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 4.1.2 Analysis of user needs, behaviours & requirements on ontology engineering tools

Deliverable Co-ordinator: Martin Dzbor (OU)

Task Co-ordinating Institution: The Open University (OU)

Contributors: Carlos Buil Aranda (ISOCO), Enrico Motta (OU), Jose Manuel Gomez (ISOCO)

In this deliverable we summarize the rationale, motivation, methodology and findings related to the observational user study that has been conducted on 28 participants using existing tools supporting ontology engineering. The study, which is a follow-up to the study carried out in summer 2006 (see report D4.1.1), considers and comments, in particular, on various aspects influencing effectiveness, efficiency and user satisfaction while carrying out an engineering task with public ontologies using two additional tools.

Document Identifier:	NEON/2008/D4.1.2/v1.0	Date due:	December 31, 2007
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	February 15, 2008
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: Jesús Contreras E-mail address: jcontreras@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	**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	**Laboratorios KIN (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed to the writing of this document or its parts:

Partner 1: The Open University (OU)

Partner 2: iSOCO (ISOCO)

Partner 3: Laboratorios KIN (KIN)

Change Log

Version	Date	Amended by	Changes
0.1	10-12-2007	Martin Dzbor	Theory, motivation, setup, variables
0.2	20-12-2007	Martin Dzbor, Carlos Buil Aranda	Summary of findings, analysis
0.3	18-01-2008	Martin Dzbor	Further analysis, discussion
0.4	13-02-2008	Martin Dzbor	Final clean-up, QA
1.0	15-02-2008	Hannah Beaman	Final QA

Table of Contents

List of tables

List of figures

1. Introduction

Ontologies are seen as one of the key technologies to support interoperability on the Web and to enable semantic integration of both data and processes. Since the notion of ontology was first proposed by Tom Gruber [8], ontologies matured; we are now in the phase when 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).This is changing continuously, and there can be seen a degree of progress since the first version of this study was carried out in July-August 2006.

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 continue with the investigation of the shortcomings and strengths of current tools for working with such advanced ontologies. This document is a follow up of report D4.1.1 [5] that has been released in summer 2006 and compared Protégé with TopBraid. At this point, we included two additional tools, Protégé version 4 and Swoop.

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.

NeOn aims to challenge this undesirable lack of rigour in studying the behaviour of ontology engineering tools. We started this with a highly successful study and its presentation to the OWL: Experiences and Directions community in 2006 [4]. With this document we elaborate on those findings and extend them to other tools on the market – as we suggested in the conclusions from the aforementioned paper.

This study is a continuation of the work carried out in June-August 2006 on Protégé v.3.2 and TopBraid. This time two other tools were observed: Protégé v.4 and Swoop. Analyses were performed not only between the two new tools but also compared to the previous results. 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 is concerned with the actual findings in three categories (effectiveness, efficiency and usability) and their analysis, the structure of the report reflects this setup of the actual user study.

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 qualitative analysis of findings appears in section 5, and section 6 contains discussion and implications from this study. Before presenting the actual study, we start with a summary of a broader motivation for observing 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

We start by pointing to some generic themes from the previous report, which inform and drive our work in the area of studying the tools and environments for ontology engineerings.

As long argued in the human-computer interaction (HCI) domain, interactions involve three parts: the user, the technology, and the ways they work together [12]. In the past, we have expanded 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. It is clear that engineering tools that fulfil at least some needs of ordinary users trying to design advanced ontologies 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 [12], 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 two models are often implicit; users and tools do not expose them explicitly. It means that model of a system/tool is often unknown to the end user and its working is established from the interaction with its 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.

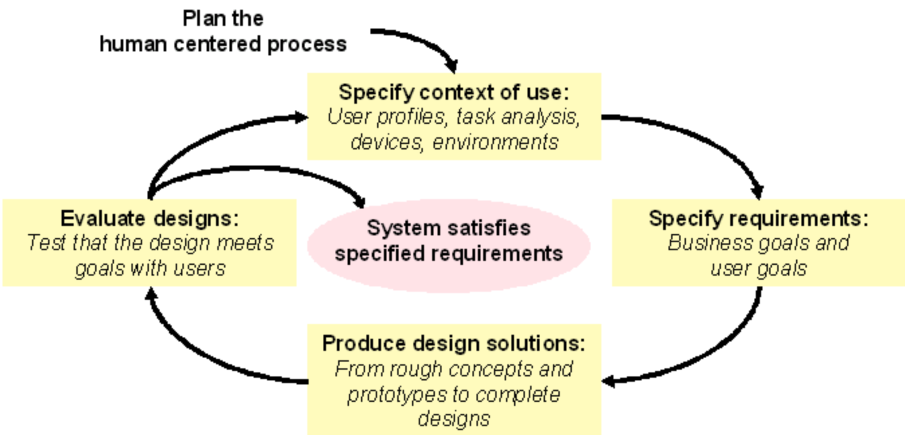

Figure 1. A typical user-centred development spiral [1]

Successful tools typically reflect an understanding of the users, their tasks, their goals, and their environments. A general process for including human-centred activities throughout a development lifecycle of tools has been standardized in ISO 13407[1]. One benefit of this principled design, as shown in Figure 1, is that it helps to bring in different aspects of user experience and needs early in the application lifecycle; thus increasing the chance to develop a successful application.

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

Many 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. Thus, it is our belief that studying other tools, as well as the NeOn products, is a valuable input to improving our understanding of how people use ontology engineering tools, what they like and dislike, and where are still the gaps and challenges, which might be taken on board in the NeOn project.

2.1 Past findings

Some work on evaluating tools for ontology engineering has been done in the past. In [3] authors conclude that the tools available in the time of their study (cca 1999) were little more than research prototypes with significant problems in their user interfaces. These included too many options for visualizing ontologies, which tended to confuse the user and hinder navigation. Moreover, the systems' feedback was found to be poor, which meant a steep learning curve for non-expert users. Finally, most tools provided little support for raising the level of abstraction in the modelling process and expected the user to be proficient in low-level formalisms.

Work described in [13] evaluated Protégé in several tasks, from the perspective of a power user. The authors found the system intuitive for expert knowledge engineers, as long as operations were triggered by them (e.g. knowledge re-arrangement). However, difficulties arose when assistance from the tool was expected; e.g. in inference or consistency checks. Weak performance was also noted in language interoperability. The survey reported in [7] also noted issues with tool support for operations on ontologies beyond mere editing (e.g. integration or re-use). In particular, the authors emphasised the limited 'intelligence' of current tools – e.g. no possibility to re-use previously used processes in current design. Tools expected the user to drive the interaction, with the tool imposing constraints rather than adapting itself to users' needs.

Finally, other researchers [18] found that visualization support in Protégé and its customization models are too complex and do not reflect users' models of what they would normally want to see. Others observed users having difficulties with description logic based formalisms in general [9]. Again, tools expected detailed knowledge of intricate language and logic details, and this often led to modelling errors.

Our own study [4, 5] has highlighted and explored some persevering issues with OWL engineering tools that reduce the appeal and adoption of otherwise successful (OWL) technology by the practitioners. Although the tools made a great progress since the evaluations reported in the previous paragraphs, issues with user interaction remain remarkably resilient. The effort was observed in making the ontological 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 were found to 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 dependencies among multiple imported ontologies or term occurrences in an ontology). Even if such user-centric algorithms exist, they were not found in off-the-shelf tools.

For instance, frequently used operations and their correlations with user errors provided us with opportunities to hypothesize the effect of improving the support for these tasks on the overall tool adoption. In our case, the support was given by facilitators, but clearly, the support for the frequent actions is likely to affect the experiences with OWL engineering. The most frequent steps in OWL development (see [4]) are the actual coding of definitions and import of ontologies (unsurprisingly), but, surprisingly, also search (71% users), re-conceptualization of restrictions and editing of logical expressions (both 54%), and locating terms in ontologies (46%).

Correlations were 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 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 suggested considering multiple ways for defining and editing axioms (to a limited extent this partly exists in Protégé). Any way, our study in [4D] seems to confirm that DL may be good for reasoning, but it is not a medium for thinking (even among ontology designers).

Another issue discussed in [4] concerned the gap between the language of users and language of tools; a high number of users were surprised by syntactically incorrect statements. In 64.3% 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 facilitator, people tended to doubt results of other operations (e.g. search or classification) if these differed from what they expected. Lack of trust is problematic because it puts the tool solely in the role of a plain editor, which further reduces tool's initiative.

The extensive use of features in the tools is also an issue increasing complexity of user interaction. Both tested tools showed most of possibly relevant information on screen at all times. There was little possibility to filter or customize this interaction. 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) reasoning tool, but one cannot modify or filter the components of user interaction.

2.2 Conclusions from the past evaluations

To summarize, there is a number of empirical work with ontology engineering tools, and many studies have highlighted issues. We carried out yet another study in the NeOn project, as none of the earlier studies provided the kind of data we wanted to inform the development of the NeOn Toolkit [20]. Specifically, we formulate the following key reasons (see [4]):

- **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.** Now that the notion of ontologies being networked, dynamically changing and shared by many applications gains more solid ground in the Semantic Web community, it is desirable to study the role of tools in the changed world. In such 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 of the ontology design lifecycle (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 (which is happening in this deliverable, as we are now reusing the methodology to bring in two additional tools).

After this summary of the reasons and our previous study, we describe the current study and its methodology in the next section, and then offer 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. To make the process of observing user needs manageable, we have based our study on a few assumptions. 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

These assumptions have been explained previously, in deliverable D4.1.1 [4, 5]. Hence, we only emphasize that had we decided to conduct a mock-up user study without any understanding of what toolkit we may offer to the users, there would have been a danger of making observations biased to a specific organization, a specific problem domain or a particular school of thought. Hence, it is possible to make a high-level distinction between studying the users in a way useful for engineering ontology design tools (NeOn view) studying *generic engineering* needs. As the distinction suggests, it is likely that NeOn scenarios provide insights in 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* [21], and in particular the engineering act of *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 integrating is when two or more agents/users choose an ontology as a common vocabulary to describe their own problems. Another way of 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.

Previous studies (e.g. [3]) provide 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 for analyzing tool effectiveness and efficiency. In our case, we opted for a clearer split between the modellers/engineers and evaluators – hence, we talk about *observational study* rather than a tool evaluation. Finally, unlike authors in [3] who evaluated several different tools, we biased our study towards OWL-based integrated environments rather than mere ontology editors; hence the choice of Protégé and Swoop in this round of study.

3.1 Study setup and methodology

The second 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. In contrast to the first study, when we opted for a fairly abstract domain of copyrights, in this round we chose more concrete ontologies from the domain of e-commerce. In particular, we selected tasks around the notion of product selling and invoicing, which is an area even non-experts are pretty much familiarized. Moreover, this choice allowed us to use this observational study for gathering the needs about the training needed by NeOn use case partners (in WP7). The following list summarizes the re-usable ontologies and lists their characteristics:

- *PharmaInnova Schema* ontology … a medium-sized ontology capturing the concepts of invoicing and product sales built by NeOn WP7 as a basis for conceptualizing the processes underlying emitting and receiving invoices
- *DOLCE Lite* ontology … a large upper-level ontology defining concepts like time, temporality, tangibility, etc.
- *EDIFact Schema* ontology … a medium- to large-sized domain ontology focusing on various aspects of transactions among business partners, such as different kinds of agency, organization, processes, etc. – all for the purposes of driving e-commerce interactions

Table 1. Features of the ontologies used in the study

Ontology	Lang	Classes	Properties	Expressivity	Notes
PharmaInnova	OWL	36	81	ALCF(D)	Mostly terminological schema with a few cardinality and domain restrictions
DOLCE Lite	OWL	242	326	SHOIN(D)	Large ontological model with a number of equivalencies, defined concepts,…
EDIFact	OWL	200	290	SHIN(D)	A model extending DOLCE Lite as a generic basis for EDI transactions

The *PharmaInnova Schema* 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, mainly *EDIFact*. 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 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. When invited to the study, we asked for participants' (subjective) level of expertise and familiarity with ontology engineering tools and representation languages. The participants were all expected to have knowledge of basic OWL features (e.g. sub-classing, restrictions,…) but not of the advanced ones (e.g. SPARQL, rules e-connections,…). At any time during the session, the participant could ask a clarifying question, and was provided with a pen and paper to formulate ideas, proposals or approaches to the task. During the study, they were observed, 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 28 participants from three different institutions, which included two academic and one industrial organization. Participants were mixed in terms of different level of experience with designing ontologies and with using ontology engineering environments. Two such environments were considered for this round of the study – Protégé v.4 from University of Manchester and Stanford University[2], and SWOOP Browser and Editor (originally) from MindSWAP, University of Maryland[3].

Participant cohorts were composed so that we had 15 people considered highly experienced or expert and 13 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 restrictions on two of the participating sites with respect to timing and familiarity of the facilator, we could not achieve a perfect balance between the subsets, but this was later taken into account in the analysis. At the end, out of 28 participants, there were 18 assigned to carry out the study with Protégé v.4 and 10 worked with SWOOP.

3.2 Tasks given to users

As mentioned earlier, participants were given three tasks – each about different aspect of integrating 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.

[2] More information on Protégé v.4 and the download of the latest version is available from http://protege.stanford.edu.

[3] SWOOP download and further details are available from http://www.mindswap.org/2004/SWOOP.

3.2.1 Task 1: Ontology import, sub-classing and property definition

In Task 1, participants were asked to introduce concept *ProductDelivery* into *PharmInnova Schema*, which would formally be defined in terms of the *EDIFact Schema* (that needed to be imported appropriately). The definition of *ProductDelivery* was to be augmented by sub-classing it to the relevant definition of *FinancialTransaction* from *EDIFact*.

The objective was to review the given ontologies, to create and locate the classes in question (i.e. *ProductDelivery* and *FinancialTransaction*, respectively), and to make an assertion that *subClassOf (ProductDelivery, FinancialTransaction)*. Participants were expected to import *EDIFact Schema* into *PharmaInnova Schema*, and were prompted to ensure that it was indeed possible to express the correspondence (i.e. equivalence) of the two concepts mentioned above. The objective of the task was to observe the approaches to searching, browsing, importing, and defining relationships among concepts from different ontologies. This task was also considered as an aid to familiarize participants with different parts of the user interface.

The second part of the task aimed to reinforce the familiarity, by essentially replicated the process of creating new entities and relating them to the imported ontology. The only difference was that this time, the request was to create two properties *involvesEmitter* and *involvesRecipient*, which were supposed to be declared as sub-properties of *involves* (defined in DOLCE Lite, which was imported by *EDIFact Schema*). Also, the users were asked to express two restrictions on the two new properties constraining their domains to *EmittingCompany* and *ReceivingCompany* respectively (both already defined in the *PharmaInnova Schema*).

3.2.2 Task 2: Import of two ontologies and axiom amendments

Ontology integration in Task 2 was motivated by the need to differentiate two types of deliveries to complete the definition of the concept introduced in the previous task. Namely, we asked the users to appropriately extend the definition of an existing concept *DeliveryPoint* in the *PharmaInnova Schema* ontology. The extension was expected in terms of turning the *DeliveryPoint* into a concept that is defined by means of satisfying the following conditions:

- *DeliveryPoint* is equivalent to a *Company*
- The *physical_location* of any *DeliveryPoint* has to be a member of class *Place*
    - Sample solution: *rangeOf* (*physical_location* , *Place*)

Furthermore, we asked the participant to introduce two new types of product delivery, which would be distinguished based on the above-introduced *physical_location* property. In particular,

- Concept *DeliveryPointDirect* would correspond to all those delivery points, which have as physical locations any place <u>except</u> elements of class *Warehouse*
    - Sample solution: *rangeOf* (*physical_location* , *Place* $\wedge$ ($\neg$ *Warehouse*))
- Concept *DeliveryPointMediated* would correspond to all those delivery points, which have as physical locations <u>only</u> elements of class *Warehouse*
    - Sample solution: *rangeOf* (*physical_location* , *Warehouse*)

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:

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. This was similar as in the previous study, so these tasks are essentially equivalent and focusing on how the tool may help people (i) with composed restrictions (e.g. conjunctions) and (ii) with translating natural language terms (like ‘except’) into logical constructs of the (OWL) language.

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 *ProductDelivery* inside the *PharmaInnova Schema* ontology so that formal definitions include what we introduced in the previous two tasks. Participants were told to make amendments only to their base – *PharmaInnova Schema* ontology, rather than to the other two.

They we first asked to locate *generically_dependent_on* – a property inherited from the *DOLCE Lite* ontology, which should have appeared in the definition of concept *ProductDelivery* – if done correctly in the previous tasks. This property, in its inherited form, was restricted to only very abstract notions from the *DOLCE Lite* ontology, and we asked people to make a more specific commitment. In particular, we wanted them to introduce two new concepts, *ProductDeliveryDirect* and *ProductDeliveryMediated*, which would satisfy the following conditions:

- Both new concepts would be sub-classes of concept *ProductDelivery*
- The two sub-concepts should be distinguishable by introducing appropriate restrictions to the property *generically_dependent_on*; e.g. “*ProductDeliveryDirect* would be *generically dependent on* elements of class *DeliveryPointDirect*”

The second half of Task 3 tackled the requirement to express in formal terms the fact that each an every product delivery made by a company has to fall <u>either</u> into class *ProductDeliveryDirect* <u>or</u> into class *ProductDeliveryMediated*. In other words, users were asked to introduce so-called covering axiom, an exclusive disjunction (XOR) into their ontology. This could have been done in the following style:

– Introducing equivalentClass (*ProductDelivery* ,
unionOf (*ProductDeliveryDirect* , *ProductDeliveryMediated*))

– Introducing subclassOf (*ProductDeliveryDirect* , *ProductDelivery*)
subclassOf (*ProductDeliveryMediated , ProductDelivery)*

The participants were to explain the construct(s) they introduced; e.g. in terms of verifying that any instances had to be introduced in one of the sub-concepts; thus there could not exist a ‘generic’, untyped product delivery.

3.3 Evaluation and analysis instruments

Similarly to the study reported in D4.1.1, the primary goal of the study was to identify capabilities and gaps in the existing support tools. Hence, we based our analysis on *observing users* carrying out a specific task using existing tools, and *expressing* their interaction in a set of measures looking at different facets. In other words, we conducted 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 (but should give a broader view across several tools). As before, it was not our objective to prove that one environment is better than other, merely to see their effect on effectiveness and efficiency of the ontology design process. Second, while generic tool usability was considered important, measures were expected not to be solely usability-centric.

Tool and expertise level made standard parameters for *grouping* users, to keep this study compatible with one carried out in the summer 2006. Note that grouping parameters are having slightly different role than standard *independent variables*, which would be needed *if we wanted* to carry out an experiment. However, we don't have our own toolkit ready in a form to allow us to hypothesize something useful, and we were 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 [15], 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. Two basic instruments for data analysis were available to us: (i) participants' questionnaires and debriefs, and (ii) facilitators' notes summarizing the user's interactions during the study. In terms of what has been observed and analyzed, we refer to [10] and in particular, to the following levels of analysis: (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 the same as in the past study; it was designed with 53 questions covering several categories of measures. 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. [16] 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 instrument (facilitators' notes) was 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 expressed in measures from the same categories may be used afterwards to attempt answering *why* the difficulties occurred.

Table 2 (overleaf) repeats the 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 [11] ranging from very useful, satisfactory, etc. (+1) to not very useful, missing, etc. (–1). The questions 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 [2]. 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 calculated 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. To assess the effect of the tool and 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.

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 : 'coding'	A/V	Estimated ratio of conceptualization/understanding : 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. 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.

In addition to reporting the observations from this year's study, we also include cross-analyses with the last year's data and/or observations.

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 *reasonable*. If we express 'not very easy' as '–1', 'reasonable' as '0' and 'easy' as '+1', the 'mean' attitude of the total population of **28 participants** was inclined more towards the positive end of attitudes, with 'mean' being **+0.032** (see Table 3). Although the perception was not negative, participants were not convinced by the tool's capability to introduce its functionality simply and effectively. However, this was an improvement on the last year, as we mention below.

A negative attitude emerged when participants were asked to rank how they felt about the support for frequently repeated operations (such as repeated definition modifications). Most participants perceived the existing support more towards the 'not very good' value; the 'mean' is **–0.387**. In other words, the tools used this year did not effectively support operations that could be potentially 'bundled' into macros or pre-defined sequences of steps. Similar observations were made as last year, e.g. searching used not to actually find a class satisfying a 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 (current study)

Measure/question	Avg. response	–1	0	+1	Total	Mean
process of getting around the tool and understand it	reasonable	25%	43%	32%	28	+0.032
support for frequently repeated operations	not very good	47%	46%	7%	28	–0.387
overall behaviour of the tool	reasonable	22%	46%	32%	28	+0.097
effectiveness of dealing with task 1 (e.g. difficulties)	very easy	46%	36%	18%	28	–0.355
effectiveness of dealing with task 2 (e.g. difficulties)	moderate	25%	64%	11%	28	–0.194
effectiveness of dealing with task 3 (e.g. difficulties)	moderate	18%	61%	21%	28	+0.032
help from the facilitator	reasonable / very useful	4%	53%	43%	28	+0.355

Table 4. Selection of a few general observations (previous study)

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 **+0.097**). This means that participants judged the tools acceptable; this is a difference on the last year’s tools – partly could be attributed to the use of significantly redesigned Protégé v.4.

The operations or activities that were most frequently found among complaints about the effectiveness are more varied than last year. Still “importing ontologies” was mentioned, but usually only in Swoop (which had a rather hard to locate option to turn a viewed ontology into ‘editable’, which would allow imports). The “importing ontology” operation is among the most frequent ones in the networked ontology context, and this has been confirmed again. Out of 28 participants, 10 mentioned this among most common operations. At the same time, 8 participants 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 debriefing afterwards, most people concurred that opening and importing was a kind of automatic click, rather than a fully-fledged meaningful operation. So if the tool imposed an obstacle (e.g. in finding the ‘import’-related buttons), they were easily thrown off the track.

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 ‘moderate’ – ‘difficult’ end; as also shown by **‘mean’ attitudes –0.355 vs. +0.032**. 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’ **+0.355**), which contrasts with the aforementioned more neutral judgement of the tool support in getting around (+0.032 or reasonable).

In terms of comparing with the previous study, we observed that the pattern from last year, when participants tended to take a neutral view towards the tool’s overall behaviour, has not been repeated. The positions on this measure were more polarized than before, which led to a weakly positive average reaction to the tools’ behaviour. This is again a good news that shows a degree of progress has been achieved over the 18 months since the last study; probably attributable to the introduction of Protégé v.4 into our tool set. Nonetheless, we will re-visit some of the finer-grained constituents of this broad measure later; e.g. people still maintain their negative attitudes toward visualizations, but an attitude towards reasoning support has improved.

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 many questions. Table 5 shows some measures that appeared in the last study, with a few more measures added, where different attitudes were observed. In cases of perceived task complexity, Protégé v.4 users generally considered task 1 **less complex** than Swoop users, but in case of task 3, the attitude were opposite. However, the significance of both variances dependent on the tool has **not been confirmed** by χ^2 test at p=0.05. Only in the case of the third task the χ^2 value got close significant levels. If we look at what functionality different people reported, at least a part of the reduction in complexity seems to be due to the capability of Swoop to easily introduce different restrictions or other axioms into its categorized user interface. This gives users all possible choices (e.g. equivalencies, unions, intersections, ranges, etc.) In fact, intuitiveness of the user interface was a common comment that appeared with both tools. Protégé was praised as a step forward from v.3.2, but still being fairly busy. Swoop was simpler, but some people had difficulties switching it from a 'viewer' into an 'editor' mode due to an obscure location of the respective switch.

Table 5. Comparison of attitudes between tools groups (SW: Swoop, P4: Protégé v.4) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	$\chi 2$	Sign.
overall behaviour of the tool	tools	SW (-0.20) vs. P4 (+0.29)	5.12	no
getting around and understanding the tool	tools	SW (-0.30) vs. P4 (+0.28)	3.14	no
subjective complexity of task 1	tools	SW (+0.10) vs. P4 (-0.50)	4.51	no
subjective complexity of task 3	tools	SW (-0.30) vs. P4 (+0.22)	5.51	no
information about ontology edited	tools	SW (+0.70) vs. P4 (-0.39)	7.07	yes
support and performance of reasoning	tools	SW (-0.06) vs. P4 (+0.50)	5.48	no

Some variance is observable in participants' perception about the tool behaviours. Most Protégé v.4 users considered the tool positively ('mean' attitude +0.29), whereas Swoop users tended to opt for more negative attitudes ('mean' attitude –0.20). 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. Here Protégé scored slightly better, perhaps due to its market leadership and many people simply getting used to its user interface, which was also slightly simplified in version 4, so a positive step for Protégé.

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, switches or mouse click interactions.

In terms of comparing this year attitudes to the ones from last year, we note an overall positive attitude prevailing for Protégé v.4 in comparison with its predecessor – although this was not found to be significant variance (χ^2 = 5.45). Using Protégé v.4 also led to more positive attitudes than using TopBraid from the last year (χ^2 = 7.43), and this variance was statistically significant. On the other hand, TopBraid was seen more positively than Swoop, and this was again a statistically significant variance (χ^2 = 7.35).

Users also appreciated that Swoop gives better information about loaded ontologies than both Protégé v.3.2 and TopBraid from the last study. Both variances were found to be statistically significant: χ^2 = 6.71 (against Protégé v.3.2) and χ^2 = 7.55 (against TopBraid). Protégé v.4 also raised more positive attitudes w.r.t. this measure than both TopBraid and Protégé v.3.2.

Both current tools (Protégé v.4 and Swoop) led to more positive attitudes regarding their reasoning functionality when compared to Protégé v.3.2 from the last study. In case of Protégé v.4 this variance was also statistically significant (χ^2 = 9.62), which can be seen in the light of some users' comments pointing to a simpler triggering of the reasoning function in Protégé v.4. For similar reasons, Swoop was considered more positively at χ^2 = 5.48.

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

Measure	Type	Outcome	χ2	Sign.
getting around and understanding the tool	experience	Le (+0.12) vs. Ex (-0.27)	1.34	no
role of tool in reducing complexity of task 2	experience	Le (-0.44) vs. Ex (+0.13)	6.46	yes
ease of concepts mapping	experience	Le (-0.06) vs. Ex (+0.47)	4.91	no

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 6 shows some measures where different attitudes were observed. E.g. in task 2 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. 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 may relate to the fact how much does tool guide the user, e.g. in terms of giving him or her different options for axiomatizing a particular statement. Here Swoop provided the categories that at least suggested the options (in the language of the tool). Yet it seems that this aspect was somewhat offset by Swoop's support for 'ontology flipping' (i.e. switching to and looking at different ontologies while devising an axiom).

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 each experienced user reported they had used Protégé v.3.2 in the past; yet they carried only a part of this past experience into getting acquainted with the new tools.

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 clicking when moving between ontologies…*". This particular quote comes from a participant using Swoop, but similar, although less strongly worded statements were noted also in the Protégé v.4 group.

If we look at the distributions of attitudes between less and more experienced users adjusted for different pairing of tools, we can see a variance in less experienced users having more neutral attitudes towards the Protégé family than the experienced ones (at χ^2 = 5.89 this was almost significant). With an exception of working with multiple ontologies where less experienced users were more positive in the Protégé v.4/v.3.2 pair (at χ^2 = 5.90), the other differences on expertise were farther from statistical significance. Similar distribution of attitudes emerged if we adjusted the set for Protégé v.3.2 and Swoop users; the attitudes of both less experienced users and experts were broadly similar, with no statistically significant variances.

Adjusted for Protégé v.4 and TopBraid pairing, the attitudes of both groups tend towards the positive end of the Likert scale. One interesting measure where more experienced users reacted more positively was a general support for concept mapping and linking between multiple ontologies – in this pairing, the variance between less experienced and experts' attitudes was statistically significant at $\chi^2 = 6.85$ (in favour of experts). As in the previous paragraph, if we adjust the set for Swoop and TopBraid, the variances in expert and non-expert attitudes dissipate and are nowhere near being statistically significant.

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 7 and give some general observations.

4.2.1 Efficiency-related findings – general

Participants still 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 this time inclined towards the neutral end of the spectrum ('mean' attitude **–0.032** as opposed to previous –0.172). This is a positive step forward, as this year's batch of tools represents version that improves on the previously used ones (esp. true with Protégé v.3.2 and v.4). Hence tool authors addressed many objections we raised in the previous study.

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 **–0.458**) and limited usefulness of the tool tips, hints etc. ('mean' attitude **–0.241**). Although, both of these parameters actually improved compared to the last study.

On the level of how often people perceived that they carried out repetitive tasks, the usual observations included the modification of concept restrictions and the repeated search for concepts – both reported by **19 out of 28 participants**. Both operations were closely followed by writing logical expressions for axioms (**13 out of 28**).

Table 7. Selection of a few general observations across population (current study)

Measure/question	Avg. response	–1	0	+1	Total	Mean
providing sufficient information about ontologies	reasonable	35%	29%	36%	28	–0.032
support provided by documentation, help	not very good	50%	50%	0%	22	–0.458
usefulness of the tool tips, hints, …	not very good	46%	42%	12%	26	–0.241
subjective time taken for task 1	very low/moderate	39%	43%	18%	28	–0.194
subjective time taken for task 2	moderate	28%	53%	19%	28	–0.097
subjective time taken for task 3	moderate/too long	14%	43%	43%	28	+0.323

The loss of focus when switching between multiple ontologies seemed to have been less problematic than last time. Here tabbed (for Protégé v.4) and combo box selection (for Swoop) user interfaces are the likely contributors to this particular difference.

Table 8. Selection of a few general observations (previous study)

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

As already mentioned above, the tools are now providing more information about the loaded ontologies. Both Protégé v.4 and Swoop have dedicated tabs giving details about ontologies, such as entity statistics, nature of entities (defined, inherited), ontology type ('expressive power'), etc. Both tools give more information than Protégé v.3.2 and TopBraid from the last study, so again a useful step forward addressing some of the negative findings from the past.

While formal documentation has little improved compared to the previous study, there is an increase in the likelihood of people perceiving tool tips as more efficient in this year's batch of tools. In terms of subjective efficiency of carrying out tasks, the overall pattern remained, but the differences between the attitudes were less sharp. Participants moved closer to the 'moderate' part of the spectrum, which was most visible for task 1.

4.2.2 Efficiency-related findings – effect of tool

On the tool level, the efficiency of the two groups was approximately the same. Table 9 shows measures from this category, including one where different attitudes were observed. One of the factors that might affect the efficiency of the tool support relates e.g. to the perception of how the tool helped to handle ontology dependencies. Here users of Protégé v.4 showed slightly more negative attitudes than Swoop users, but $\chi^2 = 5.42$ proved that the variance between the two tools was **not significant at p=0.05** (although close).

Swoop and Protégé are both roughly comparable in terms of users' attitudes to the efficiency of their ontology visualization and visual navigation support (at $\chi^2 = 1.64$), the little variance between the two tolls was not statistically significant. Another variance, also not significant according to the χ^2 test, was observed with ontology formats. Here Swoop was judged slightly more positively, but not statistically significant ($\chi^2 = 2.87$). If we look into qualitative remarks, many (expert) users had objections with the simplified syntax of their axioms in Protégé v.4, although the attitudes to this particular element were more positive this time than in the previous study – partly due to a slight style change in constructing statements from axioms in this simplified syntax in v.4. The new syntax seems to be perceived as more efficient by people and fewer complaints were raised about the lack of support for editing more complex restrictions than in the past.

One qualitative feature that was observed in both tools was related to the so-called depth of an operation in the user interface. Subjectively, **11 participants out of 28** felt that they had a problem with finding an operation within a menu or workspace. Most frequent 'offenders' this time were the 'editable' switch (for Swoop) and in some cases search within one particular ontology (which hadto be explicitly chosen in Swoop but not in Protégé v.4).

The efficiency of search has been emphasized by a **statistically significant variance** in attitudes towards the two tools: while Protégé v.4 users showed neutral attitudes, Swoop users were more negative (at $\chi^2 = 6.58$). As suggested above, one reason could be in having to choose 'working' ontology in Swoop, and in our study, the *DOLCE Lite* ontology consisting of around 12 imports, which all showed as separate search spaces, caused a drop in perceived efficiency.

In terms of efficient reuse of data from other ontologies, the variance was not significant, and in both tools was around a neutral sentiment.

Table 9. Comparison of attitudes between tools groups (SW: Swoop, P4: Protégé v.4) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
help with handling ontology dependencies	tools	SW (-0.10) vs. P4 (-0.35)	5.42	no
efficient visualization & ontology navigation facilities	tools	SW (-0.50) vs. P4 (-0.55)	1.64	no
handling ontology syntax / abstract syntax	tools	SW (+0.30) vs. P4 (+0.17)	2.87	no
efficiency of carrying out axiom modifications, mappings	tools	SW (+0.61) vs. P4 (+0.19)	3.93	no
efficiency of reusing data from imported ontologies	tools	SW (0.0) vs. P4 (-0.17)	1.51	no
efficiency of searching, locating entities in ontologies	tools	SW (-0.70) vs. P4 (-0.07)	6.58	yes

In terms of comparing this study with the previous one, there is a variance between using Protégé v.3.2 and v.4 with respect to handling ontology syntax – here, the latest version gets more positive attitudes, but this is not significant (χ^2 = 2.50). A similar observation can be made about Protégé v.4 being more efficient in terms of search and data reuse than its predecessor. Swoop was also seen more positively than Protégé v.3.2 in handling dependencies (χ^2 = 5.67), as well as in handling ontology syntax (χ^2 = 4.20). On the other hand Protégé v.3.2 was seen less negatively on efficient reuse of data (likely due to the aforementioned selection of a working ontology in Swoop).

Also, TopBraid from the last study showed more neutral attitudes to the efficiency of visualization and visual interaction than both Protégé v.4 and Swoop (at χ^2 = 4.21), which is likely due to some useful visualizations being included in the basic TopBraid package (e.g. UML-style).

4.2.3 Efficiency-related findings – effect of expertise

As last time, the level of expertise had minimal effect on the differences in perception of efficiency. Table 10 shows some measures where different attitudes were observed. Both groups concurred that they perceived available visualization and navigation support as less efficient compared, with experts being somewhat less negative (however, χ^2 = 0.53 was **not significant** at p=0.05).

As mentioned in the previous section, both groups had broadly positive attitudes to the simplified syntax and its role in improving efficiency. Probably not surprisingly, less-experienced people showed more positive attitude to this aspect than experts. The degree of confidence of this variance was fairly low though; χ^2 = 1.86 **did not confirm** the significance at p=0.05.

The only question where higher levels of confidence (though still not **significant**) of variances was observed is concerned with the efficiency and speed of carrying out the integration of imported axioms into the working ontology (at χ^2 = 4.91). Unlike in the last study, there were no major surprises, both user groups found the support for this operation moderately efficient.

On the other hand, version and change management in the working ontologies was a source of a minor variance: experts seemed to have more negative attitudes to the existing support for this task or found the feature missing. There was a difference in attitudes as shown in Table 10, but the value of χ^2 = 1.23 **did not confirm** any statistical significance of this variance.

Table 10. Comparison of attitudes between expertise groups (Le: less experienced, Ex: expert) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
ease/speed of carrying out mappings	experience	Le (+0.07) vs. Ex (+0.13)	4.91	no
level of visualization and navigation support	experience	Le (-0.61) vs. Ex (-0.46)	0.53	no
specifics of ontology representation languages, abstract syntax, etc.	experience	Le (+0.40) vs. Ex (+0.29)	1.86	no
management of versions for engineered ontologies	experience	Le (0.0) vs. Ex (–0.31)	1.23	no
efficiency of reusing data from imported ontologies	experience	Le (0.0) vs. Ex (-0.22)	0.98	no
efficiency of searching, locating entities in ontologies	experience	Le (0.0) vs. Ex (-0.40)	1.57	no

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.*"

If comparing these results with the previous study, the effect of expertise on speed of creating mappings and integrations, was similar in cohorts including Protégé v.3.2 and TopBraid. The only variance appeared if the cohort was adjusted for TopBraid and Protégé v.4 pairing – here, more experienced users showed more positive attitudes to the efficiency of this operation (χ^2 = 6.85). This might be likely due to a strong engineering focus of both environments, which seems to be more appealing to the expert users.

In other measures, the cohorts adjusted for different pairings of tools (i.e. Protégé v.3.2/v.4, Swoop/Protégé v.3.2, Protégé v.4/TopBraid, and TopBraid/Swoop) show very similar attitudes, and the levels of variances are not significant.

4.3 Design and user experience related findings

With respect to user experiences with the offered functions, several questions were asked. Since the number of the experiential findings may be large, we followed in the tracks of the previous study, and only report on experience with selected aspects that are deemed as relevant to NeOn. 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 the *attitudes towards future developments*. We start with Table 11 and give some general observations regarding the user experiences with the existing functionalities.

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

Although, on the positive note, there are also some improvements in the attitudes towards the reengineered versions of the tools that were used in the current study – this is particularly visible in the comparison of older and newer members of the Protégé family.

4.3.1 User experiences findings – general

Our participants tended to being dissatisfied with the usability of tool documentation, tool tips and various other tool-initiated hints played in easing them into the tool. We also observed difference between this year's answers compared to the previous one, as follows: A 'mean' attitude towards the usability of the tool's help system remained negative ('mean' **–0.458**), which was a slight improvement on the previous opinions ('mean' **–0.500**). Usefulness of tool tips and hints was also negative, but the improvement was more obvious ('mean' **–0.241**) against previous 'mean' **–0.423**. This seems to confirm that tool designers are increasingly taking into consideration the powers offered to them by integrated programming environments (as e.g. Eclipse) and make use of the features helping users to get around easier.

Similarly, participants considered the support for tool customization – e.g. its user interface or functionality – mostly negatively. This marginally improved: 'mean' **–0.333** from the previous value of **–0.400**. A common reason for this low score remained the lack of opportunity to automate some actions, lack of support for keyboard-centric interaction, lack of support for more visual interactions. Reasons remain diverse, and to some extent reflect a user's personal preference in interacting with ontological models. We touch on these tool- and experience-specific differences later.

Table 11. 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	50%	50%	0%	22	–0.458
usefulness of the tooltips, hints, …	not very good / reasonable	46%	42%	12%	26	–0.241
intuitiveness of customizing the tool, its GUI or functionality	not very good	39%	43%	8%	28	–0.333
usefulness of showing ontology dependencies	not very good / reasonable	29%	60%	11%	27	–0.200
intuitive visualization of imports, constraints & dependencies	not very good	57%	39%	4%	27	–0.516
support for [partial] ontology import	not very good	73%	27%	0%	26	–0.655
useful tool interventions in establishing mapping	not very good	58%	42%	0%	24	–0.577

Other aspects where marginal improvements have been observed include the intuitiveness of how the system shows and presents ontology dependencies to the user and how it visualizes imports and other dependencies. While generic capability of showing dependencies shows a sharper sinking tendency (from 'mean' **–0.259** to 'mean' **–0.200**), the visualization was less well received (previous 'mean' **–0.536** only made it to current 'mean' **–0.516**). We believe the main reason in improved attitudes towards the former measure is substantially due to improving the amount of information tools provide about loaded ontologies – in both studied tools there is an explicit page dedicated to such ontology-level descriptions (see also Table 5 and related discussion).

One qualitative feedback related to the usability that occurred less frequently this time was the complaint about too many actions and options being available at any given point during the integration tasks. In the previous study, this comment referred to the amount of information displayed and the number of window segments needed to accommodate this information. While users were willing to accept constant presence of e.g. properties in Protégé v3.2, one could not easily see only those properties relevant to a given concept – i.e. in a slot-like manner. This seems to have improved, partially due to new versions distinguishing 'class views' from 'entity views'. In other words, some customization has been taken on board, albeit still not on the level of individual users (i.e. no personalization opportunities).

Interestingly, features like drag&drop have somewhat disappeared from people's complaints, too. In the past, 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 in the previous study actually reported that they missed drag&drop features. In this study, the capability was hardly mentioned. Similarly, the capabilities to work with multiple ontologies simultaneously became somewhat default; people barely commented on it and were more willing to accept different solutions to the issue – tabs in Protégé v.4 and pull-down combo box on Swoop. Although, the tab-based metaphor seemed to have come more naturally to the users.

Another general point affecting the usability of the user interfaces that remains is about the non-standard and variable use of icons, buttons and various switches. What remained confusing was the fact that some key interaction pathways to carry out a central operation (e.g. switching on the editing mode in Swoop) were positioned in a rather obscure fashion among other switches, which were semantically unrelated with each other.

Other comments related to usability are summarized below:

- More intelligent handling of mistakes in axioms or incomplete axioms – e.g. if a user was editing a complex axiom (e.g. intersections to conceptualize the 'except' statements) manually in a dedicated dialog, s/he could not check on the concept spelling; confirming an incomplete axiom led to it being ignored and essentially dropped by the tool, which meant starting it again;
- Possibility to add axioms to one of the available 'categories' (Swoop) was found helpful by some users; e.g. the tool offered them a slot where they can express equivalencies between classes and where (say) range restrictions). If this guidance was not provided (Protégé), a hint from the facilitator was often needed to help people to interpret the difference between two similar 'add' action buttons, which led to different conceptual consequences;
- Translation of the conceptual statement into the tool's half-natural and half-description logic – DL – formalism was still a matter causing difficulties to some users that in most cases required direct intervention of the facilitator. This seemed to have been improved and the current form of the semi-formal language used looks more acceptable (e.g. when compared to the one used by Protégé v.3.2). Yet, this is still a matter of preference, which cannot be toggled: for experts it was too non-DL, for less experienced it was too much DL;
- Intuitiveness of the tool in terms of finding and locating an operation in the menu or on screen, finding the concept in the ontology and flagging it so that it does not disappear remained among those occurring in the feedback from our users. Particularly, it was noted that more visibility should be given to the key actions taking place at key stages of the user interaction (e.g. 'apply changes' once modifications were made or 'editable' upon loading ontology);
- 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).

Thus, in general, responses about the usability of the existing tools in many key functional areas remained inclined towards the negative end. In addition to aforementioned dependencies among ontologies and their visualizations; support for partial import (or at least notification from the tool of being unable to do so) branch received only 'mean' **–0.655**, and tool's intervention in establishing mapping was also considered of limited use, with a 'mean' **–0.577**. Clearly, even newer versions of ontology engineering tools do not focus on this particular scope substantially differently from the ones studied in our previous study.

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 people tended to show more neutral or positive attitudes toward Protégé v.4 than Swoop (but the χ^2 test **did not confirm significance** at p=0.05 in any variance). The ones being more obvious include overall satisfaction with the user interface and the intuitive way of handling ontology dependencies. Table 12 summarizes some other measures where different attitudes were observed.

Table 12. Comparison of attitudes between tools groups (SW: Swoop, P4: Protégé v.4) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
level of overall satisfaction with the tools	tools	SW (-0.06) vs. P4 (0.0)	5.19	no
overall satisfaction with tool's graphical environment	tools	SW (-0.50) vs. P4 (-0.06)	3.11	no
intuitive handling dependencies among ontologies	tools	SW (-0.35) vs. P4 (-0.10)	5.42	no
intuitive visualization and navigation support	tools	SW (-0.55) vs. P4 (-0.50)	1.64	no
ease of carrying out context/ontology switches	tools	SW (-0.06) vs. P4 (+0.40)	3.93	no

Comparing with the previous study, there is a minor variance (yet **not significant** at χ^2 = 2.76) in user satisfaction being in favour of Protégé v.4 against its predecessor v.3.2. Also, a similar variance in favour of TopBraid against Swoop can be noted (though also **not significant**, χ^2 = 5.0). Variances in user satisfaction with other tool pairings are not notable.

Another feature that was perceived as subjectively better in Swoop against Protégé v.3.2 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. Value χ^2 = 5.67 is not significant at p=0.05, but it is in favour of Swoop. Interestingly, TopBraid was considered more satisfying in the same measure than Protégé v.4 (with χ^2 = 7.43); yet at the time of writing we were not able to point to a likely feature that caused this variance.

From other responses, we highlight slightly better performance of TopBraid in visualizing te dependencies among ontologies and mapping the concepts. TopBraid interface was perceived more intuitive, so it was slightly easier to see what is imported, what is a defined class, etc. than Swoop and to a lesser extent Protégé v.4 (χ^2 = 4.21 and χ^2 = 1.54, respectively). Possibly due to the same factor of a less crowded interface and easier navigation in tabs, also the context switching was seen as more intuitive in TopBraid and Protégé v.3.2 (both using the tab metaphor) than in Swoop (but values χ^2 = 2.80 and χ^2 = 2.46, respectively, **did not reach threshold of significance** at p=0.05).

4.3.3 User experiences findings – effect of expertise

Looking at variances based on the users' experience, we note several responses to be consistent with the numbers from the previous study. Table 13 shows some measures where different attitudes were observed. In general, the variances are further from being statistically significant than we observed in the past; hence only a brief summary of this perspective.

For instance, less experienced users felt that the tools were too rigid for them and found the possibilities to customize them not satisfying. Value χ^2 = 3.85 means **this is not a statistically significant** variance at p=0.05. As mentioned earlier, this observation is more generic, often coupled with the complexity of GUIs, so it is not surprising the experience made little difference.

Table 13. Comparison of attitudes between expertise groups (Le: less experienced, Ex: expert) – significance threshold: χ^2=5.99 at p=0.05

Measure	Type	Outcome	χ2	Sign.
overall satisfaction with tool's graphical environment	experience	Le (0.0) vs. Ex (-0.07)	1.46	no
availability of customization of the tool, its GUI or functionality	experience	Le (-0.50) vs. Ex (-0.23)	3.85	no
intuitive tool guidance during editing	experience	Le (0.0) vs. Ex (+0.13)	1.34	no
overall satisfaction with tool functionality	experience	Le (-0.31) vs. Ex (-0.13)	0.86	no
intuitiveness of reasoning and inferences	experience	Le (0.0) vs. Ex (+0.07)	1.95	no

Another observation, where minor variance based on experience is in favour of experienced users, relates to the intuitiveness of the tool guidance in carrying out certain actions (e.g. 'apply changes'). Less experienced participants felt less satisfied with this aspect, but χ^2 = 1.34, this **was not a significant variance** at p=0.05. Probably the key factor affecting this question were the aforementioned ambiguous operation labels, and also an overall depth or obscurity of some key operations in the tool.

From other functionalities, more experienced users were somewhat more satisfied with how intuitive the reasoning was in the tools. Value of χ^2 = 1.95 **did not confirm significance** of this variance. Yet this may not be very surprising finding, due to experts being more likely to have deeper understanding of OWL- and DL-level consequences and expectations from reasoning.

4.4 Functionality-related findings

In the previous study, this section summarized participants' expectations with respect to various functionalities, which were associated with the individual measures reported on in section 4.3. The motivation of the section in the previous study was to assess how satisfied the users are with the status quo of the existing tools, and how they relate to a selection of potential amendments or improvements. Since this part of requirement gathering was done on a tool-independent basis, we decided not to include this part of analysis in the current study, as little new insight was expected to be obtained.

5. Qualitative analysis of findings

The purpose of this section is to consider relationships and correlations between different observations made during the user study. This section cannot be considered *explanatory* in terms of clearly identifying causes of findings from the previous sections; it is purely *exploratory*. 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 Exploring issues with user interaction and navigation

In Table 14 (overleaf) we summarize frequently observed mistakes, problems and confusing situations from the user interaction perspective. These observations were made by facilitators, 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) reported by the users. The data in Table 14 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”.

Compared to the previous study, when 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), this year’s observations are somewhat different. For instance, instead of 80% of users needing a hint about searching in the past, this time the frequency of this specific intervention from the facilitator dropped to 34%. This may be seen as search dialogs and options being implemented in a more robust and consistent manner in the tools studied in the current study, and thus, it is a step in the positive direction.

From the current observations it is also visible that fewer users were observed editing random or unwanted concepts (32% in the past reduced to current 12%), which may again be attributable to a higher level of tool robustness in terms of managing the focus (and maintaining or resetting if an ontology changes). On the other hand, the frequency of issues related to locating particular functions, buttons or commands has slightly risen (from 39% to 49%) – some contributions to this type of confusion were already discussed earlier (e.g. the obscurity of some switches or difference between saving and applying changes). Locating concepts (usually in hierarchies) remained an issue despite improved search facilities (37% of users affected this time, compared to 39% in the past). One aspect that seems to have affected this particular issue is whether the tool constrains itself to searching in the ‘active’ (i.e. currently showed) ontology as opposed to searching in the entire ontology network (i.e. currently showed ontology with all its imports).

Table 14. Observations of issues related to navigation

Observation	Frequency	% affected	Examples
Dialogs, buttons,… (confusion, inconsistency)	23x	39%	Buttons/icons after axioms misleading; Single/double clicks to select, edit, etc.
Locating import, search, edit, etc. buttons, widgets, options	20x	49%	Where is “edit” option?; Where is “equivalence” definition?; How to add “intersectionOf” axiom?
Locating the class in a hierarchy	20x	37%	Native classes 'hidden' under imported ones.; Losing working class while previewing another one.
Searching for the class (partial text search on labels)	19x	34%	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,…)	18x	56%	How to get in the edit mode?; Where is it alerting me about error?;
Visualization issues	11x	27%	RDF graphs only, not really ontologies.; Seeing equivalences ‘next to each other’
Working with incorrect concept (concept edited without explicitly selecting anything)	5x	12%	Are you changing/editing concept in the right ontology? Is it the right concept? (see also single vs. double clicks)

A functionality that caused a lot of trouble in the past was *importing an ontology*, which was explicitly highlighted by 43% of participants as one of the most frequently used. The reason for this was an absence or obscurity of the actual menu label corresponding to this operation. This confusion in menu labels was not observed in this year’s cohort – the currently studied tools made away with different types of importing ontologies, which substantially simplified the interaction. Also, dialogs like “Open ontology” automatically defaulted to OWL formats, without any additional specification or choice needed from the users. So, in this respect, we can conclude that our flagging of the incoherent use of the “Import” label may have helped to achieve this trivial fix.

One issue that sits between navigational and conceptual relates to the translation of a conceptual model of a restriction into DL-style formalism. Leaving at the moment the question of suitability and/or complexity of DL notation aside, a commonly occurring situation observed by the facilitator related to the user's confusion between the axiom editing options. In particular, both tools offered in principle three key facets of editing an axiom: (i) simple definition (e.g. sub-classing), (ii) simple restriction (e.g. cardinality), and (iii) more sophisticated axiom editing (e.g. manually or by means of composing operators, properties, etc.)

A navigational issue related to this feature is that users were somewhat randomly trying to figure out which of the three tabs (as summed up above) in the axiom definition window may actually apply to a particular task they embarked on. In Protégé, after initial familiarization, they often reverted to manually typing semi-formal statements and let the tool to check their validity. In Swoop's visual user interface, this issue was harder to resolve, and participants opted for a trial and error approach to see if what they need, can be done in a given tab.

While we appreciate conceptual correctness of treating the three approaches to axiom definition as producing alternative types of restrictions on concepts, more navigational difference should be considered. This may e.g. comprise re-labelling of tab titles to reflect whether a user is editing an axiom *defining* a sub-class relationship, as opposed to editing (say) an axiom *restricting* a property for a given class. Some tabs may appear in the former case, but not in the latter.

5.2 Issues of prior expertise with formalisms

Much has been written and said about making ontology engineering tools usable and accessible to the users unfamiliar with the description logic. Similarly, 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]. Among other findings, this study argues that use of logical vs. visual notation is often expertise-dependent, problem and context dependent and also dependent on individual preferences. As stated in the past, an implication of such generic studies would suggest the need to consider *multiple ways* for defining and editing axioms.

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. However, the tools still have a tendency to use a single style of interacting with ontologies, at the expense of alternatives. For example, Swoop tends to rely solely and directly on OWL constructs (such as "unionOf" or "subClassOf"). On the other hand, Protégé v.4 tends to express axioms in a semi-formal variant of DL constructs, e.g. DL intersection of a concept and a restriction on its property can be expressed in two alternative ways:

- [InformationForm equivalent to]
  intersectionOf (PhysicalRealization,
  (allValuesFrom (realizes, InformationObject)))
    - PhysicalRealization <u>and</u> (realizes <u>only</u> InformationObject)
    - PhysicalRealization <u>that</u> (realizes <u>only</u> InformationObject)

The implications from this general knowledge in the visual programming community, combined with the responses of our participants, shows that neither native DL, nor OWL constructs, nor protégé syntax are generic enough to fit all needs. While DL and OWL-based interaction are often a preferred choice of *our* expert users, this is unlikely to be the case with experts and modellers in other domains, and with people who are less experienced in a particular formalism or style of conceptualizing their modelling ideas. The tools still lack any possibility for the user to opt for a different variant of expressing ontological commitments, and to do it in a dynamic, seamless way.

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 in the previous study and also in the previous sections, the gap between conceptual formulation of the activity and formal realization of the same in the tool was often too big.

Table 15. Observations of issues related to structural differences between the user and tool

Observation	Frequency	% affected	Examples
Carry out (translate, formulate, compose) logical operation (e.g. equivalence, inheritance)	31x	64%	How to start complex axiom?; Stepwise definition?; Is “same” equal to “equivalentTo”?
Incompatibilities between task and formal definition, errors in definitions	26x	57%	Property type/range elsewhere than cardinality; Wrong translation of “X except Z”
Syntactic check (brackets, logic,…) → user not alerted or not noticing	11x	26%	Buttons/icons after axioms misleading; Single/double clicks to select, edit, etc.
(Re-)definition of a class A in different branch (“double definition”)	8x	24%	Defining new sub-class for Y and labelling it X vs. re-defining existing X as “subClassOf” Y

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 syntactically incorrect axiomatization. In 57% of user sessions we observed at least one issue due to syntax when translating a correct conceptual idea into a given formalism (see Table 15). This number is comparable with last year’s 54%. More contrasting drop was observed in confusions related to lack of notification about syntactic mistakes in an axiom. While this was an issue in 64% of cases in the previous study, this time it was 26%. It is likely that a major role here plays the inclusion of syntactic checks directly into axiom editing windows. This, however, does not remove a related issue: underlined with red a term such as bracket in a definition tells little about *what can be wrong*; i.e. the feedback to the user is not complete.

Other differences in comparison with the previous study include an absence of major issues with namespaces in this year’s study – while internally this notion was retained, the presentation to the user treated all entities as ‘equal’ concepts (without any additional namespace qualifier, e.g. dolce:InformationObject). This simplification also addresses one of the gripes mentioned in the previous study. Similarly, confusions related to choosing which type of reasoning users need have diminished. Both tools are now much clearer about triggering this functionality. Indeed, Protégé v.4 retained a single option “Classify…” under menu “Reasoner”, and Swoop distinguished between reasoning, debugging, and testing ontologies.

5.4 Observation and feedback from newcomers

In this section we briefly summarize and comment on some of the feedback we obtained from one specific sub-group of user study participants – namely, staff members of a NeOn industrial partner (KIN). The reason we highlight this group in a separate section, away from a more generic division into experienced and less experienced, is twofold.

First, the user study with group of participants was coordinated with WP8 members in such a way that the results obtained may be used as a basis to identifying more obvious, urgent training needs and opportunities, which are planned as a part of WP8 effort. Hence, more detailed understanding of where people foundered was desirable. This has been carried out in terms of users interpreting the observations of the facilitator during the debriefing session. Second, after completing the tasks with a usual tool setup, this group was offered to replicate the same behaviour in the prototype implementation of the NeOn Toolkit 1.0[4]. Hence, there was a direct opportunity to obtain feedback from non-expert users of ontological tools vis-à-vis each other.

From the former perspective, several issues were identified, and we grouped them into several, more abstract types or categories: (i) general accessibility of the tool, (ii) various aspects and issues related to usability and user friendliness, (iii) capability of supporting user metaphors for interacting with knowledge models, and (iv) adoption of (knowledge modelling and DL) terminology and syntax during the tasks. We believe the participant cohort involved in this study is representative of potential users in commercial organizations; in this case they came from the financial department – i.e. the department likely to handle invoicing and invoice management, which was the theme of our tasks.

On the accessibility front, the users did not show major difficulties, although four times an issue of small size of either text or buttons arose, which had to be addressed by the facilitator pointing to the operation in question. An issue closely related to accessibility that appeared in every user's interaction was an occasional difficulty to find a particular operation on the screen. As we already reported for a general user setup, some key operations (e.g. switching to 'Edit' mode or searching) were placed according to some users in unusual places. An additional issue for this user group was an uncertainty about their activities – for instance, if they failed to locate a particular operation, their initial reaction was: "*Am I doing something wrong? I don't understand the task…*"

In terms of usability, frequent complaints were made about the view styles offered by the tools. In particular, ontology tree was difficult for two users, mainly because by default, the tree was collapsed; hence, they have not noticed any immediate effect of e.g. import operation. This, again, led to doubts about their understanding of the task. More on this subject is noted below, under our discussion of interesting metaphors used.

Features that were liked by users included, for instance, the auto-complete (Ctrl+Space) function (reported by three users on ten occasions), and the error notification function during the operation of editing axioms (reported by two users on six occasions). On the other hand, some features our users found confusing included, for instance:

(i) an unclear distinction between "a link" and "a record" in the definitions (here some of them got a bit lost after clicking on such item as class label in the restriction definition, which transferred them to the definition of that class, often deeply into DOLCE or EDIFact ontology)

(ii) lack of indication about which particular mode of interaction currently applies (to two users a choice of "double click" vs. "right click" looked somewhat random)

(iii) hierarchy as a default presentation technique, style were not very popular, at some point each user in this group complained about multiple occurrence of entries at different points of the hierarchy (which confused them, as they thought their previous action unintentionally removed a part of ontology – whereas in fact, they looked into a wrong branch)

(iv) one user could not get over the options in the import dialog – i.e. found it hard to relate them to "given ontologies"; the others admitted they kind of bypassed this step in the import dialog by trial and error, without knowing what other options meant

[4] To obtain the toolkit, visit http://www.neon-toolkit.org, "Download → Toolkit" section.

We mentioned above that with this group of users coming from non-IT background, we were quite interested in seeing how they adapt themselves to the visage and speech style of the tools. One reoccurring metaphor for a hierarchy considered ontological classes as *folders*. The view strongly resembled what they saw on their PCs. Another metaphor treated ontological properties as a form of hyperlink allowing one to move from a working class to another one. This particular notion was likely reinforced by the feature of the tools that actually opened the definition of the target class, e.g. for the *domainOf* or *involves* properties. Otherwise people thought it too abstract.

Overall, this group of people preferred interacting in a graphical mode – only one user (with IT background) opted for typing the definitions of axioms rather than picking them from menus and combo boxes. However, what people were often unclear with, was the terminology of knowledge modelling and DL. For instance, they grasped the notion of class – sub-class relation, but then tried to apply it also to sub-properties, and it was hard to explain the notion of "sub-property". Also terms like domains, ranges, or object properties did not trigger any immediate response from three users. In such cases, the facilitator had to explain the key principles of a particular term and relate it to the domain they were familiar with (e.g. in finances it was an idea of master budget with smaller budgets for each cost centre).

Also the notions of "super-" and "sibling …" caused some difficulties, which are to some extent attributable to the fact that for our users English was a second or often third language. Yet, even taking multilinguality in account, the problem was to some extent exacerbated by the use of slightly different terminology in the task descriptions ("*find concept X*") and in the labels on screen ("*Class view*").

In terms of commenting on the prototype implementation of NeOn Toolkit, the users were overall positive. They particularly stressed the fact that NeOn Toolkit did not ask them to choose a tab from among "obscure" labels, as one said: "*… all was in one hand, one view or max two…*" (here, a view is corresponding to a tab). Also, the graphical capabilities of NeOn Toolkit were highlighted; especially in defining more complex mapping axioms. Two people stated they learned NeOn Toolkit faster than Protégé, but that might not be indicative, as Protégé was an entirely new tool for them, whereas NeOn Toolkit was shown to them after Protégé, in a more relaxed atmosphere.

When people were told that this is a new tool that is currently under development, almost unilateral response was, whether it would be developed also in Spanish (rather than English GUI). This might be something for the developers to consider.

5.5 Other aspects of observational user study

In the previous study we introduced this work in terms of user observation for the purposes of requirements gathering and gaps identification. We highlighted three factors of such studies: (i) the *needs* (where the gaps are and how great they are), (ii) the *effectiveness* (where potential impact could be made), and (iii) the *evaluability* (what makes sense to (re-)evaluate). So far we focused on category (i) the gaps. Let us consider briefly the other two aspects also in this report.

In the previous study, we ventured into guessing which issues observed in our study might be among those making an impact on the user; i.e. what the user may notice if it is addressed. Among the factors identified in [5] were improvements in searching facilities, improvements and simplifications in axiom editing facilities, importing ontologies, locating entities in ontology networks or different browsing metaphors (trees, lists, entities in general, classes, etc.) Rather than repeating these observations in this study, we opted for presenting a delta in the perceived effectiveness or satisfaction with these aspects of user interaction. In other words, although we cannot claim that our previous study 'made' developers to address these issues, it was widely shared and discussed with the tool developing community. Hence, some of the issues we identified as low-hanging fruits indeed acted as motivators for prioritizing development of next generation tools.

Table 16 contains a list of most frequently executed activities and operations as reported by the participants. The numbers shown in the table in the '2006' column appeared in the previous study [5], those in the '2007' column were mentioned in the previous sections, but are adjusted here to reflect the responses only from the participants who gave a non-empty answer. These numbers are taken from the debriefing sessions with the participants that were carried out after the observational sessions and questionnaires

Table 16. Most frequent actions – where some impact was achieved?

Operation	% of users observed as having problems with	
	in 06/2006	in 11/2007
Modify ontological definition of concept, logical expression	71%	57%
Search for a concept, property,…	71%	34%
Define, introduce new axiom (e.g. concept, restriction,…)	54%	34%
Import, network ontology/file	50%	5%
Locate an item in the ontology, browse ontology	46%	37%

As can be seen above, in each of the currently observed dimensions there is some reduction in a frequency of problematic situations. From a conceptual point of view, it is encouraging to see that the modifications to the DL-like syntax in Protégé v.4 and probably also consistent and visually supported use of OWL constructs in Swoop made the operation of creating new axioms by a third simpler than in the previous tools (54% → 34%). Less reduction is seen in modifying existing definitions, but this may be something to do with the need to interpret an expression that needs amendment prior to carrying out the actual modification.

Even more striking is the reduction of problems related to importing and networking ontologies. While it used to be a common issue in 2006, the removals of redundant uses of labels like "open ontology" and "import" helped to bring the frequency of this operation being a source of confusion and dissatisfaction to 5%. Similarly, search was causing dissatisfaction due to its inconsistent use. Commitment of the currently studied tools to a single style search dialog led to reduction from 71% to 34% – and a majority of the remaining confusions was not due to search inconsistency but due to missing or omitting to set a particular parameter (in case of Swoop this was the ontology to be searched).

Smaller but nonetheless, positive improvement has been achieved in locating and browsing items in the ontological hierarchies. From 46% this went down to 37%. To some extent this is partly due to simplified search, partly due to the possibility to opt for a class view as opposed to a more generic entity view, and partly due to a more consistent treatment of namespaces and their inclusion in the ontology presentation to the user.

Where can we see further opportunities for improvements? To some extent, the idea explored by Swoop of restricting search to a specific sub-space of a complex model is interesting. It is close to our earlier proposition of a *constrained – 'contextualized' – search* (incl. searching for an item solely in the scope of a particular branch, particular neighbourhood, etc.) Maybe it would be worth developers' effort to allow for a more explicit and customizable formation of search dialogs, so that a user may decide how much additional search parameters s/he wants to or can provide.

With regards to axiomatizing conceptual commitments, there is still some scope in deploying wizards, which were already mentioned in our previous study. A potential extension to consider 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. In addition to more complex ontology pattern wizards (e.g. as explored in the WonderWeb project [14]), also simpler wizards for re-formulating a particular axiom (e.g. DL → OWL → Protégé syntax → visual) may be helpful to further placate the users' attitudes to different formalisms. However, as we argued in the previous study, it might not be desirable to keep adding new and new wizards to the user interface to facilitate each and every specific transaction. So, here is a definite scope for further research.

Another aspect that achieved certain level of progress since the time of writing the previous study is the knowledge of ontological patterns. These might be associated not only with the structure of models and modelled problems, but also with parameters from the user's environment; e.g. user's participation in a particular group or community (what are typical conceptualizations, how other users did a similar thing), or a procedural context of the operation (when, during which task an operation is performed).

With regard to navigating through ontological structures, one inevitably gets to the point having an appropriate visualization framework that supports metaphors particular users are happier with. 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 and its content, rather than its formal structure. 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 navigation context.

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

This report (as well as the previous one [5] and the corresponding paper [4]) explored *some issues* with ontology engineering tools (particularly those working with OWL) that affect the appeal and adoption of otherwise successful (OWL) technology by the practitioners. As we showed in the previous sections, although many issues remain, they are not as resilient as we saw them last time. The effort was spent to make the ontological formalisms more accessible to users without a strong DL background, and some reductions in confusion and difficulty has been seen already in this study with reengineered version of Protégé.

The translation of a conceptual axiom into a DL-style formalism was a separate issue: still a number of users were observed to stumble during such definitions. As we already suggested, considering *multiple ways* for defining and editing axioms might be a way forward. We do not suggest inventing an obscure translation of DL or OWL notation into a controlled natural language. The suggestion is merely to allow for multiple choices in defining, writing or previewing axioms, and to open the choice to the user by means of some customization or personalization.

On the other hand, tools still provide little help with some obvious user-centric tasks – e.g. visualization: There are many plug-ins realizing different visualization techniques; yet most of those currently available in the tools are variations of the 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). Perhaps directions initially explored by e.g. Jambalaya in providing different views on the ontological neighbourhoods [6] would be worth re-visiting. 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 original outcome was full of boxes and overlapping edges.

An issue with the ontology engineering tools remains in (i) their relative complexity and (ii) their relatively closed nature; i.e. difficulties to further customize or adapt to a preferred style or form. Certain flexibility is embedded in the environments such as Protégé; however, this customization seems to be done on a rather abstract level: 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. A good example is the class vs. entity views. These are two complementary predefined views that can be already used, but any further customization or filtering of the content within these panes is not possible. 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[5]) reasoning tool, but one cannot *modify or filter the components of user interaction*.

6.2 Implications for NeOn

As already stated, there was some progress in the tool design, but 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 followed up a benchmark established in our 2006 study. In line with the conclusions from the past works [4], we included other OWL engineering tools (Swoop and Protégé v.4) this time. 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 web-based or face-to-face tutorial, but perhaps a practical deployment of so-far largely theoretical body of knowledge on ontological patterns [22]. The users tend to perform operations in a mechanical way, using the same or similar steps. Providing a "Show me how…" pattern and applying to a specific case may help users to adapt themselves to the tool. Alternatively, this could be seen as users adopting certain 'best practices' for doing tasks demanded by the tool. Hence, our first point concerns the often-repeated position about a tool being able to adapt to the user. While this may

[5] 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.

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 realized by means of applying simple content patterns.

From this recommendation follows another opinion that the designers of NeOn tools and techniques could take on board: namely, to take more interest in the "languages of users", not only in the obvious technological requirements (e.g. on ontology 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- and/or task-specific labels. One possible way of achieving this is the support for the functionality of wrapper modules that would carry out usual operations, e.g. discovery of ontologies in a repository, but daisy-chain them into a process meaningful for the user, e.g. partial import (or better inclusion) of a branch or node from an existing ontology.

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 [19], so-called controlled evaluations tend 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 of the existing tools, since our own NeOn toolkit is still under development with regard to the OWL fragment and OWL features that exist in the studied ontologies.

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 both our studies – if the task were chosen too narrowly, its capacity to generalize would be very limited. Similarly, if the task was too broad, it may bring in too many open ends into the user interaction, which makes the subsequent analysis nearly impossible. In our case, the determinant of the study breadth was the composition of the actual ontology integration tasks.

Although more effort was spent on designing tasks that would be non-trivial, yet understandable by a variety of users and executable within a short time span, there were still difficulties in conveying the substance of some task activities to the participants. First, the tasks were to some extent *ill-defined* [17]. This was deliberate because if we gave step-by-step instructions of what users should do, this would significantly weaken and bias our findings. Yet, the basic form and shape of the tasks chosen in the previous study proved to be appropriate, since it was possible to switch to an entirely different domain this time, while still maintaining task similarity for the purpose of comparison and analysis.

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 to any single, controlled feature. Typically, in the observational studies, there is a mesh of features that affect usability, user acceptance or satisfaction. A range of our questions was chosen so as to tease out responses to different aspects of the tool and consisted of a mix of toolkit-level and function-level questions.

Another reason for having some difficulties may be attributed to the language of 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 subtle hints in 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 "corresponding to" and "equivalent", or such statements as "constraining property X to members of a class A, except those that are known to be in class B". Particularly, the second kind of statements was considered by many as vague, yet this form of competency statement is fairly common in knowledge modelling. So again, maybe a match between linguistic patterns and ontological pattern structures might be a way out of this.

References

[1] ISO 13407, *User-centred design process for interactive systems*. 1998.
[2] Colman, A.M., *A Dictionary of Psychology*. 2001, Oxford: Oxford University press. 864.
[3] 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.
[4] Dzbor, M., Motta, E., Buil Aranda, C., Gomez-Perez, J.M., Goerlitz, O., Lewen, H. *Developing ontologies in OWL: An observational study*. Workshop on OWL: Experiences and Directions, November 2006, Georgia, US. (http://owl-workshop.man.ac.uk/acceptedLong/submission_30.pdf)
[5] Dzbor, M., Motta, E., Buil Aranda, C., Gomez-Perez, J.M., Goerlitz, O., Lewen, H. *Analysis of user needs, behaviours & requirements wrt interfaces and navigation of ontologies*. Deliverable report D4.1.1, NeOn Project Consortium, August 2006.
[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] Kalyanpur, A., Parsia, B., Sirin, E., *et al.*, *Debugging Unsatisfiable Classes in OWL Ontologies.* Journal of Web Semantics, 2005. **3**(4).
[10] Kirkpatrick, D.L., *Evaluating Training Programs: the Four Levels*. 1994, San Francisco: Berrett-Koehler Publishers. 289.
[11] Likert, R., *A technique for the measurement of attitudes.* Archives of Psychology, 1932. **140**: p. 5-55.
[12] Norman, D., *The Invisible Computer*. 1998, Cambridge, MA: MIT Press.
[13] 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.
[14] 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.

[15] 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.

[16] Shneiderman, B. and Plaisant, C., *Designing the User Interface: Strategies for effective human-computer interaction*. 4 ed. 2004: Addison-Wesley. 672.

[17] Simon, H.A., *The structure of ill-structured problems.* Artificial Intelligence, 1973. **4**: p.181-201.

[18] Storey, M.A., Lintern, R., Ernst, N.A.*, et al. Visualization and Protégé*. In *7th International Protégé Conference*. 2004. Maryland, US.

[19] Twidale, M., *Redressing the balance: the advantages of informal evaluation techniques for Intelligent Learning Environments*. 1993, University of Lancaster.

[20] Waterfeld, W., Weiten, M., Haase, P., Cunnigham, H., Dzbor, M., de Palma, R., Munoz Garcia, O. *Specification of NeOn reference architecture and NeOn APIs*. Deliverable report D6.2.1, NeOn Project Consortium, March 2007.

[21] Wang, Y., Haase, P., Rudolph, S., de Palma, R., Euzenat, J., d'Aquin, M. *Networked Ontology Model*. Deliverable report D1.1.1, NeOn Project Consortium, November 2006.

[22] Presutti, V., Gangemi, A. *Ontology Patterns*. Chapter 10 in Handbook of Ontologies, Staab, S. and Studer, R. (eds), Springer Verlag, The Netherlands (in press).

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 v.4 with FaCT++[6] classifier/reasoner [or Swoop Viewer/Editor]
- 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, nor 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 ... PharmaInnova Schema ontology (OWL)
This is a small-size ontology derived from a schema capturing the concepts related to a typical invoicing process that would need further refinement during the experiment
O-2 ... EDIFact Reference Schema ontology (OWL)
This is a fairly large ontology combining specific terminology related to EDIFACT interactions with various terms related to business operations (like transaction, payment, etc.)
O-3 ... DOLCE Lite ontology (OWL)
This is a large size upper ontology forming a backdrop for the EDIFact terms and containing many generic terms (like object, realizedBy, expresedBy, agency, etc.)

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

Task 1

1.1 Motivation

The PharmaInnova invoicing schema contains many useful terms for describing invoices (e.g. *EmittingCompany*, *InvoiceHeader*, etc.) However, these terms are currently very shallow, and to make them ontologically more complete, you need to integrate your initial invoicing schema with a rich reference model of invoicing (based on EDIFACT).

At the moment, you may notice that there is no concept related to product deliveries and purchases in the initial invoicing schema. The first task is about adding a new concept ("inno#ProductDelivery") and defining it in terms of the rich EDIFACT ontology.

1.2 Specification

- Start the toolkit and *load* the initial invoicing schema ontology
- Ensure the initial invoicing schema has *imported* the EDIFACT reference ontology
- Add a new concept into the initial invoicing schema ontology with name "ProductDelivery" into an appropriate place of the ontology, satisfying the following:
    - ProductDelivery *corresponds* (i.e. is equivalent) to a financial transaction from the EDIFACT ontology
    - Add a statement that ProductDelivery has to be expressed by an emitted invoice, which is a concept known in the EDIFACT ontology
- Next, you need to express that each product delivery has two specific relationships:
    - involvesEmitter … which can only be an instance of and emitting Company (this concept is defined in the PharmaInnova schema and you need to locate it)
    - involvesRecipient … which can only be an instance of and receiving Company (this concept is defined in the PharmaInnova schema and you need to locate it)
- Once the two new properties of concept ProductDelivery are created, link them to a more generic property of financial transactions in the EDIFACT ontology called "involves".

1.3 Participant's notes

……………………………………………………….. empty space …………………………………………

Task 2

2.1 Motivation

Now you conceptualized the notion of product deliveries. Next, it is common that each delivery has a specific destination or a specific delivery point. In this task, we will ask you to assume that your company wants to differentiate between deliveries made directly to end customers (e.g. pharmacies) and deliveries made to wholesalers.

However, before this can be accomplished, you need to update the definition of concept DeliveryPoint that exists in the PharmaInnova Schema ontology.

2.2 Specification

Your task is to extend the vague and incomplete existing definition of 'delivery point' in the initial PharmaInnova invoicing schema ontology by integrating it with more formal and deeper conceptualizations from the EDIFACT reference ontology. In particular, concepts 'Place', 'physical-location', and similarly are useful to start with:

- Convert an existing, simple concept "DeliveryPoint" into a defined concept,…
- …and define it in terms of the following definition:
  - a delivery point always corresponds (is equivalent) to a "Company" (which is a concept defined in the PharmaInnova schema and you need to locate it)
  - and furthermore, its physical location has to always be an instance of class "Place" (this class is also a known concept in PharmaInnova schema)
- Next, distinguish between a direct and mediated product delivery points; the former goes directly to the client, the latter must go to a warehouse:
  - Define a new concept "DeliveryPointDirect", so that it restricts the "physical location" property to any Place <u>except</u> Warehouse.
  - Define a new concept "DeliveryPointMediated", so that it restricts the "physical location" property only to elements being classified as Warehouse(s).
- Show the facilitator that the integrated and newly defined concepts are used appropriately and appear in correct places of the improved initial invoicing schema ontology.

2.3 Participant's notes

………………………………………………….. empty space ………………………………………

Task 3

3.1 Motivation

Having defined concepts for describing two types of delivery points and appropriate conditions, your next task is to distinguish two types of product deliveries, too.

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

Do the following integration steps in the current version of the initial PharmaInnova invoicing schema ontology, with which you were working in the two previous tasks:

- Locate concept “ProductDelivery” and its property called “generically dependent on” in the initial invoicing schema ontology.
  - You will see that this property is restricted to an abstract concept from the EDIFACT reference ontology, but this needs to be made more specific.
- Your task is to define two new concepts, namely “Mediated Product Delivery” and “Direct Product Delivery” using the following guidance:
  - Both new concepts should be also product deliveries (i.e. specializations of the concept called “ProductDelivery”)
  - Suggest a way to complete the definitions of the two concepts, so that they are sufficiently distinguishable by the property “generically dependent on” being either a “delivery point mediated” or “delivery point direct”.
- Next, express in your initial invoicing schema ontology that any “Product Delivery” has to be either direct or mediated.
  - In other words, ensure that the two newly added concepts do not overlap and completely define (cover) concept “Product Delivery”.

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/Swoop … 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. EDIFACT would automatically import all relevant DOLCE Lite files. PharmaInnova uses Dublin Core, but this is not needed for import, so the user can ignore the tool request to find a file with its definition.

Once in the tool, 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 (*)

Task 2: guidance for facilitators

Again, one needs to find the ontology in which the concepts like 'Transaction') 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 EDIFACT into the PharmaInnova ontology. This means going to 'Metadata' tab in Protégé and declaring a new import from an existing source.

When pre-viewing EDIFACT, people may notice that it also imports many DOLCE Lite ontologies. So, they may suggest having only one direct import in the PharmaInnova ontology and the rest would be imported indirectly through EDIFACT. In other words, we aim to recreate more 'network-like' character here.

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 EDIFACT and PharmaInnova ontology in order to verify the statements we gave them in the objectives.

Further amendments are needed for concept 'ProductDelivery'; in line with section 2.1.

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:

<u>Participant 0</u>

- *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: [……]'

<u>Participant 1</u>

- *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:

Protégé v.4	*Swoop*	*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?

PharmaInnova ontology	
EDIFACT ontology	
DOLCE Lite 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*[7]

[7] 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 colors 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 behavior 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 componentes 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 netoworked ontologies-compliant editor:

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

Versioning

E-5a. Do you find the support for creating and mantaining 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.

…………………………