D5.3.2 – The Dicode multi-perspective ontology engineering tool (enhanced version)

Due date of deliverable: 31 December 2012
Actual submission date: 18 January 2013
Lead Partner: UOL
Contributing Partners: UOL, CTI, UPM, NEO

Nature: □ Report □ Prototype □ Demonstrator □ Other

Dissemination Level:
□ PU: Public
□ PP: Restricted to other programme participants (including the Commission Services)
□ RE: Restricted to a group specified by the consortium (including the Commission Services)
□ CO: Confidential, only for members of the consortium (including the Commission Services)

Keyword List: Ontology engineering, Knowledge capture, Dialogue for ontology authoring, Semantic augmentation, Semantic data browsers, Linked Open Data, Exploratory search.

The Dicode project (dicode-project.eu) is funded by the European Commission, Information Society and Media Directorate General, under the FP7 Cooperation programme (ICT/SO 4.3: Intelligent Information Management).
The Dicode Consortium

Computer Technology Institute & Press “Diophantus” (CTI) (coordinator), Greece

University of Leeds (UOL), UK

Universidad Politécnica de Madrid (UPM), Spain

Neofonie GmbH (NEO), Germany

Image Analysis Limited (IMA), UK

Biomedical Research Foundation, Academy of Athens (BRF), Greece

Publicis Frankfurt Zweigniederlassung der PWW GmbH (PUB), Germany
This deliverable provides an update on the suite of semantic services developed in WP5 since the last deliverable D5.3.1. These services aim to make it easier for domain experts to be involved in the creation and use of ontologies and for users to interact with semantically enriched data sets. There are two main parts in this deliverable.

Firstly, the main features of the extension (called Perico) of the ROO tool, consisting of a dialogue for ontology authoring, are described. Perico, a dialogue framework, enables the user (i.e. a domain expert) to better understand the semantic feedback provided by the tool while authoring the ontology. Perico extends Entendre (presented in D5.3.1) by adding interactive features for communicating with ontology authors to facilitate the knowledge elicitation process.

Secondly, a semantic data browser, Pinta, has been developed for the end users to effectively explore the semantically enriched datasets. Conceptually, Pinta is a progression from Augmentor (reported in D5.3.1). While Augmentor was tailored for a specific domain, Pinta is a generic shell for a semantic data browser that supports exploratory search through heterogeneous content. An experimental study is reported using an instantiation of Pinta for exploratory search tasks related to preparing a creative brief for advertising a product (i.e. suitable for Use Case 3).

The prototypes accompanying this deliverable are available online:
- Perico: https://sourceforge.net/projects/perico/
- Pinta: http://imash.leeds.ac.uk/services/pinta/
Table of Contents

1 Introduction ........................................................................................................................ 5

2 Intuitive Ontology Engineering Tool ROO: Dicode Extension ........................................ 5
   2.1. Overview ....................................................................................................................... 5
   2.2. Perico: Dialogue-based Interaction for Ontology Authoring ....................................... 6
   2.3. Functional Components of Perico ............................................................................... 6
   2.4. Implementation ............................................................................................................ 9
   2.5. Interaction Example .................................................................................................... 10
   2.6. Summary: ROO Extension in Dicode .......................................................................... 12

3 Semantic Support for Exploratory Search ...................................................................... 14
   3.1. Using Linked Data to Support Exploratory Search ...................................................... 14
   3.2. Architecture of Pinta .................................................................................................. 15
   3.3. Instantiation: MusicPinta ............................................................................................ 17
   3.4. Experimental Study ..................................................................................................... 21
   3.5. Findings ....................................................................................................................... 23
      3.5.1 Task Success ........................................................................................................... 23
      3.5.2 Browsing Behaviour ............................................................................................... 26
      3.5.3 Usability Feedback ................................................................................................ 28
   3.6. Discussion .................................................................................................................. 28

4 Summary and Conclusions ............................................................................................. 30

References ........................................................................................................................... 31
1 Introduction

Ontologies have been proposed and studied in the last couple of decades as a way to capture and share people's knowledge about the world in a way that is processable by computer systems. Ontologies have the potential to serve as a bridge between the human conceptual understanding of the world and the data produced, processed and stored in computer systems. However, ontologies so far have not gathered widespread adoption, failing to realise the original vision of the semantic web as a next generation of the World Wide Web (where everyone would be able to contribute and interlink their data and knowledge as easily as they can contribute and interlink their websites).

One of the main reasons for the lack of adoption of ontologies is the steep learning curve for authoring them; most people find it too difficult to learn the syntax and formal semantics of ontology languages. Most research has tried to alleviate this problem by finding ways to help people collaborate with knowledge engineers when building ontologies. This approach however, requires the wide availability of knowledge engineers, who are, in practice, scarce. Recent research has started looking for intuitive tools to directly capture knowledge from domain experts. The first part of this deliverable (Section 2) presents an update of the multi-perspective ontology engineering tool since the last DICODE deliverable D5.3.1. A dialogue framework, Perico, extends ROO (an Intuitive Ontology Engineering Tool developed in Leeds) by helping the user understand the semantic feedback provided by the tool.

Another reason for the lack of adoption of ontologies is that end users do not appreciate their benefits. To this end, the second part of this deliverable (Section 3) presents an experimentation of a semantic data browser, Pinta, which has been developed and instantiated for exploratory search tasks related to writing a creative brief for advertising (i.e. suitable for Use Case 3). The domain of music is chosen due to the availability of appropriate Linked Open Data sources. The outcome of the study has provided insights into further work and demonstrated the added value of semantics if an easy to use browser is available for exploring the vast semantic datasets underpinned by a domain ontology. Conceptually, Pinta is a progression from Augmentor (reported in D5.3.1). While Augmentor was tailored for a specific domain, Pinta is a generic shell for a semantic data browser that supports exploratory search through heterogeneous content.

2 Intuitive Ontology Engineering Tool ROO: Dicode Extension

2.1 Overview

Ontology Authoring is the process of developing ontologies. This process is performed by one or more ontology authors, people who directly contribute to the formal capture of knowledge in the form of ontological structures. Most ontology authoring tools are designed to be used by knowledge engineers (i.e. specialists with appropriate knowledge engineering skills). However, they may lack the necessary domain expertise to create the relevant ontologies [Funk et al 2007]. Finding knowledge engineers who are competent in the target knowledge domain is a luxury.

The most common approach is, while knowledge engineers encode the ontology using available ontology authoring tools, to ask domain experts (i.e. people who are subject matter experts) to provide relevant knowledge sources or apply knowledge elicitation techniques to discover information directly from the expert. Apart from creating an extra layer of
bureaucracy in the development cycle [Klischewski 2006], this approach can hinder the ontology construction process and may have a negative impact on the quality of the resultant ontology (e.g. poor documentation, inconsistency of terminology used, incorrect or incomplete knowledge constructs).

Hence, there is a need for an intuitive ontology authoring tool to enable domain experts to directly contribute their knowledge to ontologies without requiring extensive training in knowledge engineering or ontology authoring tools. Existing research has shown strong support for using a Controlled Natural Language (CNL) (an engineered language, subset of a natural language and computer processable) as a basis for making ontology authoring tools more intuitive [Davis & Iqbal 2008, Funk et al 2007, Kaljurand & Fuchs 2006, Schwitter 2010]. CNLs enable the syntactic expression of knowledge in a way that can be automatically converted into an appropriate logical formalism [Dolbear et al 2007]. CNLs seem to be a step in the right direction; however, domain experts still need support concerning other aspects relevant to ontology authoring [Kovacs et al 2006].

As reported in Deliverable 5.3.1, UOL has developed Entendre, a new semantic analysis framework, to extend ROO (a CNL-based Ontology Authoring tool). The purpose of Entendre is to (i) identify the possible cases that may happen when new axioms are added to an ontology and (ii) provide immediate feedback (e.g. warning of inconsistency or redundancy) upon entering ontology axioms. However, an evaluative study [Denaux et al., 2012], presented in Deliverable D5.3.1, showed that it is not sufficient only to understand the users but also to make users understand what is conveyed by the feedback. Hence, further work has been carried out in extending ROO by adding a dialogue agent to support the knowledge elicitation and ontology authoring process. The resulting generic dialogue framework, Perico, will be described in the following section.

2.2. Perico: Dialogue-based Interaction for Ontology Authoring

Perico extends Entendre with a new layer of input analysis to enable more advanced ontology authoring interactions. In order to make this extension generic enough to support a wide variety of ontology authoring interactions, we will use dialogue-discourse analysis to track the authoring interaction. This allows us to build on an extensive body of research on dialogue systems. Summarising the requirements for such a dialogue framework, it must:

- model the interaction from the perspective of one of the participants in the dialogue: the ontology authoring system participant;
- be ontology-driven (i.e. any agreed knowledge should be stored in an ontology language);
- describe ontology authoring interactions based on a standard representation, while taking into account ontologies and their reasoning services; it must enable the description and execution of non-trivial ontology authoring interactions;
- allow ontology authoring interactions based on a CNL;
- extend the Entendre framework for input analysis and feedback;
- provide reusable and composable interactions for enabling advanced ontology authoring interactions.

2.3. Functional Components of Perico

Perico follows the general architecture of dialogue systems. It uses a hub-like design where the dialogue is mediated by a PericoAgent, which communicates with the components of the dialogue pipeline as shown in Figure 1.
Figure 1: Functional view of Perico showing the main components, the services they provide and the functional dependencies between the components.

The rationale for using a hub-like design is that the PericoAgent is the only component that controls the dialogue state. All Perico components can provide services that are independent of the current dialogue state and the pipeline components are not aware of each other.

Perico does not impose major restrictions regarding the Input Recogniser and Output Presenter. However, since Entendre focuses on textual input analysis, in practice these two components will also be text-based. The Input Interpreter component in Perico reuses the Entendre pipeline of lexical, syntax and semantic analysers. Furthermore, in Perico we add a new type of analyser called a Discourse Analyser, a component that extends a syntactic reading and derives a Discourse Reading of the input (i.e. it attaches dialogue move annotations to the input). The output of the Input Interpreter is based on the dialogue moves in the discourse reading, which are aggregated into a Functional Segment. Figure 2 gives an overview of the input interpretation process in Perico.
Figure 2: UML Activity Diagram showing the workflow for the Input Interpreter component in Perico. The workflow is used for analysing inputs from other dialogue participants and converting them into functional segments of the dialogue.

The **Dialogue Manager** component in Perico consists of two subcomponents: the knowledge grounder and the dialogue plan executer. The knowledge grounder may use the semantic analysis from Entendre to decide whether to add or remove knowledge to the reference ontology. The knowledge grounder can also inject grounding plans to the execution agenda to provide feedback about how messages have been interpreted, or to request confirmation. The dialogue plan executer is responsible for determining Perico's next dialogue move based on the current dialogue state, which includes a history of functional segments and the current execution stack (the set of dialogue plans that are being executed). The output of the Dialogue Manager is a functional segment (consisting of one or more dialogue moves) that needs to be performed.

Finally, the **Output Generator** converts a functional segment proposed by the dialogue manager into an output that can be understood by the other dialogue participants. Since functional segments consist of both discourse and domain information, the output generator needs to be able to convert both types of information. For domain information, Perico can reuse the rendering syntax (e.g. a CNL-generator). For discourse information, Perico uses simple text-based templates since we are not aware of a CNL that we can reuse for this purpose.

An informational view (extending ISO-DIS-24617-2) of a dialogue in Perico is depicted in Figure.

---

2.4. Implementation

Scala\(^2\) has been used for the implementation. An API was created for dialogue engineers to define task trees that can be executed by Perico. Hooks are provided for dialogue engineers to define custom discourse readings based on the current dialogue state. These hooks are used, for example, to correctly analyse help requests by the user (which are not valid CNL statements but are valid moves in the dialogue).

The Perico implementation also includes a web-based chat-like interface where users can see the history of the dialogue and enter new inputs. A screenshot of the interface is shown in Figure 4.

\(^2\) Scala is a strongly typed language that combines the Object Oriented and Functional programming paradigms. The syntax of Scala is similar to Java and it also compiles onto JVM classes, which means it can be easily combined with existing Java programs. For more information on Scala, see [http://www.scala-lang.org/](http://www.scala-lang.org/).
2.5. Interaction Example

This section presents a representative dialogue between Perico and a simulated ontology author. The dialogue was created by using the implemented Perico dialogue agent. The example dialogue is using the Leeds Points of Interest ontology that we used to evaluate the Entendre feedback. We assume that the user has browsed through the ontology (and thus is aware of some of the entities in the ontology) and is familiar with some examples of Rabbit sentences. We further assume that the author has received the task to enter an initial fact (Bus Line 95 has end point Bodington Hall). The corresponding dialogue follows.

```
fs1 Perico: Tell me a new fact to add to the ontology.
fs2 Perico: You can do one of the following:
    ^ Enter a new fact or
    ^ Enter Skip if you do not want to tell a new fact or
    ^ Enter Help to receive a hint.

fs2 OA: Bus Line 95 has end point Bodington Hall.

fs3 Perico: This fact is indeed new.
fs4 Perico: Normally, entering a new fact enables me to make further inferences but I cannot do this in this case.
```

Figure 4: Screenshot of the web interface implementation of Perico.
Perico: Can you tell me something that I should be able to infer?

You can do one of the following:

- Enter a new fact or
- Enter Skip if you prefer not to answer this question or
- Enter Help to receive a hint.

OA: Bus Line 95 stops near Bodington Hall.

Perico: Actually, "Bus Line 95 stops near Bodington Hall" is a new fact.

Tell me more general facts about Bus Route or Hall of Residence or has end point or stops near .

Perico: You can do one of the following:

- Enter a new fact or
- Enter Skip if you prefer not to answer this question or
- Enter Help to receive a hint.

General facts relate concepts to each other. For example:

- Every Car is a kind of Vehicle.
- Father and Mother are mutually exclusive.
- Every School has a part a Building that has purpose Education.
- Every car has at least three Wheels.

General facts can also define relationships. For example:

- The relationship is directly connected to is a special type of the relationship is connected to .
- The relationship contains is the inverse of is contained by .
- For the relationship eats , nothing eats itself.

OA: Help.

Perico: Facts are simple sentences such as:

- Every Car is a kind of Vehicle.
- Car is a concept.
- Every car has at least three Wheels.

OA: Every Student Hall has purpose Accommodation.

Perico: This general fact is new.

I can now infer that Bodington Hall has purpose a Accommodation.

Tell me more specific facts about Bodington Hall .

Perico: You can do one of the following:

- Enter a new fact or
- Enter Skip if you prefer not to answer this question or
- Enter Help to receive a hint.

OA: Skip.

Tell me a general fact about stops near .

Perico: You can do one of the following:

- Enter a new fact or
- Enter Skip if you prefer not to answer this question or
- Enter Help to receive a hint.

OA: Help.
Perico: I know that:
  * has end point is a relationship.
  * Every Transport Route has end point exactly 1 POI.
You can use general facts to further define the meaning of has end point. For example you could enter facts similar to:
  * The relationship has end point is a special type of the relationship has purpose.
  * The relationship has end point is the inverse of stops near.
  * For the relationship has end point, nothing has end point itself.

OA: The relationship has end point is a special type of the relationship stops at.

Perico: This general fact is new.

The example dialogue shows how the dialogue plan provides support to novice authors in the following ways:

- Various elicitation moves guide the user by narrowing down the type of knowledge that the author should enter. The dialogue starts with a default elicitation move in fs1. In fs5, the ElicitExpectedInference plan is executed after the N axiom from fs2. In fs9 and fs21, the ElicitScopedTBoxAxiom plan is executed after an expected inference is detected. Finally in fs18, an ElicitScopedABoxAxiom is executed.
- The dialogue show various instances where the InformElicitationAnswerOptions is executed (fs2, fs6, fs10, fs19 and fs22). This plan is always executed right after an elicitation move.
- The author makes use of the available options by providing new facts, asking for help (fs11, fs13 and fs23) or skipping an elicitation move (fs20). The system provides hints to the author in fs12, where the HintForTBoxRbtInScope plan exploits the knowledge about the expected types of Rabbit sentences to only suggest T-Box sentences. In fs14, since the author is still asking for help, the system falls back to the default plan for helping users formulate facts. In fs24, the system again provides help based on a HintForTBoxRbtInScope; however, note that since the scope now is a single relationship (stopsNear), the hint can be more focused on relationships.
- Finally, the new simplified IntegrationFeedbackProvider results in functional segments fs3-4, where feedback is given about the N axiom from fs2. In fs8, feedback is given for the N-axiom from fs7; Perico provides an even shorter feedback in this case since it knows that fs7 is an expected inference from fs2. Functional segments fs16-17 provide feedback about an N+- axiom from fs15.

2.6. Summary: ROO Extension in Dicode

The work on ROO aims at enabling domain experts to actively contribute to the process of ontology authoring. We have focused on using the tool to support domain experts formulate their knowledge using expressive ontology languages, enabling them to contribute to heavyweight ontologies. The research followed an explorative and iterative approach to developing principled tool support approaches. Starting with syntactic support and using a Controlled Natural Language, we explored providing more holistic support by including support for ontology engineering methodologies. We also added support for becoming aware of logical consequences of new inputs and providing understandable tool support. Finally, we defined a framework to enable the analysis of ontology authoring interactions.
using dialogue systems. The main contribution of this work is the definition and development of an intelligent tool to provide support for novice ontology authors.

This research work has explored the area of intelligent tool support for enabling the direct involvement of domain experts in the process of building ontologies. We have iteratively formalised, implemented and evaluated this tool support in order to make ontology authoring more accessible. We started from the hypothesis that holistic intelligent tool support can enable the active involvement of domain experts in ontology authoring. This hypothesis is in line with existing work which suggested the application of an ontology authoring methodology that uses a Controlled Natural Languages in order to involve domain experts in ontology authoring.

The first version of the ROO ontology authoring tool was designed and developed prior to Dicode. It integrated NLP, parsing and GUI design techniques to provide appropriate tool support for ontology authors without knowledge engineering experience. ROO included intelligent support for formulating knowledge in a way that is understood by the CNL parser and tool support for adhering to the ontology authoring methodology.

In Dicode, ROO was employed to define the Dicode Ontology (DON). The practical experiences with ROO identified further aspects that could hinder ontology construction. One of these aspects was the need for improved disambiguation handling and feedback. Another aspect was the need to make knowledge engineers more aware of the logical consequences of their ontology authoring actions.

We addressed these problems by defining a framework called Entendre for facilitating the systematic analysis of ontology author's textual inputs (Deliverable 5.3.1). The Entendre framework formally defines the main syntactic analyses of an input that can be performed by an ontology authoring system and the main results of such analyses. The framework provides a categorisation of lexical and syntactic analysis results which guide the performing of disambiguation, as well as the feedback generation. We performed an evaluation with 5 novice ontology authors and 5 experienced knowledge engineers to find out what ontology authors thought of the interactive feedback, whether they understood it and whether they would know what to do after receiving the feedback. We found that both novice and experienced ontology authors liked the feedback and though it was useful and informative [Denaux et al, 2012]. The evaluation reinforced earlier results that indicated that effective interaction between the ontology authoring system and domain experts is needed to provide better feedback and guide ontology author’s actions.

This challenge was addressed in the work presented in this deliverable, where we defined Perico, a dialogue framework for ontology authoring that can be used for describing and executing ontology authoring interactions in terms of dialogue moves. The presented framework adds a layer of discourse analysis (on top of Entendre) to formalise ontology authoring interactions. We reused standards from the area of dialogue systems and integrated them with ontology engineering concepts in order to define the Perico framework. We validated Perico by using it to (i) formalise an existing ontology authoring interaction and (ii) adapt and formalise an existing ontology authoring interaction to provide better support to novice ontology authors. A final validation of the framework consisted of the implementation of the framework itself (reusing the Entendre framework) and of the formalised dialogue systems.
The following software outputs implement the ROO extensions, which were developed in Dicode:

- the **Entendre** library (presented in Deliverable 5.3.1) for analysing textual ontology authoring inputs, which understands inputs in both Rabbit and Manchester Syntax. This library also performs input-axiom analysis in order to provide textual feedback about the logical consequences of adding an input to an existing ontology.
- the **Perico** dialogue toolkit (presented in this deliverable), which provides an API for defining dialogue systems where the shared knowledge of the participants results in an ontology. The toolkit is integrated into the Entendre library to enable dialogue interactions based on the controlled natural language Rabbit.

### 3 Semantic Support for Exploratory Search

While ROO enables ontology authoring, it does not provide appropriate means for exploring large semantic datasets. This challenge was addressed by developing intelligent services for semantic data exploration utilising data sets from Linked Data. This section presents the implementation of Pinta, a semantic data browser shell, which combines state of the art semantic web technologies for semantic augmentation, semantic query and representation. The main goal of Pinta is to enable users easily tap into resources built from the Web and, in particular, explore the use of the Linked Data paradigm.

#### 3.1. Using Linked Data to Support Exploratory Search

The Linking Open Data initiative has engaged various communities to share their data for sustainable usage based on semantic web technologies [Heath & Bizer 2011]. Increasingly, applications have been developed to make use of these available resources. Although the motivation behind is to enable automated integration of data from different sources, it is the humans who are the consumers of these Linked Data. The human usage is the ultimate judgment for the benefit of information integration and linking [Brambilla & Ceri 2012]. There are arguments that Linked Data can be utilised to enable user-oriented exploratory search systems for the future internet, e.g. [Waitelonis et al 2010]. Stepping on such arguments, we apply a systematic experimental methodology to examine user exploratory search behaviour when interacting with a semantic data browser over linked semantic data.

In contrast to regular search, exploratory search gives a more complete overview of a topic. Exploratory search is open-ended, multi-faceted, iterative in nature and is commonly used in scientific discovery, learning, and sense making [Marchionini 2006, White & Roth 2009]. Exploration demands more time, effort and creativity from the user but rewards the user with deeper knowledge [Marchionini 2006]. Exploratory search is particularly beneficial for ill-structured problems and more open-ended goals, with persistent, opportunistic, iterative query processes. Exploratory tasks inherently have uncertainty, ambiguity and discovery as common aspects [Marchionini 2006]. Linked semantic data appears to offer a great potential for exploratory search. For example, earlier studies suggested that tags [Kammerer et al 2009] or some form of presentation of the knowledge space structure [Qu & Furnas 2008] could benefit browsing and learning. The work presented here starts from these claims (which did not exploit semantics and linked data), and examines the role of semantic tags and their effect on browsing and learning in a class of applications called semantic data browsers.

In Dicode, to enable better synergy between human and machine intelligence, semantic data browser offers an effective interface between decision makers and semantically marked up datasets. Furthermore, especially for Use Case 3 (PUB), the time is ripe to start...
experimentation with exposing linked semantic data from social media (e.g.
DBpedia/product reviews) to end users. This section presents such a case study, and points
at the potential of semantic data browsers to facilitate exploratory search. A semantic data
browser shell, called Pinta, providing uni-focal faceted exploration of linked semantic data,
was developed. An instantiation of Pinta in a Music domain (‘MusicPinta’) linking music
datasets from the Web of data and social content from Amazon reviews about musical
instruments was set up for a systematic study to get an initial insight of main issues related
to exploratory search in linked data, an analysis of the role of semantics and an indication of
heuristics for adding semantic nudges [Thakker et al., 2012] to facilitate user exploration.

3.2. Architecture of Pinta

Figure depicts the generic three-layer architecture for Pinta: (i) the Data Layer including
knowledge sources and content, (ii) the Processing Layer including modules for semantic
augmentation and query, and (ii) the Presentation Layer for content browsing.

![Generic architecture for Pinta](image)

**Data Layer**

This layer contains domain specific ontological knowledge resources and content assembled
from the Web (Linked Data and other sources, as appropriate). The knowledge resources
consist of graphs of ontological concepts relevant to the domain of interest. They provide
the foundation for semantic augmentation of the content and the structure for semantic
trajectories for browsing in the Presentation Layer. The content in Pinta is in textual format
and can be assembled from more than one online platform, e.g. blogs, reviews, comments
(e.g. [Thakker et al. 2012]).

**Processing Layer**

This layer has two main services: (i) semantic augmentation of the assembled content and (ii)
semantic queries to retrieve content for the Presentation Layer. Semantic augmentation (also
known as semantic tagging) is a process of attaching semantics (in the form of "concepts" in
an ontology) to a selected part of a piece of text. The augmented text can be automatically
interpreted by another software component. The semantic augmentation module in Pinta
includes: Semantic Repository, Information Extractor and Semantic Indexer. The Semantic Repository (using OWLIM\(^3\)) combines the functionality of an RDF-based DBMS and an inference engine. The Information Extractor (using GATE\(^4\) - General Architecture for Text Engineering) produces annotated sets of extracted entities with offset, ontology URI and type information. The Semantic Indexer (using Sesame SPARQL API\(^5\)) converts these annotated sets to RDF triples.

The semantic augmentation process is performed in two stages: configuration (performed manually and offline) and processing (performed automatically and on-the-fly):

- At the configuration stage, ontologies for the domain are selected and a list of known entities (called Gazetteers) is built. Mapping from ontologies to gazetteers exploits the label properties (rdfs:label, skos:prefLabel, skos:altLabel) in the various ontology standards (such as RDFS and SKOS). The gazetteers are prepared for each class (rdfs:class and owl:Class) from the ontologies. A gazetteer list is mapped to the class from an ontology by giving the same name and contains known entities in the form {label, uri}. The GATE\(^5\) JAPE grammar component at this stage contains linguistic filtering rules for pre-processing of the content and post-processing of the results from Information Extractor.

- At the processing stage, the pre-processing JAPE grammar is applied on the text input to extract the part of text for augmentation. Linguistic processing techniques (such as sentence detection and splitting, tokenisation, part-of-speech tagging, sentence chunking and word stemming) are applied on the text. This process produces a surface form (i.e. linguistically annotated text with nouns, verbs, adverbs and adjectives). The gazetteer component matches the ontological labels to the surface forms and attaches an ontology concept URI to the surface forms. The Semantic Indexer then converts the annotation sets into semantic format (i.e. set of triples), checks the existing index for the content and stores new or updated indices in the semantic repository. Semantic Queries take term(s) or concept(s) as keywords and output information relating to the matching concept(s) and content(s). Combined with the Semantic Repository, Semantic Queries implement various concept/content lookup functionalities to find related and relevant concept(s) or content(s) from the Semantic Repository. The query processor component is built using the Sesame SPARQL API and extensively utilises the SPARQL queries.

**Presentation Layer**

This layer provides a front-end for the output of semantic queries from the Processing Layer. The layout template for a focus entity (Figure) includes three main facets and a description (at the top) extracted from the knowledge data sets for the focus entity: (i) Facet 1 includes facts about the focus entity; (ii) Facet 2 includes terms related to the focus entity; and (iii) Facet 3 shows content related to the focus entity. Facts and related terms for the focus entity consist of triples from the Semantic Repositories, which include hierarchy links (denoted as “is a kind of”), membership (denoted as “is a type of”) and object properties (denoted as “other”). Hyperlinks to content and ontology entities are provided to further details for the retrieved objects.

\(^3\) [http://www.ontotext.com/owlim](http://www.ontotext.com/owlim)
\(^4\) [http://www.openrdf.org/](http://www.openrdf.org/)
\(^5\) [http://gate.ac.uk/](http://gate.ac.uk/)
3.3.  Instantiation: MusicPinta

The music domain has been selected for an instantiation of Pinta, as the Web is rich in music-related content. As of 2011, there were at least 13 datasets identified, with a diverse range of concepts covering instruments, performances, artists and music genres. The data sets used for Pinta’s case study in music include:

- DBpedia: this includes the part about musical instruments and artists. This dataset is extracted from dbpedia.org/sparql using CONSTRUCT queries.
- DBTune: includes music-related structured data made available by DBTune.org in linked data fashion. Among the datasets on DBTune.org we utilise: (i) Jamendo - a large repository of Creative Commons licensed music; (ii) Megatune - an independent music label; and (iii) MusicBrainz - a community-maintained open source encyclopedia of music information.
- Amazon reviews of musical instruments shown in Pinta.

All datasets, except the reviews, were available as RDF datasets and utilise Music ontology as schema. The Amazon reviews were converted in RDF using Pinta’s semantic augmentation. Table 1 presents the datasets that support exploration concerning instruments, artists, albums, tracks and records, and reviews.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Supported in Dataset(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Musical Instruments</td>
<td>MusicBrainz, DBpedia</td>
</tr>
<tr>
<td>Artists</td>
<td>MusicBrainz, DBpedia, Jamendo, Megatune</td>
</tr>
<tr>
<td>Albums/Tracks/Records</td>
<td>MusicBrainz, Jamendo, Megatune</td>
</tr>
<tr>
<td>Customer reviews</td>
<td>Amazon Reviews</td>
</tr>
</tbody>
</table>

The datasets coming from DBTune.org (such as MusicBrainz, Jamendo and Megatunes) already contain the “sameAs” links between them for linking same entities (see Figure ). We utilise the “sameAs” links provided by DBpedia to link MusicBrainz and DBpedia datasets.
This way, the DBpedia is linked to the rest of the datasets from DBtune.org. This interlinking allows Pinta to benefit from the abstract descriptions of entities and their images – which, in DBpedia, are generally richer.

![Figure 7: Linked data semantic datasets used in MusicPinta.](image)

DBpedia is also very rich in terms of categorisation of musical instruments. For example, the categories separate instruments according to their country of origin/use. The MusicPinta datasets has 2.4 Million entities and 19 Million triple statements, taking 2GB physical space, including 876 musical instruments entities, 71k performances (albums, records, tracks), 188k music artists. We have also added Amazon reviews as an example of social content in MusicPinta. Figure and Figure show some screenshots of the MusicPinta interface.
Bouzouki

The bouzouki is a musical instrument with Greek origin in the lute family. A mainstay of modern Greek music, the front of the body is flat and is usually heavily inlaid with mother-of-pearl. The instrument is played with a plectrum and has a sharp metallic sound, reminiscent of a mandolin but pitched lower. There are two main types of bouzouki. The three-course has three pairs of strings (known as courses), and the four-course has four pairs of strings.

### Facts about “Bouzouki”

<table>
<thead>
<tr>
<th>Bouzouki</th>
<th>is a kind of</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Plucked string instruments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>String instruments</td>
</tr>
<tr>
<td>Bouzouki</td>
<td>is type of</td>
<td>Instrument</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>Cypriot musical instruments</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>Greek musical instruments</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>Necked bowl lutes</td>
</tr>
<tr>
<td></td>
<td>subject</td>
<td>Turkish loanwords</td>
</tr>
</tbody>
</table>

### Terms related to “Bouzouki”

<table>
<thead>
<tr>
<th>Lute</th>
<th>is a kind of</th>
<th>Bouzouki</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sitar</td>
<td>Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Xalam</td>
<td>Bouzouki</td>
<td></td>
</tr>
<tr>
<td>(khalam)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oud</td>
<td>Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Cittern</td>
<td>Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Rebab</td>
<td>Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Morten Musicus performing (recorded on album Mēn...</td>
<td>instrument Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Lanvall performing (recorded on album The Grand De...</td>
<td>instrument Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Warren Ellis performing (recorded on album The Lyr...</td>
<td>instrument Bouzouki</td>
<td></td>
</tr>
<tr>
<td>Morten Musicus performing (recorded on album Mēn...</td>
<td>factor Bouzouki</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8: Extracted content from the information about bouzouki.
An electric guitar is a guitar that uses a pickup to convert the vibration of its strings into electrical impulses. The most common guitar pickup uses the principle of direct electromagnetic induction, but others use piezoelectricity or optical sensors. The signal generated by an electric guitar is too weak to drive a loudspeaker, so it is amplified before sending it to a loudspeaker.

**Facts about Electric guitar**

<table>
<thead>
<tr>
<th>Electric guitar</th>
<th>is a kind of</th>
<th>Electric lap steel guitar</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Instrument</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lap steel guitar</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plucked string instruments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Steel guitar</td>
</tr>
<tr>
<td></td>
<td></td>
<td>String instruments</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Electric guitar</th>
<th>is type of</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>subject</td>
<td>1931 introductions</td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td>American inventions</td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td>American musical instruments</td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td>Amplified instruments</td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td>Electric guitars</td>
<td></td>
</tr>
</tbody>
</table>

**Terms related to Electric guitar**

- Jay Tausig performing (recorded on album D’arcana)
- Martin Moro performing (recorded on album Step by ...
- Andrew Gold performing (recorded on album Karla Bo...
- Nils Frykdahl performing (recorded on album Family...
- Mac Aladdin performing (recorded on album The Beek...
- Aleksandr Ljapin performing
- Gary Louris performing (recorded on album You Gott...
- Pete Anderson performing (recorded on album Short ...

**Content related to Electric guitar**

- Epiphone LP Special II Les Paul Electric Guitar, Vintage Sunburst - Super value for the money. If you’re on a budget, this is an outstanding

  Tags: Electric guitar| Electric guitar| Electric guitar| Electric guitar| Guitar|

- Epiphone LP Special II Les Paul Electric Guitar, Vintage Sunburst - Good Guitar

  Tags: Guitar| Guitar| Electric guitar| Electric guitar| Electric guitar| Electric guitar| Guitar|


  Tags: Electric guitar| Electric guitar| Electric guitar| Electric guitar| Electric guitar| Electric guitar| Guitar|
3.4. Experimental Study

To get an insight of how MusicPinta can support exploratory search through linked semantic data, we conducted an experimental study within an advertising domain (i.e. can be related to Use Case 3 in Dicode). The study considers conducting exploratory search tasks through a collection of heterogeneous sources within the context of conducting research for preparing a creative brief for a product. It is assumed that the person conducting the product research (identifying main characteristics of a product or finding interesting content related to the content) is unfamiliar with the product.

The study addressed the following research questions:

Q1: How well can users without domain knowledge perform exploratory search tasks using MusicPinta, and what is the benefit/drawback of its features?

Q2: What further improvements have to be addressed to make MusicPinta (and semantic data browsers in general) suitable for exploratory search tasks.

The experimental study followed a within-subjects design method with two exploratory search tasks as the within-subjects factor. To avoid bias by the ordering of tasks, participants were divided equally into two groups; one group performed Task 1 then Task 2, and the other group performed Task 2 then Task 1.

Participants. The study involved 12 participants recruited on voluntary basis (a compensation of £15 Amazon vouchers was paid). Half of the participants were native speakers and the other half spoke and communicated in English fluently. All participants had IT background, experience in web search, and most participants had good experience in data analysis. Half of the participants have visited sites with music information regularly, while the rest did this only occasionally; 4 participants listened to online music sites daily. Half of the participants indicated that they currently practiced musical instruments (none of these were instruments they had to research in the study, see below).

Method. Each participant attended an individual session, conducted and observed by an experimenter, for about an hour, including:

- **Pre-study questionnaire** [5 min] - collecting information about the user profile and testing his/her domain awareness.
- **Introduction to MusicPinta** [10 min] – introduction to the main features of the system using tenor saxophone as an example (the participants followed a script to do so)
- **Task 1** [20 min] - identifying distinctive characteristics of the musical instrument bouzouki [15min]. After completion, filling in a task difficulty questionnaire [5 min].
- **Task 2** [20 min] – identifying the usage and features of the musical instrument electric guitar [15min]. After completion, filling in a task difficulty questionnaire [5 min].
- **Post-study questionnaire** [10 min] – testing again the participant’s domain awareness and gathering usability feedback.
- **Brief interview** [5 min] – noting the overall impression of MusicPinta.

The participants were asked to think aloud; the experimenter kept notes of the interaction noting any interesting comments made. All browser clicks were logged for further analysis.

Tasks. To design the study tasks, we have followed the main characteristics of exploratory search tasks summarised in [Wildemuth & Freund 2012]: the main goal is learning and/or investigation of a musical instrument; there is a low level of specificity about the information needed and how to find it; search is open ended, requires finding several items and involves
a degree of uncertainty; tasks are ‘not too easy’ and include multiple facets. The study required participants to complete two tasks related to researching musical instruments and positioned within an advertisement scenario of a hypothetical UK music shop (see Table 2 and Table 3). In both tasks, the participants were given an entry point to the browser and a template form to fill in their answers. The completion of Task 1 required mainly browsing through the musical instrument classification (in both DBTune and DBpedia) and reading descriptions provided from DBpedia. In contrast, Task 2 required browsing through content about music albums and artists, and reading through Amazon reviews.

### Table 2: Task 1 - Characteristics of a musical instrument

<table>
<thead>
<tr>
<th>The music shop is extending its collection of instruments with international musical instruments. You work in an advertising agency which has been asked to prepare an advertisement script for some of the new instruments that will appear in the shop. A key part of the preparation of the advertisement script is the research of the product. You have been asked to conduct a research of one of the new instruments, called bouzouki, using the information available in MusicPinta. You have to identify:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• the main characteristics of bouzouki;</td>
</tr>
<tr>
<td>• up to five similar instruments to bouzouki;</td>
</tr>
<tr>
<td>• features that make bouzouki distinctive from the similar ones you have chosen.</td>
</tr>
<tr>
<td>Go to ‘Semantic Search’ in MusicPinta and type bouzouki. Browse the content and follow links. Complete the provided form.</td>
</tr>
</tbody>
</table>

### Table 3: Task 2 - Usage and features of a musical instrument

<table>
<thead>
<tr>
<th>The music shop wants to increase the sales of its traditional musical instruments, such as electrical guitars. It intends to do this by adding links to creative commons album recordings with electric guitars, together with some interesting information about these albums to inspire customers to play/buy electric guitars or other musical instruments. Furthermore, when displaying its electric guitar items, the shop wants to highlight key features people look for when purchasing electric guitars. You are asked to conduct the research to address the above requirements by using information provided in MusicPinta. You have to review the information about electric guitar and identify:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• three interesting album recordings that include electric guitars and specify what is interesting;</td>
</tr>
<tr>
<td>• key features that people look for when purchasing an electric guitar.</td>
</tr>
<tr>
<td>Go to ‘Semantic Search’ in MusicPinta and type electric guitar. Browse the content and follow links. Complete the provided form.</td>
</tr>
</tbody>
</table>

After each task, the users were asked to fill-out a short questionnaire to rate their subjective level of cognitive load using a modified version of the NASA-TLX questionnaire [Hart & Staveland 1998]

**Data collected.** The data collected in the study includes: (i) the forms with the participants’ outputs for Tasks 1 and 2; (ii) the pre- and post-experiment questionnaires and word association tests; (iii) system log data; (iv) experimenter notes. The data was analysed using qualitative and quantitative methods (including non-parametric statistical test); the results are presented in the next section.
3.5. Findings

In order to address the research questions of the study, we will examine: task success, browsing behaviour, learning, and usability.

3.5.1 Task Success

Two musical instrument experts (one for Bouzouki, one for Electric guitar) have marked the outcome of participants for the two tasks. The marking refers to measuring how successful the participants have been in completing the tasks using MusicPinta.

Task 1: Bouzouki characteristics

For Task 1, participants produced a form with 3 sections: (i) characteristics of bouzouki; (ii) instruments similar to bouzouki; and (iii) summary of distinctive features of bouzouki (see Table 4 for a sample answer by a participant). The expert examined the information about bouzouki (using both the description and the semantics presented in MusicPinta) and identified main characteristics. The participants’ answers were compared with the experts’ answer and scored according to the overlap.

For similar instruments, the expert considered whether the instruments identified by the users were appropriate taking into account information shown in MusicPinta. For example, all similar instruments listed in Table 4, except ukulele, are marked as appropriate (ukulele is considered inappropriate as its resemblance to bouzouki is that they both are string instruments, regarded as a rather generic link for this task). The similarity and difference specified by the user were scored regarding appropriateness, as well as the summary of distinctive characteristics. The answer in Table 4 does not provide sufficient description about the bouzouki’s difference to mandolin and tambura. The average score of similarity-difference was 70% (st dev 14). The lowest score was 44% where the participant listed guitar, lap steel guitar, table steel guitar, electric guitar, which were seen as rather generic to be selected as similar to bouzouki for this task.
Table 4: Sample answer to Task 1 from a participant

Characteristics of bouzouki: Greek musical instrument, lute family, string instrument, nice looking, sharp sound, mandolin

Instruments similar to bouzouki

<table>
<thead>
<tr>
<th>Instrument</th>
<th>Similarity</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandolin</td>
<td>Similarly looking</td>
<td>Bouzouki</td>
</tr>
<tr>
<td>Tambura</td>
<td>Kind of bouzouki</td>
<td>No information about tambura</td>
</tr>
<tr>
<td>Banjo</td>
<td>Plucked string instrument, lute</td>
<td>Banjo has 4 pairs of strings, Bouzouki can be with 3 or 4 pairs of strings, shape is different – bouzouki looks more elegant</td>
</tr>
<tr>
<td>Ukulele</td>
<td>Plucked string instrument, 4 pairs of strings</td>
<td>Ukulele is more like a guitar (looks a small guitar)</td>
</tr>
<tr>
<td>Balalaika</td>
<td>Plucked string instrument, folk instrument</td>
<td>Balalaika - is in Russia, shape is different, 3 strings</td>
</tr>
<tr>
<td>Sitar</td>
<td>Plucked string instrument, folk instrument</td>
<td>Indian, Pakistan, has many strings</td>
</tr>
</tbody>
</table>

Summary of distinctive features of bouzouki: Greek, plucked string instrument, 3 or 4 pairs of strings, elegant looking.

The learning outcome of all participants on the different components in Task 1 is given in Table 5. All together, the participants identified 44 characteristics (70%, individual score median 4) from the description section of bouzouki (including the picture), and 19 descriptions (30%, individual score median 1.5) from the semantic tags. The difference between the two sets is significant (Wilcoxon test, W=60, p<0.01), i.e. participants tended to use the instrument links in answering the task. Also, 3 participants pointed out characteristics from the bouzouki picture (e.g. elegant looking or tear-drop shape).

Table 5: Average percentage scores for Task 1

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Similar instruments</th>
<th>Similarity-difference</th>
<th>Distinctive features</th>
<th>Overall score</th>
</tr>
</thead>
<tbody>
<tr>
<td>65%</td>
<td>80%</td>
<td>70%</td>
<td>68%</td>
<td>70%</td>
</tr>
</tbody>
</table>

Task 2: Electric guitar usage and features

For Task 2, the participants produced a form with 2 sections: (i) interesting albums, and (ii) key features of electric guitar people look for (see Table 6 for a sample answer by a participant).
Table 6: Sample answer to Task 2

<table>
<thead>
<tr>
<th>Album title</th>
<th>Interesting things about the album</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devon Graves – murder of crows</td>
<td>Has a lot of different instruments - having a combination of flute, electric guitar, and base, and acoustic guitar – will be interesting for people who can get inspired to listen to these instruments and perhaps be inspired to buy</td>
</tr>
<tr>
<td>Jennifer Turner, Tiger Lili</td>
<td>Has a combination of acoustic and electric guitars. Can be appealing for women</td>
</tr>
<tr>
<td>Bostian Andrejec, Casica</td>
<td>Classical album; Mixture of classical guitar, flute and electric guitar, may be more of a gentle sound.</td>
</tr>
</tbody>
</table>

Key features people look for when buying electric guitars

Sound, look, echo, price

The electric guitar expert examined the electric guitar content presented by MusicPinta - which presents a total of 31 unique albums - with information of artist, biographical detail, photo, other albums by the same artist, country and range of instruments. Against the knowledge presented by MusicPinta, each album described by a user was marked considering: no attempt, superficial fact (i.e. can be obtained by one look at the page), interesting insight on one aspect, and the maximum score was for indication of deeper thinking on more than one aspects. For Amazon reviews, the expert conducted a content analysis on the Amazon reviews related to electronic guitar available in MusicPinta to extract features that people referred to. The extracted list was used to mark the list produced by each user. To compare task success, the scores for both tasks were computed as percentages.

As the task was timed, no deep insight was collected (if time were not an issue, one could found out more about individual artists, may notice two albums which had electric guitar & bouzouki/mandolin, more than one female artist; the type of music – rock, jazz, folk). This is also problematic as the participants were unfamiliar with the interface/terminology and took a while for them to contextualise what information could they expect while pursuing certain links (2 out of the 6 participants who did Task 2 first had poor replies for this task). Most participants identified that cost, quality and material might be something to consider when buying an electric guitar.

The average performance on Task 2 was 48%, which was significantly lower than Task 1 (Wilcoxon test, W=74, p<0.005), yet there was no significant difference in the participants’ confidence scores (the subjective perceptions of both tasks are given in Figure ). Task 2 was more frustrating with border line significance (Wilcoxon test, W=.34, p<0.05). There was no correlation between scores and confidence – the participants seemed confident that they did as much as they could in the given time.
3.5.2 Browsing Behaviour

The interaction log files, which recorded the user clicks when browsing data sets in MusicPinta, provided an insight into the browsing behaviour. The log data was preprocessed and each link was assigned an abstraction level based on the ontology (Table 7).

The distribution of user clicks on both tasks according to the abstraction levels is presented in Figure 10.

**Abstraction level.** In both tasks, the participants were asked to start browsing from an entity classified at a classification low level, i.e. a concrete musical instrument (bouzouki and electric guitar, respectively). Participants clicked much more on entities from instrument classifications in Task 1 than in Task 2 (high significance, Wilcoxon test, W=74, p<0.005). Similarly, participants clicked much more on content entities in Task 2 than in Task 1 (high significance, Wilcoxon test, W=-66, p<0.005), the average click distribution is given in Table 8. The result is not surprising; tasks were set expecting such user browsing behaviour. Further analysis of the log files showed some unexpected cases outlined below.

### Table 7: Abstraction level assigned to the user clicks

<table>
<thead>
<tr>
<th>Abstraction level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification - upper level</td>
<td>Clicks on abstract concepts, such as instrument, performance, artist, from the Music Ontology</td>
</tr>
<tr>
<td>Classification - middle level</td>
<td>Clicks on categories for musical instruments, e.g. string instruments, plucked string instruments, steel guitars, Greek musical instruments</td>
</tr>
<tr>
<td>Classification - low level</td>
<td>Clicks on entities representing musical instruments, e.g. bouzouki, mandolin, lute, electric guitar.</td>
</tr>
<tr>
<td>Content - albums</td>
<td>Clicks on descriptions of music albums</td>
</tr>
<tr>
<td>Content - artists</td>
<td>Clicks on descriptions of music artists</td>
</tr>
<tr>
<td>Content - reviews</td>
<td>Clicks on Amazon reviews of musical instruments</td>
</tr>
</tbody>
</table>
Confusion of abstract concepts. Two participants clicked on abstract concepts, such as instrument, performance and performer, from the Music Ontology. In both cases, the participants were looking for concrete information (e.g. participant-12 clicked on instrument in Task 1 when seeking for more detail about a musical instrument, while participant-05 clicked on performer and performance in Task 2 when seeking more detail about an album). Clicking on abstract concepts led to confusion, and the participants quickly pressed the back button on their browsers. This implies that it would be beneficial to avoid showing links at high abstract level.

Existence of ‘empty clicks’. Another interesting case is the high number of ‘empty clicks’ - the user clicks on a link and is taken to a page with no information, sees that this link is not helpful and quickly returns to the previous page. In Task 1, such clicks concerned similar instruments, e.g. there was no information about bajitar, xalam, rebab. In Task 2 such clicks concerned performances (music albums) and happened quite often. ‘Empty clicks’ leading to pages with no information was seen as one of the key usability problems. Some of the links, however, were perceived as empty and the users missed to click on important information (e.g. pages about musical instruments were abandoned, although there was useful information about relevant instruments; or interesting facts about an album artist were overlooked as the users did not click on the corresponding link). A possible way to prevent such problems is to reduce the number of links shown to the user (which will avoid clutter and confusion) and to signpost links with a certain browsing value.

Table 8: Distribution of user clicks in both tasks (average per individual participants’ clicks)

<table>
<thead>
<tr>
<th></th>
<th>Classification-upper level</th>
<th>Classification-middle level</th>
<th>Classification-lower level</th>
<th>Content-albums</th>
<th>Content-artists</th>
<th>Content-reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1-AVG</td>
<td>0.08</td>
<td>2.58</td>
<td>11.08</td>
<td>0.25</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Task 2-AVG</td>
<td>0.38</td>
<td>2.92</td>
<td>6.42</td>
<td>5.71</td>
<td>3.25</td>
<td>2.50</td>
</tr>
</tbody>
</table>

Selection strategies. Both tasks (deliberately) put the users in situations where they had too many choices. The bouzouki page included 12 different links in the facts facet (11 links to concepts in the middle classification level and 1 link to the abstract concept instrument) and 51 links in the terms facet (43 links to musical instruments and 8 links to performances).
There were four main strategies observed when selecting a link from the bouzouki page: (i) click on the nearest classification link (e.g. plucked string instruments or string instruments) to see general characteristics; (ii) click on instruments mentioned in the description – lute and mandolin; (iii) click on an instrument that ‘sounds familiar’ (e.g. sitar, banjo, pipa); (iv) click on something that sounds interesting or unusual (e.g. oud, xalam).

The electric guitar page included 18 links in the facts facet (to concepts in the middle and upper classification levels), 78 links to albums in the terms facet, and 8 links to Amazon reviews in the content facet. The users rarely clicked on links from the facts facet (as the task did not require this), and adopted some strategies for selecting from the list of albums: (i) clicking on something that looks important (e.g. an artist has several albums); (ii) clicking on something that is interesting or unusual (e.g. noticing a women artist or something interesting in the album name); (iii) clicking at random (which happened after exhausting the first two strategies). The observations about participants’ behaviour with MusicPinta provide useful information about possible improvement of the browser by adding signposting techniques.

3.5.3 Usability Feedback

As MusicPinta is still at a prototype level, its usability hinders performance at times. The answers to the usability questionnaire indicated key usability issues that have to be addressed. The main criticism stems from lack of content after clicking on a link (e.g. for electric guitar, 12 out of 25 artists linked to electric guitar had no extra information; only 3 artists had extra links to other albums; 6 artists had biographical information and photos; there was no extra information on content of albums e.g. tracks, and there was no way to listen what the album included). Participants were overwhelmed by the long list of links and baffled by link duplications (when showing the same concept from DBTune and DBpedia). There is evidence that small font text used for description was easily missed by participants despite usefulness of content and position of text. It took a while to get familiar with the terminology and navigate to useful information (as seen from performance in Task 2 when it was done before Task 1). Slow system performance at times was an issue (only one participant did not have issues with performance, as he regularly opened new windows when clicking on browser links). These usability issues will be addressed in the next MusicPinta version.

3.6 Discussion

The findings from the study provide insight into exploratory search behaviour when browsing through linked semantic data. We will summarise by revisiting the main questions of the study.

**Use of semantic data browsers for exploratory search** [Question 1]. The study provides evidence that *semantic data browsers are suitable for exploratory search* (which is in line with current trends in knowledge-enriched browsing environments – see Section 2). Users without domain knowledge could complete investigation tasks with MusicPinta fairly quickly and create meaningful answers (provided that there was suitable content). None of the users indicated that they were lost in the information space, and they used the browser facets to *easily navigate through linked semantic data*. Users performed better and felt less frustrated on the *analytical Task 1* that required browsing at the classification level (which was more about exploration on structure of concepts/knowledge). MusicPinta *did not appear very beneficial for tasks requiring creativity and exploration at content level, such as Task 2* – see improvement discussed below.
Showing the three facets together (facts, terms and content) provided a flexible way to explore the richness of data types and embedded information in linked data sets. Facts and terms gave starting points for classification/content exploration, while descriptions from DBpedia or from Amazon reviews offered additional details. There were different ways people consume knowledge enriched information. When browsing at classification level, some people preferred to look at the upper classification categories (like plucked string instruments, Greek musical instruments), while other people used the suggested list of related terms. When browsing at content level, people looked for ‘important’ links or anything that was ‘interesting’.

Future improvement to realise the potential of semantic data browsers for exploratory search [Question 2]. Apart from a list of usability issues, the study indicated the need for further algorithmic support to realise the exploratory search potential of semantic data browsers. We have recently experimented with an approach to use semantics to generate ‘semantic nudges’ (signposts and prompts) to facilitate user browsing [Thakker et al 2012]. The signposts include a list of (up to) seven links to related entities which are seen as valuable for the user browsing experience. The prompts include remarks (pop-ups) from the system to show any interesting facts (e.g. to broaden the user’s exploratory space).

The MusicPinta study provided empirical evidence for deriving heuristics for what and when semantic nudges [Thakker et al., 2012] may be needed:

- For more analytical tasks which require browsing through the classification levels (knowledge-rich tasks), entity signposts from the semantic graph can be beneficial: (i) entities in the signposting list can be extracted from the related terms and should refer to ‘important’ links; (ii) ‘importance’ can be judged on the basis of amount of possible navigation paths or steps from the current entity or availability of rich content that can be seen if the link is followed; (iii) one size would not fit all users, some adaptation to the users’ previous knowledge would be beneficial (e.g. some of the signposts may relate to familiar concepts, while others may introduce new knowledge – putting familiar and new items together can deepen the learning by association); (iv) generally, detecting prior knowledge is challenging, especially when the freedom of exploration has to be preserved; a way to ‘sense’ previous knowledge is to analyse the user clicks on the low classification level links – clicking on an instrument can indicate some familiarity with its most specific classification category (e.g. in the study, users familiar with Russian musical instruments clicked on balalaika and users familiar with Chinese musical instruments clicked on pipa).

- For more creative tasks requiring browsing through a large amount of content, the study appeared to provide indication that it will not be very beneficial to limit the user entity choices, as this can affect the free content exploration. Instead, signposting can include some indicators about the ‘importance’ or ‘value’ of a content item, e.g. if there is any description (or any multimedia content), its source of the content (e.g. DBpedia, MusicBrainz), if further semantic links are available in the content (e.g. albums that have several musical instruments) to facilitate user choices. There can be some ordering based on the value. Furthermore, content level browsing experience can be enriched with prompts to point at diversity of paths or unexplored connections to stimulate curiosity and to facilitate serendipitous discoveries, a dimension often linked to exploratory search [Makri et al 2011].

The above suggestions for signposts can be implemented using density metrics for the semantic graphs, while the prompts can exploit similarity and diversity metrics. Further experimental studies are needed to examine the validity and benefits of the above semantic nudging heuristics for improving exploratory search experience with semantic data browsers.
Study results are inevitably influenced by experimental design choices. This study did not use a base line system to compare interaction behaviour. We deliberately selected this design, as most of the data sets were already semantically enriched and because MusicPinta provides standard exploration facets (there is no base line to compare with, as the base line is indeed a semantic data browser). The current version of MusicPinta will be used as a base line for comparing user interaction behaviour with and without semantic nudges and will enable us to validate the above heuristics. The number of participants was relatively small, which is appropriate for qualitative studies where multi-faceted analysis is done. In future work, we intend to look at larger scale studies which will exploit appropriate quantitative analysis techniques (e.g. data mining of the log data could reveal further insights about the user browsing behaviour). Similar to [Kammerer et al 2009], we gave the participants entry points for their exploration, which ensures that the key features of the system were visited for more systematic analysis. Future experiments would be needed to test the ecological validity of the findings and allow participants to decide their entry points.

4 Summary and Conclusions

This deliverable presented the extended suite of semantic services for ontology authoring and semantic data browsing.

Firstly, we presented the Perico framework for ontology authoring dialogues which extends the ROO tool presented in Deliverable 5.3.1. Perico’s functional components follow the main basic dialogue system architecture but are influenced by the choice for representing domain, discourse and task. In particular, Perico reuses and extends Entendre to describe the standard dialogue system components. Perico defines key parts of the information-state-update view:

- basic dialogue moves from ISO-DIS-24617-2 are defined as well as some specific dialogue moves made possible by the domain knowledge;
- dialogue grounding is defined in terms of changes to the domain ontologies.

Secondly, we presented Pinta – a semantic data browser which exploits data sets from Linked Data. This work starts from the position that linked semantic data can provide a rich source of knowledge that can be exposed to end users via semantic data browsers. In a study based on a Music domain within advertisement context, we created representative exploratory search conditions, including a more knowledge-rich analytical task (compare, find similarities and differences) and a more content-based creativity task (find something interesting). The study confirmed that the overview of the knowledge structure presented with the classification level tags is beneficial for the success of the analytical tasks and can facilitate serendipitous learning in the creativity tasks. Based on the findings, we drew heuristics for extending semantic data browsers with nudging. The findings can be beneficial for researchers and developers who wish to exploit linked data for browsing, as well as users who can be the brave early adopters of such technologies. Our immediate future work is to extend the social content available in MusicPinta in order to examine the benefit of knowledge-enriched exploration of social content. We are also examining sensemaking domains which require exploratory search through linked semantic data, such as advertising (relevant to Use Case 3) and clinical data browsing (relevant to Use Case 2).
References


