NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

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

D4.5.2 NeOn Toolkit plugin realizing revised and extended techniques for contextualized visualization of ontologies and ontology networks

Deliverable Co-ordinator: **Boštjan Pajntar**

Deliverable Co-ordinating Institution: **J. Stefan Institute (JSI)**

Other Authors: **Dunja Mladenić and Marko Grobelnik (JSI), Sandra Kohler (ISOCO), Martin Dzbor (OU)**

This deliverable provides a NeOn Toolkit plugin that enables contextualized visualization of ontologies and ontology networks. The idea is to enable the user to browse through an ontology inside a context of related networked ontologies as provided in D3.2.2. Visualization consists of two parts. First, a pair of ontologies is visualized in an intelligent way and second, alignment of these two ontologies is visualized as links between the related concepts. Furthermore, this software is seamlessly integrated with the other NeOn technologies. Integration inside the NeOn Toolkit provides an easy access to the loaded ontologies, as well as possibility to edit them. Next, this software consumes the NeOn Alignment server (D3.3.2), which enables the user an access to several different alignments between ontologies of current interest. This deliverable is used for evaluation of mappings in D3.4.1. We also show its potential usage on NeOn case studies in an illustrative example.

Document Identifier:	NEON/2007/D4.5.2/v1.0	Date due:	November 30, 2008
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	November 30, 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 the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following 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 11 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.ump.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 SL–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 – 665 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 Kolenz-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. Marino 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 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, S.A. (KIN)** C/Ciudad de Granada, 123 08018 Barcelona Spain Contact person: Antonio López E-mail address: alopez@kin.es

Contents

List of Figures

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:

- Jožeph Stefan Institute (JSI)
- Open University (OU)
- Intelligent Software Components S.A. (ISOCO)
- Food and Agriculture Organization of the United Nations (FAO)

Change Log

Version	Date	Amended by	Changes
0.1	11-10-2008	Boštjan Pajntar	Definition of chapters and sections
0.2	18-11-2008	Boštjan Pajntar	Chapters 1, 2, 3
0.3	20-11-2008	Dunja Mladenić	Revision of content
0.4	5-12-2008	Boštjan Pajntar	Figures, revision of Chapter 3
0.5	4-1-2008	Sandra Kohler	Added to Use Cases
0.6	5-1-2008	Martin Dzbor	Revision
0.7	6-1-2008	Boštjan Pajntar	Chapter 4, Executive summary
0.8	9-1-2008	Boštjan Pajntar	Appendix, Overall revision
0.9	10-1-2008	Marko Grobelnik	Overall revision

Executive Summary

This deliverable presents a Neon Toolkit plugin named OntoConto. The main goal of this plugin is to provide context for a selected ontology by providing related parts in another. This is achieved by visualizing and manipulating the chosen and a context ontology side by side, with mappings connecting nodes across both ontologies. For each concept in the chosen ontology the context is defined by an aligned concept in the context ontology and its neighborhood of concepts. The user can simply switch contexts by either changing the context ontology or by changing the alignment connecting the two ontologies. Apart from the main goal, OntoConto also provides a simple and intuitive graphic user interface (GUI) for visualization and editing of mappings, that is suitable to be used even by non-ontology experts.

The basic visualization technique was proposed in D3.2.2 [Mar08]. The following contributions have been made as part of this deliverable:

- A lot of engineering work has gone into integrating the visualization part (written in Flash technology) inside a Neon Toolkit (Java) plugin. Now that this is done Neon Toolkit boast an advanced yet intuitive GUI for a visualization and editing of alignments. Both of this task are a requirement from both use case partners and they were not previously satisfied.
- OntoConto consumes a local instance of an Alignment Server and it also consumes the advanced methods provided by the Alignment Server's web service. OntoConto is also prepared to add any other service (a plugin or a webservice) providing alignments. The possibility of an automatic proposing of the mappings inside the mapping editor is also a requirement of one of the case study partners.
- New visualization techniques were added to both alignment and ontology visualization.
- User interface has been rewritten in order to work optimal inside the Neon Toolkit platform.

The plugin is available in both binary and source version at http://kameleon.ijs.si/ontoconto. To install it, binary version must be copied inside the plugins directory of the Neon Toolkit. No additional plugins are required.

Chapter 1

Introduction

Networked ontologies in general can be a very complex structure. There are many ways how the knowledge stored inside them can be obtained. One possible way is to focus on a part of the networked ontologies i.e. a pair of related ontologies and in this setting try to understand the context that one ontology provides for the other. The featured plugin named "OntoConto" enables the user to achieve such a scenario through a visualization of such a related ontology pair inside the NeOn Toolkit.

The need for visualizing a complex structure is basic to any human computer interaction. The reasons for this are discussed in different resources and publications, for instance, Wikipedia includes the following statement:

> Scientific visualization is the transformation, selection or representation of data from simulations or experiments, with an implicit or explicit geometric structure, to allow the exploration, analysis and understanding of the data. [wik08]

Due to the complexity of an ontology network (also referred to as networked ontologies) it seems natural to provide the user with a way that would help with exploration, analysis and understanding of the network. In our specific solution, several visualization techniques were utilized in order to provide a very detailed visualization of an ontology pair and of the mappings linking the two ontologies.

The rest of this deliverable is sectioned as follows. First, the proposed approach is described in detail (Chapter: 2). Next, motivation for such a plugin is presented, together with the requirements and scenarios provided by the case study partners (Chapter: 3). Last, some discussion of further possibilities is provided in the final Chapter 4. In appendix A a work flow of a concrete scenario is presented together with screenshots of each action the user takes.

Chapter 2

Approach Description

The main idea behind the proposed approach is to provide a context of a concept in one ontology by providing a part of the related ontology that is similar/related to this concept. This is knowledge that is inherently present in every network of ontologies and harnessing it can be beneficial in several ways. Ontology creator/editor gains help by finding out how certain parts of an ontology were done in a different ontology and this helps with making editing decisions. An ontology user on the other hand, is presented with a tool that can help understand an ontology. For example if the user has a task of annotating some data, this tool can help understand each specific concept in the ontology by providing a context in another more familiar ontology.

Detailed description of the proposed approach follows. If a concept A in the first ontology is aligned to the concept B in the second, then since the "near" concepts of B are similar to B, this provides context for A. To define the "near" concepts of B, different approaches could be used. For example any type of a relation inside of an ontology could be used to define a unique metric between concepts, by counting how many hops over specified relation are needed to get from one concept to another. Every such metric would provide a specific context defined by the relation used.

In our solution we took the most basic *SubConceptOf* relation of the ontologies to measure this distance between the concepts. The reasons for picking it is that this relation can be considered the most basic relation in an ontology, it is a relation found in most if not all ontologies. Apart from this, *SubConceptOf* relation defines an underlying tree structure which can be very easily and intuitively visualized to the user.

The fundament of this visualization was proposed in the WP3 deliverable D3.2.2 [Mar08]. Two selected ontologies are visualized side by side, and the aligned concepts are visually linked across ontologies. Detailed description of the visualization and the user interface is given in Section 2.1. This application is also tightly integrated with other NeOn technologies. First, OntoConto is integrated inside of the NeOn Toolkit as a plugin (Section 2.2). Second, alignments are automatically obtained from the Alignment Server (Section 2.3).

This setting provides the user with a complete solution for several different tasks. Most notably the user can edit ontologies inside an automatically generated context of another ontology. Apart from that, OntoConto also provides means that help with the evaluation and manipulation of the mappings. More detailed review of usage and scenarios is given in Chapter 3.

2.1 OntoConto visualization

The primary goal of the visualization is to enable the user to view one ontology in the context of another. This is achieved by visualizing mappings between the ontology of interest and the context ontology. This is also the core added value of OntoConto. To our knowledge it was not possible to visualize mappings and therefore a context defined by an ontology alignment prior to our work.

OntoConto visualization is based on the prototype developed in NeOn deliverable D3.2.2, however several changes have been made, most notably in the user interface with the mind of simple and intuitive plugin. Therefor the description of the techniques is given in full. The visualization can be broken into three parts:

- visualization of an ontology
- visualization of the mappings
- user interface

2.1.1 Visualization of an ontology

To achieve effective visualization of mappings it is necessary to first visualize ontologies. Our approach explained below could also be replaced by a different (preexisting) approach found elsewhere. Another possibility would be to enable the user to visualize ontologies based on an ontology relation of his choosing.

For the visualization of an ontology, OntoConto is visualizing concepts ordered by an underlying tree structure stemming from *SubConceptOf* relation. For this purpose the following visualization technique was developed in D3.2.2 [Mar08]. First, due to the demand to visualize a context for a selected concept and also due to the limitation of how many concepts can be distinctly visualized at the same time our approach visualizes only four layers of the ontology at the same time. The definition of the layers follows. The selected node (concept of interest) is visualized at the top. All the nearest concepts defined by *SubConceptOf* relation (children and a parent) are visualized bellow the selected node, all on the same level. This is the second layer. Below this, on the third layer, the concepts with the distance 2 to the selected node are visualized and similar for the fourth layer. As demonstrated, the defined metric disregards the direction of the *SubConceptOf* relation. The rationale behind this decision is two fold. First the hierarchy of what is a sub concept of what is not retained across different ontologies, so there is no great need to visualize this information. The second and more important reason is that the proposed visualization can be easily upgraded with a different one, in which another - possibly symmetric - relation replaces the *SubConceptOf* relation. The basic visualization of an ontology is depicted in figure 2.1. Notice the vertical displacement of the concepts on the same level. This has been a small modification from the prototype developed in D3.2.2 in order to enhance the visibility of a single concept in the case of overcrowding.

Often visualizing four layers at one time still proves too much of an information to visualize distinctly, so another visualization technique has been implemented. Its effect is somewhat similar to using lens. Visual entities near the center of the screen are enlarged, on account of reduced entities that are closer to the borders. This well known visualization technique enables the user to clearly see the detail of interest, while retaining the big picture. The user can control the magnitude of this effect by scrolling the mouse wheel. This is not unlike choosing lens with different convexities.

As an example of use we demonstrate this functionality on the case study ontology: http://www.fao.org/aims/aos/asc (figure 2.2). For the user interested in the sub concepts of a concept “p” the following interaction suffices: First visualization must be centered on the point of interest - namely around “p”. This is achieved by dragging the background. Next, the user sets the convexity of the visualization by rolling the mouse wheel. With this two very basic mouse operation, the user is enabled to distinctly read all fourteen sub concepts of the concept “p” while still being able to understand the overall structure of this part of the ontology.

For the visualization of the mappings two selected ontologies are visualized side by side in this way.

2.1.2 Visualization of the mappings

This is the main part of the OntoConto visualization. It enables the user to acquire contextual information in networked ontologies across two selected ontologies. Even though this is the core added value of OntoConto, the basic idea for the visualization is very simple. For the visualization of a mapping, a simple link is visualized spanning the space between the mapped concepts. This has already been proposed in D3.2.2 [Mar08]. Such a simple solution is only possible because great care was taken when devising the visualization of the ontologies.

While this simple idea already fulfills most of the requirements stated by the case study partners (Chapter 3) several detail improvements have been made. First the visualization of the mappings must be done dynamically because the visualized part of the ontologies changes through the user interaction (already done in D3.2.2). Second, as per request of the consortium partner, the intensity of the visualized mapping follows the stored or calculated value for each mapping. This enables easy selection of the mappings where more than one mapping per concept is proposed. It also helps with the assessment of the quality of different alignments. Some possible future visualization functionalities are presented in Chapter 4.

Next, a lot of effort has gone into connecting OntoConto with the Alignment Server (Section 2.3). This allows the user to obtain auto-generated alignments (Section 2.3). It must be noted the Alignment Server only serves as one source to obtain the alignments and the modular design of OntoConto makes addition of other such services easy.

Currently it is also possible to obtain alignments that are stored inside the Neon Toolkit's workspace. This alignments could be manually created by the user, obtained by editing and storing an existing alignment or obtained from a third party application exported and stored as an owl ontology. Loading and storing of the alignments in such a manner also enables interoperability with other plugins (Section 2.2).

An example of the main OntoConto visualization follows: A simple Named Equality Alignment on two FAO ontologies can be viewed on the Figure 2.3. The top three orange lines show mapping between the two example ontologies, where concepts *c_classification_shema, c_category, c_noun* in the left hand-side ontology are linked with the concepts having the same names in the right hand-side ontology.

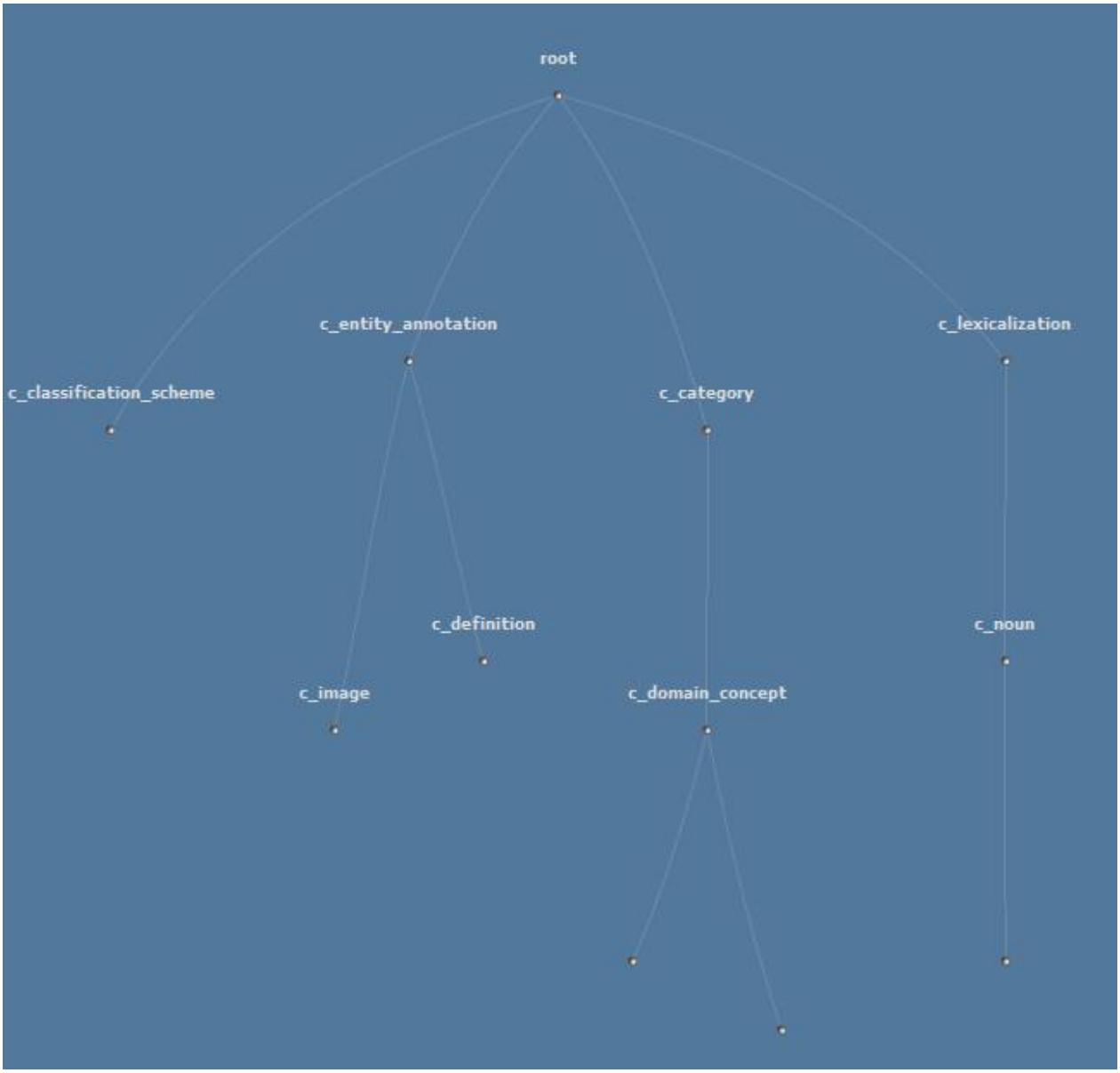

Figure 2.1: Layered visualization of an ontology. Concepts on the same layer are slightly vertically displaced in order to enhance the visibility of a single concept. For the same reason, label of concepts are removed altogether from the forth layer. They are visible on mouseover.

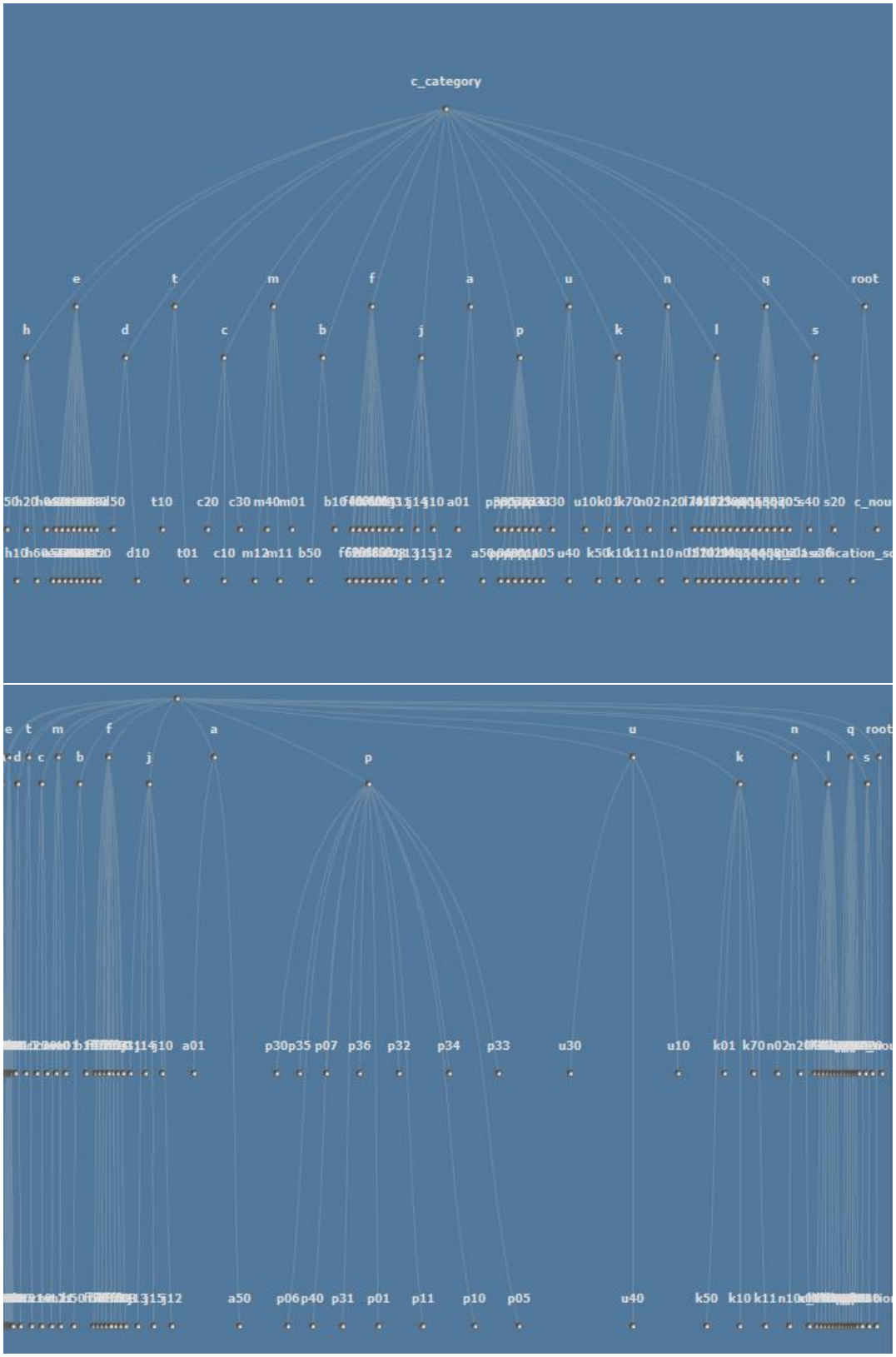

Figure 2.2: If many sub concepts for a single concept are present (upper picture) it is possible for the user interested in the sub concepts of the concept “p” to easily visualize it by setting the convexity via mouse wheel and dragging the point of interest into the center of the field (bottom picture).

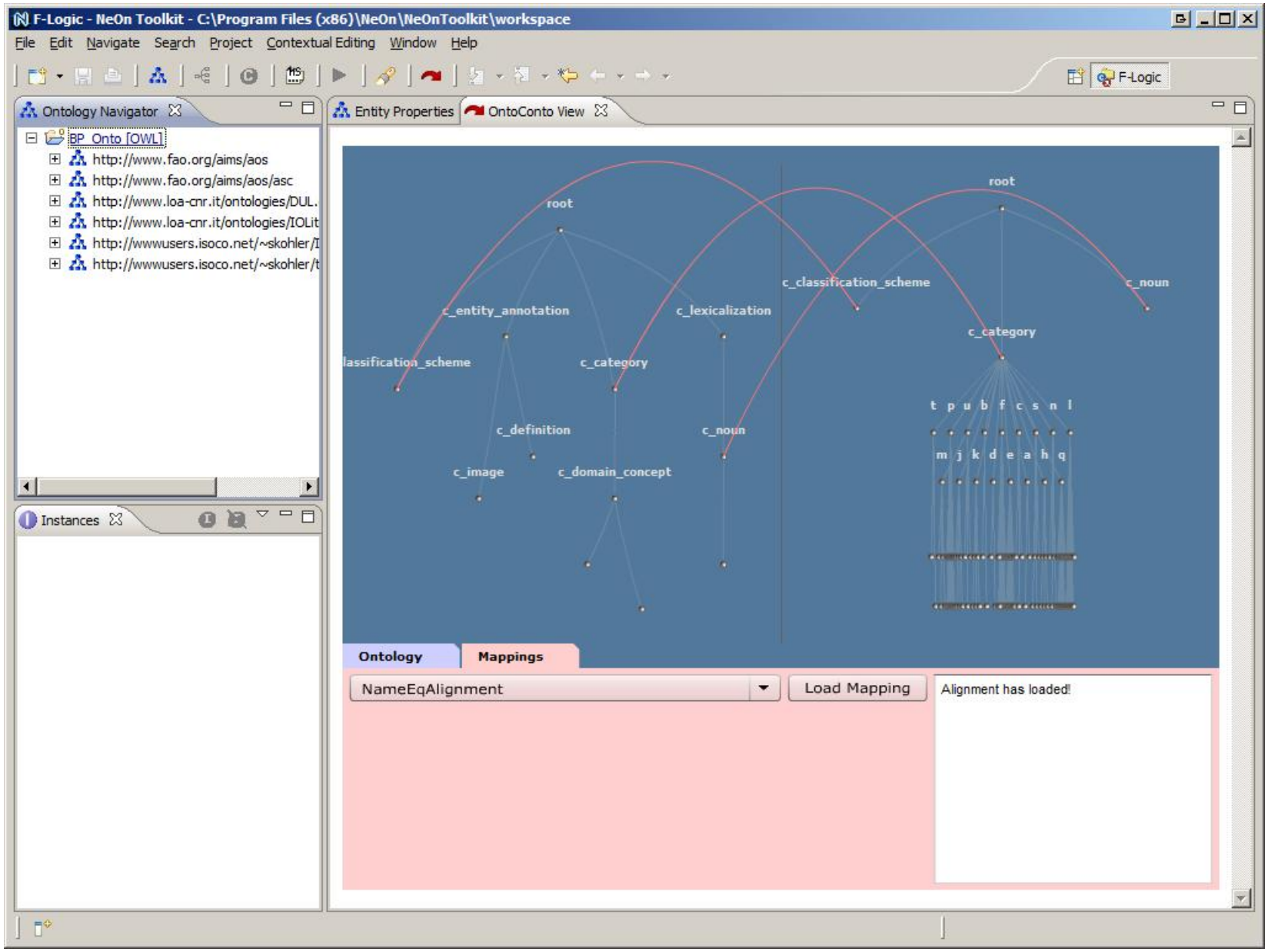

Figure 2.3: Integration of OntoConto inside of the NeOn Toolkit.

2.1.3 Graphical User Interface

Great care was taken in this deliverable to make an easy and intuitive graphical user interface. When OntoConto starts, the user is guided by the interface only providing actions that must be taken. The first step for the user is to choose to load two ontologies. This is done by selecting each from a respective combo box, upon which ontologies get visualized. Notice that any owl ontology loaded inside the NeOn Toolkit is available to load into OntoConto. Actually OntoConto is prepared for F-logic ontologies also, but currently this is disabled, because Alignment server supports only OWL ontologies. If it will become possible to access automatically generated mappings for F-logic they will also be easily added to OntoConto.

After both ontologies are loaded and visualized the user interface permits to request an alignment. This is done by selecting one of the Alignment Server's methods for automatic generation of an alignment, by loading a prestored alignment saved as an ontology inside the Neon Toolkit's workspace or by starting a new alignment. As soon as the alignment is generated or loaded, it gets visualized on the screen.

The user is further enabled to easily browse through both ontologies. This is done by simply clicking on any concept, which selects it as the selected node, and the whole ontology gets dynamically redrawn in to the new position, with selected node on the top. When this is done, the concepts may be visualized or hidden, the mappings also follow the dynamic redrawing. If there are too many visual entities on the screen, the user can implore the "lens" functionality. The magnitude of this effect can simply be set by using mouse wheel while hovering over the field where ontology is drawn. The user can further drag the visualized part of ontology on the screen in order to move the part of interest into the center where the zoom-in effect is the greatest. Dragging of the ontology is done by dragging the underlying field.

No editing of ontologies is possible in OntoConto itself, as editing is supported by the NeOn Toolkit. Notice that the prototype proposed in D3.2.2 [Mar08] had this functionality which has now been disabled for the OntoConto plugin. The reason for this is that one of the core functions of the NeOn Toolkit is editing of the ontology, so it seems reasonable not to include the editing of the ontology also inside OntoConto as it would only confuse the user. On the other hand, retaining the editing of the mappings makes sense, since this functionality is not well supported by other Neon Toolkit plugins, yet is required by the case study partners. Saving of the mappings had to be reimplemented in order to make it accessible to Neon Toolkit. Mappings are stored (and can be loaded from) Neon Toolkit's workspace.

Editing of the mappings is done with the following actions. By clicking on a mapping it is possible to select it. When a mapping is selected it is possible to edit its weight, the two concepts it links and it is also possible to delete the mapping altogether. It is also possible to create a new mapping by clicking two concepts to be connected while holding the key “m”. For a demonstration see Appendix A.

2.2 Integration within the NeOn Toolkit

OntoConto is a plugin inside the NeOn Toolkit application / platform. This provides seamless integration with other plugins and core functionalities of the Toolkit. To install it, user must simply copy the binary version of the OntoConto into the “plugins” folder of the Toolkit. Both binary and source version are accessible at: http://kameleon.ijs.si/ontoconto. OntoConto does not require any custom plugins to work, the core Neon Toolkit application is all that is needed.

Architecturally wise OntoConto is integrated inside the NeOn Toolkit in a non-trivial way. Primarily OntoConto functions as one of the graphical user interfaces of the Toolkit, with several advanced visualization techniques and this is why it was implemented with Adobe's Flash application. Flash is one of the leading dynamic visualization development applications, allowing easy development of advanced visualization techniques. The architecture of the integration (Figure 2.4) is done by first implementing a regular java plugin named OntoConto. This plugin connects to the Toolkit's data model, and extracts all the information of the ontologies and stored alignments. Second, this plugin consumes the Alignment Server and could also easily consume any other alignment providing plugin or webservice. Last, it implements a browser which in turn includes the OntoConto's visualization part written in Flash. The communication between Flash application and the java

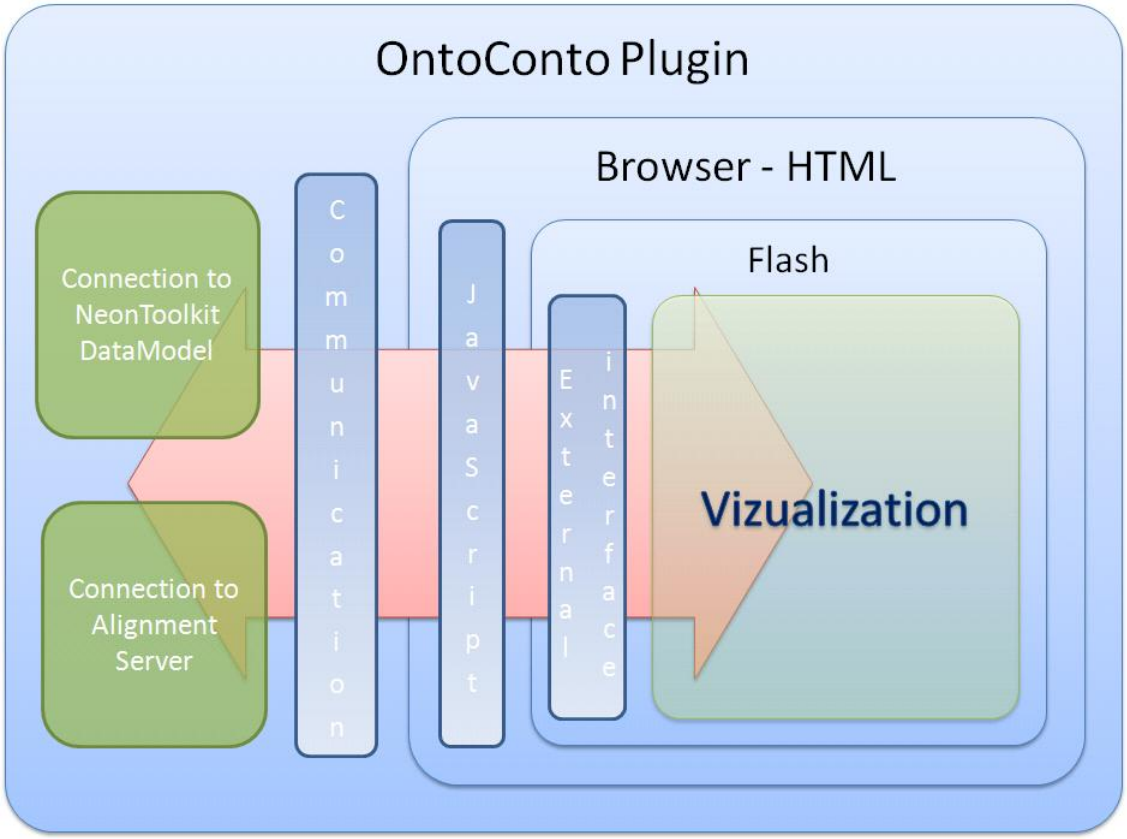

Figure 2.4: Architecture of OntoConto Plugin. On the left connection to external modules is presented. Currently this are the NTK-datamodel and the Alignment Server. Another alignment service could easily be added as a separate module. This information stored in java plugin is communicated with the Flash visualization application on the right through three distinct layers.

plugin is done through:

- java, with invoking browser methods (to send information from java through javascript to flash) and events (to load information passed from flash through the browsers title).
- browser, done by javascript as a way to forward information from java to flash and window title being changed by flash and caught by java browser event.
- flash application, with its external interface exposing some methods being called from java via javascript and calling a javascript command to change the title of the browser.

This nontrivial communication is required in order to fully integrate the Java plugin and Flash applet. Now that this has been done, two great benefits are made possible. First the OntoConto visualization part can easily be ported outside of the Neon Toolkit. Flash is an extensively used technology spanning mediums (internet, intranet, desktop applications), platforms (Windows, Mac, Linux...) and devices (PCs, PDAs, phones) and OntoConto is modularly built so it would be easy to connect the flash visualization to some other source of data and port this visualization to another medium. The second benefit is that this communication can be reused by any future plugin that would desire to incorporate any flash application.

2.3 Consumption of the Alignment Server

Alignment Server has been developed within the NeOn project in order to serve several different methods that automatically generate alignments (D3.2.2). To consume it, it must be called with two OWL ontologies and a method. It returns alignment in the RDF format. This makes it a perfect tool to be used by OntoConto. However, OntoConto could easily consume another such service if available (i.e. an application that would provide F-Logic alignments)

OntoConto is consuming a local instance of the alignment server. This ensures that OntoConto works on an isolated NeOn toolkit installation (without internet access). Due to request of a consortium partner OntoConto is now also able to consumes the advanced alignment services provided by the Alignment Server's web service. In order for this to work internet connection must be available and the two ontologies must be available online at the designated URL.

Chapter 3

Usage and Scenarios

In this chapter we describe usability of OntoConto. First requirements of case study partners are considered and later several scenarios of use are described. For a demonstration of a concrete scenario of use see Appendix A.

3.1 Requirements

At the beginning of the NeOn project the case study partners produced extensive User Requirements reports [Mar06] and [Jos06]. What follows is the assessment of the case study requirements that OntoConto fulfils.

OntoConto was implemented with great care to simplify user interface with very simple actions to select and browse ontologies and alignments. The user is not required to understand ontology and mapping notation nor existing ontology editing software. This in fact is a requirement from both case study partners. FAO defines several groups of users one being “subject experts”:

> Ontology editors are either subject experts or information management specialists, who will be in charge of the everyday editing and maintenance work of the networked ontologies. Their duties also involve the creation of multilingual versions of ontologies. Subject experts know about the domain to be modelled, but usually know little or nothing about ontology design issues or software for ontologies. . . . Subject experts should be provided with more intuitive interfaces than those available to ontology experts and application developers; in particular interfaces for subject experts should conceal much of the purely ontological and engineering decisions. [Mar06, page 27]

OntoConto with its combo selections, one click user interface and no ontology notation fits perfectly to serve this user group. Notably, this user group had no way of using mappings prior to OntoConto plugin.

Similarly Pharma Domain requirements state under the “5.2.3 Requirements for creators of invoice models and standards”:

> . . . On the other hand, it is important to stress the relevance of intuitive GUIs that release this kind of users from ontology-related complexity. As described in 5.2.2, since their knowledge of ontology engineering is certainly scarce, it is fundamental to apply techniques and methods that hide specifics of the ontological realm and maximize the use of terms from the pharmaceutical business. This issue is certainly a challenge for human-ontology interaction in WP4. [Jos06, page 57]

Again, OntoConto is the only plugin that provides this requirements with regard to mappings. In general, for this type of the user, it is also very useful to visualize an ontology, since this helps with its understanding and editing.

Apart from the clear need of OntoConto for users unfamiliar with ontologies, OntoConto also provides for expert users. Visualization in general provides easy exploration, understanding of any complex model, which an ontology network definitely is. This is also one of the requirements by FAO partner [Mar06, page 31]:

> In order to work with a network of ontologies, ontology experts should be able to visualize more than one ontology at a time, including ontologies for which they do not have editing rights. They should be able to:
>
> - ...
> - visualize mappings, list of relations used in each of them

3.2 Use Cases

Next we turn our attention on the scenarios which would benefit from OntoConto plugin. FAO has provided a list of scenarios. The relevant two that OntoConto provides for, are listed below.

The first simple scenario OntoConto fulfils deals with visualizing an ontology: “4.6.6 Visualizes an ontology as a diagram” [Mar06, page 40]. This is completely fulfilled by OntoConto. The only difference is when visualizing an ontology too large to fit to a screen, where instead of prompting the user which part of ontology should be visualized, this is later provided by the simple browsing through the ontology. We believe this is an improvement of the scenario, due to the simplified user interface and also the fact that the user might not know which part of the ontology is interesting for the issue at hand.

The main scenario OntoConto fulfils is: “4.6.10 System-supported creation of mappings between concepts”[Mar06, page 46]. The desired basic flow is fully fulfilled:

- User selects option "search for candidate mappings among concepts"
- System requests to choose a type of mapping
- User inspects the proposed candidates one by one (by clicking on them and visualizing the candidate concepts and the surrounding ontology fragment)
- User selects the appropriate candidate and confirms the proposed mapping

Apart from the pre-thought scenarios, OntoConto has already proved to serve other needs as well. In WP3, deliverable D3.4.1 Evaluation of mappings, OntoConto already proved to be a useful asset.

Deliverable D3.4.1 evaluates the algorithms for ontology mapping provided by the alignment plugin. The experiments tend to answer how good these algorithms detect alignments in comparison with an expert of the domain. The expert defines the matchings between concepts of the two respective ontologies. With this control set precision and recall are calculated for the alignments detected by the different algorithms.

To be able to detect concepts which represent the same aspect of the real world it is necessary to take the whole context into account. Concepts labeled with the same name do not necessarily have to describe exactly the same element; one concept might be more general or might describe a different part. To detect such ambiguities one has to check the concept within its hierarchy. OntoConto visualizes the structure of both ontologies and permits the comparison of two concepts within their context. It provides help for the evaluation of the similitude of the two respective elements and therefore helps to avoid wrong similarity assumptions.

OntoConto also simplifies the qualitative analysis of the mapping algorithms. Some algorithms do not give concrete alignments. The semanticmapper-algorithm for example proposes semantic relations and the user has to choose the correct ones out of this pre-processed set. For algorithms of this kind the strict quantitative analysis calculating recall and precision is not sufficient. With OntoConto it is possible to check the matching proposals and compare the different possibilities for each mapped concept. The decision which subset of the suggested alignments is correct is easier to take. Moreover OntoConto helps to get a quick first impression of the available algorithms. It is quite probable that a new user who uses the Alignment Plugin does not

know the theory of all of them and neither the differences between them. With the visualization OntoConto provides the user sees immediately which algorithm provides more matchings, which one only defines two concepts as equal if their labels are identical, which one detects not only the equivalence matchings etc.

Chapter 4

Discussion

When visualizing complex structures there is no definite way that visualization should follow. A careful equilibrium between simplicity of the user interface, selection of the information to be presented and ways of presenting it must be sought. OntoConto plugin leaned towards simplicity in order to fulfil the need of non ontology experts. This user group is defined by both case studies; however, it has not been well provided for up till now.

When implementing and integrating OntoConto many different functionalities were considered. At first only basic functionalities were implemented and then the consortium partners were asked to request what is needed. On the one hand, with the time available it was not possible to implement every functionality possible, so it seems sensible to implement those really needed and on the other hand a functionality that is not needed yet implemented burdens the user interface so it seems prudent not to include anything unnecessary.

Due to partner requests, the following functionalities were added already at the time OntoConto was developed:

- Visualization of the weights of the mappings. This is generally needed in order to assess the importance of a mapping.
- Alignment Server online methods are consumed. Initially only a local instance of Alignment server was supported. Work on the deliverable D3.4.1 required the advanced online methods so this was added.
- Alignment editing was implemented. Due to the fact this is a use case requirement - not provided for elsewhere - a solution for storing the alignments as owl ontologies was devised. This is not optimal since OntoConto was primarily a visualization plugin.

Several other functionalities were considered and can be implemented if they will be required by the partners:

- Visualization of an ontology could follow a specific - user picked - relation inside the ontology instead of the *SubConceptOf* relation. For the same alignment the user would gain a different context for each relation inside the ontology. This would not be a trivial upgrade of the existing solution, because the nice properties of the *SubConceptOf* relation also serve as the means to navigate through ontology, which is not true for a general relation.
- It would be possible to visualize several alignments at the same time. This would be easily implemented; however it would burden the user interface.
- A threshold that would alow the user to select amount of the mappings to visualize could be implemented. This also would not be hard to implement.

Appendix A

Example scenario

In order to showcase the features of OntoConto as well as to demonstrate the workflow of usage, we have devised a scenario which involves a user who is charged with a specific task. In the remainder of the appendix, first the scenario will be defined. It will involve a user with a selected set of expertise and a realistic task he has to do. Finally a step by step actions of the user will be described explaining the motivation for each action as well as demonstrating the usage of OntoConto.

A.1 Definition of the scenario

Following the case study partners definition for one user type (Chapter 3): we define the user as an expert of the domain being modeled; however, with very little knowledge about ontologies. This user is responsible for the maintainance of a small part of some ontology which will be defined shortly. Furthermore, he is given another ontology. His task is to link concepts in his part of the ontology with concepts in the new given ontology where appropriate. Such a task seems realistic. Mappings will most likely be done by a domain (and not an ontology) expert.

For this scenario we have selected the following use case ontologies. The initial ontology which the user is maintaining is http://wwwusers.isoco.net/-skohler/InvoicerefOntology.owl. He is in charge of the part starting with “InformationObject” concept, together with all of its sub concepts, instances and so forth. The new given ontology is http://www.loa-cnr.it/ontologies/IOLite.owl. The user is unfamiliar with this ontology.

The user is finally provided an installation of the Neon Toolkit with an installed OntoConto plugin.

A.2 Initial steps

When the user starts the Neon Toolkit and the OntoConto plugin a relatively empty screen together with a very light user interface is returned. This can be seen on Figure A.1. The user has an option to select an ontology, to refresh the ontology list and to load a context ontology. It does not take him long to select the correct ontology from the drop down menu on the left side.

Once the user presses the “Load” button the top part of the ontology gets immediately visualized. This can be seen on the Figure A.2. Since the user is not interested in the complete ontology, he first locates the concept “InformationObject”. If he has problem locating it, he can use the mouse wheel to increase the lens functionality, which aids him in his exploration of the ontology. Once it is located, the user can click on it and the whole ontology gets dynamically redrawn. The animated transition helps the user maintain the mental image of the ontology’s structure.

After the user is satisfied with the part of the ontology that is visualized, he loads the new given ontology to the right side of the screen (Figure A.3). By clicking on its nodes, the new (never before seen) ontology can be easily explored. Notice also that once both ontologies are loaded the user interface changes to a new tab which is designated to manipulation of alignments.

Figure A.1: Starting OntoConto: Choosing from a combobox, which stored ontology to load.

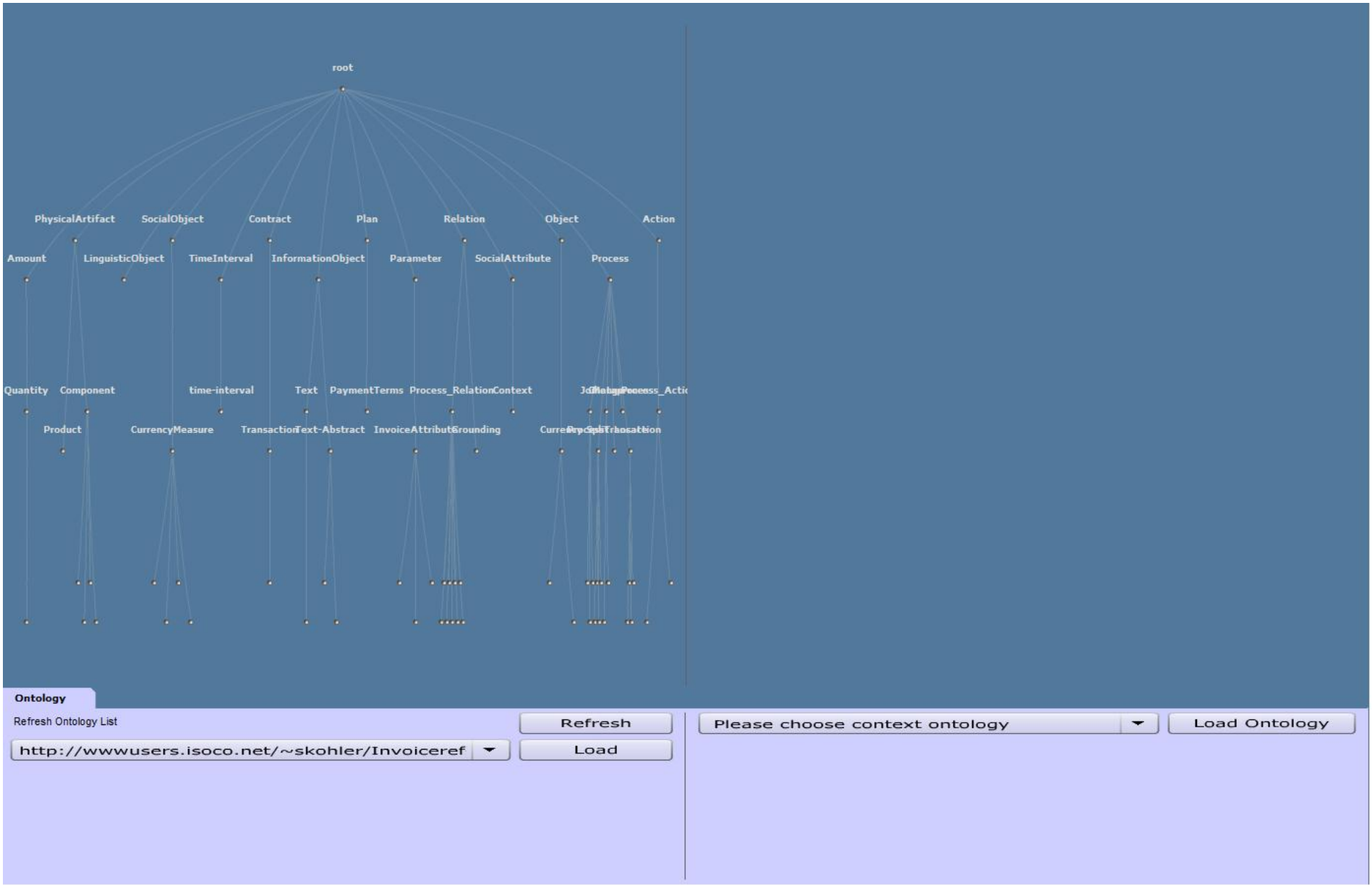

Figure A.2: The chosen ontology is visualized.

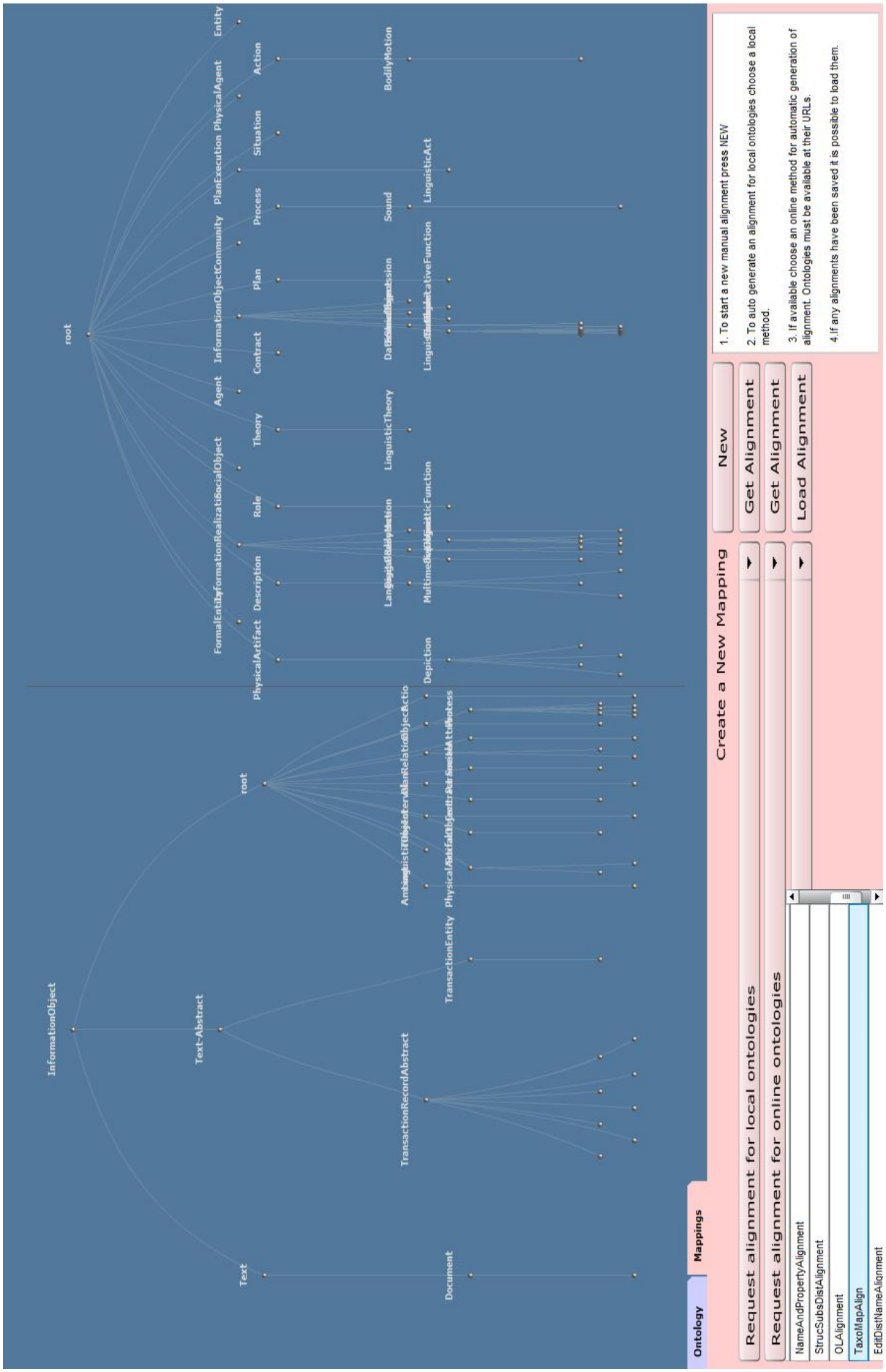

Figure A.3: Working ontology rerooted to “InformationObject” concept. Context ontology loaded.

A.3 Alignment visualization

As can be seen on the Figure A.3 the user is presented with four ways of generating alignments. First, it is possible to start a new empty alignment by pressing the “New” button. Even with the easy exploration of ontologies the user is still hard pressed to find all the mappings on his own. This is why he chooses one of the methods that automatically generate them. There are two types of this methods. The Local methods work without internet connection and with local ontologies. More advanced online methods on the other hand require internet connection as well as online accessible ontologies. Finally, if there were any saved alignments between this two ontologies the user would be able to load them.

The user can try out several methods in order to choose the one that suits him most. Alternatively, he can seek an advice for the choice of the method from an ontology expert. In our example the user chose “TaxoMapAlign” online method. Depending on the ontologies and the choice of the method it can take some time before an alignment is generated. Alignment is then quickly loaded and visualized on the screen. Mappings are showed as red lines spanning between concepts from one ontology to the other. The number one on the left side of the alignment designates the confidence of the alignment. Some algorithms return probabilities (values between 0 and 1) which would also show by the mapping being less visible. In any event, the user can now easily finds the part (or parts) of the context ontology that are most similar to his part.

A.4 Alignment manipulation

With the good starting point (Figure A.4) of the automatically proposed mappings, the user can now start to manually correct them. The following changes are meant only as an illustrative example and may not be correct. First the user notices, two sub concepts (Text, Text-Abstract) of the “InformationObject” link to the concept “Text”, which is a sub concept of ”LinguisticObject”. He decides to remove the latter mapping. He does that by first clicking on it which selects it. Once the mapping is selected the user interface changes (Figure A.5). The user can read the two concepts the mapping links, he can edit the measure (strength) of the mapping, change one or both of the concepts by clicking on some other while holding the key “m” and finally as in our case, the user can click on the “Delete Mapping” button to delete it.

The user decides to add a new mapping spanning “InformationObject” in both ontologies. He can do this only when no map is selected. Action is similar as when editing a selected mapping. User must click on two concepts while holding down the key “m”. Next, the users adds another mapping between the concept “Document” and “ContractText”. However, the user is not certain in this relation so he changes the measure to 0.3. This is reflected in less visible mapping. All of this changes can be seen on Figure A.5. After he is satisfied he can name by filling “Alignment Title” input box and save it by pressing the “Save” button (Figure A.6).

Finally the user decides to use another automatic method in order to check on his work (Figure A.6). His saved alignment is unloaded (it is easy and fast to reloaded it) and this time the “StringDistAlignment” is chosen. After closer inspection the user notices a mapping spanning “LinguisticObject” in both Ontologies (a white mapping due to mouse over). This concepts is placed outside the part of the ontology he is responsible for, so he was not even aware of it before. However, on the right side ontology this concept lies between “InformationObject” and “Text” which are directly linked in his ontology. He takes this information to the ontology design expert and they together decide if it makes sense to change the hierarchy of their ontology to the way this is done in the concept ontology.

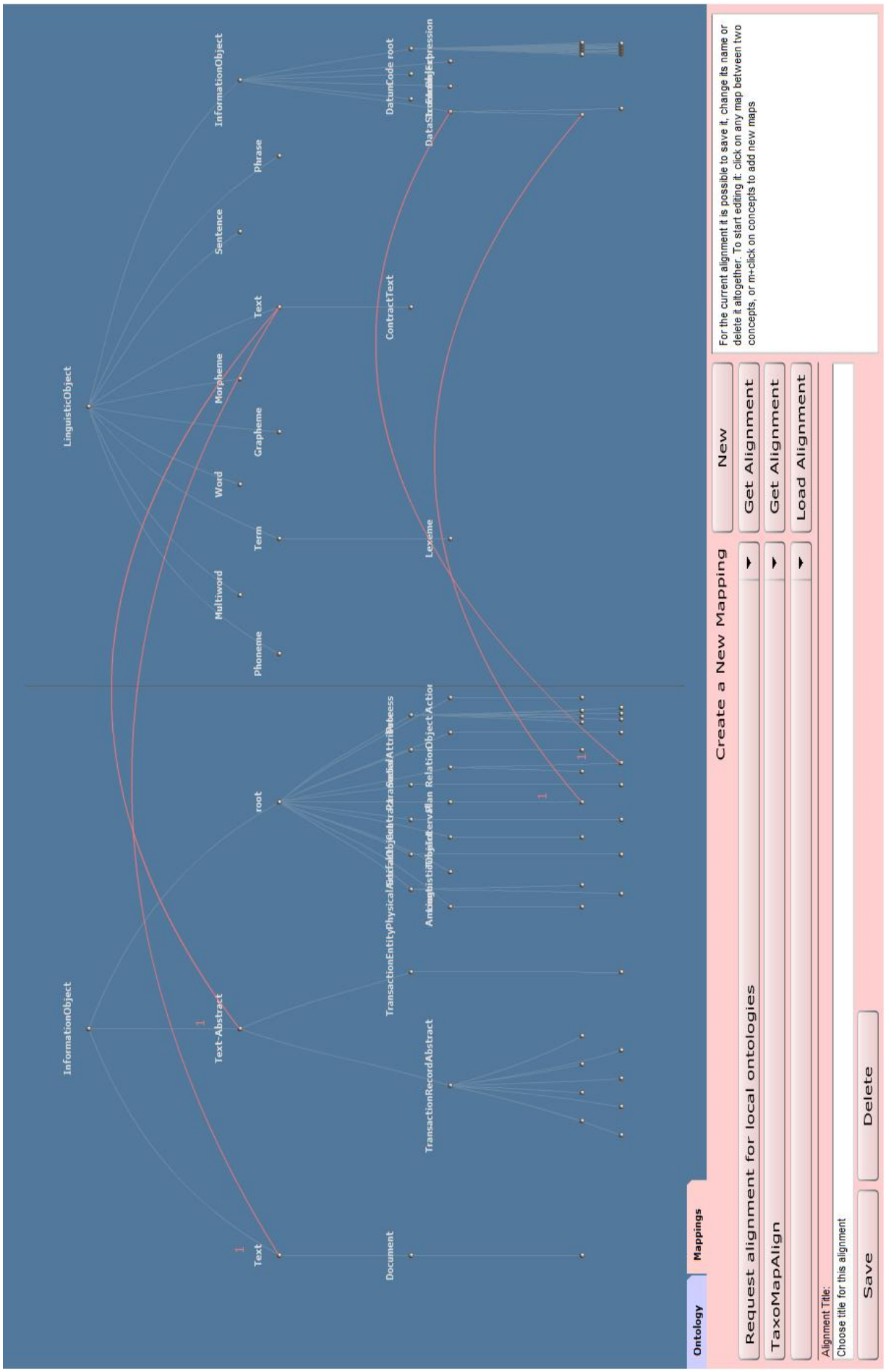

Figure A.4: Alignment provided by the "TaxoMapAlign" method.

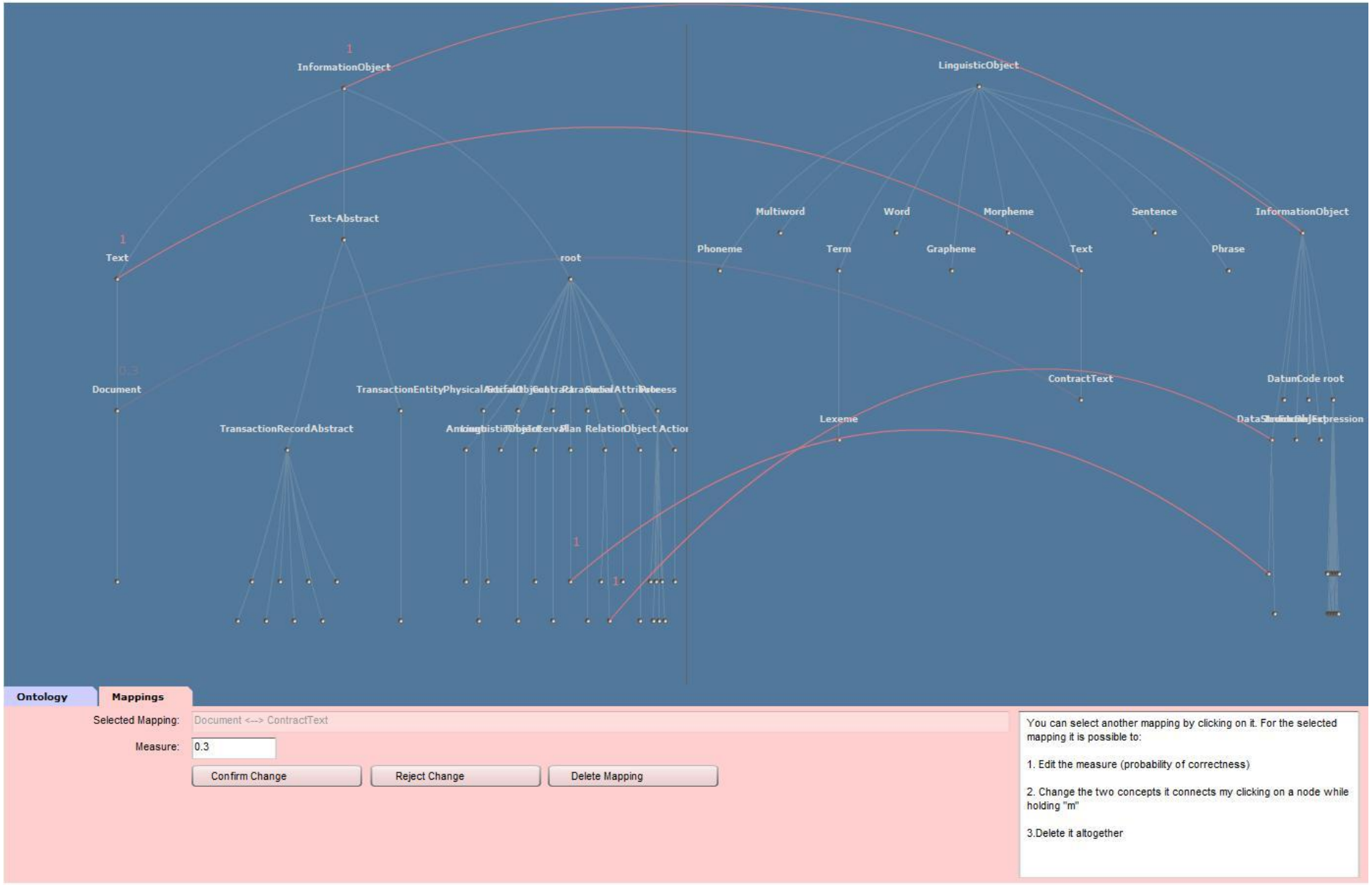

Figure A.5: Manually edited alignment.

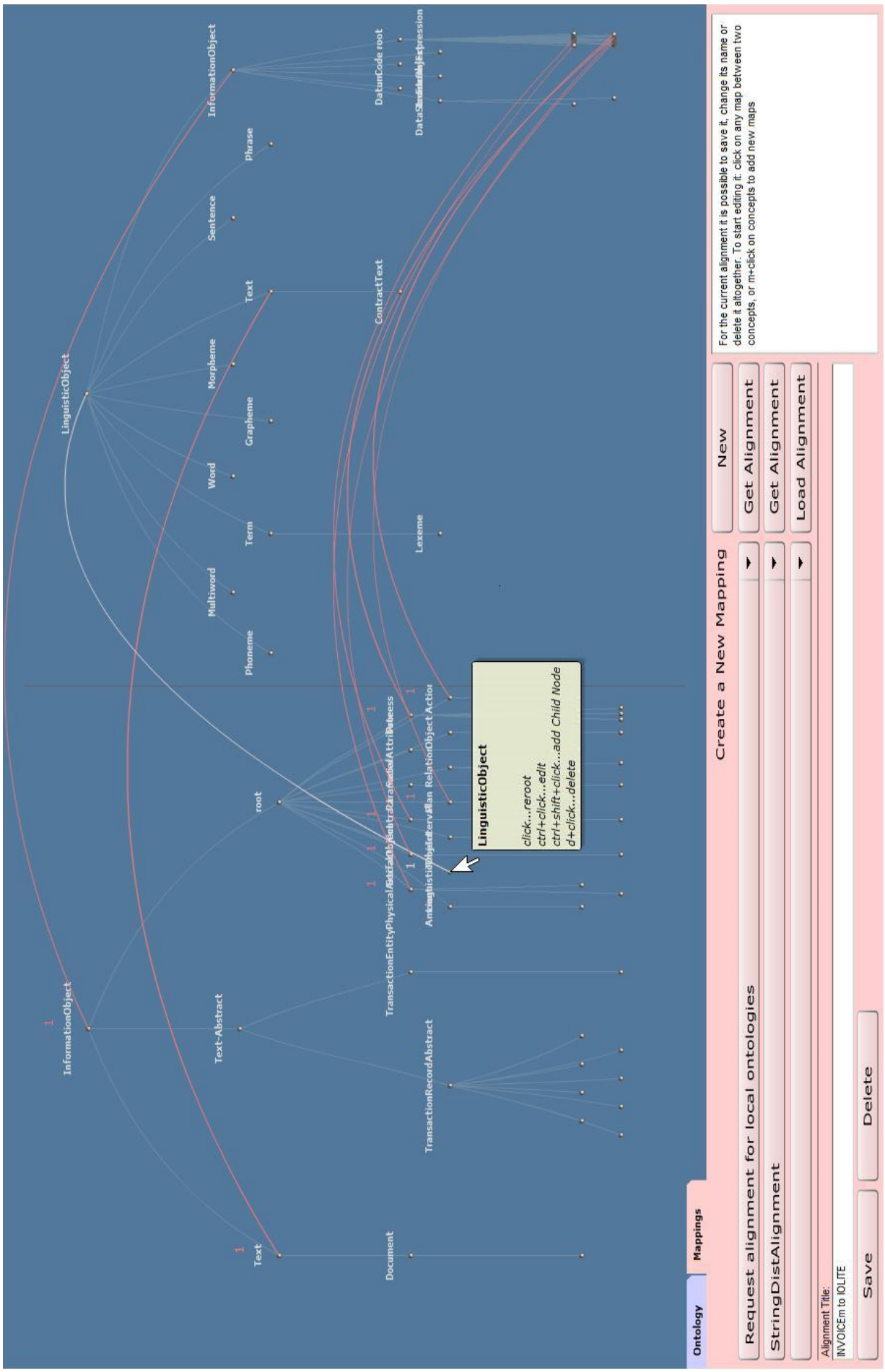

Figure A.6: Alignment provided by the StringDistAlignment method for another context.

Bibliography

[Jos06] Jose Manuel Gómez and Claire Daviaud and Berta Morera and Richard Benjamins and Tomás Pariente Lobo and Germán Herrero Cárcel and Gloria Tort. Analysis of the pharma domain and requirements. deliverable 8.1.1, NeOn, 2006.

[Mar06] Marta Iglesias and Caterina Caracciolo and Yves Jaques and Margherita Sini and Francesco Calderini and Johannes Keizer and Fynvola Le Hunte Ward and Malvina Nissim and Aldo Gangemi . User requirements. deliverable 7.1.1, NeOn, 2006.

[Mar08] Marko Grobelnik and Boštjan Pajntar and Dunja Mladenić. Reasoning with context. deliverable 3.2.2, NeOn, 2008.

[wik08] Visualization - computer graphics. *Wikipedia*, 2008.