode). On the other hand, there was a substantial demand for text-based interaction that would be structured so as to assist with more complex, multi-step activities.

Our study cannot give a definite answer to the question which of the two interaction modalities would be more likely to help the users. Indeed it might not even be possible to simplify the complex subject of personal preferences to the choice between graphics and text. The two visual modalities could be also expressed in terms of tool modalities – users preferring graphical interaction were typically happy with using mouse; users demanding textual interaction wanted more opportunities to apply the keyboard. Whichever of the two interaction modalities the user of the existing tools prefers, our study shows that none is capable of giving adequate support in ontology integration tasks. This is particularly disappointing for the visual metaphor, as there is a substantial amount of research and development focusing on graphics and visualization (e.g. [6, 27]).

The primary shortcoming that we would associate with rather negative attitudes of our user population would be an attempt to use a one-size-fits-all visualization for several different purposes. In other words, in the tested tools, visual metaphor is typically activated by switching to a dedicated "visualization" tab (one can have several different tabs active but typically not simultaneously). This is, in our opinion, key mistake and key reason of negative attitudes. Visual metaphor would only work if it is addressing a specific problem (or perhaps a set of problems). Thus, rather than switching to a graphical interface, it is more appropriate to solve or respond to a particular problem using graphical and visual means. This embedding of the visual metaphor in the tackled problem could make the appreciation of its role in the system more positive. In particular, visual tabs cease to be 'yet another complicated graph' and have a chance of becoming tools and means for appropriate situations.

This position is not as theoretical as it may seem. There is already ongoing development of many specialized visualization techniques that are much more problem-centric than medium-centric. For instance, fish-eye projection is another metaphor familiar to all interested in photography, but also to car drivers. This technique is used, for instance, in [12] to fit large semantic networks into limited-size display and in parallel to make them user-navigable. As with Jambalaya, the fish-eye metaphor enables customizable navigation; it uses different properties of the objects in a knowledge base to create clusters of different granularity and of different semantic meaning.

In a similar tune, another more recent example of actually addressing specific user needs is the 'crop circles' metaphor [18]. As with fish-eye, this metaphor also shows some implicit topography in

an overview mode. The idea draws on what mathematicians call Venn diagrams, which for the ordinary user means overlapping circles. The difference between visualizing generic graphs and task-specific graphs is in their utility. The former methods show the underlying technology, such as triple-based RDF, in a slightly simpler form. The latter show the implicit or explicit structure and topography of a particular problem (e.g. mutual imports of the ontologies or concept neighbourhoods) only drawing upon semantic commitments.

Another view on this discussion of visual vs. textual interaction is offered by [22], who argues that knowledge spaces on the Semantic Web are in general n-dimensional. Therefore, choosing two-dimensional graphs to show all the remaining 'n-2' dimensions can be very confusing to the user (whether they naturally prefer interacting with visual information or not). Instead of boxing the user into a pre-defined two-dimensional space, for example, a technique called mSpace [22], but also an earlier technique called Lois [4], exploits a simple idea of forming the view stepwise. This step by step visualization is familiar under the name of faceted browsing and is realized by choosing the order in which the different semantic 'dimensions' are explored.

The point we want to make in this more abstract discussion is not to focus on a vaguely defined middle ground where the tool supports a little bit of visual interaction and a little bit of textual (or other forms of) interaction. This generality does not seem to improve users' attitudes to the tool friendliness. It might be therefore more worthwhile to look at both, slightly better defined extreme modalities, and attempt to identify the situations and contexts, where they can be applied to the maximum benefit for the user.

5.5 Other aspects of observational user study

In section 3.3 we introduced the reported user study in terms of user observation for the purposes of requirements gathering and identification of gaps. We highlighted three factors of such studies: (i) the *needs* (where are the gaps and how great they are), (ii) the *effectiveness* (where potential impact could be made), and (iii) the *evaluability* (what makes sense to evaluate in the future). So far we devoted substantial space to the core category (i) – the user needs, gaps and perceptions. Let us consider briefly the other two aspects.

Which of the issues we discussed earlier are more important? Is it possible to create some form of ranking mechanism that would enable NeOn designers to prioritize their work, or at least to assess the potential impact of developing a particular technique to address a given need/gap? When the identification of needs and gaps is rather open-ended, their ranking is even more difficult. Nevertheless, one possible way of how we can approximate the potential impact (and thus effectiveness) of a particular functional feature, technique or methodological advance is to take in account the frequencies of different interactive operations people typically carry out. Before providing further details, we have to note that the action typicality is to some extent dependent on the tasks carried out, so we cannot claim that *any ontology engineering* task would encounter the same operations. Nonetheless, it is possible to estimate that in principle *an ontology integration* task may exhibit approximately similar composition of activities and operations.

Table 13 contains a list of most frequently executed activities and operations as reported by the participants. The numbers shown in the table were already used in the previous sections, but are adjusted here to reflect the responses only from the participants who gave a non-empty answer (4 users failed to respond to the question). As the labels of some operations may seem similar, we also include an example of what people used to associate with those labels (acquired from the debriefing sessions with the participants that were carried out after the observational sessions and questionnaires).

Let us next look at opportunities for making an impact related to the operations mentioned in Table 13. With respect to modifying ontological definitions, we would most likely associate the issues

discussed above – e.g. user interaction customization in section 5.3 or visual interaction in section 5.4. The measure of success to be aimed at in this category would be (i) to accelerate the operation, and (ii) to enable achieving the same effect through *multiple modalities* (or simply interactive strategies). Compare this to our earlier discussion on drag&drop functionality or sub- vs. super-class conceptualization.

Table 13. Most frequent actions – where to achieve quick impact?

Operation	% reported	Examples
Modify ontological definition	71%	Re-define inheritance for a concept
Search for a concept, property,…	71%	Full text, partial text search for item satisfying given pattern
Define, introduce new statement (e.g. concept, restriction,…)	54%	Conceptualize a new restriction or logical statement
Create or edit logical expression	54%	Translate conceptualization into appropriate logical formalism
Import, network ontology/file	50%	Re-use other ontologies during engineering
Locate an item in the ontology	46%	Select the working object, highlight the working object
Browse hierarchy, lists of concepts	29%	Using hierarchical trees, alphabetical lists,…
Expand existing definition	21%	Add a statement to an existing restriction
Undo action	4%	

With respect to searching, the usual details associated with this operation were about making search more flexible and dynamic. That may mean a capability to search for terms in different areas of an ontological definition (e.g. concept label vs. description). Another frequently mentioned point related to inadequacy of exact match between string terms – with exact string matching there was limited incentive for people to rely on search. They either need to know what the class name could be or simply go and browse the hierarchy or alphabetical list. Search without some more flexibility is consequently perceived as a hindering rather than assistive feature, and contributes to reducing people's satisfaction. Possible ways to make an impact on this front may include support for *similarity-based search* (incl. simple synonyms, typos or language variations of "colour vs. colour" style), *descriptive search* (incl. searching for items satisfying certain inheritance conditions or associated with certain properties), or *constrained – 'contextualized' – search* (incl. searching for an item solely in the scope of a particular branch, particular neighbourhood, etc.)

With respect to conceptualizing and defining more complex logical statements and assertions, the key aspect were impact may be made is to guide users through the conceptualization process. This could be achieved by using the notion of 'wizards', which are not completely novel e.g. in the context of Protégé. WonderWeb research project has contributed some targeted wizards that simplify the definitions of complex ontological statements such as partition classes, exhaustive enumerations and similarly [21]. These interfaces have effect not only on the subjectively perceived simplicity of an operation but also reduce the chance to commit syntactic errors in the definition. However, in line with section 5.4 it might not be desirable to keep adding new and new wizards to the user interface to facilitate each and every specific transaction. A potential solution might be in using *dynamically configurable* and *adaptive* wizards, whereby the users were allowed to (i) activate a particular form of support and (ii) fine-tune its performance.

Moreover, such adaptation might be associated with so-called external environment aspects – e.g. user's membership of particular groups and communities, or topical and procedural context in which a particular difficulty is observed. Thus, no assistance may be provided for a trivial definition of a simple cardinality restriction, but when the user indicates they are after conceptualizing intersections, complements or suchlike, the mechanism may be 'contextually' activated.

On the level of support import into the ontology engineering environment, the primary aim is to adopt as far as possible the standard assumptions and established practices of the external environment. This means using terminology that is close to operating systems and common applications (see the earlier discussion of notions like “open vs. import” or “import file vs. import ontology”). Obviously, a share of blame belongs to the cavalier use of these notions within the Semantic Web research community – e.g. in many situations, researchers work with an *implicit equivalence* between a physical file and ontology as a conceptual model. While this might be clear to the experts, it may throw less experienced users off the track.

In addition to this terminological mismatch, another scope for making an impact is to enable alternative ways of networking the ontologies. The <imports/> statement is currently carried out on the level of entire ontologies, and by definition makes the imported terms an integral part of the importing ontology. From the feedback of users it seems worthwhile to either extend the notion of networking to differentiate conceptually different ways of linking ontologies (e.g. reference of terms, extension of terms, full adoption of terms,…) Another aspect already recognized in NeOn is related to the partial import – or probably better, to the creation of ontological networks from ontology sub-sets or modules rather than entire ontologies.

With regard to navigating through ontological structures, one inevitably gets to the point having an appropriate visualization framework that supports metaphors particular users are familiar with. As we noted in the earlier sections (mainly 5.3 and 5.4), one direction would be to decouple the representational from the conceptual visualizations – i.e. an RDF graph based on triples is a representational visualization, whereas ontology visualization methods are more concerned with the actual model rather than its formal language. Another direction where navigation through ontological structures may be made more efficient for the users is to consider on-demand visualizations of specific aspects, relations or functionalities, and associate methods with these needs. In other words, a ‘visualization tab’ in the tool GUI is too generic and broad. Perhaps strategies already explored by Jambalaya might be a way forward – rather than visualizing whole ontology, it depicts a specific neighbourhood or a specific view of a neighbourhood. Obviously, the notions repeated above regarding better customization and variability to suit people’s cognitive make-up are also applicable in the context of addressing this particular category.

6. Discussion and Conclusions

This report presents a summary of findings, but also lessons learnt, from the observational user study we conducted in order to improve our understanding of the user needs and the gaps in tool support for the tasks involving ontology integration and networking. Since majority of the report was concerned with the actual findings and their analysis, a part of this section is devoted to the reflection on the actual user study.

6.1 Summary of findings

Technology (such as OWL), no matter how good it is, does not guarantee that the application for its development would support users in the right tasks or that the user needs in performing tasks are taken on board. At a certain stage, each successful tool must balance the technology with user experience and functional features [15]. This paper explored *some issues* with ontology engineering tools (particularly those working with OWL) that reduce the appeal and adoption of otherwise successful (OWL) technology by the practitioners.

As shown above, although the tools made a great progress since the evaluations reported in section 2, issues with user interaction remain remarkably resilient. The effort was spent to make

the formalisms more expressive and robust, yet they are not any easier to use, unless one is proficient in the low-level languages and frameworks (incl. DL in general and OWL's DL syntax in particular). Existing tools provide little help with the user-centric tasks – a classic example is visualization: There are many visualization techniques; most of them are variations of the same, low-level metaphor of a graph. And they are often too generic to be useful in the *users'* problems (e.g. seeing ontology dependencies or term occurrences in an ontology).

In sections 4 through 5 we highlighted a few gaps with the existing support for engineering ontologies in the scenario developed by NeOn. That is in the scenario when ontology engineers are developing complex ontologies by reuse, i.e., by integrating existing semantic resources. Along the gaps we also acquired people's opinions with respect to how would certain features and functional primitives help to improve their experiences. It is fair to say that some of these 'wishes' actually do exist already – to a lesser or greater extent. For instance, Prompt is a plug-in intending to address some of the issues with version management or comparing different versions of ontologies [16, 17].

Similarly, Jambalaya is capable of providing different views on the ontological neighbourhoods [6]. Jambalaya's visualization is still based on a graph metaphor, but already allows more customization of what can be visually depicted. Particularly its FilmStrip metaphor shows an interesting compromise between data overviews and its specific context. Yet, due to realizing this idea through showing the relevant information as nodes, the outcome is full of boxes and overlapping edges. These often achieve the opposite of a positive user experience because they may easily obscure much of the underlying semantic structure.

Nevertheless, these two examples show that there are specialist components capable of tackling some of the gaps we highlighted. One issue with these systems is (i) their relative complexity and (ii) their relative closeness; i.e. difficulties to further customize or adapt to a preferred style or form. Obviously, certain flexibility is already embedded in the environments such as current versions of Protégé. However, this customization seems to be done on a rather abstract level. In other words, one can customize the Protégé environment as a whole – for instance, by adding and activating a particular plug-in with the desired functionality. Yet, one cannot easily modify the predefined behaviours of these plug-ins; that is, not without going into their code, which is probably not what most users would do. The granularity at which tools are customizable is set fairly high. For instance, one can add new visualization tabs into Protégé or use different (DIG-compliant[11]) reasoning tool, but one cannot *modify or filter the components of user interaction*.

Correlations were also observed between e.g. incorrect logical conceptualization and confusion caused by ambiguous labels or dialogs. Other correlations were between problems with importing an ontology and the absence or semantic ambiguity of appropriate widgets in the workspace, and between difficulties with definitions and the failure of tools to alert users about automatic syntactic checks (e.g. on brackets). The translation of a conceptual model of a restriction into DL-style formalism was a separate issue: 70% were observed to stumble during such definitions. From our data, we suggest considering *multiple ways* for defining and editing axioms (to a limited extent this partly exists in Protégé). Its 'property' vs. 'logic' viewpoints partly removes the issue with some restrictions, but in principle, this is only a minor visual arrangement. Underneath, there is still a DL-style formalism, which may not be easily comprehensible by all designers. Anyway, DL may be good for reasoning, but it is by no means the preferred "medium for thinking" (even among ontology designers).

[11] DIG-compliance means that a reasoning tool implements a common application programming interface (API) enabling the communication between the description logic reasoners and third-party tools such as ontology editors. The DIG interface has been designed by the Description Logic Interest Group (see http://dl.kr.org/dig) and its specification defines a concept language and a minimal set of operations that must be supported by a reasoning tool.

Another issue following from this is the gap between the language of users (i.e. language for presentation and expression of knowledge) and language of tools (i.e. language of representation and storage). A high number of users were surprised by syntactically incorrect statements. In 64.3% of sessions at least one issue due to syntax (e.g. of complex restrictions) was observed. Because of these minor issues they had to be alerted to by the facilitator, people tended to doubt the results of other operations (e.g. search or classification) if these differed from what they expected. Lack of trust is generally problematic because it puts the tool solely in the role of a plain editor, which further reduces the tool's initiative. The problem becomes more visible if the lack of trust seems to be more due to insufficient knowledge about what is happening inside the tools and how a particular conclusion (e.g. a correction of an axiom) has been reached. In an attempt to restore 'user trust', some tools (e.g. SWOOP) started recently moving towards trying to justify their results [10].

An interesting anecdotal comment came up in trying to reconcile the different perceptions of the users. Although the comment is applicable only to the two tools we tested, there is one lesson that can be generalized. Namely, the users tend to complain about Protégé because they have used it in so many situations and found so many things that they did not like. Thus, tool familiarity is a double-edged sword – it improves the user interaction (esp. its efficiency and effectiveness), but may also cause a projection of past negative experiences. On the other hand, when participants started to use TopBraid they may have projected their Protégé experiences onto this new tool. In other words, the initial perception or expectation is an important factor of the overall satisfaction. If users think that what they are going to find will be performing similarly to what they are used to, and on the contrary find that the tool or product is better than expected, their satisfaction improves. The key potential lesson for NeOn is to take advantage of this low expectancy, and make sure that in the areas where most friction occurs the performance is better. In this way, the tool may quickly gain some ground, despite its novelty and lack of familiarity.

6.2 Implications for NeOn

Clearly, there is some way to go to provide the level of support needed by 'normal' users engineering OWL ontologies. Our analysis highlighted some shortcomings, esp. the flexibility and adaptability of user interfaces and lifting the formal abstractions. With this study, we obtained a benchmark, which we plan to use to assess the support provided by our own future tools in 18-24 months. Obviously, it may be useful to include other OWL engineering tools (e.g. SWOOP or ontoStudio) as well. However, let us first discuss more direct implications and advice that can be drawn for the work carried out in the context of the NeOn project.

Among the gaps, we highlighted the issue with user interaction, the use of languages that are not familiar and natural for the users, and several issues with non-standard user interfaces. We would like to suggest that the aspect of easing the user into the tool, and possibly to any non-standard or proprietary features, should be taken more seriously. This does not mean yet another tutorial on the web page, but perhaps a series of short mini-tutorials well-focused on a particular action or topic (e.g. ontology import). The users tend to perform the basic operations in a mechanical way, using the same or similar steps. Providing a "Show me how…" mini-tutorial with these most common (but not necessarily trivial) operations may help users to adapt themselves to the tool. Alternatively, this could be seen as users adopting certain peculiar ways of doing things that may be imposed by the tool architecture. Hence, our first point concerns the often repeated position about a tool being able to adapt to the user. While this may be the ultimate aim with the NeOn toolkit, it may be more effective to assist the user in getting used to the tool by a series of mini-tutorials.

Having claimed there is a need for such mini-tutorials, another compulsory implication for NeOn from the study is a fairly rigorous following of standard practices. These include common icons,

dialogs, error messages, hints, positions, etc. of various interactive components. While the graphical part of the user interface is often the best place where a new entrant can differentiate its product from the established players, this shallow novelty often works for a limited time. We would suggest that our developers of modules and techniques follow established standards from the existing operating systems or common office applications, and seek an opportunity to distinguish the NeOn toolkit in the quality of its user support, the flexibility or the customization.

From this recommendation follows another opinion that the designers of NeOn tools and techniques could take on board: namely, to take into account the "languages of users", not only the objective technological requirements on any particular function (e.g. ontology selection, alignment or mapping). While the language of users may be completely different from the representational languages used in the tools, and as such, very hard to acquire and understand, there is an option to cater for it anyway, for instance, by simply allowing to remap the low-level operations and user interaction features using more user-specific labels. One possible way of achieving this is the support for the functionality of macros – similar to those users may be familiar with in Lisp or in the Office tools. The "Lisp-style macro" metaphor is perhaps even more interesting because it allows the user to create a macro using visual and interactive means. For instance, the user can show how an action is done officially using standard steps, record this interaction, and then replay it upon request on a different object, a different part of screen, etc.

Another important lesson that was proven by this study is the fact that the development of ontologies by their integration and creation of ontology networks is not necessarily the same as the establishment of mappings on the level of ontological concepts. While conceptual alignment may play an important role in the NeOn scenario of developing ontologies by re-use and networking, there are multiple interpretations of what integration may mean. Hence, it may be useful to annotate developed plug-ins, modules or methods in terms how they contribute to the broad process of ontology integration. This task-centric description of methods (but also modules and plug-ins) may then, in principle, assist the users in choosing the right tool for the right task. This recommendation is in line with the previous point on the adaptation, which, to some extent, assumes that people find and add new modules to the toolkit to suit their needs. To do this one has to be able to search and locate appropriate downloadable components.

6.3 Discussion of the methodology

Observational user studies have a goal to acquire requirements that would improve product development. The issue with such studies is that they work with existing tools (or interfaces), but aim to project the findings to the future design and development of possibly new tools. As analyzed by [29], so-called controlled evaluations tend to be useful in scenarios where the aim is to assess the overall effectiveness of a *completed system*. On the contrary, formative evaluations are more associated with incremental development of existing systems. In our case, we conducted an observational study aimed at user needs that could be applied to the future NeOn toolkit. Rather than reporting on a specific hypothesis, the observational study, such as ours, brings to the front deficiencies, gaps and various other shortcomings, which might aid product re-design or improvement.

What characterizes both observational and formative studies is often *a narrow focus* on a particular area, domain or task. This is also the key issue with such studies – if the task is chosen too narrowly, its capacity to generalize may be very limited. Similarly, if the task is chosen too broadly, it may bring in too many pathways into the user interaction, which makes the subsequent analysis and identification of implications nearly impossible. In our case, the determinant of the study breadth was the composition of the actual ontology integration tasks.

Although effort has been put into designing a task that would be non-trivial, yet executable within a short time span, there were still difficulties in conveying the substance of some task activities to the

participants. There were two reasons for this. First, the tasks were to some extent *ill-defined* [26]. This was deliberate because if we gave step-by-step instructions for what users should do, this would significantly constrain the range of potentially useful findings. Too detailed a task description may, for instance, have prevented us from observing surprising effects or strategies people deploy to tackle the problem. Thus, one aim for future research would certainly be a more principled design of an evaluation task. In particular, the Semantic Web community seems to be rather unsure of how to assess the utility of various semantic and semantic web technologies.

Looking at usual measures of precision or recall is inappropriate as soon as we leave the domain of information extraction or knowledge discovery. Attitudinal evaluations express people's satisfaction, happiness or acceptance of certain features, yet as we noticed in some of our questions, the formulation of attitude (e.g. towards the tool as a whole) may not really reflect the acceptance of the tool's components. It is particularly difficult to link satisfaction with the single, controlled feature. Typically, in the observational studies, there is a mesh of features that affect usability, user acceptance or satisfaction. Hence, much work needs to be done with respect to designing a task that would truly assess the *semantic benefit* of a tool rather than the qualities of its design team.

Another reason for some participants having difficulties with a task may be attributed to the language – this time we mean natural language of human participants. Most of the participants in the study were not native English speakers and also, the core ontology they were working with was designed in a non-English environment. Yet, the tasks were formulated and illustrated in English. Although all our users were able to communicate in academic-level English, this did not mean that they interpreted the tasks in the same way. This was particularly visible in observing some users struggling with subtle differences in the semantics of such notions as "legal" and "legal agent", or such tasks as "re-defining property X in the context of class A so that it is broader than it was in case of class B". Particularly, the second kind of statements was considered by many participants too vague, yet their suggestion to simply say "give property X range B that is a super-class to A" was not acceptable from the point of our objectives.

6.4 Conclusions

In this report we discussed the findings from the observational user study performed with the aim to improve our understanding of how users carry out tasks using networks of ontologies. Compared with previous studies, our study involved not only 'power' users and also users who have experience in the studied domain but cannot be considered knowledge engineering experts or power users.

As the previous sections showed, there is clearly some way to go to provide the level of support expected by typical users who are creating and re-using ontologies in the OWL formalism. As we mentioned in the motivation to the study, it is increasingly likely that the ontologies of the future would be designed predominantly by this user group – a typical feature of which might be the lack of formal training in standard, description logic centred approaches to knowledge modelling. The established models of this user group are sometimes at odds with the models driving the actual engineering tools, which might lead to reduced effectiveness of the tool, its efficiency in supporting a particular operation, or in overall user experience with the human-computer interaction.

Our analysis highlighted some shortcomings, especially the flexibility and adaptability of user interfaces. This includes lifting of the formal abstractions that are presently used in the existing tools and are essentially re-used low-level representational formalisms. We observed limited role, and indeed intention, of the existing engineering environments in actively contributing to the human-computer interaction. Where such active role was observed, it was in most cases very coarse, without much 'intelligent adaptation' (a typical example of this situation is the syntactic check and automated bracket inclusion, which instead of helping often caused more confusion).

With this study, we obtained a benchmark, which we plan to use to assess the support provided by the developed NeOn tools and techniques. These follow-up evaluations will play the role of formative user studies informing the actual tool development – the formative nature of the future studies will come more clearly to the fore once early prototype NeOn tools reach sufficient level of maturity. The first formative studies on the NeOn tools are expected to take place in 18-24 months. Obviously, it may be equally useful to expand the scope of the study and include other OWL engineering tools as well (in particular, SWOOP from MindSwap lab and OntoStudio from ontoprise, GmbH which were not included in the current batch).

Finally, we appreciate the contribution from Holger Lewen (UKARL), Olaf Görlitz (UKO-LD) and Carlos Buil Aranda (ISOCO) for facilitating the studies in their organizations, also the contribution from Diana Maynard (USFD), Enrico Motta (OU) and anonymous reviewers from the ISWC 2006 workshop on "OWL: Experiences & Directions" who kindly reviewed earlier drafts of this work.

References

[1] ISO 13407, *User-centred design process for interactive systems*. 1998.
[2] Blackwell, A.F., Whitley, K.N., Good, J.*, et al.*, *Cognitive Factors in Programming with Diagrams.* Artificial Intelligence Review, 2001. **15**(1-2): p. 95-114.
[3] Colman, A.M., *A Dictionary of Psychology*. 2001, Oxford: Oxford University press. 864.
[4] Domingue, J. and Motta, E., *Planet-Onto: From News Publishing to Integrated Knowledge Management Support.* IEEE Intelligent Systems, 2000. **15**(3): p. 26-32.
[5] Duineveld, A.J., Stoter, R., Weiden, M.R.*, et al.*, *WonderTools? A comparative study of ontological engineering tools.* Intl. J. of Human-Computer Studies, 2000. **52**(6):1111-1133.
[6] Ernst, N.A., Storey, M.A., Allen, P.*, et al. Addressing cognitive issues in knowledge engineering with Jambalaya*. In *Knowledge Capture Conference (K-Cap)*. 2003. Florida, US.
[7] Fensel, D. and Gomez-Perez, A., *A survey on ontology tools*. 2002, OntoWeb Project.
[8] Gruber, T.R., *Towards principles for the design of ontologies used for knowledge sharing.* Intl. Journal of Human-Computer Studies, 1993. **43**(5/6): p. 907-928.
[9] Jackson, M., *Problem Frames: Analyzing and structuring software development problems*. 1 ed. 2001, London, UK: Addison-Wesley. 390p.
[10] Kalyanpur, A., Parsia, B., Sirin, E.*, et al.*, *Debugging Unsatisfiable Classes in OWL Ontologies.* Journal of Web Semantics, 2005. **3**(4).
[11] Kirkpatrick, D.L., *Evaluating Training Programs: the Four Levels*. 1994, San Francisco: Berrett-Koehler Publishers. 289.
[12] Komzak, J. and Slavik, P. *Scaleable GIS Data Transmission and Visualisation*. In *Intl. Conf. on Information Visualization (IV)*. 2003. UK.
[13] Likert, R., *A technique for the measurement of attitudes.* Archives of Psychology, 1932. **140**: p. 5-55.
[14] Motta, E., *Reusable Components for Knowledge Modelling*. Frontiers in AI and Applications. 1997, The Netherlands: IOS Press.
[15] Norman, D., *The Invisible Computer*. 1998, Cambridge, MA: MIT Press.
[16] Noy, N.F. and Musen, M.A. *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. In *Proc. of the 17th National Conference on Artificial Intelligence (AAAI)*. 2000. Texas, US.
[17] Noy, N.F. and Musen, M.A., *The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping.* International Journal of Human-Computer Studies, 2003. **59**(6): p. 983-1024.
[18] Parsia, B., Wang, T., and Golbeck, J. *Visualizing Web Ontologies with CropCircles*. In *Proc. of the ISWC 2005 Workshop on End User Semantic Web Interaction*. 2005. Ireland.

[19] Pinto, S., Peralta, N., and Mamede, N.J. *Using Protégé-2000 in Reuse Processes*. In *Evaluation of ontology-based tools (EON)*. 2002. p. 15-25.
[20] Rector, A.L., Zanstra, P.E., Solomon, W.D.*, et al.*, *Reconciling users' needs and formal requirements: issues in developing a reusable ontology for medicine.* IEEE Transactions on Information Technology in Biomedicine, 1998. **2**(4): p. 229 - 242.
[21] Rector, A.L., Drummond, N., Horridge, M.*, et al. Designing User interfaces to Minimise Common Errors in Ontology Development*. In *UK eScience All Hands Meeting*. 2004.
[22] Schraefel, M.C., Carr, L., De Roure, D.*, et al.*, *You've Got Hypertext.* Journal of Digital Information, 2004. **5**(1): p. 253.
[23] Schreiber, A.T., Wielinga, B.J., and Breuker, J.A., *KADS: A Principled Approach to Knowledge-Based System Development*. Knowledge-Based Systems Book Series. Vol. 11. 1993, London: Academic Press.
[24] Scriven, M., *Beyond Formative and Summative Evaluation*, In *Evaluation and Education: A Quarter Century*, McLaughlin, M.W. and Phillips, D.C., Editors. 1991, University of Chicago Press: Chicago. p. 19-64.
[25] Shneiderman, B. and Plaisant, C., *Designing the User Interface: Strategies for effective human-computer interaction*. 4 ed. 2004: Addison-Wesley. 672.
[26] Simon, H.A., *The structure of ill-structured problems.* Artificial Intelligence, 1973. **4**: p.181-201.
[27] Storey, M.A., Lintern, R., Ernst, N.A.*, et al. Visualization and Protégé*. In *7th International Protégé Conference*. 2004. Maryland, US.
[28] SWiK.net, *Dialogue-Driven Development*. 2006.
[29] Twidale, M., *Redressing the balance: the advantages of informal evaluation techniques for Intelligent Learning Environments*. 1993, University of Lancaster.

Appendix A. User instructions and brief

Welcome to the NeOn WP4 user study!

Thank you for your participation in this study. You will carry out the tasks on your own in the presence of a facilitator. The facilitator would be giving hints, asking questions why a particular decision has been made, etc. The study will comprise a series of tasks, to be completed one at a time; i.e. once the facilitator, who would be observing your interactions, agrees that you have concluded one task, you will be presented another one.

The study is about the integration of definitions coming from several non-trivial ontologies. You will be asked to carry out tasks that involve simple ontology engineering activities, but you would but restricted to a small number of tools you can use.

During the session you have the following tools at your disposal:

- Protégé 3.2 with FaCT++[12] classifier/reasoner [OntoStudio 1.4.0/TopBraid Composer]
- Text editor and viewer (e.g. Notepad, Word,…)
- Web browser
- Notepad, pens and pencils

You are not asked to follow any particular ontology engineering methodology, but the facilitator may ask you how and why you have made a specific decision. In this experiment we are not evaluating your expertise on a specific engineering tool, or your expertise in using OWL language. You are expected to have knowledge of basic OWL features, but not of the advanced ones (e.g. SPARQL, C-OWL, e-connections,…).

Your work will be recorded using Camtasia and at the end you will be asked to fill in a simple questionnaire. This will help us understand how people use ontology engineering tools, how efficient and effective the existing tools are, what are the gaps in the support they provide, and similarly.

During the experiment, you may ask questions if you feel you are stuck and cannot continue the task. You may also formulate your ideas, proposals or approaches to the task, and ask the facilitator how these could be achieved in the specific environment. We would appreciate if you provided brief comments on what you are doing using the questionnaire. Altogether the task shall take up to one hour.

You have access to three source ontologies:
O-1 ... Copyright ontology (OWL)
This is a mid-size ontology capturing the concepts of having different kinds of rights and being able to perform different actions with those rights
O-2 ... AKT Support ontology (OWL)
This is a small upper-level ontology defining concepts like time, temporality, tangibility, etc.
O-3 ... AKT Portal ontology (OWL)
This is mid- to large size domain ontology focusing on the organizational aspects such as agency, organization, etc. and then academic activities (publications,...)

[12] FaCT++ can be downloaded from the following: http://owl.man.ac.uk/factplusplus/

Task 1

1.1 Motivation

The Copyright ontology is based around the notion of creating the object on which its holder exercises one of the rights (e.g. the right to lend). The ontology contains the concept 'CreationProcess'. According to its definition, the concept 'CreationProcess' refers to temporal aspects of the creation.

However, the Copyright ontology does not contain any formalization of these temporal aspects. The Copyright ontology needs to be augmented with the relevant definitions from other ontologies, which already contain conceptual definitions of time and temporal aspects (e.g. AKT Support)

1.2 Specification

- Review current definition of the concept 'CreationProcess' in the Copyright ontology and the definition of concept 'Temporal-Thing' in the AKT Support ontology.
- Make the 'CreationProcess' a 'Temporal-Thing' by importing the concept of 'Temporal-Thing' from the AKT Support ontology.
- Note that you may need to also import all other concepts and/or axioms that are needed to properly use the 'Temporal-Thing' in the combined ontology.

1.3 Participant's notes

…….. empty space …….

Task 2

2.1 Motivation

Western copyright laws are based on the identification of an individual owner or creator (who is always a 'Person'). These individuals are entitled to economic reward for their creativity.

However, this is a very narrow view. For instance, a community or organization may have a collective right (rather than all individuals). Similarly, economic rights can in practice be exercised only by legal agents; whereas recipients of the benefits could be any agents (individual and collective).

2.2 Specification

Your task is to extend the very weak existing notion of 'Person' in the Copyright ontology by integrating more formal and deeper conceptualizations from the AKT Portal and AKT Support ontologies. In particular, the concepts of 'Legal-Agent', 'Generic-Agent', 'Organization' and similarly may be useful places to start with.

- Import the concepts describing various types of agency into the Copyright ontology to reflect what has been mentioned in section 2.1.
- Integrate the concepts of 'Person', 'Legal-Agent', etc. from the AKT Portal ontology into the Copyright ontology.

- Ensure that the integrated concepts are used appropriately in the axioms of the Copyright ontology.

Additional Information

Please, do these integration amendments in the following concepts:

- 'Economic-Rights' … the type of property 'agent' shall be extended
- 'Moral-Rights' … in this case, property 'agent' shall be broader than in the case of economic rights; its type shall be appropriately extended
- 'Distribute' … here the type of property 'agent' shall not be formally the same as the type of property 'recipient'; choose an appropriate modification
- 'Distribute' … express the fact that recipients can be any generic agents except geopolitical entities

For instance, each 'Copyright' has currently a property called 'agent', which is untyped. However, in the case of 'EconomicRights' this property is restricted only to 'Person(s)', which is not consistent with our interpretation in section 2.1. You need to replace the type of this property with a concept of e.g. 'Legal-Agent'…

2.3 Participant's notes

…….. empty space ……

Task 3

3.1 Motivation

In many cultures, no single individual person can be identified as the author or a recipient of a work since it is based on cultural traditions, which have been handed down over many generations. These traditions are passed by communicating them, often from one person to another. In other words, communication rights apply as much to the collective of recipients as they apply to individual people.

3.2 Specification

This task is about reconciling the differences in a network of ontologies. It includes simpler sub-tasks, such as finding the right concepts that are needed, and focuses on creating more complex types or restrictions

Note that you are working with networked ontologies, where you can amend the Copyright ontology but should aim to keep the AKT Support and AKT Portal as 'read-only'.

Do the following modifications:

- Find concept 'Communicate' and its property 'recipient' in the Copyright ontology
- You will see that this property is restricted to 'Collective(s)'
- 'Collective' is not defined properly and also is not related to the notions of agency you imported in Task 2.
- Suggest a way to complete the definitions of concept 'Collective' (e.g. you may try and define it in terms of such concepts as 'Organization', 'Organization-Unit', etc.)

- Add appropriate properties and restrictions to the concept ‘Collective’ that would enable you to express the fact that collectives include at least two people.
- Return to concept ‘Communicate’ and modify the type of property ‘recipient’ so that communications could be received either by individuals or by collectives.

For example, you may want to make ‘Collective’ a defined class, which would express the fact that all ‘Organization(s)’ and ‘Organization-Unit(s)’ are also ‘Collective(s)’. If something is an organization, this shall be a sufficient condition for considering it a collective as well… but not the other way round (i.e. ‘Collective’ ⊄ ‘Organization’)!

3.3 Participant’s notes

…….. empty space …….

Appendix B. Guidance for facilitators

Task 1: guidance for facilitators

People need to explore both ontologies, ideally one next to another and clearly point to which concepts are re-used. In Protégé one needs to import these ontologies as new projects using OWL wizards. This is fairly trivial but not intuitive, and people usually complain that they can't have more than one window open. The correct steps are:

- Launch Protégé … a window to open/create projects appears
- Click button 'Create new project…'
- Then check option 'From existing sources' and 'OWL/RDFS files'

If ontologies have imports people need to resolve potential dependencies manually. They simply point to the files where the actual imported ontology resides. AKT Support does not import anything else, so it should be easy to load up and browse. Copyright uses Dublin Core, but this is not available, so the user can ignore Protégé's request to find a file with its definition.

Once in Protégé, people browse, find the right concepts, consider definitions, etc. At this point they may need a hint to tell them about buttons at the bottom of the Protégé screen that toggle an alphabetical display of concepts or enable search. It may take a few attempts to find the right class: search is done on matching full labels, incl. XML namespaces. However, they can use wildcards (*)

Then the simple way to solve the task is to include only the 'Temporal-Thing' concept with its entire sub-branch. Ideally by cutting/pasting, but this will not work. Similarly, they may try 'File → Export to…' option, but this would fail, so the user concludes they need to import the entire AKT Support into Copyright.

Import can be done by editing the Copyright OWL file directly, adding the <import /> property the <owl:Ontology> tag. This means people have to reload the Copyright ontology, go through the OWL wizards, resolve dependencies, etc.

Cleaner way is to use the 'Metadata' tab in Protégé, where you can point them to the 'Imports'. They simply import an existing ontology, all namespaces are created automatically.

Once this is survived, it's fairly simple to redefine the 'CreationProcess' and make it a subclass of 'support:Temporal-Thing'. Since AKT Support was imported as a whole, this shall be consistent and complete in terms of all additional concepts are there to conceptualize the time-duration of a creative process.

For those less familiar with Protégé: people have to go to the 'CreationProcess' class and display its definition. Then in the 'Asserted conditions' pane they have to add a new record for which they choose the appropriate superclass.

Typical complaints that may come up are about losing the concept while looking at another concept. People would do repetitive searches and the alphabetical and hierarchical views are not automatically synchronized, which is confusing

Task 2: guidance for facilitators

Again, one needs to find the ontology in which the concepts like 'Legal-Agent') are actually defined. This shall be trivial, but it may require loading another ontology into a text viewer or into Protégé to make sure = time consuming, repetitive...

As before, people repeat what they learned in Task 1 and import the AKT Portal into the Copyright ontology. This means going to 'Metadata' tab in Protégé and declaring a new import from an existing source.

When pre-viewing AKT Portal, people may notice that it also imports AKT Support. So, they may suggest to have only one direct import in the Copyright ontology and the AKT Support would be imported recursively through AKT Portal. In other words, we aim to recreate more 'network-like' character here.

Once AKT Portal is imported and the Copyright ontology, one should properly clarify which ontology/namespace the concepts belong to. AKT Support is properly labelled, but not AKT Portal. In Protégé it would show as an arbitrary namespace 'p1'. So, the user may want to explicitly change this into 'portal'... which can be done in the 'Metadata' tab.

Next people can start local amendments of the Copyright ontology.

As advised, people should explore concepts of 'Copyright' and 'EconomicRights' first, and then a few other rights that are required ('MoralRights', 'Distribute' and 'Communicate'). People should be able to formulate what they want to do; how they want to make the changes we require. *Ask for explanations, pls.*

Then they should start changing the previous (inconsistent) commitments from 'Person' to appropriate agency (*always ask for explanations, pls.*):

E.g. in case of 'EconomicRights' we suggest:

[allValuesFrom 'agent' 'portal:Legal-Agent']

As a first pass on the 'Distribute' concept, people may do e.g.:

[allValuesFrom 'recipient' 'portal:Generic-Agent']

At this point, they shall realize that 'Generic-Agent' is too broad and also includes 'Geopolitical-Entities', 'Countries',... which is not appropriate. So, they should be nudged towards making the conceptualization more precise; e.g.

[allValuesFrom 'recipient'
[unionOf 'portal:Legal-Agent' 'portal:Organization-Unit']]

In case of 'MoralRights' they are modifying property 'agent', but this time the context is different than in the 'EconomicRights'. They may not be able to express this in OWL but can mention this verbally. Thus, 'Person' type is replaced as follows:

[allValuesFrom 'agent' 'portal:Generic-Agent']

Task 3: guidance for facilitators

Task 3 is a continuation of Task 2, so no additional ontology loading is needed. People only navigate through different branches of the AKT Portal and Copyright ontology in order to verify the statements we gave them in the objectives.

Further amendments are needed for concept 'Communicate'; in line with section 2.1, traditions that are copyright-able by local communities (e.g. native aborigines) can be communicated to individuals not only to a collective (i.e. audience in a classic, western sense).

This means amending the concept of 'Collective'. This change is more substantial and one way to do it is to simply say that concepts 'portal:Organization' and 'portal:Organization-Unit' are sub-classes of the 'Collective' from the Copyright ontology. The issue you should point to the participant is that we demand AKT Portal/Support ontologies to be 'read-only'; i.e. they should not make the sub-class amendments to those ontologies!

A possible way around may be to define two new sub-classes of the 'Collective' in the Copyright ontology (e.g. 'OrganizationalEntity' and 'OrganizationalSubEntity'), and make both of them equivalent to (i.e. defined in terms of) the relevant concepts in the AKT ontologies) as follows (*ask people to formulate this point, pls.*):

[equivalentClass 'OrganizationalEntity' 'portal:Organization']
[equivalentClass 'OrganizationaSublEntity' 'portal:Organization-Unit']

People may want to verify that these statements are consistent with the AKT Portal ontology. To do this, you may need to start for them the FaCT++ or another classifier and choose menu option 'OWL → Classify taxonomy'. Point to the output of the classification correctly recognizing our local 'OrganizationalEntity' as a 'portal:Legal-Agent' and local 'OrganizationalSubEntity' as a 'portal:Generic-Agent'… (*ask people to do this, pls.*)

You may point to the participant that the above approach follows the 'networked style'; simply AKT Portal and Support ontologies are left untouched, because anybody else may be importing the same ontologies without being aware of your modifications. A simpler way is shown below, but this statement has different consequences = we shall not demand that all collectives are organizations or units!

INCORRECT:

[equivalentClass

[unionOf 'portal:Organization' and 'portal:Organization-Unit']]

Finally, people may add a property to 'Collective' = e.g. 'hasMember', and adding a cardinality restriction on this property for the concept 'Collective' saying that we need min 2 people:

[intersectionOf

[allValuesFrom 'hasMember' 'portal:Person']

[minCardinality 'hasMember' 2]]

Record sheet for facilitators

Please record requests made during the experiment = either from the participants or your own. Also, note the observations of people's behaviour, actions, frustration, etc.

Example:

Participant 0

- *Repeatedly edits incorrect class – he is confused because the alphabetical list highlights the right class, but the hierarchy highlights something different*
  → advised him to double-check field labelled 'For Class: [……]'

Participant 1

- *Why can't I simply delete 'Person' from the Copyright ontology?*
  → he would lose all references to this type in axioms (e.g. in EconomicRights), he tried but had to reload ontology ☺

Appendix C. Questionnaire for the participants

In the following questionnaire we have five main goals:

- To register the preferred ontology engineering environment for the tasks presented in this experiment.
- To get some of your impressions about the experiment itself.
- To collect your perceptions regarding the usability of the application interfaces used during the experiment
- To measure the hands-on experience with the available editor and other tools.
- To get your impressions on some of the goals of NeOn, based on your experience during this experiment.

In summary, our purpose is to get your valuable feedback about your experience with ontology engineering tools and about some of the goals of the NeOn project.

We would also appreciate your impressions on practical issues and any other comment or criticisms that you may find interesting.

Please send your completed questionnaires via email to jmgomez@isoco.com

Thanks for your time.

The NeOn project

Setting

A-1. The ontology editor used during the experiment is:

Protege	*OntoStudio*	*Other*

A-2. Please, list briefly other tools used during the experiment.

……

A-3 How would you rate your previous experience with the tools used in the test?

Beginner	*Moderate*	*Expert*	*NA/DK*

A-4. If any, please list what additional tools you would have found useful.

……..

Tasks observation

B-1. How would you rate your previous experience in ontology engineering?

Beginner	*Moderate*	*Expert*	*NA/DK*

B-2. Did you already have experience with the ontologies used during the test?

Copyright ontology	
AKT Support ontology	
AKT Portal ontology	

Please indicate how you perceived the amount of time needed to execute each of the tasks of the experiment:

Task1:

Low	*Average*	*High*	*NA/DK*[13]

[13] NA/DK=Not Appplicable/Don't Know

Task2:

Low	*Average*	*High*	*NA/DK*

Task3:

Low	*Average*	*High*	*NA/DK*

B-2. Your understanding of the tasks comprised in the experiment was:

Low	*Average*	*High*	*NA/DK*

B-3. Please, briefly describe your approach to Task 1 of the experiment.

……

B-4. Please, briefly describe your approach to Task 2 of the experiment.

……

B-5. Please, briefly describe your approach to Task 3 of the experiment.

……

B-6. The difficulties you needed to overcome due to the ontology editor and tools used during the experiment in order to complete each task were:

	Low	*Average*	*High*	*NA/DK*
Task 1				
Task 2				
Task 3				

B-7. How did you find the support provided by the facilitator?

Inadequate	*Adequate*	*Excellent*	*NA/DK*

Usability

Help and documentation

C-1a. Please indicate how useful you found the documentation in the tools and editors used.

Not very	*Reasonably*	*Very*	*NA/DK*

C-1b. Did you find the tooltips provided by the editor were sufficient?

Not very	*Reasonably*	*Very*	*NA/DK*

Interface design/accesibility

C-2a. Please indicate how well designed you felt the system interface was

Not very	*Reasonably*	*Very*	*NA/DK*

C-2b. Did you find the graphic elements, e.g. icons, of your editor clear and legible?

Not very	*Reasonably*	*Very*	*NA/DK*

C-2c. Do you find it necessary greater customization regarding fonts or colours in your editor?

Not very	*Reasonably*	*Very*	*NA/DK*

C-2d. Did you find support for the following languages in the available tools:

	Satisfactory	*Not present but useful*	*Not needed*
English			
Spanish			
German			
French			
Others			

C-2e. What other forms of customization do you find necessary?

……

C-2f. Are you satisfied with the interface design of the editor?

Not very	*Reasonably*	*Very*	*NA/DK*

C-2f. Please, briefly list any suggestion to improve accessibility

……

Hands –on experience

Effectiveness

D-1a. Did you find any problems loading the ontologies? If so, briefly list them.

……

D-1b. Please indicate how easy you found to get acquainted with the experiment ontologies by means of the available tools.

Not very	*Reasonably*	*Very*	*NA/DK*

D-1c. Did you find the editor used allows to set a clear and simple sequence of steps to accomplish each necessary action, e.g. create a new instance of a concept:

Not very	*Reasonably*	*Very*	*NA/DK*

D-1d. What was the major obstacle that you found during hands-on work with the system?

……

D-1e. Was the overall behaviour of the ontology editor and tools:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

Efficiency

D-2a. Please, in case you consider it necessary, describe how the ontology editor should be improved in order to facilitate some specific ontological task.

……

D-2b. Please write a list with approximately the five most repeated operations during your interaction with the tools available during the experiment:

Operation	*Description*

D-2c. Did your ontology editor allowed you to have all the necessary information about the different ontologies handy:

Not very	*Reasonably*	*Very*	*NA/DK*

D-2d. Was there any of the tools described in A-2 apart from the ontology editor that was essential to you during the experiment? Which one(s)?

……

Design Experience

D-3a. Did you find capabilities to flag ontology entities worked on, e.g. in order to allow quick location of important concepts later:

Satisfactory	*Not present but useful*	*Not needed*	*NA/DK*

D-3b. Did you find support provided by the editor to handle nested/dependent ontologies:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

D-3c. Did you find support to handle heterogeneous namespaces of the different ontologies:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

D-3d. How easy was it to perform search and/or replace in multiple places of the ontologies:

Not very	*Reasonably*	*Very*	*NA/DK*

D-3e. Did you find capabilities provided by the ontology editor to momentarily hide parts of the ontologies:

Satisfactory	*Not present but useful*	*Not needed*	*NA/DK*

NeOn Objectives

Visualization

E-1a. How did you find the visualization of interdependencies between different components of the ontology?

Inadequate	*Adequate*	*Excellent*	*NA/DK*

E-1b. Do you think it would be useful to visualize several branches of the ontology/ies simultaneously?

Not very	*Reasonably*	*Very*	*NA/DK*

E-1c. Do you think it would be useful to graphically realize operations between several branches of the ontology/ies? For example, create a mapping between two concepts.

Not very	*Reasonably*	*Very*	*NA/DK*

Reuse

E-2a. Did you find that the support provided by the ontology editor allowed to reuse existing ontologies:

Not very	*Reasonably*	*Very*	*NA/DK*

E-2b. Was the support provided for partial ontology import, e.g. a selected branch:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

Context

E-3. How easy was it to perform contextual changes in the ontologies. For example, in the "Copyright" ontology, the transition from "Person" to "Legal-Agent" after import of the "AKT Portal" ontology:

Not very	*Reasonably*	*Very*	*NA/DK*

Mapping

E-4a. How useful do you find to establish mappings between concepts of different ontologies:

Not very	*Reasonably*	*Very*	*NA/DK*

E-4b. Do you find the support for establishing mappings between concepts from different ontologies:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

E-4c. How useful would you find an automatic mechanism to ensure mapping consistency in a networked ontologies-compliant editor:

Not very	*Reasonably*	*Very*	*NA/DK*

Versioning

E-5a. Do you find the support for creating and maintaining versions of ontological knowledge:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

E-5b. How useful would you find an automatic mechanism to propagate updates through dependencies across ontologies. For example, between the Copyright ontology and the AKT Support ontology, where concept "CreationProcess" in the first depends from concept "Temporal-thing" in the second:

Not very	*Reasonably*	*Very*	*NA/DK*

E-5c. How useful would you find to apply the CVS metaphor to the ontology editor:

Not very	*Reasonably*	*Very*	*NA/DK*

E-5d. How useful do you find to be able to visually compare different versions of the same ontology:

Not very	*Reasonably*	*Very*	*NA/DK*

Reasoning

E-6a. Do you find the reasoning capabilities of the framework used:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

E-6b. If any, please briefly list those features that you find missing.

……

Storage

E-7a. Do you find the different formats (RDF, OWL, Flogic...) available for ontology storage:

Inadequate	*Adequate*	*Excellent*	*NA/DK*

E-7b. If any, please briefly list those that you find missing.

……

Practical matters

F-1. What functionalities would you like to see in next versions of your ontology editor?

……

F-2. Please, add any critical comments or positive suggestions on how the system might be improved.

……

Any other comments or suggestions

G-1. Finally, could you add any comments, criticisms or suggestions about any aspect of the system not covered in the above questions. Thanks for your cooperation in this.

……