



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”

D4.5.4 NeOn Toolkit plug-in for visualization and navigation in ontologies and ontology networks based on concept summarization and categorization

Deliverable Co-ordinator: Martin Dzbor (OU)

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

Contributors: Silvio Peroni, Enrico Motta and Mathieu d’Aquin (all OU)

In this deliverable we summarize the rationale, motivation, and functionality of the NeOn Toolkit plug-in for visualizing ontologies in the NeOn Toolkit based on the notion of ontology summaries and key concepts that are most likely to describe the topical focus of a given ontology. This approach offers an alternative to the usual top-down browsing of extensive ontological trees, and our preliminary studies show that people are more likely to comprehend the topicality of a given ontology from the key concept than they would be from the top-level classes.

Document Identifier:	NEON/2009/4.5.4/v1.0	Date due:	February 28 th , 2009
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	February 28 th , 2009
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:

<p>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</p>	<p>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</p>
<p>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</p>	<p>Software AG (SAG) Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com</p>
<p>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</p>	<p>Institut 'Jožef Stefan' (JSI) Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si</p>
<p>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</p>	<p>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</p>
<p>Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de</p>	<p>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</p>
<p>Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de</p>	<p>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</p>
<p>Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid, Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientalobo@atosorigin.com</p>	<p>Laboratorios KIN (KIN) C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es</p>

Work package participants

From the WP participants, only The Open University (OU) was involved in the implementation of the plug-in and elaboration of this document.

Change Log

Version	Date	Amended by	Changes
0.1	10-02-2009	Martin Dzbor	Initial setup, key principles, introduction
0.2	21-02-2009	Martin Dzbor	Details of implementation and user interaction
0.3	28-02-2009	Martin Dzbor	Limitations, use case motivations, user interaction
0.4	10-03-2009	Martin Dzbor	Discussion, further work, exec.summary
0.5	15-03-2009	Martin Dzbor	References, cleanup, corrections, for review
0.6	03-04-2009	Ricardo Melero	Review of the plug-in and documentation
1.0	09-04-2009	Martin Dzbor	Addressing review comments, finalization

Executive Summary

This report accompanies a software deliverable whose main purpose is to provide an alternative means for visualizing larger and conceptually deeper ontologies in a way inspired by a natural human approach to tackle large problems at different levels of analysis. These different levels of analysis are instantiated here in the form of (i) calculating conceptual summaries of loaded ontologies, (ii) enriching these with topological summaries, and (iii) providing a selection of interactive means to get a quick snapshot of what the ontology in question is about. These user interaction elements include, among others, visualization of ‘information value’ of different concepts summarizing the ontology, manual expansion and contraction of nodes to facilitate simple browsing and navigation in the ontology, conceptual and visual zooming, export to JPEG, etc.

The purpose of this deliverable is to present the rationale, motivation, and functionality of the NeOn Toolkit plug-in for visualizing ontologies in the NeOn Toolkit based on the notion of ontology summaries and key concepts that are most likely to describe the topical focus of a given ontology. This approach offers an alternative to the usual top-down browsing of extensive ontological trees, and our preliminary studies show that people are more likely to comprehend the topicality of a given ontology from the key concept than they would be from the top-level classes.

The report is also intended to act as an impromptu user reference guide explaining how to install, initialize, use and interact with the proposed plug-in.

Table of Contents

1. Introduction	6
1.1 Motivation – theoretical rationale	7
1.2 Motivation – use cases rationale	8
1.3 Scoping of the deliverable	9
2. Key principles of the approach	11
2.1 From key concepts to ontology summary	11
2.2 Beyond ontology summaries, towards ontology navigation	12
3. Implementation as a NeOn Toolkit plug-in	13
3.1 Plug-in installation	13
3.1.1 <i>Repository based installation</i>	13
3.1.2 <i>Manual installation</i>	13
3.1.3 <i>Installing core configuration files</i>	13
3.2 Initializing the OntoSumViz tab	14
3.3 Interacting with OntoSumViz tab	16
3.3.1 <i>User controls for ontology summary visualization</i>	17
3.3.2 <i>Navigating in ontology summaries</i>	19
3.3.3 <i>Additional navigational opportunities</i>	24
3.3.4 <i>Other user interaction functions</i>	25
4. Discussion and Conclusions	27
4.1 Technological points	27
4.2 Current limitations	28
6.2 Future extensions	29
References and bibliography	32

List of figures

Figure 1. A typical user-centred development spiral [1].....	8
Figure 2. Initializing OntoSumViz tab as an Eclipse View.....	14
Figure 3. View selection dialog in NTK.	15
Figure 4. Triggering OntoSumViz for a particular OWL ontology.....	15
Figure 5. Empty OntoSumViz tab waiting for the selection of a visualized ontology.....	16
Figure 6. Ontology summary with key concepts having the highest information value ('importance') shown.....	17
Figure 7. Example of a conceptual zoom from level 1 to include level 2.	18
Figure 8. Ontology summary showing key concept in a flat, tree-like view.....	18
Figure 9. Ontology summary conceptually zoomed to level 3 (with traditional zoom cut-out).	19

Figure 10. A possibility to hide a specific node from the summary.	20
Figure 11. Reduced ontology summary to alleviate the 'busy' screen from Figure 10.....	20
Figure 12. Exploring the topological neighbourhood of a key concept (superclasses).	21
Figure 13. Exploring the topological neighbourhood of a key concept (subclasses).	21
Figure 14. Ontology summary enriched with non-key concepts responding to user requests from Figure 12 and Figure 13 to expand concepts 'Record' and 'Instrument'.	22
Figure 15. Exploring the topology of key concept 'Agent' (sub- and super-classes).....	22
Figure 16. Singleton key concept 'Agent' with its actual topological neighbours.	23
Figure 17. Topological sub-branch of the ontology summarized in key concept 'Agent'.	23
Figure 18. The functionality of pane centering on concept 'Genre' with tooltip hints on additional interactions.	24
Figure 19. Activating the export of the ontology summary as JPEG image.	25
Figure 20. A sample JPEG image of the visualized ontology summary.	26
Figure 21. An early vision of using ontology summaries to drive topical viewpoints: a top-level summary (A) with a cut-out showing the conceptual zooming facility.	30

1. Introduction

Many past projects developing semantic tools 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.

Among the functionalities that capture the eye of most users, many existing tools provide reasonable editing facilities and some visualization/navigation support. However, as we studied in the two larger-scale studies of the main ontology engineering environments [4, 5, 6], this user-facing support is far from ideal. Let us highlight a few issues we identified in our previous studies in the area of visualization and navigation of large ontologies first:

- Studies before ours (e.g., [3, 9]) showed that facets like complex network graphs and the lengthy trees starting at the abstract levels as the primary visual elements, were generally found to be a poor metaphor for user needs, and only tended to add to the already steep learning curve for a non-expert user.
- Similarly, authors in [12] surveyed Protégé users about their ontology visualization experiences and observed that such tools are too complex and do not reflect users' models of what they would normally want to see in unfamiliar ontologies.
- Our own χ^2 test proved that e.g., the variance between Protégé and TopBraid was significant at $p=0.05$, where some inefficiencies in Protégé were attributed to the limited visualization and ontology navigation facilities.
- We also observed in [4, 5] that non-experts found themselves less efficient due to the lack of simple visualization and navigation support compared with experts, especially in terms of obtaining overviews of provided ontologies. Although the variance was high, the χ^2 test did not confirm its statistic significance at $p=0.05$.
- Our previous studies also showed that existing support for visualization was perceived mostly negatively, with up to 57% of users seeing it in the “not sufficient” category. Among the suggested improvements, 43% endorsed the idea of visualizing knowledge at the level of ontologies (e.g., their summaries, overviews, etc.)
- Ontology hierarchies in particular received their above-average share of negative comments in [4, 5]; consider the following quote: “*These hierarchies are killing me*” which sums up the attitude of 32% of participants seeing an issue with visual interaction.
- Generally, visual interaction was probably the area where participants had the strongest feelings and reactions in the debriefing sessions, with aspects like hiding of certain concepts or ‘diffing’ two ontologies being among the lacking features.

The above selection of observations from our previous user studies in WP4 motivated us to propose the idea of visualization and visual interaction based on *ontology summaries*. The reason why summaries caught our attention is simple – with the tools like Watson and the emphasis on the reuse of ontological resources, it is more likely the user comes across third-party ontologies, which s/he needs to make sense of, to quickly preview and decide whether they are worth downloading, importing, etc.

As argued by Peroni et al. [8], the current tool support for ontology development by reuse is rather limited. They illustrate this on the case of the Watson plug-in for the NeOn Toolkit (one of the purpose-built ontology reuse frameworks) – although Watson helps the user to locate entities on the Semantic Web and import them into an edited ontology, it provides little support for navigating and making sense of the ontologies in which these entities reside. Thus, in order to respond to the issues found in our previous studies and to tackle the new challenges identified in [8] we decided to contribute to the challenge of ontology understanding. That is, how to make a quick sense of the content and organization of an unknown ontology, in order to make decisions about its suitability for a specific ontology development project. A brief summary of the key concept identification is in chapter 2. This is followed by the functional description of the plug-in.

1.1 Motivation – theoretical rationale

We start by pointing to some generic themes from our previous reports on user studies, which actually motivated, informed and drove our work in the area of investigating human-ontology interactions in this, dedicated work package.

In the past, we have expanded the notion of HCI (human-computer interaction) to human-ontology interaction with the aim of better understanding (and supporting) the human user, 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 Norman [7], at a certain development stage successful applications are required to balance technology with user experience and functional features. We started this chapter by quoting several findings pointing to the contrary status of the existing visualization support.

Good user experience for non-technical users is often best achieved when the technology, for the purpose of this deliverable, *the general ontology graph*, is hidden from the users in its entire complexity [11]. Alternatively or at least, the systems and tools supporting the user should subscribe to the same or similar models as the user. Thus, a user engaged in an interaction with an ontology has a *task model* that reflects the user's subjective understanding and expectations about the activities that need to be performed to reach a goal. In our case, if the user wants to decide whether or not it is worthwhile opening, downloading or otherwise reusing ontology, would normally expect a kind of outline, summary or preview showing what it is about.

On the other hand, there is a user's *model of the system*, which 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. Normally, the user's model of (any) visualization technique specifically built for large ontologies would include the expectation of showing partial visualizations or visualizations that are both flexible and guiding in terms of starting points.

The two models are often implicit [7, 11]; 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. Comparing the two previous paragraphs and linking them to the bullet points with the analysis of shortcomings in the existing tools, we can conclude that the user interface of the existing tools (with respect to the visualization techniques) reflects the view of system designers and knowledge modelling theoreticians, rather than the view and the need of the actual users. This bias of the existing techniques leaves the ordinary 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. Also, it leaves the user to emulate, second-guess or approximate many operations they would normally consider in their own task model.

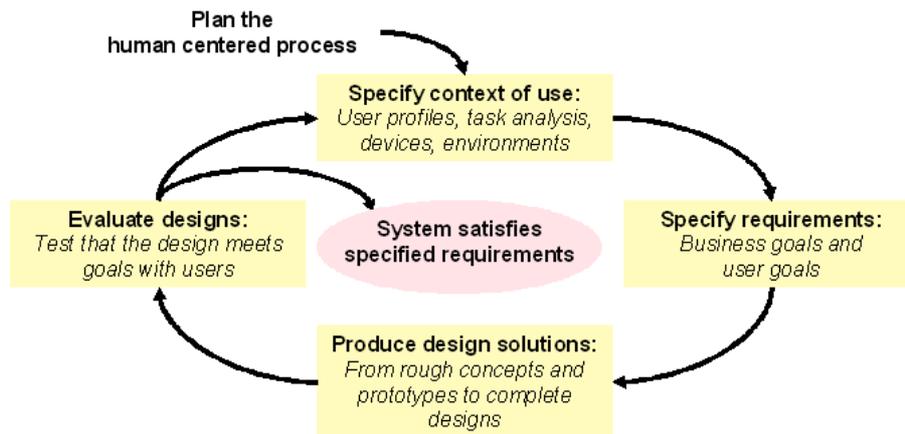


Figure 1. A typical user-centred development spiral [1]

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 13407¹. 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.

Many technology-driven models for problem solving, such as computational models [10, 13] 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 bringing in the visual interaction and navigation to the NeOn Toolkit that at least in some directions reflects the subsets of our users' task models is a valuable input to improving our position in how people use ontology engineering tools. Moreover, we believe that this is also one way how our theoretical propositions motivating our user studies – an the entire work package 4, back in 2006 – may be found an outlet and instantiated with a particular response.

1.2 Motivation – use cases rationale

In early phases of the project, we analyzed user requirements coming from the two use cases – the pharmaceutical and agricultural domains. In the wide array of requirements, we would like to flag the following ones as directly motivating and bootstrapping the work relevant to this deliverable:

- Section 4.4.1 of deliverable D7.1.1 highlights the need to provide summaries and overviews for edited and validated ontologies. As a guidance, the authors of the requirement suggest that the functionality of expressing content summaries and ontology coverage, in addition to statistical and structural summaries, would be beneficial to reduce the complexity of ontologies. In the proposed plug-in we are addressing the need to reduce complexity by means of conceptual 'key concepts' and topological overviews that reduce the entire ontology into top N concepts with the highest information value and respective links. Also, the conceptual zoom allows the user to further reduce the complexity by unwrapping the content of the ontology in steps from middle outwards.

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

- Section 4.4.4 in D7.1.1 demands that ontologies and their fragments/modules should be visualized in different ways, depending on the task to be performed (e.g., editing vs. revision of the ontology, indexing, domain browsing, etc.), and the purpose and the preference of the editor/author. The following view modes should be allowed: diagram-like, indented tree, node by node, possibly together with parents and children. In the proposed plug-in we address the first and last suggestions, as the indented trees are standard means to show the content of ontologies.
- According to the same section, it should be possible to display more than one visualization type at the time (in individual windows) and switch among them. This is achieved by having our plug-in as an independent Eclipse View that can be opened alongside other visual views, e.g., Relationship Browser, Hierarchy Browser, and so on.
- Similarly, according to discussions with the authors of D8.1.1, it would be desirable to select two or more ontologies from the same domain, for example, and then to visualize both by means of summarizing them that allow for their quick comparison. This functionality is in principle possible, but not fully implemented in the current version of the plug-in. We plan to include it into the future release of the plug-in.
- Another requirement coming out of D8.1.1 concerns the possibility to browse not only hierarchical relationships (that is, *subClassOf*) but also show and browse other types of relations. While this requirement is highly relevant and interesting for ontology summaries, we decided to first implement the prototype without non-hierarchical relations – mainly because of the existing of the specialized Relationship Browser that is to some extent complementary to the proposed visualization mode.
- A new requirement has arisen from the recent work in WP1 on integrating components of NeOn Infrastructure into the Cupboard framework – visualization of ontology summaries in response to the user's search query is one of the planned features, and to this extent, the proposed plug-in has been developed in a modular fashion allowing carving out the visual component out of the NeOn Toolkit and porting it to Cupboard. The actual integrative work is in progress and is likely to be reported in one of the future revisions of the plug-in or of the Cupboard system. In the context of Cupboard, a particularly relevant feature is the option to create ontology summaries that include the notion of 'informative importance', say at the conceptual zoom level #3, during batch processing of the ontologies, and use the resulting JPEG pictures as quick previews that can accompany the search result sets.

1.3 Scoping of the deliverable

This report accompanies a software deliverable whose main purpose is to provide an alternative means for visualizing larger and conceptually deeper ontologies in a way inspired by a natural human approach to tackle large problems at different levels of analysis. These different levels of analysis are instantiated here in the form of (i) calculating conceptual summaries of loaded ontologies, (ii) enriching these with topological summaries, and (iii) providing a selection of interactive means to get a quick snapshot of what the ontology in question is about. These user interaction elements include, among others, visualization of 'information value' of different concepts summarizing the ontology, manual expansion and contraction of nodes to facilitate simple browsing and navigation in the ontology, conceptual and visual zooming, export to JPEG, etc.

The purpose of the following chapters is to present the rationale, motivation, and functionality of the NeOn Toolkit plug-in for visualizing ontologies in the NeOn Toolkit based on the notion of ontology summaries and key concepts that are most likely to describe the topical focus of a given ontology. This approach offers an alternative to the usual top-down browsing of extensive

ontological trees, and our preliminary studies show that people are more likely to comprehend the topicality of a given ontology from the key concept than they would be from the top-level classes.

The report, in particular chapter 3, is also intended to act as an impromptu user reference guide explaining how to install, initialize, use and interact with the proposed plug-in and some of its innovative, empirically grounded functionalities. Most of the grounding for this work comes from our earlier studies on shortcoming of ontology engineering support tools [4, 5, 6]. The content of chapter 3 is also intended to be used as a basis for online documentation and also as a basis for the Eclipse/NTK Documentation accessible via the *Help* menu option.

2. Key principles of the approach

The work in this report and the associated plug-in extend the approach taken by Peroni et al. [8] that is based on identifying so-called *key concepts* in an ontology, to generate a meaningful snapshot of an ontology and facilitate the process of ontology understanding. The notion of key concept is highly subjective, but one easily follows the rationale of choosing n ‘best descriptors’ of the content of a particular ontology, where n is a number smaller than the total number of concepts defined in the ontology. The motivation of key concept identification was, to some extent, to reflect the role of so-called natural categories of the cognitive science.

While specific details on the transition from the abstract notion of ‘natural categories’ to specific ontology measures can be found in [8], we summarize the main aspects used in our visual extension of the original algorithm. The two core cognitive measures in choosing key descriptors for an ontology are: (i) concept centrality and (ii) label simplicity. The former is common e.g., in social network analysis, where it indicates how many times a given node appears in all paths between the root(s) and leaves. The latter measure reflects the number of compound terms forming the key concept label. Whereas the algorithm maximizes the former measure, the latter measure is minimized (preferring simpler names to more complicated ones).

Two additional measures are considered in the identification of key concepts: (iii) concept density and (iv) concept coverage. The former reflects how richly (frequently) is a given candidate concept described in the ontology in terms of its individuals, sub-classes, etc. The latter measure carries the usual meaning, trying to maximize the number of concepts in the entire ontology that belong to the branches headed by a particular key concept candidate.

2.1 From key concepts to ontology summary

Whilst the key concepts on their own are a good approximation of the ontology content, they form essentially a list, a vector of labels. One can then choose a vector with five, ten, twenty or more items to reflect the breadth and granularity of coverage. In order to move to proper ontology summaries, we add to the labels appropriate topological data, on one hand side, and introduce the notion of “conceptual zooming” first time explored in D4.2.1 [2].

The introduction of topology is essentially about linking the currently selected key concepts taking in account the actual *isA* hierarchy of the visualized ontology. Three types of connections are calculated. In particular, (i) a direct specialization (*isA* or *subClassOf*) and (ii) a direct instantiation (*typeOf*) are the obvious types of links between the concepts in any ontological hierarchy. Since however, key concepts have tendency to be sparse (in order to maximize ontology coverage), it is rather rare direct *isA* links would be showing. Hence, we introduce the type of “*distant ancestor*” link that forms the third and most common visualized link between the concepts. Its simple definition is: A distant ancestor of concept C_0 is concept C_N , such that a chain of specialization axioms $C_0 <isA> C_1 <isA> C_2 \dots <isA> C_N$ exists in a given ontology, possibly within maximum k number of hops, where only C_1 and C_N are key concepts.”

The principle of visualizing not only direct, but also more distant *isA*-type relationships between the key concepts in the ontology, it is possible to convey *ontology summary as a union of content descriptors and reduced topological structure*. In this way we are presenting to the user more data without actually increasing the screen estate or diverging from the nature of visualized information.

Another innovation that is implemented in our summary-driven visualization is inspired by the notion of term (tag) popularity that is pretty familiar in the Web 2.0 application. The original idea from tagged weblogs increases the size of font for tag $T_i \in \{T_1, T_2, \dots, T_i, \dots, T_K\}$ if T_i is used more

frequently by the bloggers or taggers than other terms in the set. The size of a term thus indicates its popularity in the respective tag cloud, and often, this metaphor is pushed a bit by declaring tag T_i as being 'more important' than tags in smaller fonts. Nonetheless, the popularity of a tag in the Web 2.0 context is based on a statistical sample of users with a certain size that is making use of the respective tag. In the context of an ontology – often newly designed – such a socially derived popularity is not the best indicator for the purpose of giving the user a meaningful summary of what a particular ontology is about.

Nonetheless, the idea of associating some ontological measure with the size of visualized concepts is pretty much reusable. In our case we associate the sizing of the (key) concepts with the abstract notion of 'conceptual zooming'. Conceptual zooming can be easily explained using the map metaphor in any of the popular online mapping sites, such as <http://multimap.com>. The main principle is that the amount of data (e.g., cities or streets) shown in the map depends on the zoom level, and that in turn corresponds to the map scale. The same notion used with the key concept takes advantage of the subjective nature of how much does one need to describe an ontology. In particular, one can generate a sequence of, say, ten, twenty, thirty, etc. key concepts for the same ontology and assign them 'importance levels'. Since the concepts that show among the top ten, also show among top twenty, thirty, etc. one can consider them 'more important' – in the sense of being more likely to be used by human users in summarizing what a given ontology is about.

In our extension we thus emulate key concept importance by the information value it adds to the ontology summary. Once we calculate the importance for each concept per se, one can then introduce the action of conceptual zooming or zooming at the conceptual level – i.e., adding or removing key concepts (and the related direct or distant links) to the ontology summary – all this in addition to the usual zoom of showing closer details of an image. In order to cater for different tastes and to provide this information to users with a range of (dis-)abilities, we propose two ways of realizing the notion of adding importance to the ontology summaries. The first follows the metaphor from Web 2.0 and adjust the size of the key concept node based on its importance. The second adjusts the brightness/transparency of the corresponding key concept node based on its importance. One can use only transparency mode, or can add node sizing as needed or preferred.

2.2 Beyond ontology summaries, towards ontology navigation

There is a fine line where ontology preview ends and where ontology navigation starts. Majority of scalable ontology visualization toolkits, e.g., OntoViz or IsaViz focus on the visualization side and neglect the interactive component and navigation in the shown ontologies. In our visualization based on ontology summaries and the fact that summaries actually indicate to the user where some missing (i.e., *non-key*) concepts may be, we decided to include a simple interactive opportunity to unfold the depicted concepts and show their direct super-classes, their direct sub-classes, or all directly (hierarchically) linked neighbours. To indicate that some of the 'unfolded' concepts may not be a key for the purpose of ontology summary, we decided to indicate that information by a different colour and size of the expanded nodes/concepts.

Similarly, allowing the user to expand, i.e., to add new concepts to the visual summary of the ontology, one has to support a reverse operation to avoid overcrowding of the screen estate. In our visualization approach we realize this actually by providing two operations. First, the user can 'reset' the summary graph – that is to restore it solely to the key concepts up to the current level of conceptual zoom. Second, one can hide specific nodes by simply selecting them and choosing option 'hide' from the contextual menu.

3. Implementation as a NeOn Toolkit plug-in

OntoSumViz is a plug-in developed for the NeOn Toolkit (NTK) and using the Eclipse platform. This provides a seamless integration with other plug-ins and core functionalities of NTK, in particular ontology and project browser. OntoSumViz does not require any other, custom plug-ins to work at the moment; it only relies on the core NTK application and its plug-ins and data models. However, there is a distinct possibility to split in the future the key concept generator component from the visual component and bind them using an 'ontology summary visualization manager' component, thus producing three mutually linkable plug-ins rather than one. The primary rationale for such a re-design is the fact that the visualization widget is implemented with the loosest possible bridge to the Eclipse windowing toolkit (SWT) in the more usual Java Swing and Java AWT technology. Whilst this does not affect the performance of the plug-in in NTK, it allows including the ontology summary and/or visualization also in non-toolkit systems, especially in web applications, such as Watson, Cupboard and other infrastructural components. More on this opportunity is described in the future work.

3.1 Plug-in installation

There are two possibilities to include this plug-in in a specific instance of the NeOn Toolkit. The first and long-term supported option is to use the repository of NeOn Toolkit plug-ins. The second and, at the moment, the main installation method is manual.

3.1.1 Repository based installation

Ontology Summarization Visualization plug-in can be installed as any other plug-ins in Eclipse. First, one needs to select the OntoSumViz plug-in from the repository of available updates and new installs. To do this and eventually download the plug-in, one has to use option sequence from the NeOn Toolkit menu "*Help*" → "*Software updates*" → "*Find & install*". A confirmation of the connection to the plug-in repository may be needed, upon which the dialog shows the plug-ins available for download. In the category "Other" choose OntoSumViz and confirm.

If you were using the NeOn Toolkit before, you may be prompted to close it and restart it. If you re-open the NeOn Toolkit one should have access to the new OntoSumViz View and Perspective. This means of installing the plug-in is, at the time of concluding the report, subject to review and shall appear online in the near future.

3.1.2 Manual installation

To install it, one needs to simply copy the binary version of the OntoSumViz JAR (located in the "*plugins*" sub-folder) distributed in the archive downloadable from the URI below into the "*plugins*" folder of your NTK installation. Both binary and source versions are accessible at:

<http://www.neon-project.org/resources/2009/ontoSumViz>

3.1.3 Installing core configuration files

After unzipping the archive mentioned in section 3.1.2 you will notice the resulting folder contains a README file and an additional sub-folder titled "*KeyConceptCache*". It is critical that this extra folder is copied to your NTK Workspace, which is usually located in the following places:

- MS Windows: C:\Program Files\NeOn Toolkit\workspace
- Mac OS X: /Applications/NeOnToolkit/NeOnToolkit.app/Contents/MacOS/workspace

Alternative paths may exist in your instance of NTK, and in general, these workspace paths are fully under user control – one can point the workspace root to any preferred folder on your computer. Whatever the location of your NTK workspace root is, please, copy the entire folder titled “*KeyConceptsCache*” to that location.

This folder serves three purposes: First, it contains configuration files for the four different sub-algorithms calculating the key concepts for the submitted ontology. Without these configuration files, the key concepts and ontology summaries cannot be calculated and nothing can be visualized! Second, the folder contains cache for already analyzed concepts alongside their degree of importance – this local cache helps to accelerate the visualization, as not all concept labels need to be referred to online resources such as Yahoo! and Watson. A side effect of cache is that as the plug-in is used and the ontologies are developed by means of reusing previous, presumably at least once visualized ontologies, each new ontology can be processed faster and more efficiently. Third, in this location is also the default image store, where the plug-in exports the JPEG versions of the conceptual summaries.

3.2 Initializing the OntoSumViz tab

One of the first actions of the user is to activate the OntoSumViz view or perspective in the NTK user interface. To do this one has to go through “*Window*” → “*Open perspective*” → “*Other*” sequence of menu options. This opens the dialog in which one may then choose “*OntoSumViz*” and open a new perspective (tab) with the OntoSumViz view (shown in Figure 5).

Another way is to already have a perspective open, e.g., the OWL Perspective of NTK, and go through “*Window*” → “*Show view*” → “*Other*” sequence of menu options (Figure 2). This opens the dialog in Figure 3, and choosing “*OntoSumViz*” one opens an empty OntoSumViz view (Figure 5).

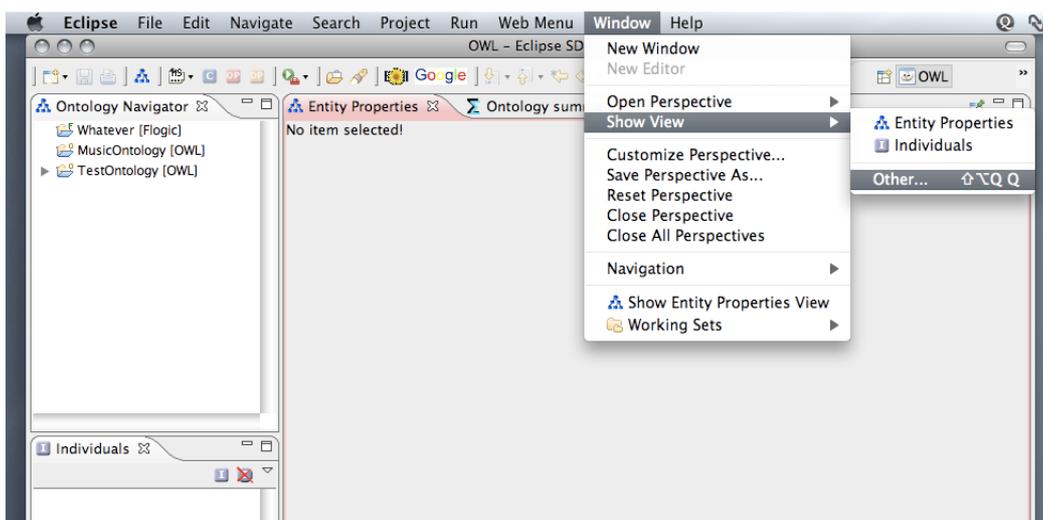


Figure 2. Initializing OntoSumViz tab as an Eclipse View.

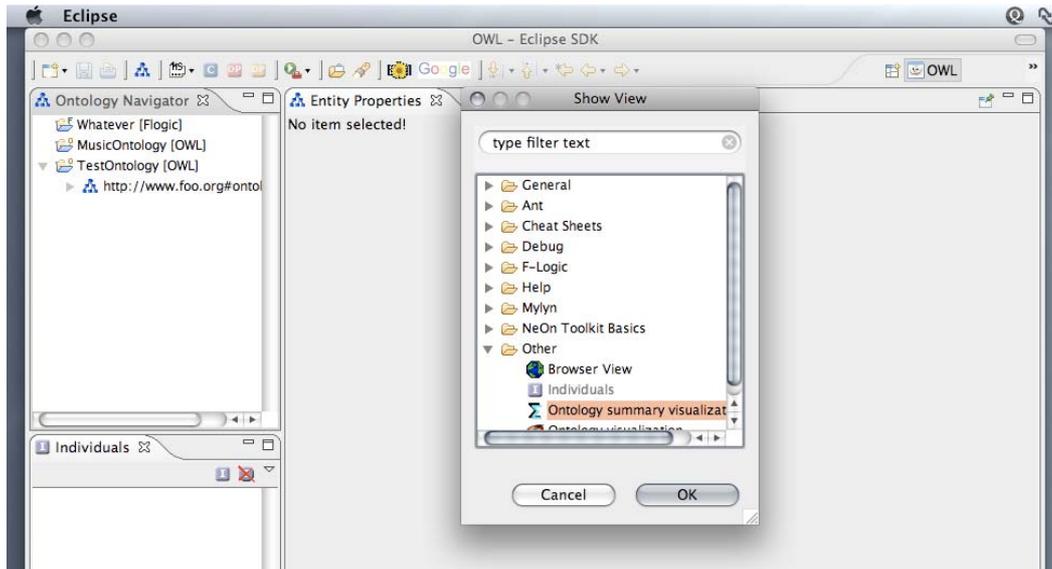


Figure 3. View selection dialog in NTK.

The third way to activate this plug-in is via a right-click with a mouse on the ontology loaded in the NTK workspace and showing on the left. As shown in Figure 4 one merely selects option “*Show ontology summary*”, which has the same effect as the above actions. Note that the plug-in actually works only for OWL ontologies, and at the moment will not process FLogic ontologies.

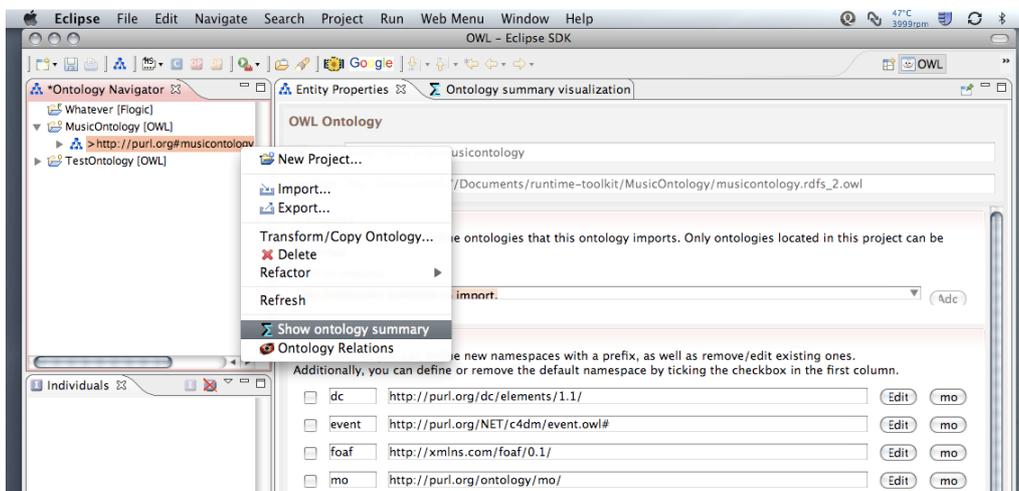


Figure 4. Triggering OntoSumViz for a particular OWL ontology.

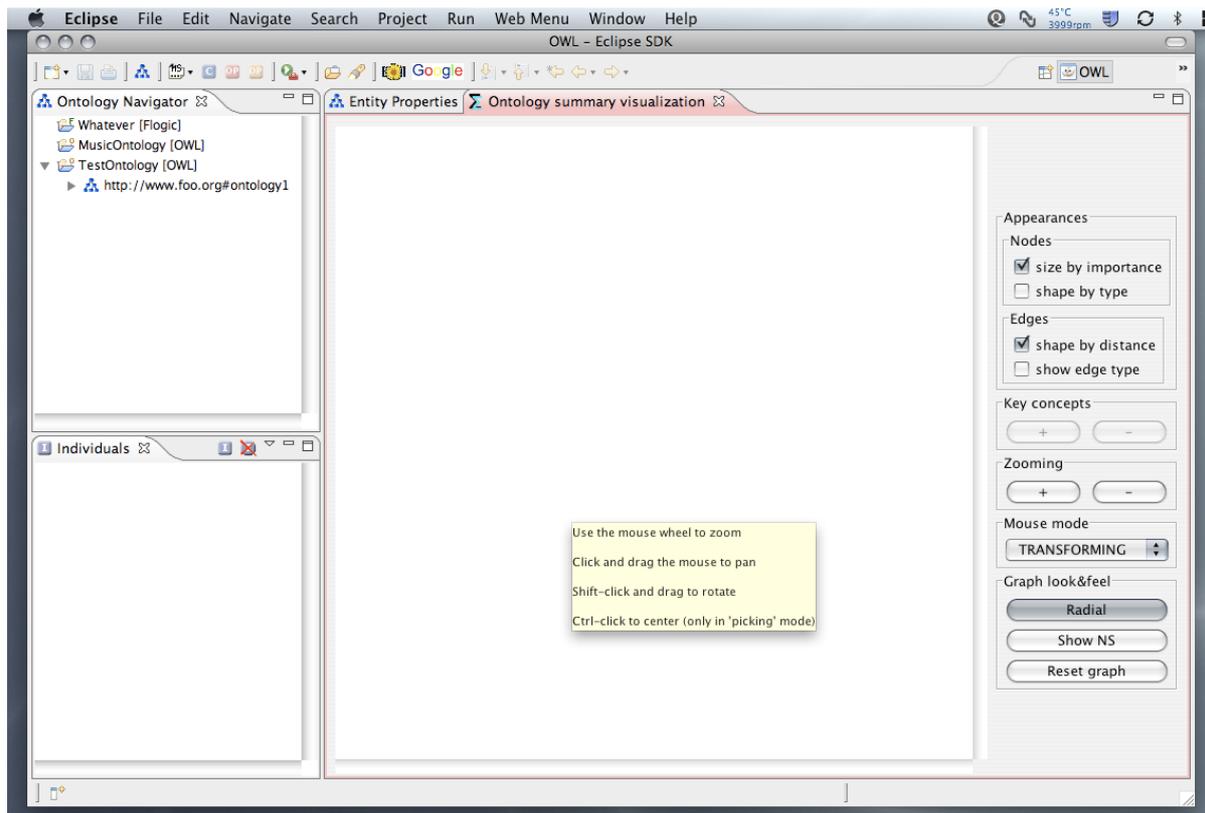


Figure 5. Empty OntoSumViz tab waiting for the selection of a visualized ontology.

3.3 Interacting with OntoSumViz tab

Once the ontology summary visualization tab is showing in the NTK environment, it needs to be fed a specific ontology from Ontology Navigator section of the NTK user interface. It is sufficient to click on an ontology listed on the left, and this ontology will be shown on the right. Before any ontology can be visualized, its key concepts need to be derived. This is done in the background and the results are stored in a local cache to accelerate future interaction. Thus, in most cases, the click on an ontology in the Navigator would read the key concept list from the cached file (as is the case in our example). The result of the feeding a new ontology to the toolkit is shown in Figure 6, in the shape of key concepts with the highest information value ('importance') being shown.

As one can see in Figure 6, the main difference between our visualization and the traditional view of ontologies is in the information value one can obtain from the left column (the Navigator) and the information value of the ontology summary on the right. From the left, it is hard to tell what the ontology is really about, whereas the view on the right summarizes the ontology in terms of its top n concepts in terms of information value. In Figure 6 one can also see the earlier-mentioned notion of including different types of relationships in the ontology summary (see the bold vs. dashed arrows), as well as the notion of adjusting the node size/shade by its importance.

Before showing additional capabilities of the ontology summary visualization tab, let us briefly say a few words about the user controls – visible in Figure 6 on the far right in a dedicated box.

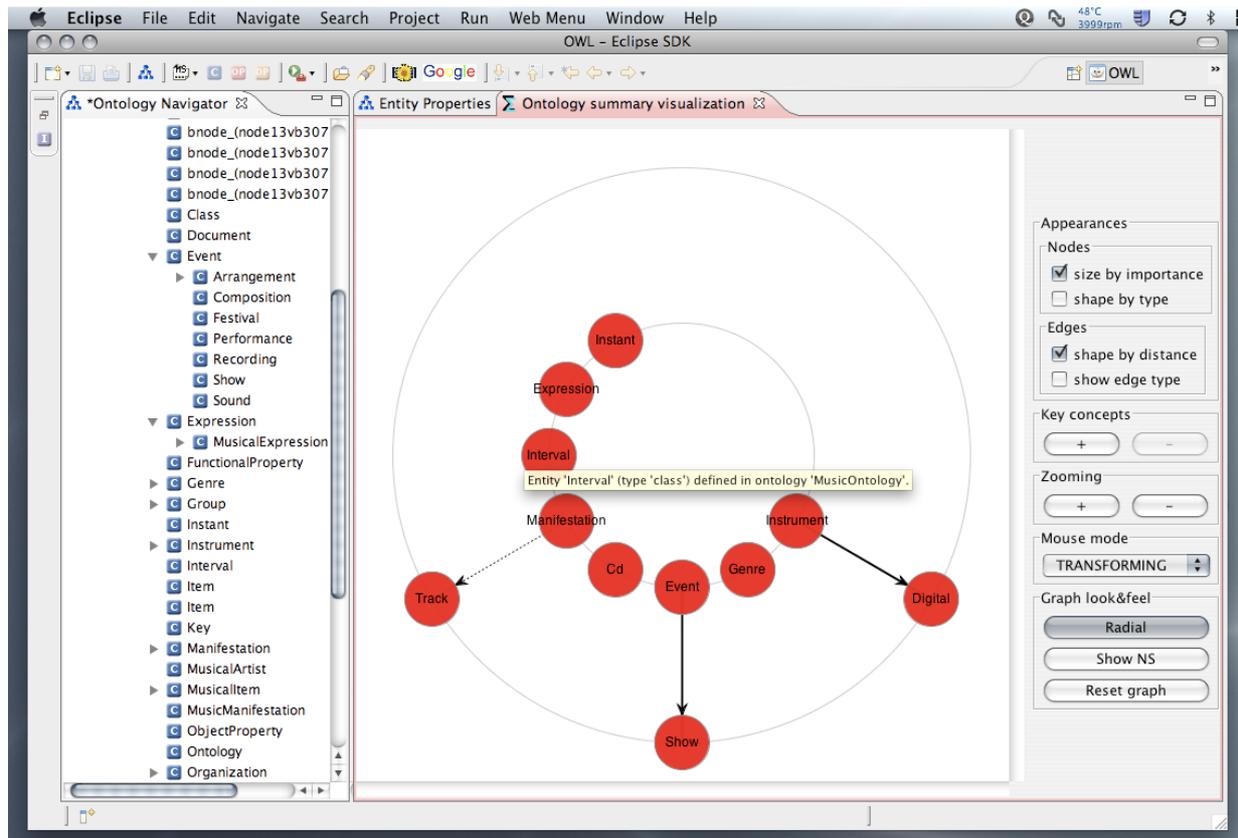


Figure 6. Ontology summary with key concepts having the highest information value ('importance') shown.

3.3.1 User controls for ontology summary visualization

User controls for ontology summary visualization appear on the far right hand side of the tab, in the form of a box with several categories. The five main control categories are: (i) generic look&feel of the visualized graphs, (ii) mouse function switch, (iii) traditional zooming, (iv) key concept expansion or so-called conceptual zooming, and (v) various other controls to modify the appearance of nodes and links. Let us start with the key functionalities – that is conceptual zooming and expansion of the key concepts.

As mentioned earlier, the point of conceptual zooming is to add to the visualized ontological summary key concepts of more or fewer levels of importance depending on whether the user interacts with the '+' button (= more details) or with the '-' button (= fewer details). There is a minimum level of conceptual zoom allowed, always corresponding to level 1 – one cannot reduce the number below level 1. There is also a maximum level of conceptual zoom, in our case set to level 3 – this limit has been chosen to keep our focus on visualizing *ontology summaries* rather than entire ontologies. The outcome of the user clicking on the '+' button in the 'Key concepts' box is the depiction of key concepts of levels 1 and 2 in the visual pane, as shown in Figure 7.

While showing the addition of the 'level 2' concepts, Figure 7 also illustrates the role of the check box controlling "(node) size by importance". In this case, all nodes are shown in the same shape and size; the only indicator of different information values of e.g., concepts 'Genre' and 'Rock' is a slightly lighter hue of red colour. The third functionality shown in Figure 7 is the use of traditional zoom to bring the entire image a bit closer, thus to get show neater gaps among the key concepts shown (mainly the region around 'Organization', 'Work' and 'Person').

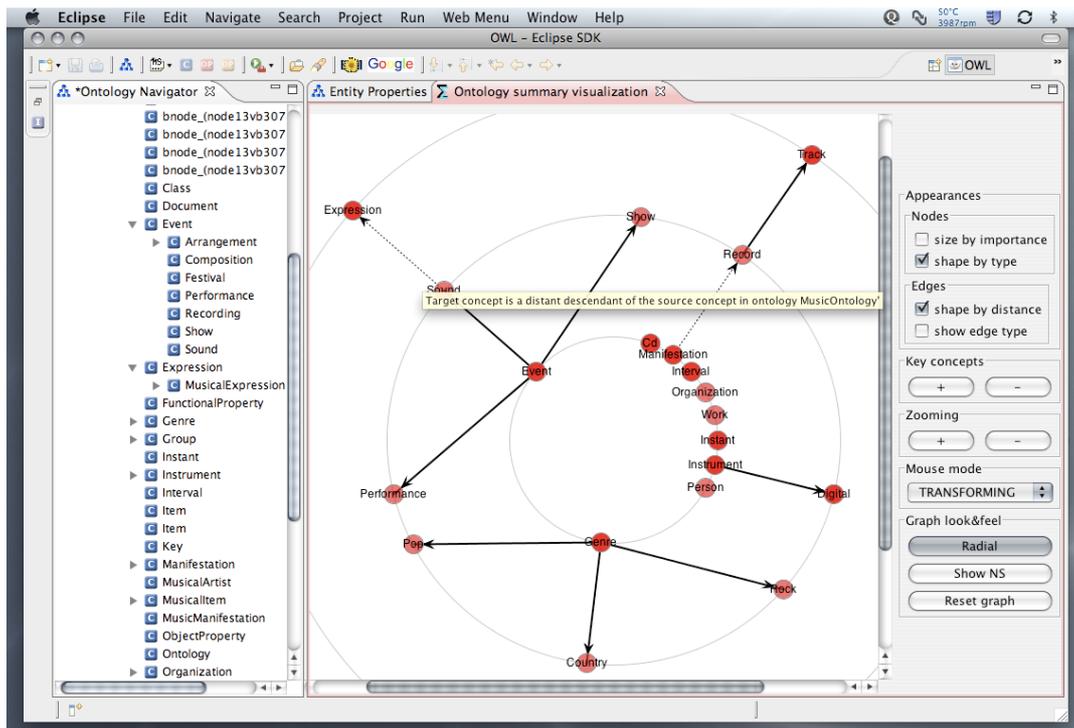


Figure 7. Example of a conceptual zoom from level 1 to include level 2.

The basic view for the ontology summary takes shape of the radial forest of partial sub-trees, which is illustrated in Figure 7. If the radial view becomes too busy, or one wants to take a more traditional tree-based view, the button labelled “Radial” acts as a toggle – switching between the radial and flat view of the same key concept forest. A sample non-radial view is in Figure 8.

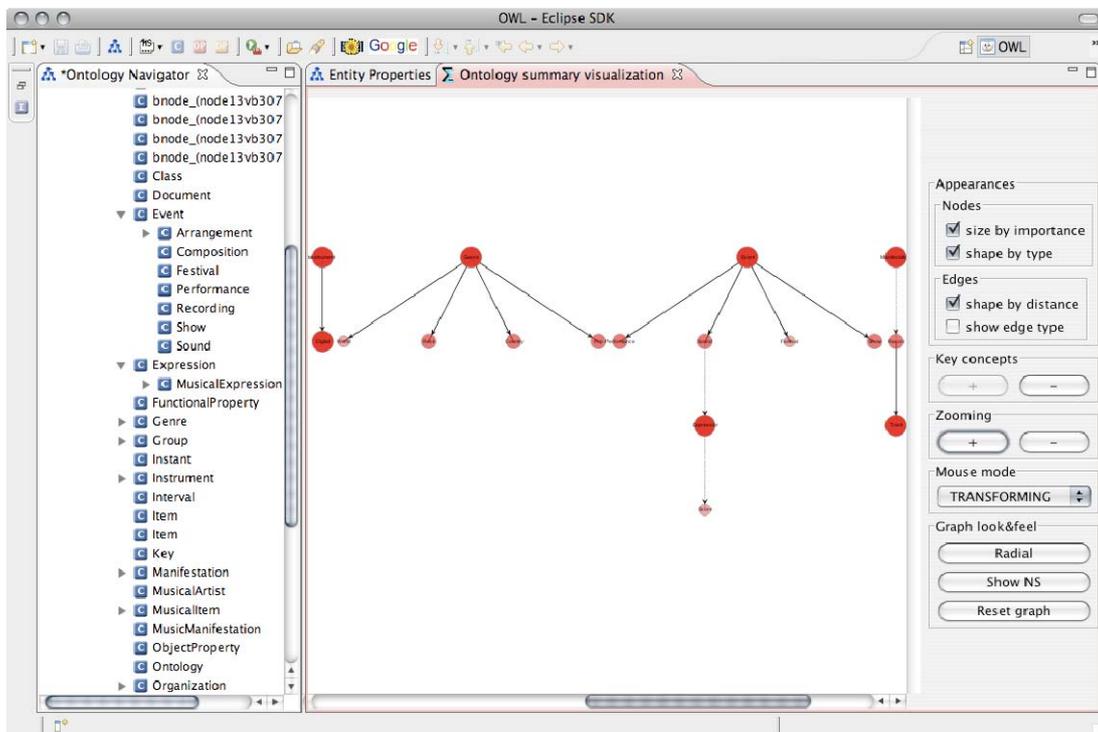


Figure 8. Ontology summary showing key concept in a flat, tree-like view.

In addition to tree view, Figure 8 also illustrates the function of the ‘-’ in the “Zooming” box. Whilst zooming out makes it hard to read the labels, one can obtain a better overall picture of the key concept sub-trees and their spans/coverage.

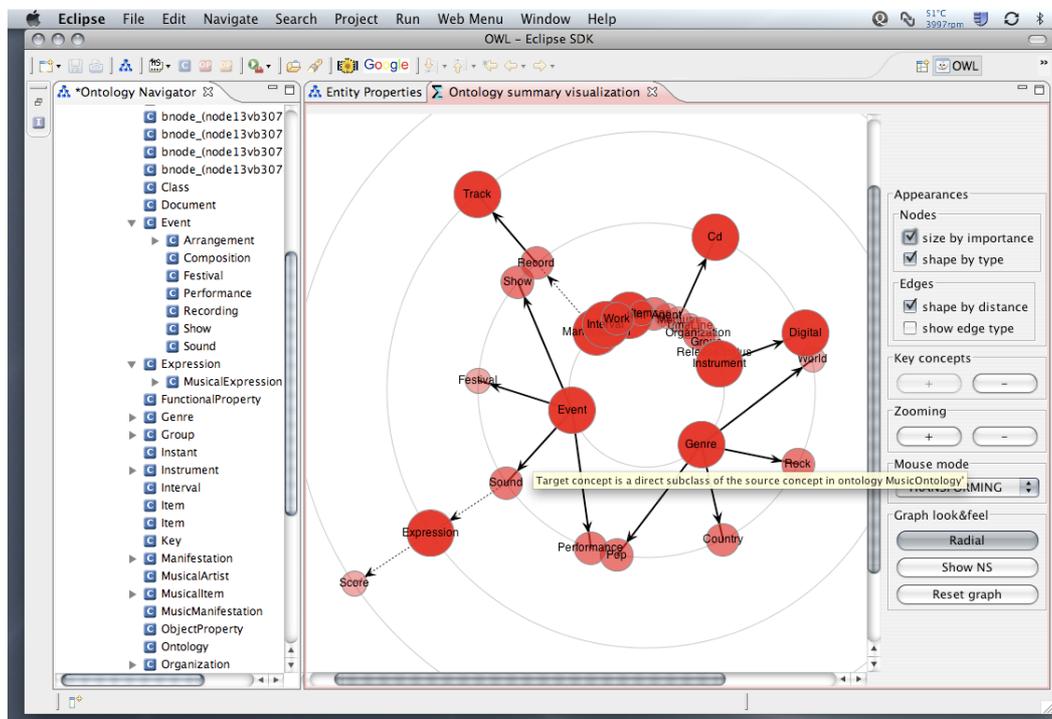


Figure 9. Ontology summary conceptually zoomed to level 3 (with traditional zoom cut-out).

Figure 8, alongside Figure 9, also illustrates in a better detail the use of node sizing and shading metaphors to indicate the information value of different key concepts. Thus, one can readily see that ‘Genre’, ‘Event’ and ‘Expression’ are more important for summary than e.g., ‘Sound’ or ‘Performance’, and those are more important than ‘Score’ or ‘Festival’. However, one side-effect of the sparse nature of key concepts is also seen in Figure 9 – there are quite a few singletons that get jammed so closely together that they make visual navigation hard. In the next section we explain how this shortcoming has been tackled.

3.3.2 Navigating in ontology summaries

First, to reduce the overall number of key concepts shown in the visual pane, one may opt to hide a specific node. Figure 10 shows that by right-clicking on a specific node (in this case, ‘Instant’) it is possible to remove it from the visualized ontology summary. Since we want to get rid of the singletons, we will repeat the “Hide this node” operation also for key concepts ‘Person’, ‘Group’, ‘ReleaseStatus’, ‘Organization’ and ‘Instant’, and show the outcome of all the hiding in Figure 11.

The next navigational action the user may wish to carry out is the exploration of the actual, topological neighbourhood for a specific key (and even non-key) concept. For example, key concept ‘Record’ is shown to be a direct super-class for another key concept ‘Track’ (the bold arrow), but is appears as a distant neighbour to concept ‘Manifestation’. Hence, one may want to actually “expand” or “unfold” the nature of that distant relationship, and find out what other non-key concepts actually lie on the path between ‘Record’ and ‘Manifestation’. This can be done by right-clicking e.g., on node ‘Record’ and choosing option “Show superclasses” (Figure 12).

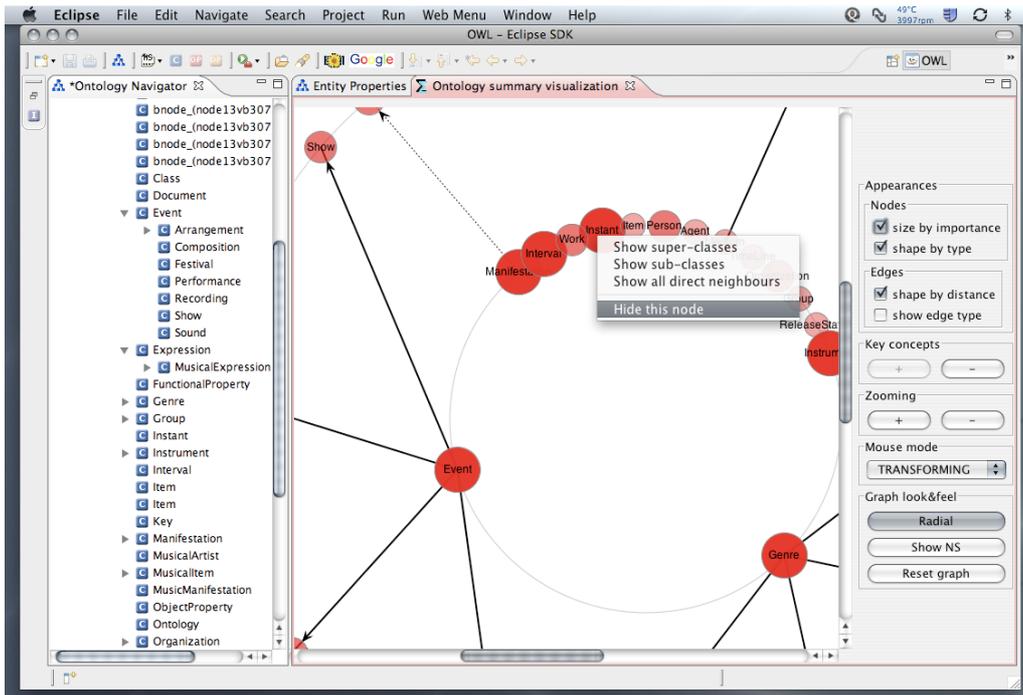


Figure 10. A possibility to hide a specific node from the summary.

Similarly, in Figure 9 we can notice key concept 'Instrument' being linked to another key concept 'Digital' that indicates a type of instrument. One may thus explore if there are any other instrument types defined in the summarized ontology – obviously, some of them being non-key, ordinary concepts. Right-clicking on node 'Instrument' and choosing option "Show subclasses" (Figure 13) enables this. The overall outcome of sub- and super-expansion is shown in Figure 14.

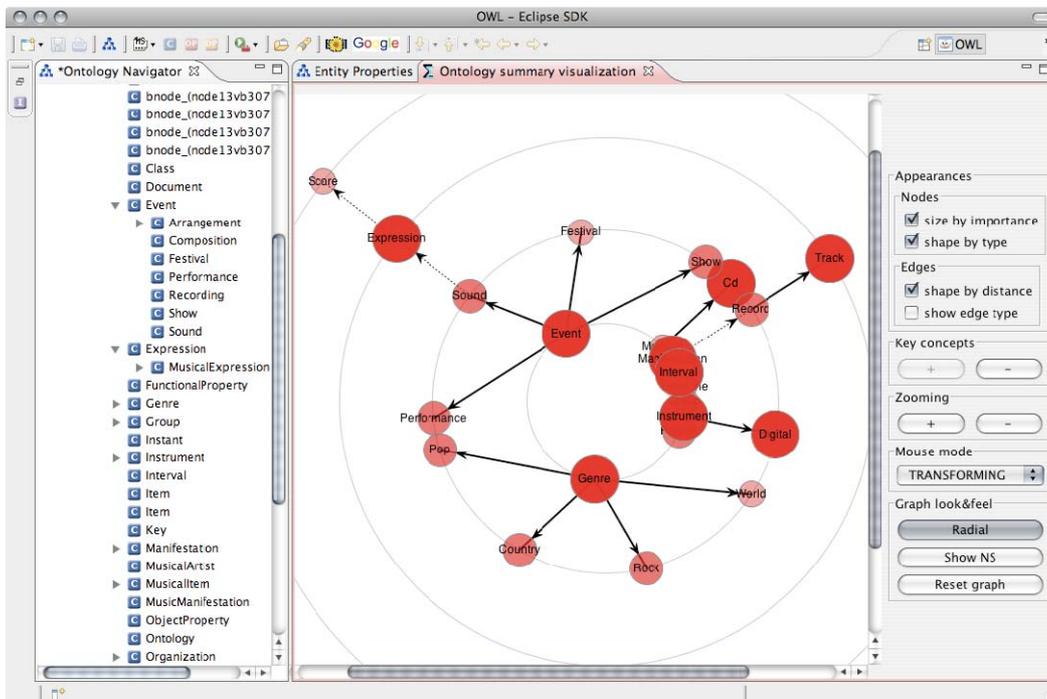


Figure 11. Reduced ontology summary to alleviate the 'busy' screen from Figure 10.

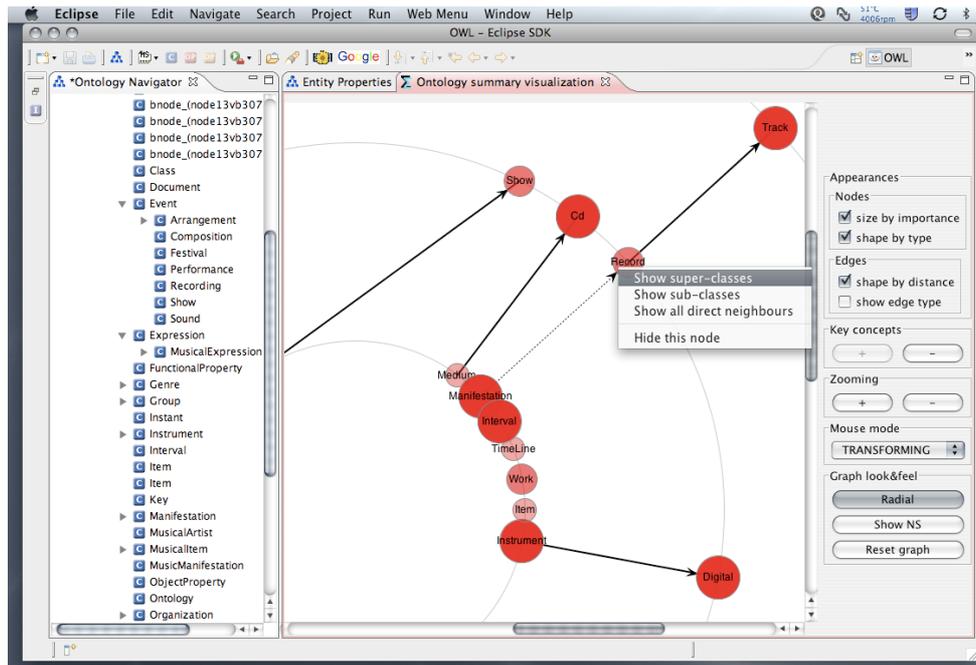


Figure 12. Exploring the topological neighbourhood of a key concept (superclasses).

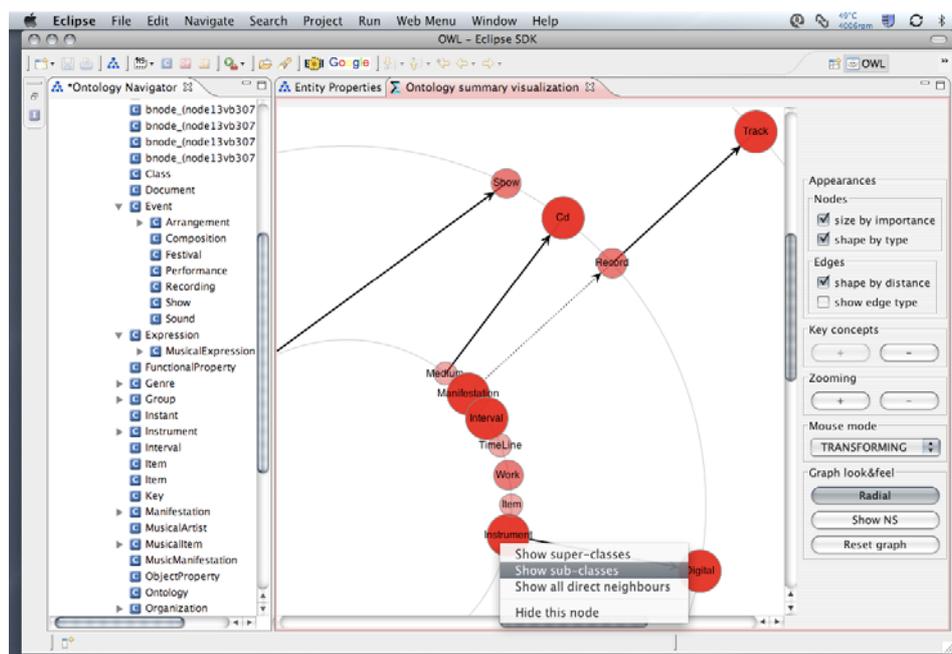


Figure 13. Exploring the topological neighbourhood of a key concept (subclasses).

In Figure 14, '*MusicalManifestation*' appeared as a direct super-class of '*Record*', and a few other instruments (e.g., '*String*' or '*Brass*') show as well. Note the grey shade of these additional nodes (non-key concepts) to distinguish them from the actual key concepts summarizing the ontology.

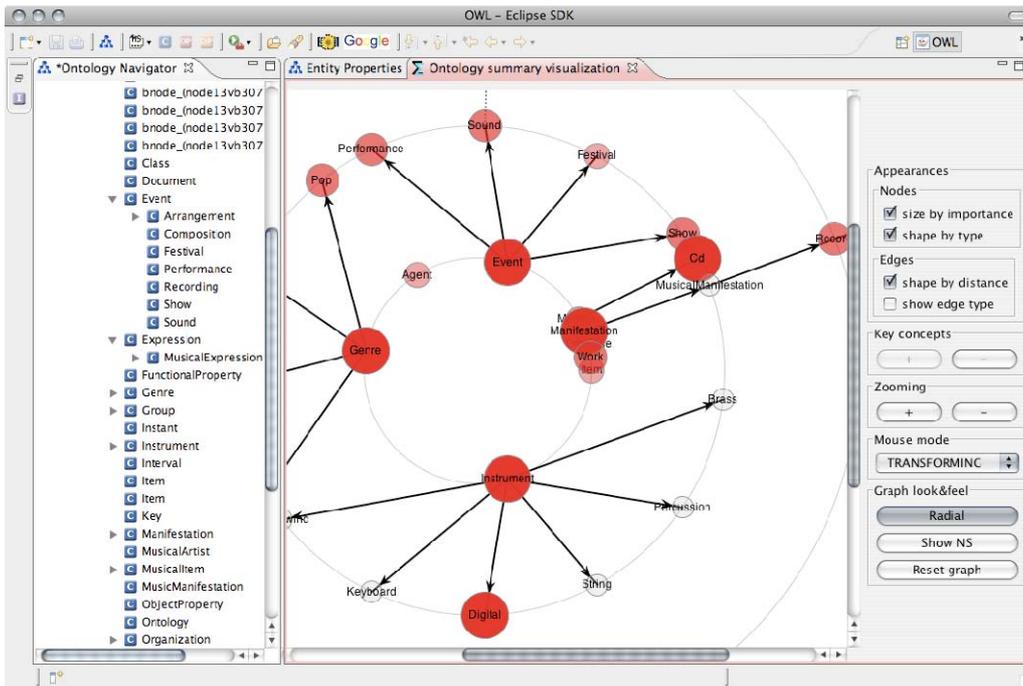


Figure 14. Ontology summary enriched with non-key concepts responding to user requests from Figure 12 and Figure 13 to expand concepts 'Record' and 'Instrument'.

Having thus exploded the ontology, one may execute the same operation also on the non-key concepts. Hence, as shown in Figure 15, we start with the singleton 'Agent' and explore both its specializations (sub-classes) and generalizations (super-classes) in one click by choosing option "Show all direct neighbours". The result is shown in Figure 16.

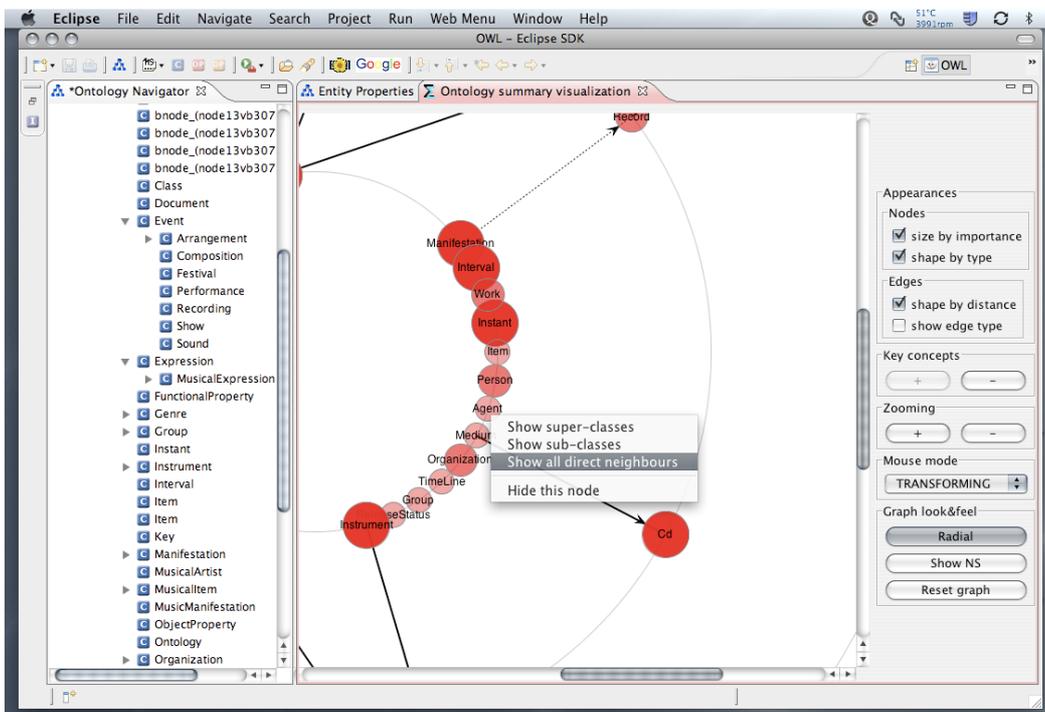


Figure 15. Exploring the topology of key concept 'Agent' (sub- and super-classes)

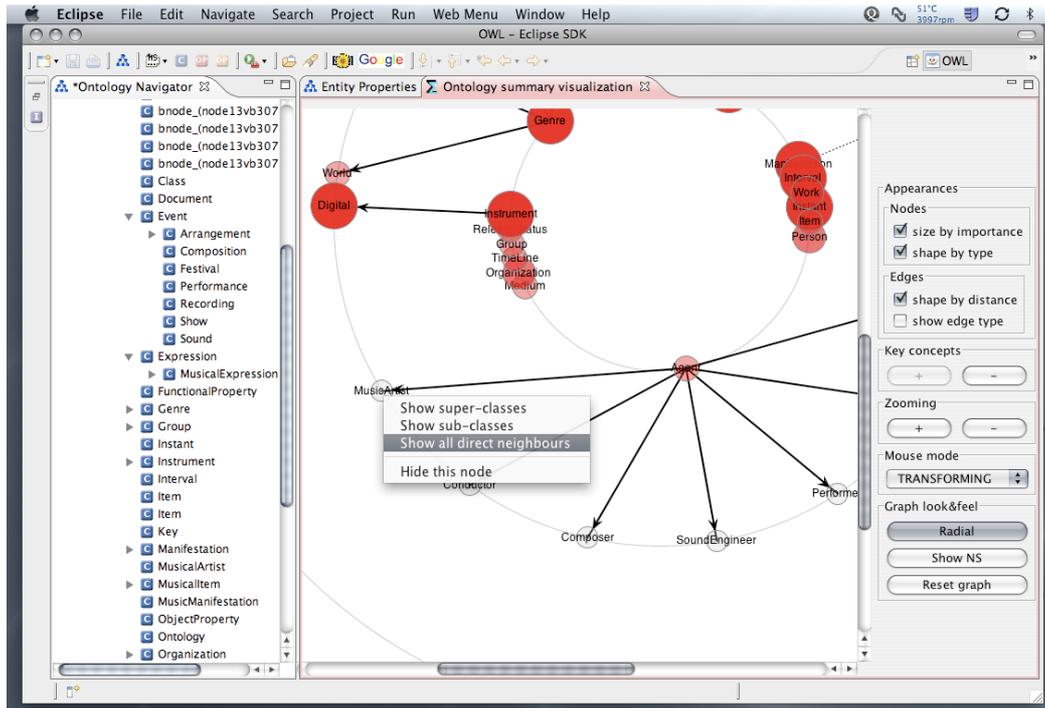


Figure 16. Singleton key concept ‘Agent’ with its actual topological neighbours.

While new nodes (e.g., ‘Composer’ or ‘MusicalArtist’) expand the previously singleton key concept, as shown in Figure 16, one may continue browsing/navigating in the emerging topological structure by invoking the “Show all...” action (say) for node ‘MusicalArtist’.

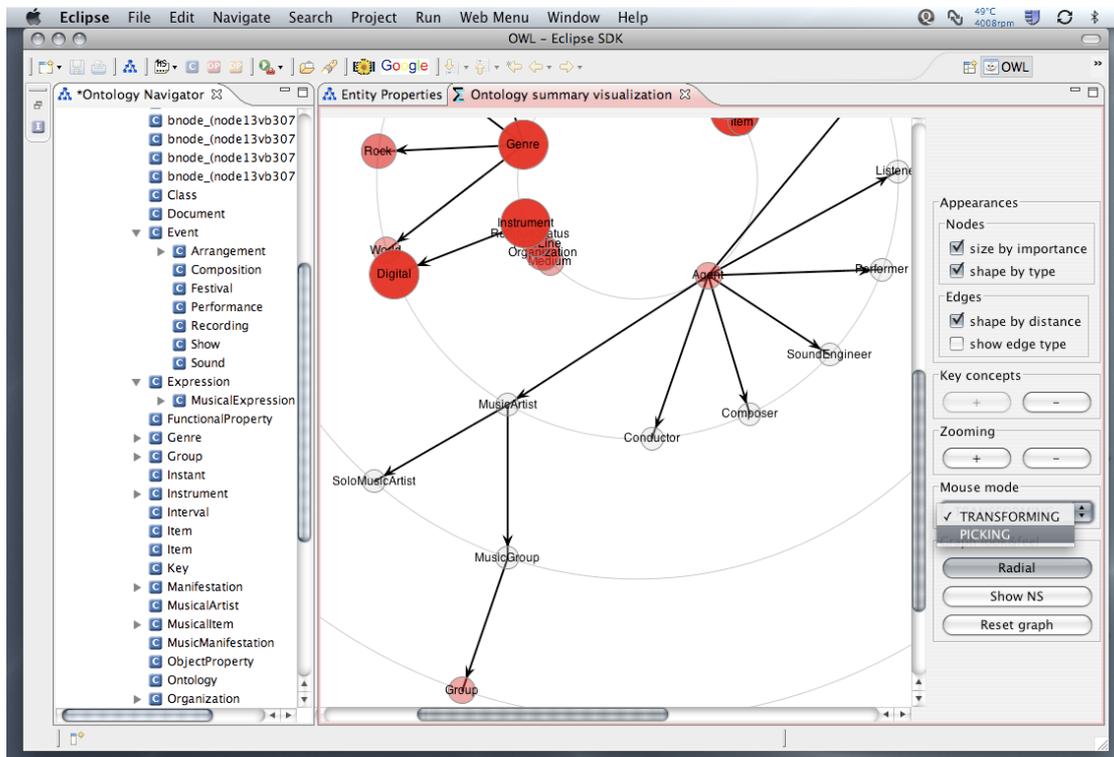


Figure 17. Topological sub-branch of the ontology summarized in key concept ‘Agent’.

The final outcome of the navigation by expansion of non-key concepts is shown in Figure 17. Interestingly, we found that upon this lengthier expansion, previously hidden key concept ‘*Group*’ re-appears – this time within the ‘*Agent*’ hierarchy and in the red hue rather than grey. Upon unfolding a node, one can obviously hide any of the showing key or non-key concepts as described earlier.

3.3.3 Additional navigational opportunities

Additional features currently implemented in the OntoSumViz plug-in include the possibility to adjust the view of the visual pane to better suit a particular distribution of nodes. In addition to aforementioned traditional zooming (see buttons labelled ‘+’ and ‘-’ in the “*Zooming*” box of e.g., Figure 17), three additional interactive aspects are available.

First, *click and drag* with the left mouse button corresponds to the global panning, i.e., shifting the entire visual pane to the left, right, top or bottom. For example, Figure 17 has been produced by panning the view to emphasize the topology of node ‘*Agent*’.

Second, *shift + click* with the left mouse button corresponds to the global rotation of the pane around the geometric centre (of the concentric circles). Again, Figure 17 also needed some rotation to get the entire sub-branch of node ‘*Agent*’ into one view.

The third interactive element is available if the user switches the mouse mode from ‘transforming’ to ‘picking’ (as shown on the right of Figure 17) in the “*Mouse mode*” box. The picking mode enables the user to left-click and *select* any shown node. The picked node will then change colour from red or grey to yellow. Once a node is selected, the combination of *ctrl + click* with the left mouse button acts as a request to re-center the visual pane so that the clicked-on node appears in the centre of the pane. This is illustrated by picking key concept ‘*Genre*’ that in Figure 17 shows towards the top of the pane. The *ctrl + click* re-positions the pane so that the picked node (‘*Genre*’ with yellow background) occupies the central spot of the pane, as shown in Figure 18.

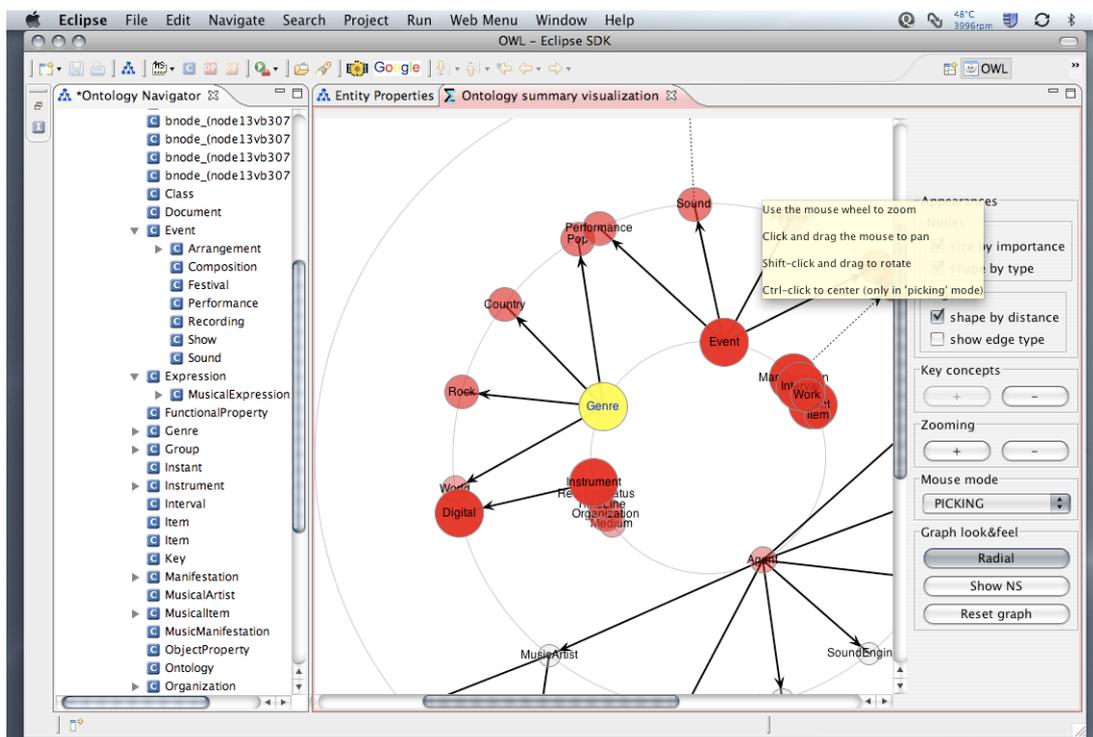


Figure 18. The functionality of pane centring on concept ‘*Genre*’ with tooltip hints on additional interactions.

As the tooltip in Figure 18 hints, one can also use the mouse wheel as a shortcut to zooming in and out – obviously, this applies to the ‘traditional’ zooming as explained earlier.

3.3.4 Other user interaction functions

One of frequently used functions of many modelling tools is the export of the resulting model into forms and shapes that can be included in reporting or documentation. To that extent, NTK plug-in for ontology summaries supports a simple export of the visualized conceptual and topological summary to JPEG format. The export is triggered by button labelled “Save as JPEG” (see pointer A in Figure 19). Upon activating this button a simple dialog (Figure 19, in the foreground) is shown to get a name of the file to export to.

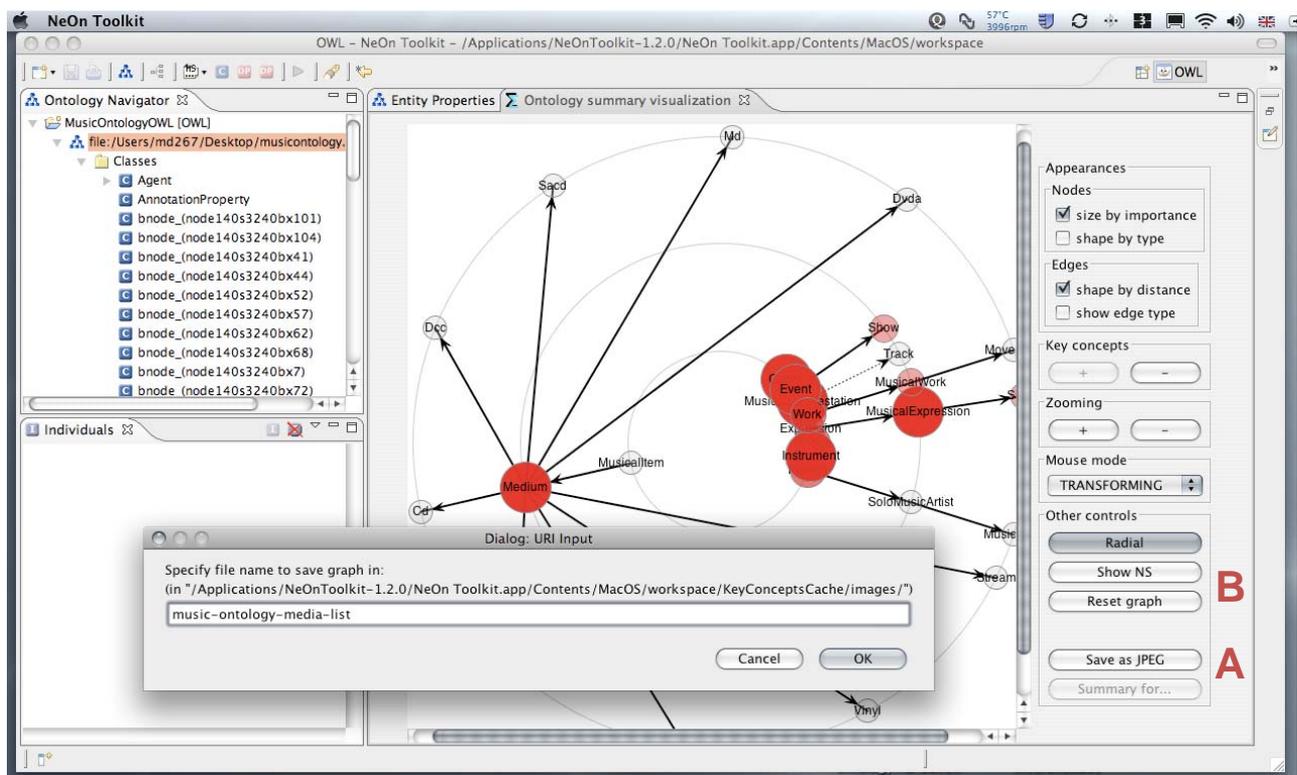


Figure 19. Activating the export of the ontology summary as JPEG image.

The results of the export process are stored in the “images” sub-folder of the “KeyConceptsCache” data store, which has been explained earlier, in section 3.1.3. Figure 20, overleaf, shows a sample JPEG image of one of the ontologies summarized and interacted with during the preparation of this report and during the tests of the plug-in.

Two other auxiliary functions are showed in Figure 19 by pointer B. Button labelled “Show NS” is self-explanatory – its purpose is to hide or show the fully qualified entity names, including their XML namespace. By default, namespaces are hidden to reduce the clutter on screen. The purpose of button labelled “Reset graph” is to return to the ‘vanilla’ summary of the ontology at a given level of conceptual zoom. As shown in Figure 19, after applying several ‘show neighbours’ and ‘hide node’ menu choices, the conceptual summary was enriched with non-key concepts. By clicking on button “Reset graph” the user may discard all shown/hidden nodes with a simple click and thus return to a ‘pure conceptual summary’ (visually, only red nodes would remain and all grey ones would be removed from the current view).

4. Discussion and Conclusions

In the report we gave motivation, rationale and explanation of several user interaction capabilities embedded in the NTK plug-in for visualizing ontologies based on their conceptual summaries. The core of the presented approach is in working with ontology summaries instead of top-down hierarchical trees (or forests). To this extent an algorithmic suite for identifying 'key concepts' was developed and extended by topological and navigational summaries. Concept layout for a given ontology established by the prevailing meaning of the concepts with respect to the Web as corpus (via Yahoo! web service search).

Different types of navigation from ontology summaries are enabled and described in this deliverable. First, we support standard topological 'node expansion' similar to the common trees – however, the key difference here is that instead of starting from the top-most (abstract) concept we expand from the centre outwards, from the conceptually top-most, most informative concepts.

Also realized in a conceptual navigation by means of zooming in. Two types of zoom are supported – standard visual enlargement and so-called conceptual zoom. As the level of conceptual zoom increases, range of key concepts included in the ontology summary is extended and thus, more details of an ontology fragment are shown on screen. Currently the plug-in supports three levels of zoom and visually shows the perceived 'importance', information value of a particular concept to the summary by adjusting the size and hue of the respective graph node.

4.1 Technological points

The plug-in as described in the previous chapters has been developed in line with the guidelines for NTK plug-in development, and thus relies on Java Virtual Machine v1.5.x. Whilst being developed in the Mac OS X environment, the current testing showed it was fully compatible with MS Windows environment. These results are encouraging as we managed in this development to resolve some of the issues with widgets like buttons or combo boxes not being presenting consistently in different operating systems. Technologically, the solution has been achieved by developing the plug-in's GUI in the common AWT/Swing windowing toolkits as opposed to Eclipse native SWT framework. Since AWT and SWT are not compatible, a bridging component was deployed to facilitate event pumping and proper mouse functionality. The bridge has been provided by the `org.eclipse.swt.widgets.SWT_AWT` object.

The visualization relies on several third-party libraries that are distributed in the archived JAR package. The libraries used can be divided into several categories:

- *Watson and web services support* ... needed for a part of the functionality of the key concept algorithm, whereby summaries can be calculated for ontologies known by their URI and residing in Watson. These libraries together with the Watson API are also available from the Watson home page (http://watson.kmi.open.ac.uk/WS_and_API.html).
- *Graph visualization* ... graph visualization is built on top of the JUNG suite (Java Universal Network/Graph library), which is an Open Source product supporting basic graph models, rendering algorithms, etc. More information on JUNG and updates can be found on the project web site (<http://jung.sourceforge.net>).
- *Other proprietary libraries* ... the third category of included files comprises a range of previously developed software modules supporting XML parsing, work with taxonomies, command line option processing, and similarly. These were contributed by the co-author of the work, Silvio Peroni (<http://www.essepuntato.it/blog>).

The plug-in described in the report has been developed based on Open Source branch of the NTK, which is available from <http://ontoware.org/projects/neon-toolkit>. The development was done with NTK v1.2.1, which was later upgraded to v1.2.2. The plug-in was also tested on both official releases of NTK – v1.2.1 and v1.2.2. Furthermore, the plug-in has been deployed into the Extended NTK alongside several non Open Source plug-ins and it did not exhibit any interference.

4.2 Current limitations

Due to dynamic nature of software development in the NeOn Project and the dual nature of several components, the plug-in in its current shape and form exhibits some limitations. The first and most obvious one is the dependency on the OWL Model used by NTK. This dependency has two corollaries: (i) the plug-in relies on KAON2 as the primary ontology and data store, and (ii) the plug-in thus only works with OWL ontologies maintained by KAON2 datamodel. There is also another possibility to use Watson as an alternative datamodel manager instead of KAON2, but since Watson is an online framework created by analyzing existing ontologies, it is not the most suitable means to provide data for visualizing newly built ontologies or ontologies still under construction and residing purely in user's computers. The key limitation of this approach is that the existing plug-in will only work for OWL ontology projects, but will not work with RDF ontology projects that are supported in NTK v1.2.2! In fact, the menu from Figure 4 is only visible for OWL ontologies (as KAON2 interprets them).

Another in-built issue we became aware of when scaling the tests is the latency for extra large ontologies (cf. CYC). Since the algorithms for calculating key concepts rely on the ranking and frequency of term occurrence in the Yahoo! document base, there is potentially a large number of Yahoo! service invocations needed to obtain a picture of term importance and information value. This sheer number of web calls may lead to Yahoo! rejecting any further requests, which in turn fails the algorithm. This issue is one of the more serious ones to be investigated in the near future.

Next limitation is due to how KAON2 maintains datamodel of a respective ontology. During the pass through an ontology, it may happen that 'meta-concepts' are returned by KAON2 (e.g., owl:Class or owl:ObjectProperty). Since these are numerous and occur almost in each and every ontology, they are likely to be included as "key concepts" summarizing a given ontology, which is conceptually wrong. After studying other approaches, it has been decided to filter out these 'meta-concepts' and treat them as labels defining the language rather than belonging to the ontology we want to visualize. In a similar manner, 'bnodes' were filtered out, as these correspond to unnamed entities defined by axioms/restrictions but otherwise carrying little explicit information value that can be gleaned from their name or label. At the moment, these filters are hard-wired, which might be a potential limitation – e.g., when visualizing ontologies describing OWL meta-model concepts like *Class*, *ObjectProperty*, etc. would be incorrectly excluded. This is a minor limitation, however, as it could be handled by a simple user-controlled switch or a preference.

At the time of writing this report, the functionality of visualizing web-based ontologies that are known to Watson engine is only partially supported, and it has been decided to disable it. This is not a major downside, as this functionality is not really of much use in NTK. It has been started as a means to bridge NTK visualization to the developed web-based systems like Cupboard [14]. In fact, the limitation shall be completely removed by the proposed integration of the visual and interactive component with the developed Cupboard, which we believe will make ontology discovery substantially more accessible to the users who want to directly interact with the system (as opposed to using its web service API).

Next, from the user interaction (and control) perspective, it would be more desirable to give the user a greater control over the aspects of the visual frontend than the current set of checkboxes

and buttons allows. In this context, an obvious way forward is the inclusion of these and additional choice in the standard *Preferences* panel of the NeOn Toolkit. The main reason this has not been realized as yet is the prioritization in functional behaviours and decisions which are 'critical' and need to be accessible via shortcuts directly in the plug-in UI and which are auxiliary for an ordinary user and could be delegated to the preferences panel of the toolkit. The most obvious aspects to open up for the user include the choice of the directories where images are stored, setup of the default layout, zoom, number of levels for conceptual zoom, etc.

Finally, from the conceptual point of view, the approach piloted here relies on the *key concepts* – that is often singletons representing and covering the ontology in question. Especially for larger ontologies, there is a degree of likelihood that concepts shown will be indeed widely dispersed and no or very few topological connection will be shown. As an implication, during conceptual zooming many singleton entities may rapidly clutter the screen estate. This is, however, a design feature rather than a bug, and we provided means like show/hide functions to alleviate this limit, at least to some extent.

Another point to raise here, although not being entirely the limitation of the plug-in itself, is the reliance on the web connectivity of the user's computer. Although Eclipse and NTK support HTTP connection by means of a proxy, in some environments, proxies are maintained centrally for the entire operating system. As a side effect, the communication of plug-in algorithms with the network may either fail entirely or significantly slow due to missing or incorrect proxy setting. There is no single piece of advice to give here, apart from advising the users to ensure their proxy is set up correctly. One simple test of the proxy setup is the use of update features: going through menu option "*Help → Software updates → Find & Install* " one may request the updates for any of the listed remote sites. If a list of available modules shows, internet connection works, if any error occurs, the issue is most likely in an incorrectly set up connection and/or proxy.

6.2 Future extensions

Future work on the ontology visualization by means of summaries can be grouped into several categories. First, there are some limitations, low-hanging fruits, which emerged during the implementation process and were not deemed too serious to diverge from rolling the plug-in out in its prototype version. Obviously, these should be fixed at the nearest opportunity. A typical representative of this kind of further work are missing user preferences – be it about specifying which concept labels should be persistently ignored in visual summaries or about the support for editing and managing the configuration files. The timeline for addressing these aspects is 2-3 months from the public release of the prototype.

In addition to collaborating with other visualization techniques, there is a wealth of opportunities to apply the visualization in other modules and plug-ins that are more of the 'backend' nature. To mention some most obvious candidates where summary-based visualization may add value we may look at the cases of ontology modularization, ontology view customization, ontology comparison, and ontology search and discovery:

- In terms of working with modules, it may be beneficial for the user to preview the effect an inclusion or exclusion of a given model onto the topical coverage of a given ontology. If two modules, for example, can be summarized in broadly similar ways, one may decide to use different operations (say, intersections instead of unions).
- Similarly, having followed ontology view customization wizard and producing a viewpoint upon a large and complex ontology, it makes sense for the user to preview whether the effort had or had not desired impact in terms of certain concepts (dis-)appearing from the ontology summary.

- Next, there is an interesting and fertile area of using a derivation of the summary-based visualization as a basis for ontology comparison. Akin to mapping visualization piloted in OntoConto [16] it may be cognitively cheaper for the user to compare two or more ontologies visually rather than parsing each of them in isolation. Two broad approaches can be taken here, and we intend to follow up both. First, in a kind of “one on one” mode two ontologies may be displayed in terms of their respective summaries as described in this report and any action performed on one would apply to the other (e.g., in terms of conceptual zooming). Thus, the user would be given two (or more) previews that can then be compared, overlaps highlighted, node importance being flagged if there is a consensus in the information value of a given concept between more than one ontology, etc.

Secondly, we plan to take a kind of “conceptual baseline” approach, whereby a pre-agreed ‘landscape’ of topics a given user is interested in will be presented as a N by N grid, and each summarized ontology would be positioned within this grid to reflect the proximity of a specific summarizing key concept and one of the topics forming the landscape. Thus, ontology summary would act as ‘one of the possible maps’ juxtaposed over the same topical ‘landscape’ – leading us to the opportunity to show two, three or more summaries against the same baseline, the same ‘landscape’. This, in turn, enables the user to rapidly assess which topics these different ontologies seem to cover, what are the most prominent differences (or consensual points, to that matter) among the ontologies. This work is currently the theme for a PhD research at the University of Technology in Kosice, Slovakia, in which the lead author of this report and WP4 leader is participating as an external advisor. Figure 21 shows the idea of spreading the ontology summary so as to match a range of pre-selected topics [17].



Figure 21. An early vision of using ontology summaries to drive topical viewpoints: a top-level summary (A) with a cut-out showing the conceptual zooming facility.

- In terms of ontology search and discovery, the work is already under way in the shape of planning our contribution to the Cupboard framework that is positioned as a one-step-shop for publishing, finding and talking about reusable ontologies. A particular role of the plug-in described here can be seen in facilitating previews and simple browsing for the discovered ontologies – currently Watson only offers a list of found ontologies and if anything is visualized it is merely the entity matching the query and its immediate topological neighbours. That is, currently Watson/Cupboard/Oyster (i.e., none of the components involved in this development) has a visual support comparable to the visualization strategy presented here. Hence, this seems to be a good strategic complementarity, which we intend to exploit.
- Another planned feature is the capability to load more than one ontology at the same time and allow their comparison by means of visualizing their respective summaries in different colours. We plan to address this comparative and analytic capability in conjunction with the aforementioned support for using the conceptual baselines, ‘landscapes’ in the future release of the visualization plug-in(s). The ‘landscape’ strand is currently in progress and prototypes exist outside of the NTK environment.
- Another large category of further work can be described as “exploiting emerging opportunities”. In other words, having now a concrete, working prototype as a part of NTK, it makes sense to explore benefits of linking this form of visualization to other visual metaphors, e.g., ISOCO’s relationship browser [15] and investigate seamless ways of switching from one visual ‘viewpoint’ to another – precisely as conceptually argued in our user studies in [4, 5, 6].
- From the viewpoint of managing plug-ins in NTK it would be desirable to move the visual components from the ‘*Other*’ category to a dedicated category, e.g., ‘*Visualization*’. This change affects the way plug-ins are listed in the online repository, as well as how views and perspectives are listed when the user wants to initialize them. Although this is beyond the control of the plug-in authors, we will make an effort to liaise with other developers of the visualization components and ‘lobby’ the administrators of the plug-in repository and of the categorization.

References and bibliography

- [1] ISO 13407, *User-centred design process for interactive systems*. 1998.
- [2] Dellschaft, K., Dzbor, M., Gomez-Perez, J.M., Buil Aranda, C., Mladenic, D., and Kubias, A.: *Review of methods and models for customizing/personalizing ontologies*. Deliverable report D4.2.1, NeOn Project Consortium, October 2006.
- [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] Dzbor, M., Buil Aranda, C., Motta, E., and Gomez-Perez, J.M. *Analysis of user needs, behaviours & requirements on ontology engineering tools*. Deliverable report D4.1.2, NeOn Project Consortium, January 2008.
- [7] Norman, D., *The Invisible Computer*. 1998, Cambridge, MA: MIT Press.
- [8] Peroni, S., Motta, E., and d'Aquin, M.: *Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures*. In Proc. of the 3rd Asian Semantic Web Conf., Thailand, 2009.
- [9] 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.
- [10] 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.
- [11] Shneiderman, B. and Plaisant, C., *Designing the User Interface: Strategies for effective human-computer interaction*. 4 ed. 2004: Addison-Wesley. 672.
- [12] Storey, M.A., Lintern, R., Ernst, N.A., *et al.* *Visualization and Protégé*. In *7th International Protégé Conference*. 2004. Maryland, US.
- [13] Presutti, V., Gangemi, A. *Ontology Patterns*. Chapter 10 in *Handbook of Ontologies*, Staab, S. and Studer, R. (eds), Springer Verlag, The Netherlands (in press).
- [14] d'Aquin, M., Lewen, H., and Palma, R.: *Cupboard – A Place to Expose your Ontologies to Applications and the Community*. Proc. of the European Semantic Web Conference, ESWC 2009, demo track
- [15] Mendez, V. and Melero, R.: *Ontology visualization plug-in – relationship browser*. Deliverable report D8.4.2, NeOn Project Consortium, January 2009.
- [16] Pajntar, B. and Mladenic, D.: *NeOn Toolkit plugin realizing revised and extended techniques for contextualized visualization of ontologies and ontology networks*. Deliverable report D4.5.2, NeOn Project Consortium, November 2008.
- [17] Vrana, J., Machova, K., and Dzbor, M.: *Support for ontology evaluation and visualization*. In *Posters of the 'Znalosti 2009' conference*, Czech republic, 2009.