Senaid Salkić, BSc

Linked Data Driven Visual Analytics for Tracking Learners in a PLE

Master's Thesis

Graz University of Technology

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Supervisor: Priv.-Doz. Dipl-Ing. Dr.techn. Martin Ebner
Co-Supervisor: Dipl-Ing. Selver Softić, BSc

Graz, February 2017

Abstract

In the last decades the data modeling, analysis and visualization have become popular due to large amounts of produced data. Various research areas emerged in order to provide adequate solutions. One of such research areas is visual analytics which tries to combine automatic data analysis techniques and visualization techniques and as such help the analyst to draw conclusions and gain knowledge from the analyzed datasets.

In this thesis, an approach for modeling, analyzing and visualizing the data gathered by tracking user behavior and actions in an educational system called Personal Learning Environment at Graz University of Technology is presented. For modeling the gathered data a novel Semantic Web driven approach is used, which enables modeling of learning and activity-based context using eligible domain specific ontologies. Furthermore, visual analytics methods for analyzing and visualizing the modeled data are applied. The main goal of this work is getting insight in gathered data in order to understand the students response to the educational system and draw necessary conclusions for its improvement.

The result of this work is an application called "Analytical Dashboard" which is used to get deeper insight into the modeled datasets. First results indicate that the students interest in the educational system is dropping rapidly over the last few years.

Kurzfassung

In den letzten Jahrzehnten sind die Datenmodellierung, Datenanalyse und Datenvisualisierung aufgrund der steigenden Menge an erzeugten Daten sehr beliebt geworden. Verschiedene Forschungsgebiete beschäftigen sich damit adäquate Lösungen zu liefern. Eines dieser ist Visual Analytics. Dieses Gebiet versucht die automatische Datenanalyse- und Visualisierungstechniken zu kombinieren und dadurch dem/der Analytiker/in helfen bessere Entscheidungen zu treffen und relevante Erkenntnisse aus den analysierten Daten zu gewinnen.

In dieser Forschungsarbeit wird ein Ansatz für die Modellierung, Analyse und Visualisierung der gesammelten Daten präsentiert. Diese Daten wurden durch die Verfolgung des Benutzerverhaltens und Benutzeraktivitäten in einem Bildungssystem namens Personal Learning Environment an der Technischen Universität Graz gesammelt. Für die Datenmodellierung wurde ein Semantic-Web-basierter Ansatz verwendet, der die Modellierung von lern- und aktivitätsbezogenen Kontexten durch domänen-spezifische Ontologien ermöglicht. Zusätzlich wurden die Visual-Analytics-Methoden für die Analyse und Visualisierung der modellierten Daten angewendet. Das Hauptziel dieser Arbeit ist es einen Einblick in der gesammelten Daten zu erhalten, um die Reaktionen der Studierenden auf das Bildungssystem zu verstehen und die erforderlichen Schlussfolgerungen für die Verbesserung des Bildungssystems zu ziehen.

Das Ergebnis dieser Arbeit ist eine Applikation namens "Analytical Dashboard", die ein tieferen Einblink in der modellierten Daten ermöglicht. Erste Resultate zeigten, dass das Interesse der Studierenden auf das Bildungssystem in den letzten Jahren ständing sinken.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, ______________________________ ____________________________________

Date Signature

Eidesstattliche Erklärung[1]

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ______________________________ ____________________________________

Datum Unterschrift

[1] Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008; Genehmigung des Senates am 1.12.2008

Acknowledgment

I would first like to thank my thesis supervisor Priv.-Doz. Dipl-Ing. Dr.techn. Martin Ebner and co-supervisor Dipl-Ing. Selver Softić, BSc at Graz University of Technology. They were always ready to help whenever I ran into a problem or had a question about my research or writing. They allowed this work to be my own and steered me in the right direction.

I would like to thank my entire family for supporting me over the entire study. In particular I would like to thank my mother Šefika Salkić and my father Sejad Salkić for supporting me over the past years and giving me the chance to study in the first place. Big thanks also goes to my sister Zinaida and my brother Emsad.

Also I would like to thank my friend, Admir, who always helped me with whichever problems I ran into and was there the entire study period.

Finally, I have to express my gratitude to my wife Zehira Dedić-Salkić for her encouragement and support during my years of study and especially throughout the period of researching and writing this thesis. This accomplishment would not have been possible without you. Thank you.

I dedicate this work to my son who gave me the strength and motivation to successfully complete it. Tarik, happy first birthday.

Senaid Salkić, BSc
Graz, Austria, February 2017

Contents

List of Figures

List of Tables

Listings

1 Introduction

Understanding users in general is a challenge for each user-centric system. The purpose of such systems is to provide services to users, and they may not always accept it as expected. And in many cases assumptions how a system is used by users are hard to make and challenging. Other important aspects are fast changing trends because users behavior and needs change over time.

In 2010 Graz University of Technology launched an educational system called Personal Learning Environment[1] (PLE). The main goal was to provide a personalized version of university services and resources integrated with the World Wide Web. The first step towards understanding the users and their behavior while using a PLE was already done back in 2011 when the tracking module was introduced (Taraghi, Stickel, and Ebner, 2011). The tracking module captures data about the widget usage over time. It was the first and very important step towards analyzing a PLE usage and understanding the student behavior and his/her response. The first reports on students' responses to the educational system were very positive and promising (Softic et al., 2013; Taraghi, Softic, et al., 2013).

On the one side, this thesis focuses on finding a solution for structuring and retrieving the captured data and on the other side, it provides an analytical tool which needs to give clear insight into system usage and student's response over the entire period.

The main research questions this thesis is dealing with are listed bellow:

- Can captured log data be used to measure the overall PLE usage and how?

[1] http://ple.tugraz.at (last visited December 2016)

- How does the PLE compete against fast changing trends among students?
- When did the PLE reached the highest utilization degree?

In general, there are two main goals. The first goal is to semantically model the data captured from the tracking module in order to give it structure and context and to provide the possibility of sophisticated information retrieval from the entire dataset. One of the main challenges will be finding the "right" ontology to represent the learning situation. The second goal is to use the knowledge from the visual analytics research field to design and develop an analytical dashboard application which shall provide an interactive visualization of the modeled dataset.

This thesis consists of several chapters. Chapter 2 is an overview of the related work where the semantic web technologies, automatic data analysis and visual analytics are introduced. At the end of the chapter, a short overview of the Personal Learning Environment is provided. In chapter 3 the concept is presented. The following chapter 4 goes into the technical details of the practical part of this work. The results of the analysis done by the developed analytical application are presented in chapter 5. And chapter 6 concludes the thesis and presents ideas for future work.

2 Related Work

This chapter presents some important topics related to this thesis. They are introduced in order to get basic information about existing research fields and technologies which are related to this work. This serves as a good background knowledge which is very important for understanding the chosen approach and practical implementation of this thesis. Four fields related to this work are presented. These are:

- Semantic Web
- Knowledge Discovery and Data Mining
- Visual Analytics and
- Personal Learning Environment

Each of these fields is discussed in detail in the following sections.

2.1 Semantic Web

The World Wide Web together with all its standards is designed to target humans as key users. The idea behind it is to write computer applications to present information to humans which are supposed to interpret and understand it finally. And the computers did exactly that, they presented information without understanding the semantics or the meaning of it. They are capable of parsing the information on the web but have no possibility to process its semantics.

Over time the amount of information in the web grew at a tremendous rate. Due to the limited time and capability of humans, their requirements also evolved. Instead of only presenting the information, they needed computers to present only the "relevant" information, and to do that instantly.

Because of the amount of information it is hard for computers to fulfill all requirements without "understanding" the information itself. New demands required them to get "intelligent" and do more sophisticated tasks in information processing in order to answer requests and to find relevant information. Since the information itself is not self-explanatory to them, a need for new technology emerged.

The solution for the emerging problem was presented by the inventor of the World Wide Web, Tim Berners-Lee. He introduced the idea of the Semantic Web in the Scientific American magazine in an article called "The Semantic Web" where he defined it as follows:

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. (Tim Berners-Lee, Hendler, and Lassila, 2001)*

The idea of Semantic Web is that the information itself becomes self-explanatory. It gets a structure and well defined semantics. Such information is easily interpretable and understandable by computers. That is the main difference compared to the current web. With this important feature the focus is mainly given to computers and not to humans as it used to be (Tim Berners-Lee, Hendler, and Lassila, 2001). And since computers do all the heavy work it will enhance the computer-human communication because computers can process and answer requests much faster.

Also compared to the traditional Web where web-pages containing the information were linked between each other, the semantic web goes into detail by linking each meaningful piece of information to another. By doing that it creates a web of semantically interlinked information or also known as a "web of data".

The World Wide Web Consortium refers to the Semantic Web as a Web of linked data and defines it as follows:

> *The Semantic Web is a Web of Data — of dates and titles and part numbers and chemical properties and any other data one might conceive of. The collection of Semantic Web technologies (RDF, OWL, SKOS,*

> *SPARQL, etc.) provides an environment where application can query that data, draw inferences using vocabularies, etc.*[1]

The ultimate goal of the Semantic Web is to connect the data in the web which shall result in creating a huge "knowledge graph" of linked data.

2.1.1 Semantic Web Architecture

At the XML 2000 conference Tim Berners-Lee presented the Semantic Web Architecture in a so-called Semantic Web Layer Cake which is illustrated in the figure 2.1. It illustrates the hierarchy of technologies and their structure in order to make the Semantic Web possible. As already mentioned, the Semantic Web is an extension of the traditional Web and some technologies are still in place. Additionally some novel technologies have been introduced and standardized.

Based on the technologies the architecture can be separated in following layers:

- Traditional Web Technologies: Uniform Resource Identifier (URI), Unicode and eXtensible Markup Language (XML)
- Standardized Semantic Web Technologies: Resource Description Framework (RDF), Resource Description Framework Schema (RDFS), Web Ontology Language (OWL), SPARQL Protocol and RDF Query Language (SPARQL), Rule Interchange Format (RIF)
- Not-Standardized Semantic Web technologies

In the next sections the following technologies and concepts important for this thesis will be discussed in detail:

- Linked Data
- Uniform Resource Identifier (URIs)
- Internationalized Resource Identifier (IRIs)
- Resource Description Framework (RDF)
- RDF Extensions: RDFS, OWL and
- Query Language for Semantic Web (SPARQL)

[1] https://www.w3.org/standards/semanticweb/data (last visited December 2016)

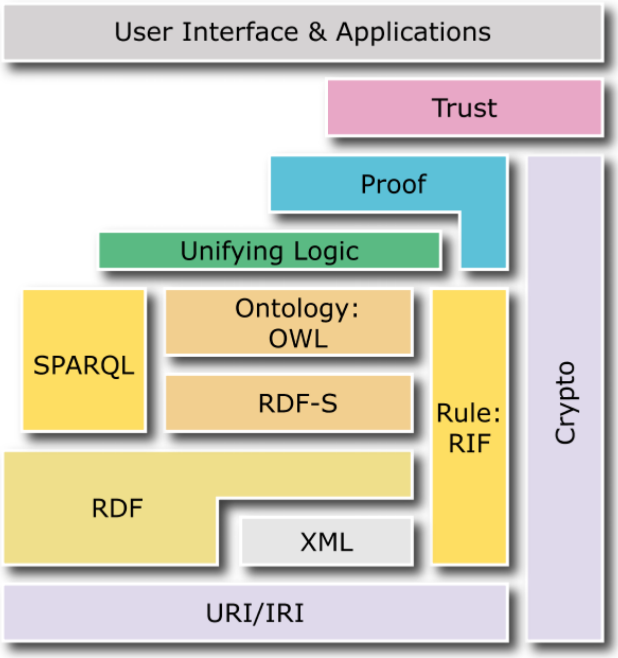


Figure 2.1: The Semantic Web "layer cake" presented by Tim Berners-Lee at the XML 2000 conference (https://www.w3.org/2007/03/layerCake.png (last visited January 2017))

2.1.2 Linked Data

The term Linked Data refers to the collection of data which is not only available in a standard format for accessing and processing it by the Semantic Web tools but also the relations among that data are known [2]. It is basically a set of rules or recommendations for publishing and structuring the data on the web.

Tim Berners-Lee coined the basic rules for the publishing of the data as Linked data on the Web in the document titled "Linked Data"[3] in 2006 (last edited in 2009), where he defined the following four rules:

- "Use URI as names for things" [3]

[2] https://www.w3.org/standards/semanticweb/data (last visited December 2016)
[3] https://www.w3.org/DesignIssues/LinkedData (last visited December 2016)

- “Use HTTP[4] URIs so that people can look up those names” [3]
- “When someone looks up a URI, provide useful information, using the standards, such as RDF, SPARQL” [3]
- “Include links to other URIs so that they can discover more things” [3]

The idea behind these rules or principles is having clearly defined standards for representing and accessing the data on the Web. For data from different sources, hyperlinks are set in order to connect all Linked Data into a single global data graph. (Bizer, Heath, and Tim Berners-Lee, 2009)

An overview of the linked datasets published in Linked Data format and interlinked with other datasets in the cloud from 2014 is depicted in the figure 2.2.

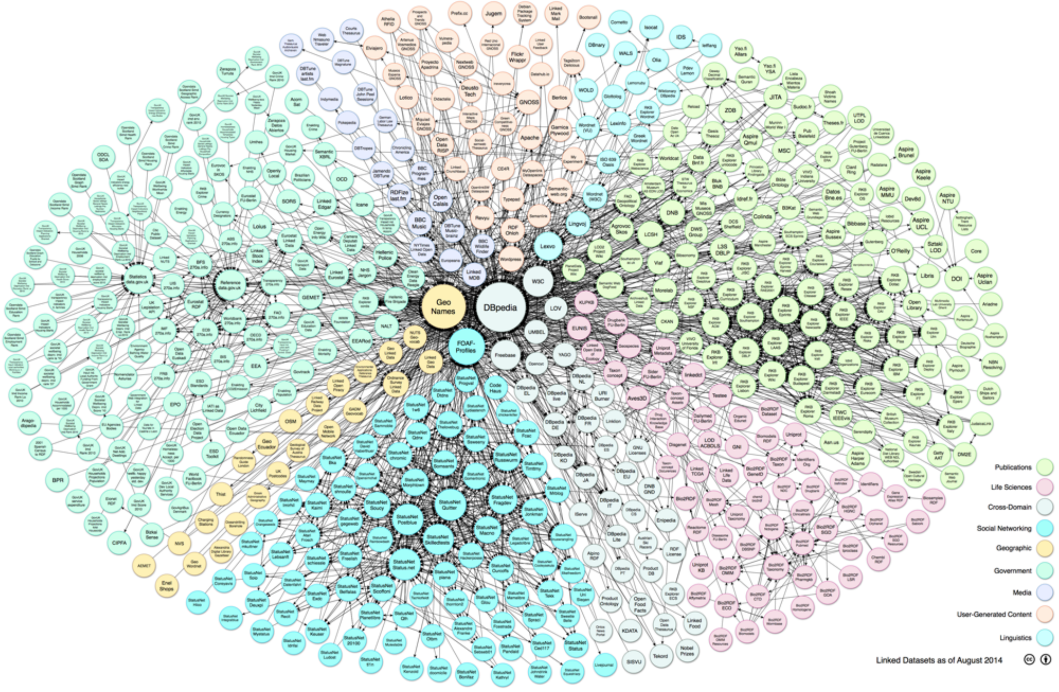


Figure 2.2: Linking Open Data cloud diagram 2014, by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak (`http://lod-cloud.net/` (last visited January 2017))

[4] `https://www.w3.org/Protocols/` (last visited December 2016)

For Linked Data to work, the technologies for accessing and querying the data are crucial. Such technologies are provided by the World Wide Web Consortium[5] (W3C) and are described in the following sections.

2.1.3 Uniform Resource Identifier (URI)

An URI is located at the bottom of the *layer cake*, see figure 2.1, and is the foundation of the Web. It is defined as "a compact sequence of characters that identifies an abstract or physical resource (T. Berners-Lee, Fielding, and Masinter, 2005)". URI is simply a unique global identifier which is used to identify the data resource which can be any piece of information.

An URI is not a new concept introduced with the semantic web, it's just being extended and used on another level additionally. Two types of URIs are popular in the traditional Web, the Uniform Resource Locator (URLs) and the Uniform Resource Names (URNs). They are used at a document level as unique identifiers for pages, documents or books. For example, the URL http://www.w3.org uniquely referrers to the World Wide Web Consortium (W3C) website and the URN urn:isbn:0596153813 uniquely identifies the Book *Programming the Semantic Web* by referencing its International Standard Book Number (ISBN). (T. Berners-Lee, Fielding, and Masinter, 2005)

In the semantic web context they are not only used for whole documents or pages but also to uniquely identify the data entities within that document or page like places, people, cars, or even abstract ideas. These are also called real-world objects or *things*. (`https://www.w3.org/TR/cooluris/#oldweb`, (last visited December 2016))

An object or a *thing* can link to another object or to the external source by referencing its URI. Also, the relationship of the link can be described. The model which enables it is called the *Resource Description Framework*.

[5] `https://www.w3.org/Consortium/` (December 2016)

2.1.4 Internationalized Resource Identifier (IRI)

Internationalized Resource Identifier (IRI) is a new standard which is basically a complement of the URI. It is defined as "a sequence of characters from the Universal Character Set (Unicode/ISO 10646) (Duerst and Suignard, 2005)". An IRI may contain characters from the Universal Character Set (Unicode/ISO 10646) (Duerst and Suignard, 2005), whereas an URI is limited to far fewer characters from the ASCII character set (T. Berners-Lee, Fielding, and Masinter, 2005).

It is featured in the recent semantic web standards, therefore, the notion of IRIs will be used in further discussions and definitions of this thesis.

2.1.5 Resource Description Framework (RDF)

A very important milestone towards making the semantic web a reality is having a standard data model for describing the information in the Web. It is crucial for the data interoperability. The information is usually being produced and published by various resources. Having such a standard is very important to be able to interchange data on the Web. For that purpose the Resource Description Framework is introduced.

Back in 1998 Eric Miller defined the RDF as follows:

> *The Resource Description Framework (RDF) is an infrastructure that enables the encoding, exchange and reuse of structured metadata (Miller, 1998).*

The RDF has been adapted as a W3C recommendation in 1999. Since then W3C published two RDF specifications, first in 2004, RDF 1.0, and second in 2014, RDF 1.1.

The W3C defines the RDF as "a framework for expressing information about resources (Schreiber and Raimond, 2014)". It is a standardized model for describing the relations between the Web resources.

RDF Data Model

The resources in the RDF terms are objects that are uniquely identified by a IRI (T. Berners-Lee, Fielding, and Masinter, 2005; Miller, 1998). They have certain properties or characteristics which are expressed by RDF. The RDF describes not only the resource properties but also the relations among the resources. It makes statements in the form of the RDF *triple*. That is 3-tuple of the RDF terms. The RDF term can be:

- IRI
- literal
- blank node

As already mentioned in the section 2.1.4, an IRI is used as a global identifier to identify resources such as people, documents, physical objects, etc. For example an IRI for a person, lets say, Leonardo da Vinci, in DBpedia[6] is `http://dbpedia.org/resource/Leonardo_da_Vinci`. This is a global identifier for this resource and can be reused to identify it, in this case for a person. An IRI may occur in all three positions of a triple. (Schreiber and Raimond, 2014)

Literals are used for values which are not IRIs, e.g. strings, dates or numbers, and may appear only in the third, object, position of a triple (Schreiber and Raimond, 2014).

Blank nodes indicate the existence of a resource without identifying it. They represent a resource for which an IRI or a literal is not given and may be used in the first, subject, and the third, object, position of a triple. (Schreiber and Raimond, 2014)

The RDF Triple is comprised of:

- A Subject
- A Predicate
- An Object

[6] DBpedia is a crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web.
http://wiki.dbpedia.org/ (last visited December 2016)

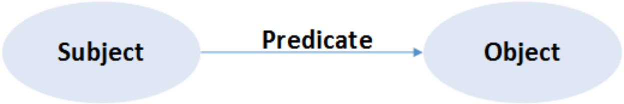


Figure 2.3: RDF Triple (based on https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/Graph-ex.gif)

A Subject is an information resource which is being described by a triple. It can be either an IRI or a blank node. A Predicate must be an IRI and denotes the nature of the relationship between the object and the subject. An object fills the value of the relation. Basically it can be any RDF term, IRI, blank node or literal. The triple must be written in the following order: subject, predicate, object. (Schreiber and Raimond, 2014)
The figure 2.3 illustrates the RDF triple and the relationship from a subject to an object.

For better understanding take a look at a statement: `<Bob> <is a> <person>`. In this simple statement `<Bob>` is a subject, `<is a>` is a predicate and `<person>` is an object. Basically more information about the subject could be added, like his birthday date, interests, age, friends, etc. All this information could be expressed with multiple RDF statements. The collection of such RDF triples represents a labeled and directed multi-graph or so-called the RDF graph (Miller, 1998).

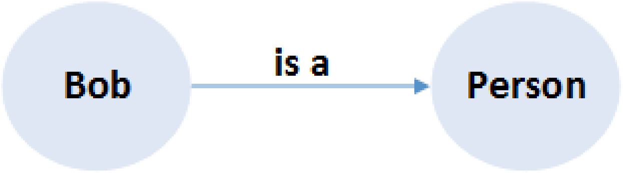


Figure 2.4: Informal graph of simple statement

In the figure 2.4 a simple statement is visualized as a connected graph. The RDF graph consists of nodes and arcs (Schreiber and Raimond, 2014). Nodes are subjects and objects and the arcs are formed by predicates.

RDF syntax

There are different formats for writing the RDF Graphs. Even though they are written in different ways, they still lead to exactly the same triples (Schreiber and Raimond, 2014). These formats are:

- Turtle family of RDF languages: N-Triples, Turtle, TriG and N-Quads
- JSON-LD: JSON based RDF syntax
- RDFa: HTML and XML embedding
- RDF/XML: XML syntax for RDF

In this thesis only the N-Triple and the Turtle syntax are used and explained later on. Detailed information regarding this and other formats are available at *https://www.w3.org/TR/rdf11-primer/ (last visited December 2016).*

N-Triples are a line-based way of serialization where each line represents a triple. Each RDF term is enclosed in angle brackets(<>). To signal the end of the line a period is added. The data-type can be appended to the literal through a ^^ delimiter. An example of the n-triple syntax is illustrated in the listing 2.1. (Schreiber and Raimond, 2014)

The Turtle syntax is an extension of N-Triples. It introduces syntactic shortcuts to increase readability. E.g. an IRI for a birth date is <*http://schema.org/birthDate*>. Instead of repeatedly writing it, a PREFIX for schema can be defined, like this: *PREFIX schema:* <*http://schema.org/*>. Then it can be used to express any schema property, in this case *schema:birthDate*. (Schreiber and Raimond, 2014)
An example of the Turtle syntax is illustrated in the listing 2.2.

RDF Example

This RDF example is given by W3C[7]. Sample statements are contained in the table 2.1.

[7] https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/#example1

Subject	Predicate	Object
Bob	is a	person.
Bob	is a friend of	Alice.
Bob	is born on	the 4th of July 1990.
Bob	is interested in	the Mona Lisa.
the Mona Lisa	was created by	Leonardo da Vinci.
the video 'La Joconde a Washington'	is about	the Mona Lisa

Table 2.1: RDF sample triples (informal) (https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/#example1 (last visited December 2016))

Now these statements can be written in a machine-readable way. An IRI for each term will be added. To express the subject in the first row the following example IRI is defined: `<http://example.org/bob#me>`.

```
<http://example.org/bob#me> <http://www.w3.org/1999/02/22-rdf-
    syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/knows>
    <http://example.org/alice#me> .
<http://example.org/bob#me> <http://schema.org/birthDate> "
    1990-07-04"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/
    topic_interest> <http://www.wikidata.org/entity/Q12418> .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/
    terms/title> "Mona Lisa" .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/
    terms/creator> <http://dbpedia.org/resource/
    Leonardo_da_Vinci> .
<http://data.europeana.eu/item/04802/243
    FA8618938F4117025F17A8B813C5F9AA4D619> <http://purl.org/dc/
    terms/subject> <http://www.wikidata.org/entity/Q12418> .
```

Listing 2.1: N-Triple example (https://www.w3.org/TR/rdf11-primer/#section-n-triples (last visited December 2016))

For expressing the relation between the subject and the object the RDF *type*[8] can be used. Using RDF *type* gives the following IRI: `<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>`. And with the help

[8] https://www.w3.org/TR/rdf-schema/#ch_type (last visited December 2016)

of the FOAF[9][10] vocabulary the object can be expressed as follows `<http://xmlns.com/foaf/0.1/Person>`. Usage of these IRIs is illustrated in the first line of the listing 2.1. This illustration also presents the N-Triple syntax, which has been introduced in the section 2.1.5.

The listing 2.2 illustrates the Turtle syntax, introduced in the section 2.1.5. Looking at these two listings, 2.1 and 2.2, it is clear that the syntax is different, although the Turtle syntax is based on the N-triple syntax. But for the machine they are completely the same. Both listings have the same meaning and will be interpreted exactly the same. Various syntaxes exist only to facilitate writing and understanding for humans, see the section 2.1.5.

```
BASE   <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX schema: <http://schema.org/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX wd: <http://www.wikidata.org/entity/>

<bob#me>
    a foaf:Person ;
    foaf:knows <alice#me> ;
    schema:birthDate "1990-07-04"^^xsd:date ;
    foaf:topic_interest wd:Q12418 .

wd:Q12418
    dcterms:title "Mona Lisa" ;
    dcterms:creator <http://dbpedia.org/resource/Leonardo_da_Vinci> .

<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
    dcterms:subject wd:Q12418 .
```

Listing 2.2: Turtle example (https://www.w3.org/TR/rdf11-primer/#section-turtle (last visited December 2016))

These triples as well as relations among them are graphically illustrated in the figure 2.5. On the RDF graph seven arcs are visible, which correspond to a number of the predicates in the example.

[9] An acronym for 'Friend of a Friend'

[10] http://xmlns.com/foaf/spec/ (last visited December 2016)

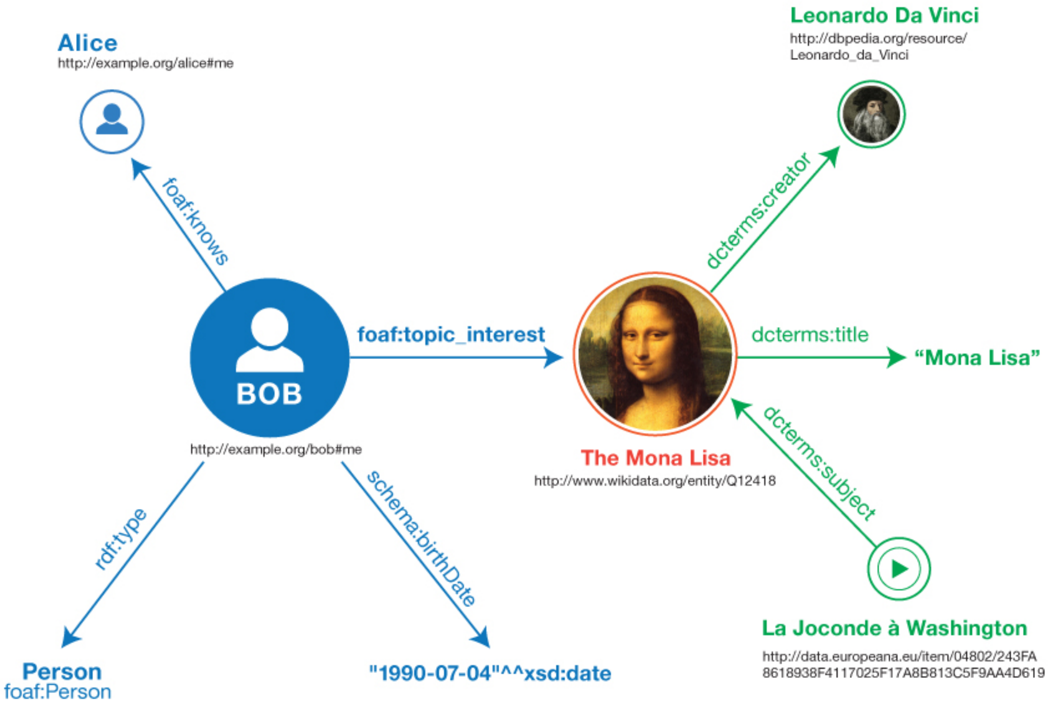

Alice
http://example.org/alice#me
foaf:knows
Leonardo Da Vinci
http://dbpedia.org/resource/Leonardo_da_Vinci
dcterms:creator
BOB
http://example.org/bob#me
foaf:topic_interest
dcterms:title
"Mona Lisa"
The Mona Lisa
http://www.wikidata.org/entity/Q12418
dcterms:subject
rdf:type
schema:birthDate
La Joconde à Washington
http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619
Person
foaf:Person
"1990-07-04"^^xsd:date

Figure 2.5: RDF graph resulting from the N-Triples/Turtle example (https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/example-graph-iris.jpg (last visited December 2016))

2.1.6 RDF Schema (RDFS)

In order to support the definition of vocabularies the RDF introduced the RDF Schema Language (Brickley and Guha, 2014). The RDF Schema is the W3C standard defined as follows:

> *RDF Schema is a semantic extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources (Brickley and Guha, 2014).*

It is written in the RDF using the terms introduced in the section 2.1.5.

The RDFS uses a class and a property system which makes it possible to define the type and the property of the resource being described. Although it is similar to the object-oriented programming (OOP), RDFS uses the *domain* and *range* mechanism. (Brickley and Guha, 2014) In that way the properties are described "in terms of the classes of the resource to which they apply to" (Brickley and Guha, 2014).

To simplify, take a look at a simple example. Typically in the OOP a class *Book* with an attribute called *author* would be defined. The type of the author would be another class called *Person*. In the RDFS a property *author* which has the domain of *Document* and a range of *Person* is defined. This allows subsequently defining other properties without changing the original description. (Brickley and Guha, 2014)

Construct	Syntactic form
Class (a class)	C rdf:type rdfs:Class
Property (a class)	P rdf:type rdf:Property
type (a property)	I rdf:type C
subClassOf (a property)	C1 rdfs:subClassOf C2
subPropertyOf (a property)	P1 rdfs:subPropertyOf P2
domain (a property)	P rdfs:domain
range (a property)	P rdfs:range C

Table 2.2: RDF Schema Constructs
(`https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/#section-vocabulary` (last visited December 2016))

An IRI `http://www.w3.org/2000/01/rdf-schema#` identifies a namespace which defines the RDFS core vocabulary. It is used with the prefix **rdfs:**. Also the RDF namespace `http://www.w3.org/1999/02/22-rdf-syntax-ns#` with the prefix **rdf:** is being used.

Some of the most important classes and properties from the RDFS are summarized in the table 2.1.6.

A resource can be defined as a class using a property **rdf:type** and a class **rdfs:Class**. Defining a Book as a class would look like this:

Book rdf:type rdfs:Class.

There are many other classes and properties of the RDF Schema. The full list and detailed information is available at https://www.w3.org/TR/rdf-schema/ (last visited December 2016).

2.1.7 Web Ontology Language (OWL)

The OWL is at the next level of the Semantic Web technology stack, see figure 2.1. It is another W3C standard defined as follows:

> *The W3C OWL 2 Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be reasoned with by computer programs either to verify the consistency of that knowledge or to make implicit knowledge explicit. (Krötzsch et al., 2012)*

It extends the RDF and the RDFS by adding more constructs for the class and the property description. The OWL became a standard in 2004. In 2009 OWL 2 was introduced and recommended by the W3C. OWL 2 is similar to OWL 1 and is fully backwards compatible. OWL 2 added some new functionalities and descriptive capabilities [11].

An overview of OWL 2 language is illustrated in the figure 2.6. It shows the structure of OWL 2 and how each part relates to another. In the center of the figure is OWL 2 ontology. It can also be seen as an RDF graph. Than there are two layers, syntax and semantics layer.

At the top of the figure is a syntax layer. There are various syntaxes visible which are used for the ontology serialization and exchange. The RDF/XML is the only mandatory syntax because it provides the interoperability among OWL 2 tools. Other optional syntaxes are:

- OWL/XML
- Functional Syntax
- Manchester Syntax and

[11] https://www.w3.org/TR/owl2-overview/#Relationship_to_OWL_1 (last visited December 2016)

- Turtle

At the bottom of figure 2.6 is a semantic layer. Two boxes represent the semantic specifications which define OWL 2 ontology meaning [12]. These specifications are Direct Semantics[13] and RDF-Based Semantics[14].

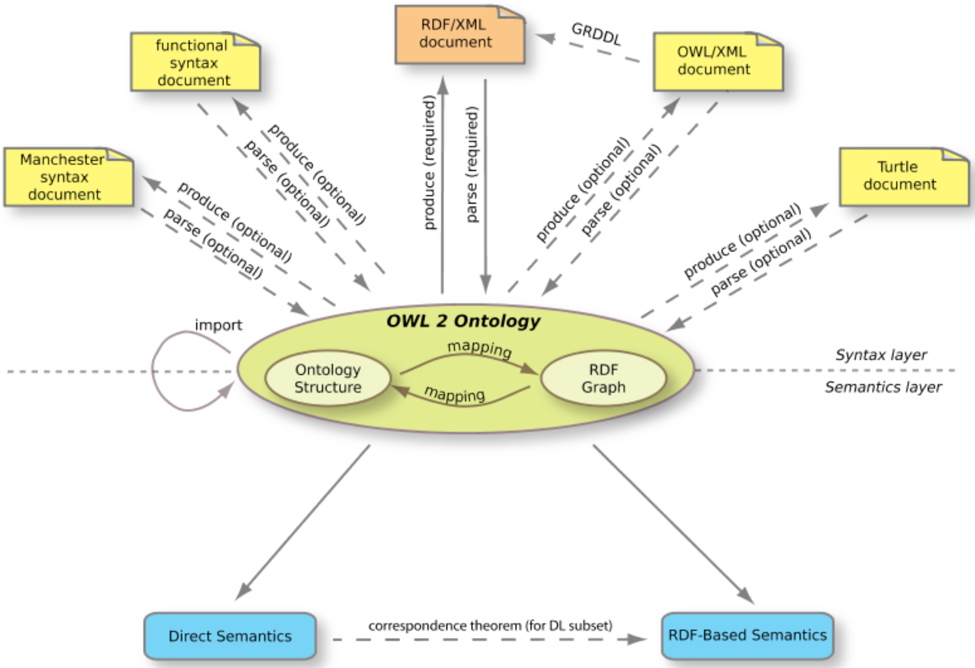


Figure 2.6: OWL 2 Structure (https://www.w3.org/TR/owl2-overview/OWL2-structure2-800.png (last visited December 2016))

In most cases only one syntax and one semantics are used. In that case the structure looks much simpler.

OWL 2 is a general and a very expressive language and as such can be very hard to work with. For various application scenarios three profiles or sub-languages are designed. These are:

[12] https://www.w3.org/TR/owl2-overview (last visited December 2016)
[13] https://www.w3.org/TR/owl2-direct-semantics/ (last visited December 2016)
[14] https://www.w3.org/TR/owl2-rdf-based-semantics/ (last visited December 2016)

- OWL 2 EL
- OWL 2 QL and
- OWL 2 RL

OWL 2 EL “enables polynomial time algorithms for all the standard reasoning tasks” [15]. It is designed for very large ontologies where performance is more important than the expressiveness. OWL 2 QL “enables conjunctive queries to be answered in LogSpace (more precisely, AC0) using standard relational database technology” [15]. It is suitable for applications with smaller ontologies where accessing the data via relational queries (e.g. SQL[16]) is necessary. OWL 2 RL “enables the implementation of polynomial time reasoning algorithms using rule-extended database technologies operating directly on RDF triples” [15]. Also suitable for applications with smaller ontologies where operating on data in form of the RDF triples is needed. Each of these profiles can be interpreted using either Direct or RDF-Based semantics.

2.1.8 Specialized Vocabularies

The vocabularies or ontologies[17] are built on RDF, RDFS or OWL. In the Semantic Web context they “define the concepts and relationships (also referred to as *terms*) used to describe and represent an area of concern”[18].

Some of the well known ontologies are:

- Friend of a Friend (FOAF)
- Dublin Core
- schema.org
- Simple Knowledge Organization System (SKOS)

[15] https://www.w3.org/TR/owl2-overview (last visited December 2016)

[16] Structured Query Language

[17] W3C does not distinguish between vocabulary and ontology. Generally *ontology* is used for more complex collection of terms.

[18] https://www.w3.org/standards/semanticweb/ontology (last visited December 2016)

Most of them have added special class definitions, constraints or properties which are needed depending on the subject area or the knowledge domain.

2.1.9 Simple Protocol and RDF Query Language

In order to be able to extract useful information from the RDF data a method to search and manipulate it is needed. For that purpose Simple Protocol and RDF Query Language (SPARQL) was developed. It is a standardized query language for the RDF graphs. Basically what SQL is to the relational database, SPARQL is to the RDF graphs.

The SPARQL provides four forms of queries:

- SELECT: "Returns all, or a subset of variables bound in a query pattern match." [19]
- CONSTRUCT: "Returns an RDF graph constructed by substituting variables in a set of triple templates." [19]
- ASK: "Returns a boolean indicating whether a query pattern matches or not." [19]
- DESCRIBE: "Returns an RDF graph that describes the resources found." [19]

In this standard also the SPARQL Protocol is important. It conveys the SPARQL queries and updates from clients to the SPARQL processors[20]. Queries are executed through a SPAQRL endpoint which is an interface into RDF data.

The SPAQRL supports four exchange formats. These are:

- XML[21]
- JavaScript Object Notation (JSON)[22]
- Comma Separated Values (CSV)[23]

[19] https://www.w3.org/TR/sparql11-query/#QueryForms (last visited December 2016)

[20] https://www.w3.org/TR/sparql11-protocol/ (last visited December 2016)

[21] https://www.w3.org/TR/rdf-sparql-XMLres/ (last visited December 2016)

[22] https://www.w3.org/TR/sparql11-results-json/ (last visited December 2016)

[23] https://www.w3.org/TR/sparql11-results-csv-tsv/ (last visited December 2016)

- Tab Separated Values(TSV) [23]

The listing 2.3 shows a sample RDF dataset in the Turtle syntax.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

<http://example.org/alice#me> a foaf:Person .
<http://example.org/alice#me> foaf:name "Alice" .
<http://example.org/alice#me> foaf:mbox <mailto:alice@example.org> .
<http://example.org/alice#me> foaf:knows <http://example.org/bob#me> .
<http://example.org/bob#me> foaf:knows <http://example.org/alice#me> .
<http://example.org/bob#me> foaf:name "Bob" .
<http://example.org/alice#me> foaf:knows <http://example.org/charlie#me> .
<http://example.org/charlie#me> foaf:knows <http://example.org/alice#me> .
<http://example.org/charlie#me> foaf:name "Charlie" .
<http://example.org/alice#me> foaf:knows <http://example.org/snoopy> .
<http://example.org/snoopy> foaf:name "Snoopy"@en .
```

Listing 2.3: RDF graph (https://www.w3.org/TR/sparql11-overview/ (last visited December 2016))

This graph can be loaded into the RDF store and than queried in various ways. For instance, to find out who knows the most people a query shown in the listing 2.4 can be used.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name (COUNT(?friend) AS ?count)
WHERE {
    ?person foaf:name ?name .
    ?person foaf:knows ?friend .
} GROUP BY ?person ?name
```

Listing 2.4: SPARQL Query Example (https://www.w3.org/TR/sparql11-overview/ (last visited December 2016))

The result is shown in the table 2.3.

Name	Count
Alice	3
Charie	1
Bob	1

Table 2.3: SPARQL Query Result (https://www.w3.org/TR/sparql11-overview/ (last visisted December 2016))

This is a simple example. Usually datasets are bigger and much more complex queries are needed.

2.2 Knowledge Discovery and Data Mining

Already in the 1990s the amount of data was growing at such a pace that it was unreachable for humans as information or knowledge extractors at that time. Nowadays so much data is being produced that even storing it has become a challenge. According to the study carried out by the International Data Corporation (IDC) in 2013 the volume of data in the world was 4.4 zettabytes or 4.4 trillion Gigabytes (GB). And it is doubling in size every two years. The same study predicts that by 2020 it will reach 44 zettabytes. It grows by a factor of ten from 2013 to 2020[24].

According to Internet Live Stats, a website of the international Real Time Statistics Project, in 2014 there were 1 billion websites on the Internet. To have better understanding how much data is being produced every single second on the internet have a look at the table 2.4.

Name	Count
Internet traffic	40.500 GB
Emails sent	2.549.274
Tweets	7459
Instagram photos	757
Tumblr posts	1.190
Skype calls	2.391
Google searches	57.824
YouTube videos viewed	64.488

Table 2.4: Produced data on Internet in one second (based on results from http://www.internetlivestats.com/one-second/, 12/20/2016 at 14:45)

Storing all the available data in some cases is not an option at all. It is clear that extracting, describing and storing only useful data is required. New techniques and tools are needed in order to catch up with the amount of data being produced.

[24] https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm last visited December 2016

2.2.1 KDD

In the mid 1990s the solutions for automated knowledge discovery and knowledge extraction gained on popularity. In that context the term Knowledge Discovery in Databases (KDD) was popular in particular. Originally at the first KDD workshop in 1989 (Piatetsky-Shapiro, 1991) the term Knowledge Discovery in Databases was introduced to point out that the "knowledge" is the result of data-driven discovery. Later on it became popular in the machine learning and artificial intelligence.

KDD is defined as follows:

> *Knowledge Discovery in Databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. (Usama Fayyad, Piatetsky-shapiro, and Padhraic Smyth, 1996)*

In this definition the term *process* implies that finding knowledge is not a single step. It involves multiple steps like data preparation, finding patterns and knowledge evaluation. And all these steps may be repeated in multiple iterations. Also finding patterns implies identifying structure of the data to designate a proper data model. From the definition four requirements for the discovered patters can be identified. These are:

- Valid
- Useful
- Novel and
- Understandable

Valid implies that there is a high probability that they also hold for the new data. The patterns are *useful* when there is some benefit from them and they may lead to further actions. *Novel* means that they are unexpected and non-obvious. And finally the pattern shall be *understandable* so that humans are able to interpret it.

2.2.2 KDD vs. Data Mining

Over time various names were used for finding useful knowledge and patterns in the data. These are data mining, knowledge extraction, information discovery, information harvesting, data archeology and data pattern processing (Usama Fayyad, Piatetsky-shapiro, and Padhraic Smyth, 1996). The term data mining also became very popular. It is very important to distinguish between the terms KDD and data mining. KDD is seen as an overall process of finding useful knowledge in the data whereas the term data mining is a specific step in this process (Usama Fayyad, Piatetsky-shapiro, and Padhraic Smyth, 1996). Data mining is defined as follows:

> *Data Mining is a step in the KDD process consisting of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data.(U. Fayyad, Piatetsky-Shapiro, and P. Smyth, 1996).*

Application of the data mining step alone is not recommended since it can easily lead to discovery of meaningless and useless knowledge.

2.2.3 Knowledge Discovery Process

The KDD process consists of several steps. Each of these steps is needed to ensure that the derived knowledge is useful. It is defined as follows:

> *KDD Process is the process of using the database along with any required selection, preprocessing, sub-sampling, and transformations of it; to apply data mining methods (algorithms) to enumerate patterns from it; and to evaluate the products of data mining to identify the subset of the enumerated patterns deemed "knowledge" (U. Fayyad, Piatetsky-Shapiro, and P. Smyth, 1996).*

This process is iterative and interactive. Many decisions are being made by the user. The KDD process consists of five stages which are depicted in the figure 2.7.

These are:

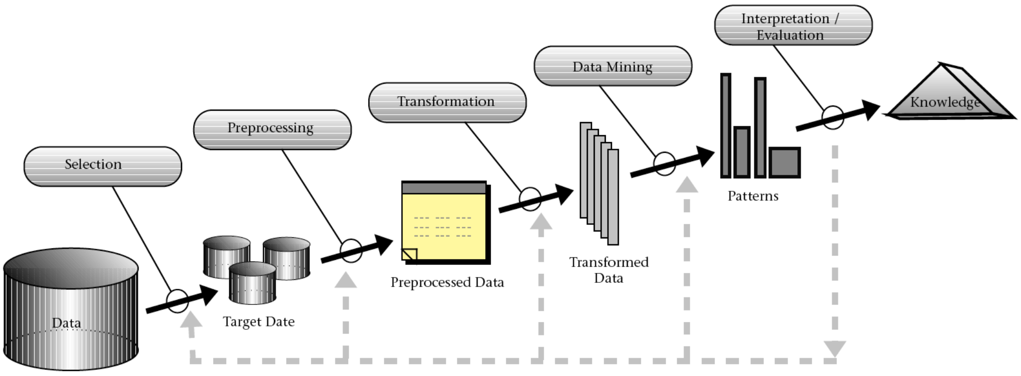


Figure 2.7: Steps in KDD process (Usama Fayyad, Piatetsky-shapiro, and Padhraic Smyth, 1996)

- Selection
- Preprocessing
- Transformation
- Data Mining
- Interpretation/Evaluation

The first stage is the *selection* stage. Its goal is selecting a dataset and is done in two steps. These are:

- Identifying the goal of KDD process
- Selecting a dataset

At the begging of the KDD process it is important to identify the goals from the user perspective. To identify them understanding of the application domain and relevant prior knowledge is needed. Once the goals are identified and defined a target set can be created. That is the second task which needs to be performed in this stage. The goal of this task is to select a dataset or a data sample based on the identified goals in order to perform knowledge discovery or knowledge extraction on it. The selected dataset is then used in the next stages of the process.

Once the dataset is selected the first operations can be performed on it.

This is already done in the next stage in the chain called *preprocessing*. The main goal of this stage is to ensure that the dataset is clean and ready for forthcoming stages. The dataset may often have missing values or contain noise, unnecessary/incorrect/inconsistent data or have formatting issues and contain redundant information. Proceeding with such dataset would have a huge impact on the next stages and on the result of the process itself. It could easily lead to having misleading or useless knowledge at the end. To prevent these issues basic operations like the removal of noise, strategies for handling missing values, etc. are usually performed on the dataset. The resulting dataset is then used for the next stage called *transformation*.

The goal of the *transformation* stage is finding the useful *features* to represent the dataset. The *feature* is "an individual measurable property of a phenomenon being observed (Bishop, 2006)". It is known as an *item* which represents knowledge suitable to Data Mining algorithms. Sometimes also called *attributes* (Machine Learning) or *variables*(statistics). Basically it is just a piece of information useful for prediction. They are extracted based on the goal of the task. Feature extraction is related to the dimensionality reduction and transformation methods. Using these methods large and redundant datasets are transformed into a reduced set of features which contain relevant information. This process is called *feature selection*. A reduced dataset, which contains only selected features, is used as an input dataset for the next stage.

The next stage is the *Data Mining* stage. It is preformed in three steps. These are:

- Selecting data mining method
- Choosing the data mining algorithm
- Data mining

There are various data mining methods like summarization, classification, regression, clustering, etc. Based on the goals of the KDD process one particular data mining method should be selected and used in this stage. There are also various algorithms which can be used in various scenarios depending on the type of the application they are used at. There is no "general" method or algorithm which can be used in any scenario. It is more important to "ask the right questions" and select an appropriate method and algorithm than to optimize the algorithmic details of a particular method. Based on the overall KDD process criteria particular data mining method and algorithm should be matched, e.g. the goal of the process may be focused more on understanding the model than on its predictive capabilities. In the last step of this stage searching for patterns in a particular representational form is performed. The result of this stage is "knowledge" which is then being interpreted and evaluated in the last stage of the KDD process.

The last stage of this process is called the *Interpretation and Evaluation* stage. Extracted knowledge is evaluated, documented and reported to interested parties. It also can be visualized for more convenient interpretation of the results. As already mentioned at the beginning of this section, this is interactive process and there is a possibility to return to any of the previous steps for further iteration.

2.3 Visual Analytics

In the last decade the speed of storing and collecting the information increased rapidly. The data is being produced at a much faster rate than the ability to process and analyze it. As a result, nowadays, in many fields there is an information overload. There are enormous data volumes being produced on a daily basis and in many cases it comes from various sources. In many application areas success depends on having the right information available at the right time. The data needs transformation to the comprehensible knowledge, in order for people to be able to understand it and act on time. Having a lot of data is not an issue. As a matter of fact it is a good thing. The technology for processing, analyzing and understanding it is needed. And that is the problem! Dealing with it properly is still an issue due to the lack of abilities. The models and methods should be identified, which can assist in understanding the data and in the decision making process. The goal of Visual analytics is "to turn the problem with the information overload into an opportunity"(Keim, Kohlhammer, et al., 2010). The idea behind Visual analytics is to help people to make better decisions by including them in the entire process. They get the power to directly interact with the information and to gain deeper insight in order to draw conclusions and make decisions. (Keim, Kohlhammer, et al., 2010)

2.3.1 Historical Background

Issues concerning the information overload are already addressed in the section 2.2. In that section KDD techniques for the automatic data analysis are introduced. These techniques are used to automatically produce useful information from the large amounts of data. The problem with the automated data analysis is that it works reliably only for well understood and defined problems. It also lacks the ability to communicate the knowledge to humans. Knowledge is presented to humans at the end, but human interaction is often needed earlier. The process of being able to make decisions from the data is complex. That is the main reason why humans must be more involved. They can use their creativity and experience to steer the knowledge discovery process and based on those results make the decisions. They should be included in the data analysis process at an earlier stage.

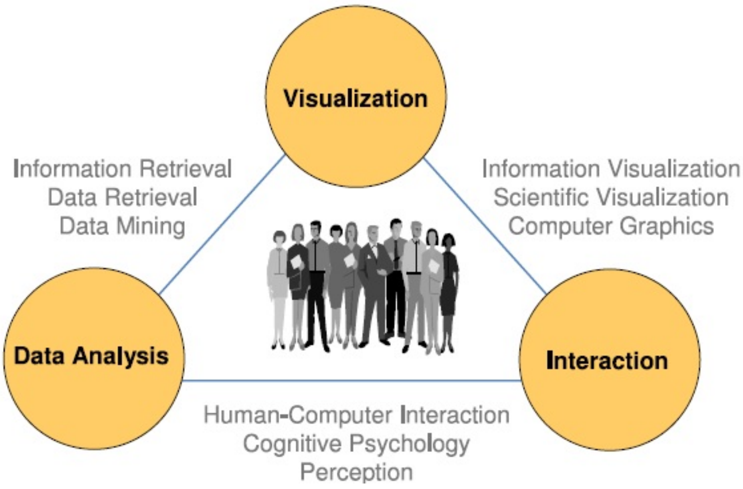


Figure 2.8: Visual Analytics as an interplay between data analysis, visualization, and interaction methods (Keim, Mansmann, et al., 2008).

Another important field in understanding the data is information visualization. Besides the automated knowledge discovery it is the most important field which led to visual analytics. This field also lacks human interaction.

The main step towards visual analytics was the need to move from the confirmatory data analysis to the exploratory data analysis. It was stated in the statistics research community from 1977 by John. W. Tukey in his book called Exploratory Data Analysis. Instead of just presenting the results using chars and other visual representations, it is needed to start interacting with the data and results. (Tukey, 1977)

Involving the humans in the KDD process together with the visualization techniques led to the visual data exploration and visual data mining (Keim, 2001). The term Visual analytics was first used in 2004 (Wong and J. Thomas, 2004) and 2005 (J. J. Thomas and Cook, 2005). Visual analytics evolved from the information visualization and automated data analysis and combines these fields together with the human interaction, see figure 2.8, in the analysis process (Keim, Kohlhammer, et al., 2010).

2.3.2 Definition

Due to the combination of different research areas Visual Analytics is not easy to define.

Back in 2004, Visual analytics was defined as "the formation of abstract visual metaphors in combination with a human information discourse (interaction) that enables detection of the expected and discovery of the unexpected within massive, dynamically changing information spaces" (Wong and J. Thomas, 2004).

In 2005 in the research and development agenda *Illuminating the Path*, with strong focus on Homeland Security in the United States, Visual analytics was defined as:

> *Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces (J. J. Thomas and Cook, 2005).*

In 2010, the definition was more specified based on current practice and is defined as follows:

> *Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex datasets (Keim, Kohlhammer, et al., 2010).*

Visual analytics tools and techniques enable the people to achieve four goals which are cited from the research and development agenda *Illuminating the Path* (J. J. Thomas and Cook, 2005). These goals are:

- Synthesize information and derive insight from massive, dynamic, ambiguous, and often conflicting data
- Detect the expected and discover the unexpected
- Provide timely, defensible, and understandable assessments and
- Communicate these assessment effectively for action

2.3.3 Scope of Visual Analytics

Visual analytics is a highly interdisciplinary combination of research areas such as "visualization, data mining, data management, data fusion, statistics and cognition science (among others)" (Keim, Kohlhammer, et al., 2010). It uses methods from the KDD, statistics and mathematics for automatic data analysis and combines them with the information visualization methods and human capabilities to perceive, relate and conclude (Keim, Kohlhammer, et al., 2010). Visual analytics acts as an integral approach combining these methods. Both of these methods, visualization and data analysis were used together before, but both lack the human interaction factor. This gap is filled by Visual analytics.

The scope of the Visual analytics is illustrated in the figure 2.9. This figure illustrates which methodologies from which disciplines are integrated in Visual analytics. From the information visualization field following methodologies are integrated:

- Information analytics
- Geospatial analytics and
- Scientific analytics

Visual analytics also makes use of three methodologies from the data analysis field. These are:

- Statistical analytics
- Knowledge discovery and
- Data Management and Knowledge Representation

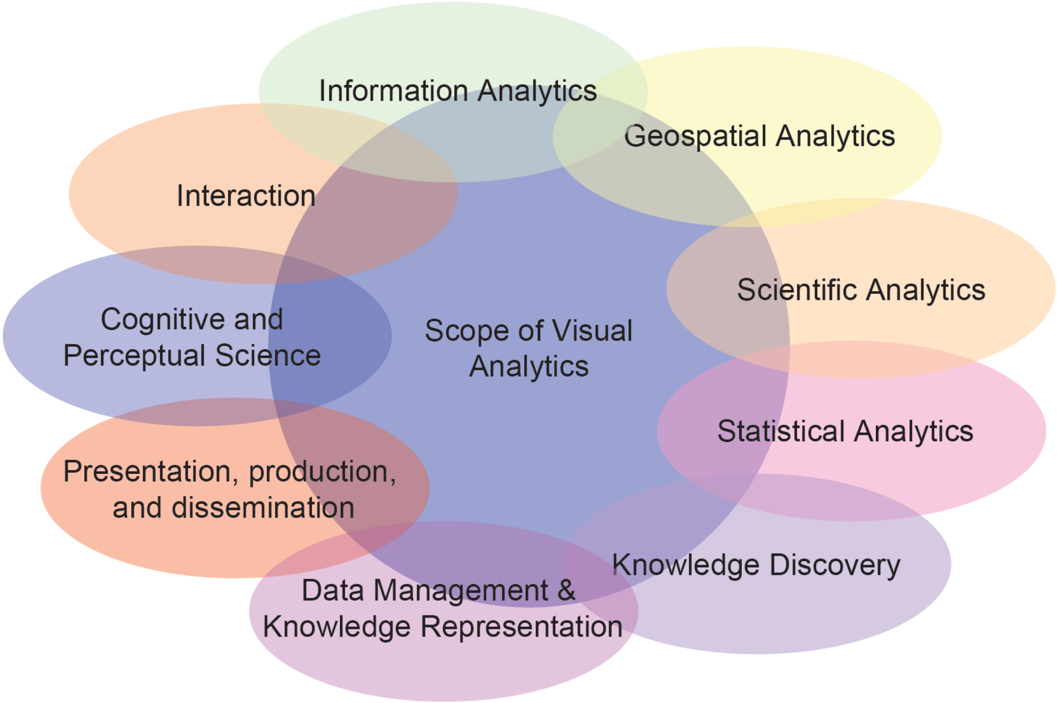

Information Analytics
Geospatial Analytics
Interaction
Scientific Analytics
Cognitive and Perceptual Science
Scope of Visual Analytics
Statistical Analytics
Presentation, production, and dissemination
Knowledge Discovery
Data Management & Knowledge Representation

Figure 2.9: The Scope of Visual Analytics (Keim, Mansmann, et al., 2008).

Human factors (e.g. interaction, cognition, perception, collaboration, presentation, and dissemination) play the major role in visual analytics (Keim, Mansmann, et al., 2008). They are important from two aspects, one is human and computer communication, and the other is the decision making process. The analyst can use his/her knowledge and experience to interact with the data and finally to make conclusions and decisions based on the results.

A more clear view on disciplines which contribute towards visual analytics is illustrated in the figure 2.10. Visual Analytics is much more than the visualization. But still the visualization is in the center of visual analytics. It is used not only for presenting the results of analysis but also to monitor the processes in other included disciplines such as data management or data mining.

Visualization can be classified into scientific and information visualization.

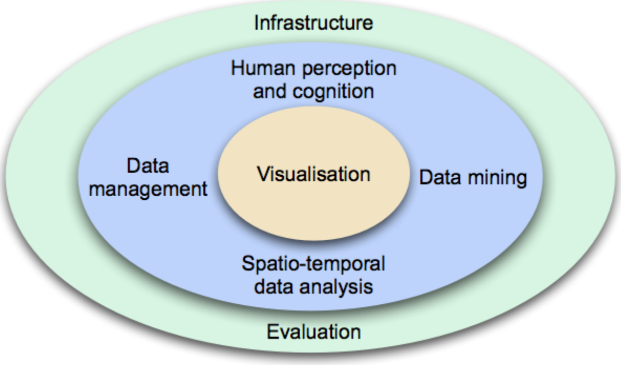


Figure 2.10: Visual analytics is interdisciplinary (Keim, Kohlhammer, et al., 2010).

Information Visualization is used for abstract data with multiple dimensions and is understood as "a graphical representation of data or concepts (Ware, 2004)". More general definition defines it "as the communication of abstract data relevant in terms of action through the use of interactive interfaces"(Keim, Mansmann, et al., 2008). Scientific visualization is primarily used for visualization of the 3-dimensional scientific data obtained from sensors, simulations etc. Compared to information visualization the scientific visualization deals with the scientific data. (Keim, Mansmann, et al., 2008)

There are three major visualization approaches. These are:

- Presentation
- Confirmatory analysis
- Exploratory analysis

The aim of *presentation* is to present the results of the analysis. It presents already fixed or known facts. The techniques used for it largely depend on the user. The *confirmatory analysis* starts with one or more hypotheses. They are than being examined and the result is visualized to either confirm or reject these hypotheses. (Keim, Mansmann, et al., 2008) The *Exploratory analysis* is more difficult. At first there are no hypotheses about the data. It is the process of analyzing and searching for useful information. (Keim, Mansmann, et al., 2008; Tukey, 1977)

2.3.4 The Visual Analytics Process

The visual analytics process is the process of discovering knowledge from the data. It combines visual analysis and automatic data analysis together with the human interaction. The major role in the process plays the analyst. It is adapted to his/her needs so that he/she can control it and based on his/her judgment interact when necessary.

The model of the "Visual analytics process" is illustrated in the figure 2.11 (Keim, Kohlhammer, et al., 2010).

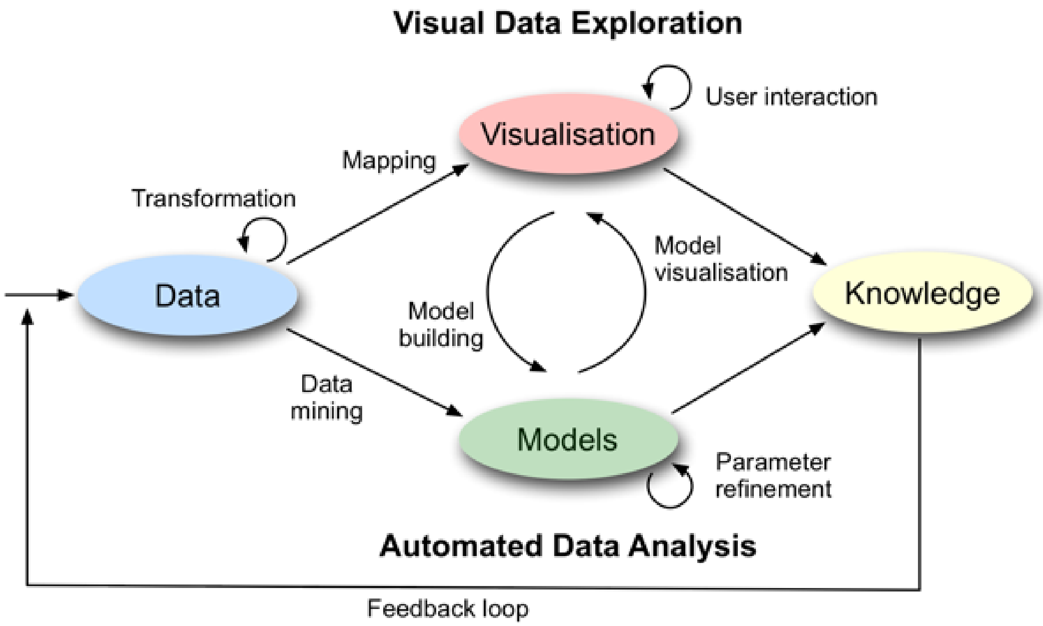


Figure 2.11: Visual Analytics process (Keim, Kohlhammer, et al., 2010).

According to this model the visual analytics process includes key factors such as the data, model, user and visualization. The analysts profit from the interaction of these factors and with some additional parameter refinement are able to gain knowledge. The process is divided into the four stages which are represented through the ovals in the figure 2.11. These stages are:

- Data
- Visualization
- Models
- Knowledge

Each stage has transitions which are represented through the arrows. The first stage is the *data* transformation and preprocessing stage. The data usually comes from different sources and needs to be integrated for further exploration. This stage includes other well known preprocessing tasks like cleaning, normalization, grouping, etc. The data is than transformed (as indicated by the *Transformation* arrow in the figure 2.11) to derive different representations important for the visual and automatic analysis (Keim, Kohlhammer, et al., 2010). This stage corresponds to the second and third stage of the KDD process, see section 2.2.

Already after the first stage the control is handed over to the analyst. He/She can choose to apply either visual or automatic analysis methods in the next stage. In case he/she decides to perform the automatic analysis first then the data mining methods are applied. These methods are used to automatically generate models of the original data. The generated models are then evaluated and if necessary refined by the analyst. He/She can interact with the data and switch between visual and automatic methods. The analyst can use visualization to refine parameters and tryout various data mining algorithms. With the help of the visualization he/she is able to evaluate the generated models. (Keim, Kohlhammer, et al., 2010)

In case the analyst performs the visualization stage first, he/she can interact with the visual representation of the data to get deeper insight in it. For instance he/she can zoom in on specific data areas or tryout different visual representations. Eventually he/she has to confirm his/her hypotheses by the automated analysis. He/She can use revealed information to steer building the models and influence the automatic analysis.

These transitions, *Parameter Refinement*, *Model visualization*, *User interaction* and *Model building*, between the Models and Visualization stage are specific for this process. This leads to a continuous loop between evaluation, refinement and verification of the preliminary results. The best thing about this is that it is driven by the analyst. He/She interacts with the process in real-time and has influence on both automatic and visual methods. Other very important characteristic of this process is the discovery of misleading results at an intermediate stage. On the one side this brings a higher confidence and better results and on the other side gives the analyst the power to act on time.

The last stage of the process is the knowledge. It helps the analyst to draw conclusions and insights based on the presented information. The analyst has the ability to go back to the first stage and start with the process again with new information (as indicated by the *Feedback loop* arrow in the figure 2.11). (Keim, Kohlhammer, et al., 2010)

Regarding data presentation on the screen Shneiderman proposed the Visual Information Seeking Mantra "Overview first, zoom/filter, details on demand"(Shneiderman, 1996) as the guide to visually explore the data. Creating an overview visualization from large amounts of data is very hard and may easily lead to loss of useful information or patterns. This issue also makes zooming and filtering useless due to insufficient amount of information provided for further examination. (Keim, Kohlhammer, et al., 2010)

For visual analytics context following mantra is presented:

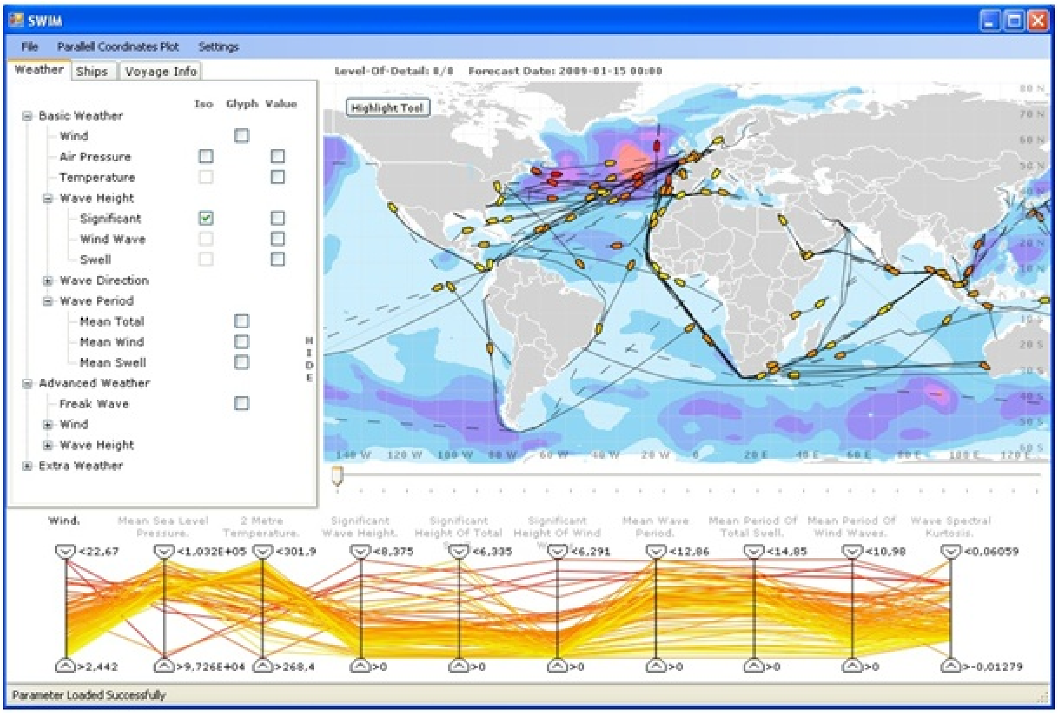
Analyze First -
Show the Important -
Zoom, Filter and Analyze Further -
Details on Demand.
(Keim, Mansmann, et al., 2008)

This clearly indicates that analyzing the data based on value of interest must be done at first. Analysis is then followed by the visualization of important data aspects and interaction models which are provided to help the analyst to get data details on demand.(Keim, Kohlhammer, et al., 2010)

2.3.5 Application of Visual Analytics

There are many application areas where large amounts of data need to be processed and analyzed. Visual analytics is crucial in such fields. For example for monitoring climate and weather. Large amounts of data are collected from different sources (satellites and sensors throughout the world) in short amount of time. Visual analytics approach helps to interpret this data and to understand dependencies between the climate factors and climate change scenarios which otherwise would remain hidden.

Another example is the shipping industry which needs to efficiently plan routes based on weather factors. For that purpose Ship and Weather Information Monitoring (SWIM) application is created. It is used for vessels monitoring and route optimization based on future weather development. (Lundblad, Eurenius, and Heldring, 2009) Figure 2.12 gives an overview of the application.



Level-Of-Detail: 8/8 Forecast Date: 2009-01-15 00:00

Parameter Loaded Successfully

Figure 2.12: Visual Analytics application for monitoring vessels (Lundblad, Eurenius, and Heldring, 2009).

Some of very important application domains are: business intelligence, biology, medicine, security, geo-graphics, emergency management, physics and astronomy (Keim, Kohlhammer, et al., 2010). Each of these application domains face a lot of challenges and need efficient solutions. Adequate visual analytics applications help them to face the challenges and play a key role in many cases.

2.4 Personal Learning Environment

The learners nowadays have plenty of options to learn and seek for knowledge. In the past decade many tools and methods have been explored in order to improve the learning process and experience. Major focus was on using new technologies to facilitate learning.

The World Wide Web is the resource that enables finding information beyond the traditional courses and classrooms. Over time the Web evolved. In 2005 Tim O'Reilly introduced the Web 2.0 (Tim, 2005) which changed the relation between users and information. Instead of only accessing information, the users had a large collection of tools which enabled creating, organizing and making meaning of the information. Users can also interact with other users in a shared environment. These Web 2.0 tools are known as social software which is simply defined as a software that supports group interaction (Allen, 2004). This also had a huge impact on learning experience and learners. It gave them plenty of possibilities to learn and gain knowledge. Learning environments were created which enabled the learners to organize the content and interact with other learners in a personalized way. This concept which puts the learner in focus is known as Personal Learning Environment (PLE) (Martindale and Dowdy, 2010).

2.4.1 Definition

There is no widely accepted PLE definition. Available definitions provide insight into the characteristics of a PLEs. The term "personal learning environment" was first mentioned in an unpublished paper by Olivier and Liber in 2001 (Olivier and Liber, 2001). Starting from 2006 PLE gets more attention. In 2006 Ron Lubensky refers to PLE as "a facility accessed by learners where content is aggregated, organized and vetted for one's own ongoing learning needs" (Lubensky, 2006). In the same year Anderson defines it as "a unique interface into the owner's digital environment. It integrates their personal and professional interests (including their formal and informal learning), connecting these via a series of syndicated and distributed feeds"(Anderson, 2006). Anderson sees PLE also as a portfolio system, a profile system and a social as well as an information environment (Anderson, 2006).

Van Harmelen also defined PLE in 2006. He sees PLEs as an information system which allow learners “to take control of and manage their own learning” (Van Harmelen, 2006). According to him PLE can be “composed of one or more sub-systems which can support learners to set and achieve their learning goals” (Van Harmelen, 2006).

Downes refers to PLE as a recognition that the “one size fits all” approach of Learning Management System (LMS) is insufficient to meet student needs (Downes, 2007).

More recent definition describes PLE as learning applications where learners on the one side can integrate and organize online content and on the other side also provide content and other things developed in the PLE to other online environments. (M. K. S. Schaffert, 2009)

2.4.2 PLE vs LMS

LMS are mainly developed to support educational institutions in student management and course organization. They make assumptions that learning occurs best in a “one-size-fits-all” context. This gives the learners no influence in learning process. Learners need to be guided and encouraged rather than being “managed” (Martindale and Dowdy, 2010). PLE gives the control of the learning process to the learner. With PLE the learner can arrange and use the learning resources and learning tools in a way that he/she prefers.

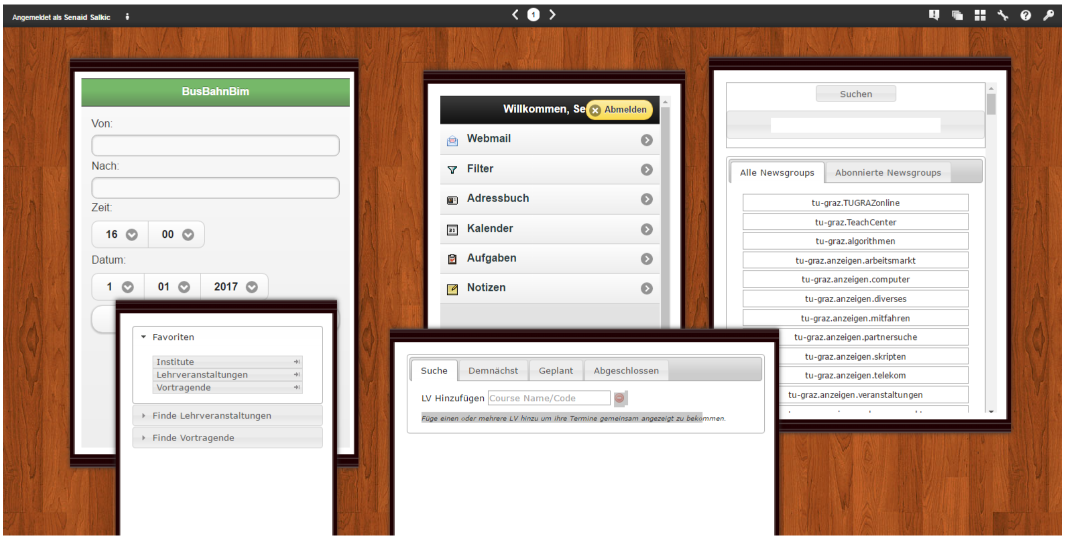
The table 2.5 illustrates some of the differences between PLE and LMS that are outlined in the literature (Wilson et al., 2006; Chatti, 2010; Mott, 2010).

PLE	LMS
Learner-centric	Course or teacher-centric
Tools selected based on learner needs	One-size-fits-all tools
Personal learning support	Formal learning support
Distributed, open	Centralized, closed, standard
open source tools	Licensed software
Knowledge push and pull	Knowledge push

Table 2.5: Differences between PLEs and LMSs (Renon, 2013)

2.4.3 PLE at Graz University of Technology

At Graz University of Technology (TU Graz) the potential of a PLE and its importance for learners was recognized quite early. The first version was launched already in 2010. The goal was to integrate the existing university services with the World Wide Web in one platform and in a personalized way (Taraghi, Ebner, Till, et al., 2010; Ebner and Taraghi, 2010). The TU Graz PLE was continuously improved since the initial launch. Already in 2011 it was redesigned where new User Interface (UI) with app-based architecture was introduced. This was necessary in order to improve attractiveness and to make it a better learner-centered application (Ebner, Scerbakov, et al., 2010; Taraghi, Ebner, and Clemens, 2012). The TU Graz PLE follows a widget-based mashup concept where learners can arrange small applications called widgets in the personal desktop space (Taraghi, Ebner, and S. Schaffert, 2009; Taraghi, Ebner, Till, et al., 2010; Ebner and Taraghi, 2010). It is very similar to mobile app environments.

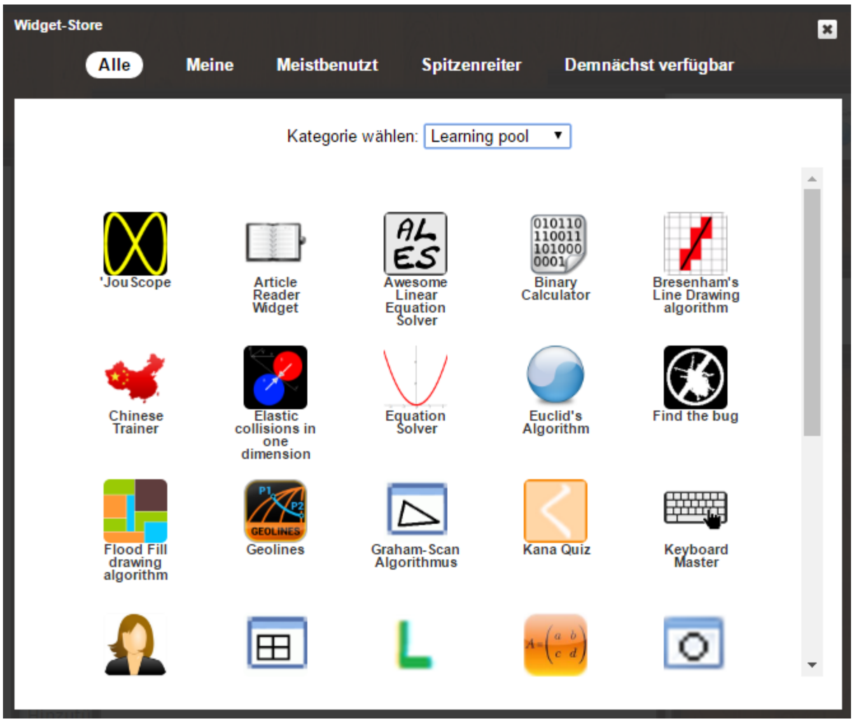


Füge einen oder mehrere LV hinzu um ihre Termine gemeinsam angezeigt zu bekommen.

Figure 2.13: PLE Space at Graz University of Technology

It consists of the widget container and the widgets themselves. The widgets are programmed by students of computer science during their courses and projects (Taraghi, Ebner, Till, et al., 2010).

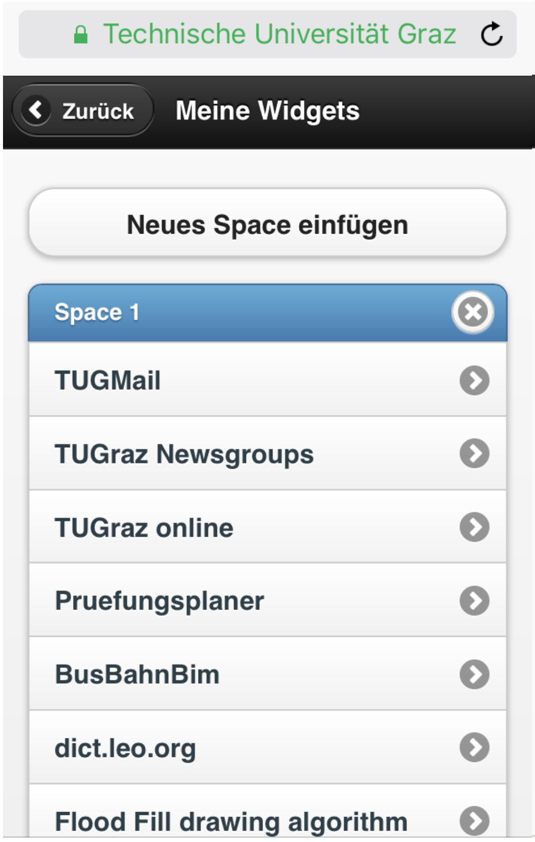
The figure 2.13 illustrates one Space in a PLE desktop view. Learners have the option to install the widgets from the widget store. The figure 2.14 illustrates the widget store where the learners can search for the widgets.



Kategorie wählen: Learning pool

Figure 2.14: TU Graz PLE Widget Store

Also the potential of ubiquitous learning was realized very soon. For that purpose the PLE was extended with the mobile interface (Taraghi, Softic, et al., 2013). The figures 2.15 and 2.16 illustrate the list of learners spaces and installed widgets as well as the widget store of the mobile version of a PLE at TU Graz.



Technische Universität Graz

Zurück

Meine Widgets

Neues Space einfügen

Space 1

TUGMail

TUGraz Newsgroups

TUGraz online

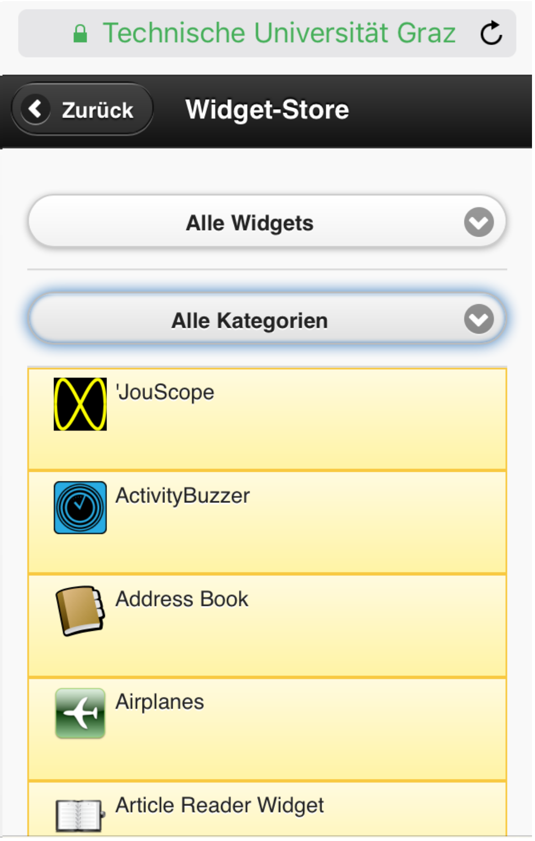
Pruefungsplaner

BusBahnBim

dict.leo.org

Flood Fill drawing algorithm

Figure 2.15: Personal Spaces in mobile PLE



Technische Universität Graz

Zurück

Widget-Store

Alle Widgets

Alle Kategorien

'JouScope

ActivityBuzzer

Address Book

Airplanes

Article Reader Widget

Figure 2.16: Widget Store in mobile PLE

According to the first reports the response of the learners to the TU Graz PLE is very positive. In first two years of its existence there were more than 4000 active learners (Softic et al., 2013; Taraghi, Softic, et al., 2013). The learners use the PLE to learn and gain knowledge but they also contribute to its content by developing various widgets needed for learning.

3 Concept

After presenting the related fields such as the Semantic Web, Knowledge Discovery and Data Mining, Visual analytics and PLE, this chapter introduces the concept definition for the practical implementation of this thesis. Understanding the solutions that each of the related fields provide is important because they are used as the basis for this work. The general goal of this work is to extract, analyze and present the knowledge from the data. This includes using techniques presented in each of the related fields such as data preprocessing, transformation, modeling, retrieval and interactive visualization.

In the following section the visualization pipeline is introduced. It aims to define and describe all steps of the development process needed for reaching the goal of this work.

3.1 The Visualization Pipeline

The concept presented here is heavily based on the visual analytics process (see section 2.3). The results of this work shall help understanding the learners and lead to improvement of the learning process at TU Graz PLE. The visualization pipeline illustrated in the figure 3.1 presents the path from the data creation to the knowledge extraction and visualization.

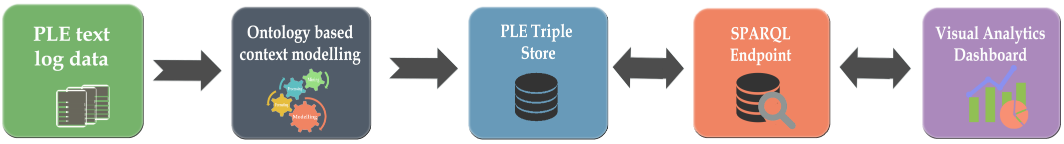


Figure 3.1: Analytical Dashboard Pipeline

In this pipeline five steps are defined and presented. These are:

- Input Data
- Data transformation and modeling
- PLE Triple Store
- SPARQL Endpoint and
- Visual Analytics Dashboard

Each of these steps are described in detail in the following sections.

3.2 Input Data

In order to be able to perform any kind of data analysis there must be some kind of data or information provided. Various works emphasized the importance of tracking activity data in learning systems (Prinsloo, Slade, and Galpin, 2012; Santos et al., 2012).

Very soon after the initial launch of TU Graz PLE (see section 2.4.3) the importance of tracking its usage was realized and a module for tracking PLE usage was implemented (Taraghi, Stickel, and Ebner, 2011).

The data produced by the PLE tracking module is used as input for the practical part of this thesis. The PLE Tracking module is introduced in the following section 3.2.1.

3.2.1 PLE Tracking Module

The PLE widget engine contains the tracking module. All captured information on the client side from all active widgets are periodically sent to the server for further processing and storage (Taraghi, Stickel, and Ebner, 2011). The captured data are ordinary text logs which include raw information about the widgets and their usage. For the purpose of better understanding of the context, in which usage of the widget occurs, the following information are captured:

- Learner

- Widget
- Activity
- Timestamp

The widget is used by a learner in a certain context or activity and occurs at a specific time period.

The input data are pre-processed into the JSON[1] format by splitting the information about the learner, widget and activity into the separate blocks (Softic et al., 2013).

3.3 Data Transformation and Modeling

Simple logging of data, as done by the tracking module, is unstructured and as such not easily queryable. Also visualizing such data implies formatting it and for each new visualization framework additional effort is needed. Importance of structuring the data in a machine understandable format is outlined in the chapter 2.1.

Therefore the next step in visualization pipeline is the transformation and modeling of the input data. In fact, the main precondition for meaningful visual analytic results is the meaningful modeling of the underlying data. This step needs to produce instanced data which are dynamically processable in order to generate responsive visualizations.

For modeling of the input data the semantic based approach is used. It is a flexible and scalable data model approach which offers the possibility to enrich the data with the context in which it was collected. This approach also provides SPARQL as a retrieval technology which delivers various output formats, such as XML or JSON, which are very wide spread as input for visualization frameworks.

The input dataset needs to be transformed to the RDF instances which give them structure and enable flexible data retrieval. Additionally an adequate modeling ontology is needed which enables reflecting the context in which

[1] http://json.org/ (last visited December 2016)

the input data is captured. This will enable revealing some hidden and non-obvious aspects of the dataset.

In order to find the adequate ontology, full understanding of the dataset is needed. As already mentioned before (see section 3.2.1) the tracking module captures information about learner, widget and activity. To get the best insights out of the input dataset ontology needs to describe these entities respectively as well as the relations and context among them.

Based on the input dataset following views are important for meaningful modeling:

- Learner view
- Widget view
- Activity view

The learner view outlines the learner and his/her relation towards the learning surrounding and the learning activity. The Widget view reflects the learning widget usage among the learners over time. And finally the Activity view represents the activity which is tightly related to the widget installed by the learner. The ontological framework for learning representation provided by the Intelligent Learning Extended Organization (IntelLEO[2]) matches the identified requirements and therefore using this ontology for modeling the input dataset is a logical choice.

3.3.1 IntelLEO EU Project

This project develops "intelligent technologies to support learning and knowledge-building (LKB) activities[2]". It tries to encourage the learners to participate in these activities by exploring the possibilities to enhance the responsiveness of the LKB environments[2]. One of the main goals of this project is building an "innovative ontological framework for learning representation which includes learners, context and collaboration models, serving to achieve the targeted synergy[2]".

[2] https://www.intelleo.eu/index.php?id=5 (last visited December 2016)

The IntelLEO ontology framework includes a set of IntelLEO ontologies. They rely on and link to already available and widely used ontologies like FOAF (ontology for modeling people and online communities), Dublin Core (ontology for content annotation)[3], etc. Two special ontologies from this framework are eminent for this work. These are:

- Learning context ontology and
- Activity ontology

Learning context ontology formalizes the learning context which is defined as follows:

> *Learning context is a specific learning situation characterized by: the learning activity that was performed or the learning-related even that occurred; the content that was used and/or produced; the individual(s) involved; the (online) environment where it took place; and the time when it took place*[4].

Activity Ontology "allows modelling different kinds of learning and knowledge building activities, both individual and collaborative ones"[5].

These two ontologies enable describing the activity, the environment (the widget) and the learner along with the events related to them as well as the time when the learning situation occurred. Therefore they are used for the modeling of the input dataset.

3.4 PLE Triple Store

Each time when the learner performs some action in a PLE a logging entry is created. With the help of RDF and presented ontologies each of these entries from the input dataset can be transformed to RDF instances.

[3] https://www.intelleo.eu/index.php?id=183 (last visited December 2016)

[4] https://www.intelleo.eu/ontologies/learning-context/spec/ (last visited December 2016)

[5] https://www.intelleo.eu/ontologies/activities/spec/ (last visited December 2016)

The figure 3.2 illustrates a sample RDF instance of the learning context which is produced from the input dataset. The learning context ontology links PLE usage as event to the widget where it occurred. The ontology namespace for the learning context is *http://www.intelleo.eu/ontologies/learning-context/ns* and its prefix is *lc*. Instead of writing *http://www.intelleo.eu/ontologies/learning-context/ns/environmentRef* to reference the environment where a specific learning situation occurred, the prefix *lc:environmentRef* can be used.

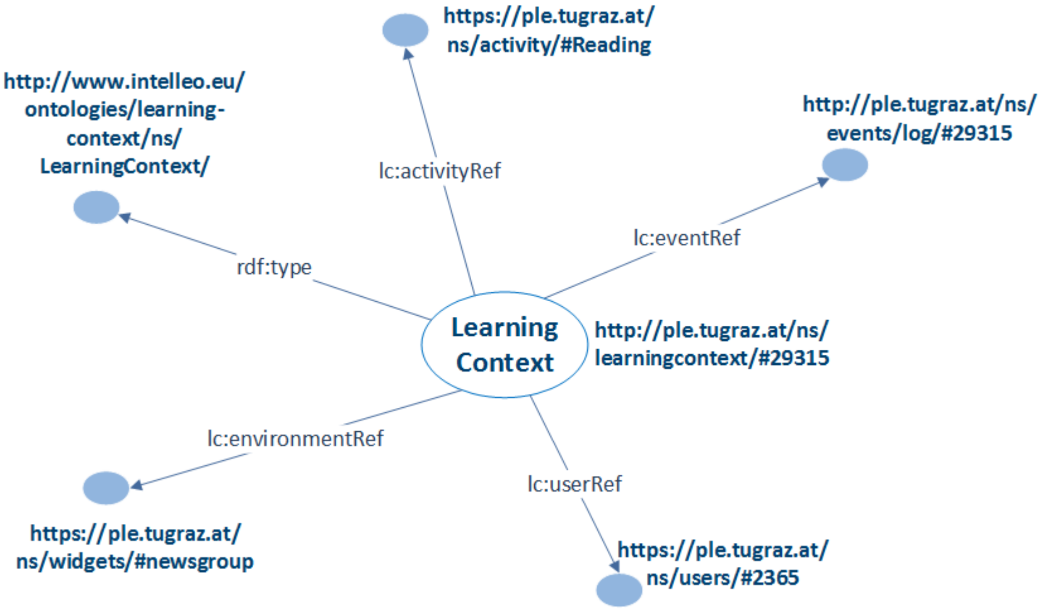

https://ple.tugraz.at/ ns/activity/#Reading
http://www.intelleo.eu/ ontologies/learning-context/ns/ LearningContext/
lc:activityRef
http://ple.tugraz.at/ns/ events/log/#29315
lc:eventRef
rdf:type
Learning Context
http://ple.tugraz.at/ns/ learningcontext/#29315
lc:environmentRef
lc:userRef
https://ple.tugraz.at/ ns/widgets/#newsgroup
https://ple.tugraz.at/ ns/users/#2365

Figure 3.2: Sample RDF instance of Learning Context

The learning context depicted in the figure 3.2 describes a log event denoted as a class *a:Logging*, which is a subclass of an event *a:Event* which occurred inside the learning widget named **newsgroup** represented through the class *a:Environment* and performed by the learner **#2365** (*um:User*). The action which was registered is **Reading** represented through the class *a:Activity*.

The resulting RDF instances represent the Linked Data and are stored in the RDF memory store called a PLE Triple Store.

3.5 SPARQL Endpoint

One of the main reasons for choosing the semantic based approach for modeling the input data is the possibility of flexible data retrieval via the SPARQL endpoint. It is briefly described in the section 2.1. A PLE Triple store is a Graph database for Linked Data and provides SPARQL Endpoint which enables querying the data from the database using the SPARQL queries. The huge advantage of the SPARQL endpoint is that it delivers results in various formats which give flexibility. For the purpose of this work the results are needed in JSON format. This format is defined as input for the visual analytics dashboard which deals with processing and visualizing of the retrieved information.

3.6 Visual Analytics Dashboard

Developing an analytical dashboard application is certainly the most important step in the visualization pipeline defined earlier in this chapter. The goal of this application is to give clear insight in the PLE usage by visualizing the results based on user interest.

One very important aspect for the implementation, as discussed in 2.3, is supporting the user interaction. Therefore, this application needs to give the user a lot of control. He shall be able to decide which data and from which aspect is visualized, or which time period needs to be considered.

In the second step of the pipeline, 3.3, based on the dataset that is provided three aspects or views from the dataset were identified. These are the learner view, the widget view and the activity view. Therefore all three views need to be visualized, separately, as well as the relation among them. Furthermore, another important aspect is the time period when certain action occurred. All of these aspects are considered in the development phase of the dashboard application.

Already at an early stage of this work a scientific paper was written which included the brief overview of the concept presented here as well as the first preliminary results. This paper was committed to the 13th E-Learning Conference in Munich in 2015 (Salkic et al., 2015).

In the next chapter the technical details are discussed. Used methods, tools and techniques that enable the extraction and visualization of the input dataset are presented. Additionally, implementation details are described in depth.

4 Technical Details and Implementation

This chapter describes in detail the practical approach used to analyze, model, query and visualize the input data based on the concept presented in the previous chapter. All technical and implementation details as well as all technologies and tools used for the practical part are described in the following sections.

Based on the visualization pipeline presented in the chapter 3, see figure 3.1, the implementation is divided into three main stages. These are:

- Creating a PLE triple store
- Configuring a server for the Analytical Dashboard
- Implementing an Analytical Dashboard client application

The first stage deals with the data transformation and modeling which as a result creates a PLE Triple Store. The PLE triple store is further used in a second stage where the Analytical Dashboard server is configured and installed. The third and most important stage is the implementation of the Analytical Dashboard client application.

Before looking at each of these stages in detail next section presents the development platform and the tools needed for accomplishing the practical part of this work.

4.1 Development Platform and Tools

All development tasks are performed on Lenovo Thinkpad T410 with the following hardware setup:

- Processor: Intel Core i5 M 520 2.4 GHz
- Memory: 4 GB 1333 MHz DDR3
- System Type: 64-bit Operating System
- Software: Windows 10 Pro

The main development IDE[1] used for writing the software is Eclipse Luna SR2, version 4.4.2. Eclipse IDE is highly customizable and provides a lot of extensions in form of plugins via the Eclipse Marketplace. The plugins enable auto-completion, code highlighting and many other functions which facilitate coding. Full documentation as well as the available versions are available at https://eclipse.org/downloads/packages/release/luna (last visited December 2016).

4.2 Creating a PLE Triple Store

In order to limit the scope of this work it was needed to perform this stage separately from the dashboard implementation. In this stage the input data is transformed and modeled from the text log files to ontology based RDF-instances. These instances are then uploaded into a PLE triple store which is further used in the second stage of implementation.

Choosing an adequate programming language to convert and model the input data was the first challenge of implementation. For this purpose Python[2] programming language was selected. One of the main reasons for using Python was the fact that it is widely used for such tasks and offers various packages for working with RDF and OWL ontologies like the one used in our case: RDFLib[3] [4].

After finishing the script and validating the results from the test datasets it was time to convert the entire input dataset. As a part of preprocessing the text log files are converted to JSON format which is further used to convert

[1] Integrated Development Environment

[2] https://www.python.org/ (January 2017)

[3] RDFLib is a Python library for working with RDF, a simple yet powerful language for representing information

[4] https://github.com/RDFLib (last visited December 2016)

the dataset into RDF-instances. The converted dataset, a PLE triple store, contains all instances and is used in a second stage. Example of converted text log entry to the RDF instance with the learning context ontology is depicted in the listing 4.1.

```
@prefix ao: <http://intelleo.eu/ontologies/activities/ns/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix lc: <http://www.intelleo.eu/ontologies/learning-context/ns/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix um: <http://intelleo.eu/ontologies/user-model/ns/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://ple.tugraz.at/ns/learningcontext/#1> a lc:LearningContext ;
    lc:activityRef <https://ple.tugraz.at/ns/activity/#Authoring>,
        <https://ple.tugraz.at/ns/activity/#DiscussAsynchronously>,
        <https://ple.tugraz.at/ns/activity/#Reading> ;
    lc:environmentRef <https://ple.tugraz.at/ns/widgets/#newsgroup> ;
    lc:eventRef <http://ple.tugraz.at/ns/events/log/#1> ;
    lc:userRef <https://ple.tugraz.at/ns/users/#6826> .
```

Listing 4.1: Learning Context in N3-notation

Since next two stages, server side solution and client implementation are closely related and together form one application, next chapter discusses the architecture of the application.

4.3 The Application Architecture

This section describes the architecture of the Analytical Dashboard application.

The knowledge gathered from the field of the visual analytics, see chapter 2.3, had a huge impact on design and the architecture of the Analytical Dashboard application. Looking at the pipeline presented in the concept, see figure 3.1, it is clear that implementing the dashboard application combines the last three steps from the pipeline. These are the PLE triple Store, SPARQL Endpoint and the dashboard client. One of the reasons for combining these steps can be justified by the requirements of this application. Besides knowledge visualization, two other requirements are very important:

- User interaction
- Simultaneous data retrieval

The user needs to interact with the results in order to get better understanding. The importance of user interaction is discussed in the chapter 2.3. Therefore these three steps are related, since each user interaction results in a request which needs to be executed on the dataset. Another important aspect is simultaneous data retrieval where supporting multiple users at the same time is needed. That is indeed a very realistic scenario where multiple users analyze the results retrieved from the same dataset simultaneously.

When designing the architecture these requirements were considered and led in the direction of a client-server model. Creating a distributed application structure is also a logical decision for these kinds of applications. In this model the database is located at a server which provides a service to one or many clients. The clients initiate the requests for such services. This model is illustrated in the figure 4.1.

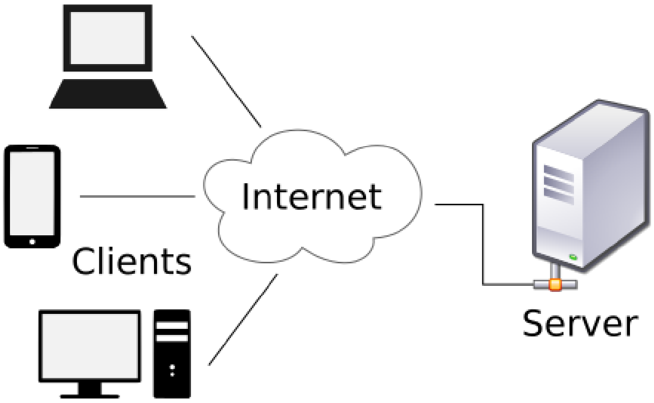


Figure 4.1: A computer network diagram of clients communicating with a server via the Internet. (https://commons.wikimedia.org/w/index.php?curid=15782858 (January 2017))

The architecture of an application is based on a client-server model. It is depicted in the figure 4.2. The server contains a PLE Triple store and provides a service called SPARQL Endpoint to the clients. Clients use the analytical dashboard application to initiate requests for the SPARQL Endpoint and retrieve information from the PLE Triple store.

Choosing the right technologies for the server side as well as for the client side solution is very important at this stage. Since the PLE triple store is a quite big database and the queries from the clients may also be complex, the performance of the server must be considered. The client application needs to provide good visualization charts which enable getting clear insight in the results and give the possibility to interact with the visualized information and explore further.

In the following sections technologies as well as the implementation details of a server and a client solutions are discussed in more detail.

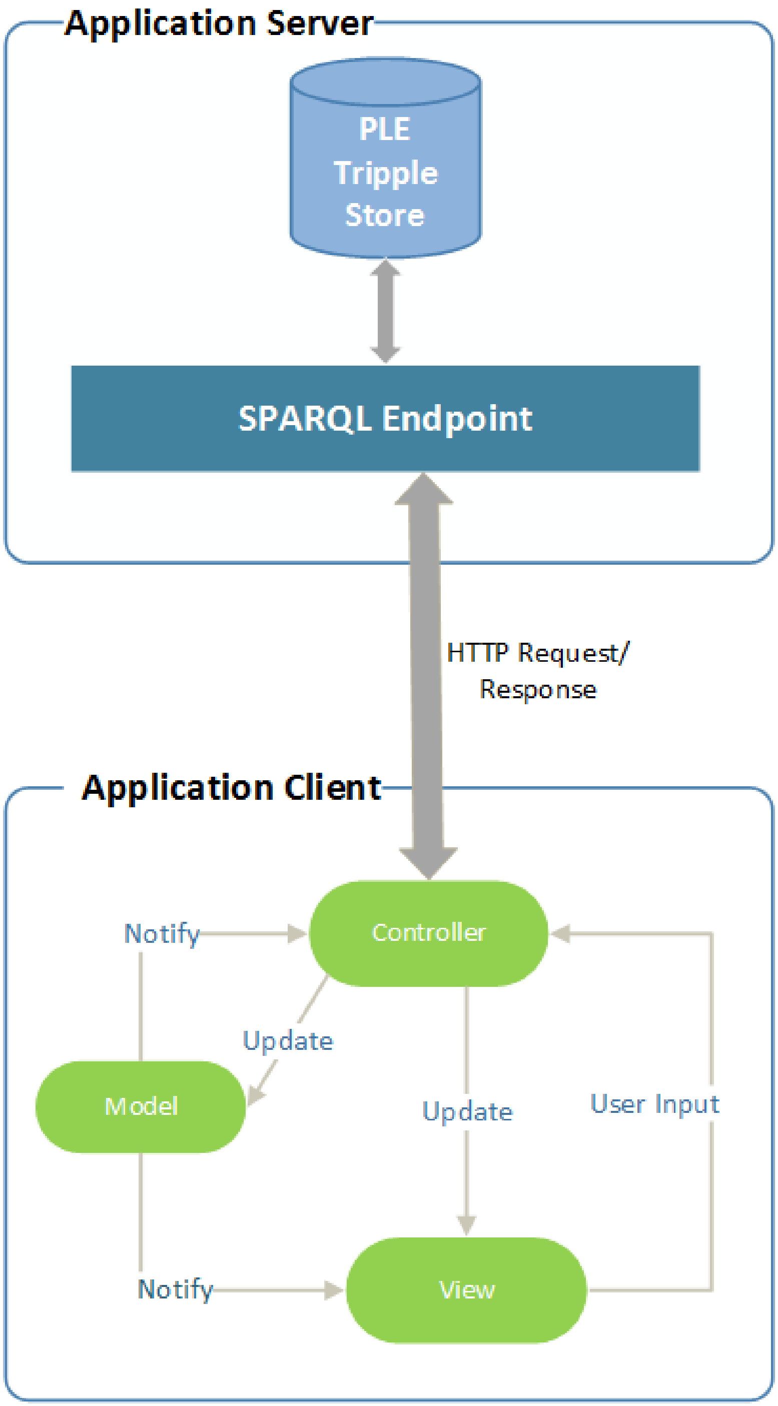


Figure 4.2: Analytical Dashboard Application Architecture

4.4 The Analytical Dashboard Server

As already mentioned, finding a suitable server side solution is very important. Luckily there are already existing solutions so the implementation was not necessary. It was important to find a software solution with good support for Semantic Web technologies. The search for possible APIs[5] resulted in a huge list. Some of the available solutions are:

- RAP - RDF API for PHP[6]
- RDF4J[7]
- ARC2[8]
- AllegroGraph[9]
- Apache Jena[10]
- and many others[11]

Although choosing the suitable API was not an easy decision, the framework from Apache Jena was selected. It is "a free and open source Java framework for building Semantic Web and Linked Data applications[10]". It offers full support for almost all Semantic Web technologies, has active community and is under successive development. Apache Jena has various extensions[10]. These are:

- RDF API: The core RDF API
- ARQ: a query engine that supports SPARQL
- TDB: Jena component for RDF storage and query
- Fuseki: a SPARQL server
- Ontology API: The Jena ontology API and
- Inference API: Reasoners and rule engines

[5] Application programming interface

[6] http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/ (last visited December 2016)

[7] http://rdf4j.org/ (last visited December 2016)

[8] https://github.com/semsol/arc2/wiki (last visited December 2016)

[9] http://franz.com/agraph/allegrograph/ (last visited December 2016)

[10] https://jena.apache.org/index.html (last visited December 2016)

[11] https://en.wikipedia.org/wiki/List_of_subject-predicate-object_databases (last visited December 2016)

The Fuseki extension fulfills the requirements for this task and therefore is selected as a server solution.

4.4.1 Apache Jena Fuseki

Apache Jena Fuseki is a SPARQL server. It enables exposing the triples as a SPARQL end-point accessible over HTTP[12] and provides REST[13]-style interaction with the RDF data[10]. Fuseki can be run in several ways. These are:

- As a standalone server
- As an operating system service
- As a Java web application (WAR file) and
- As an embedded SPARQL server

It also provides a user interface for server monitoring and administration. Security is provided by Apache Shiro[14].

For this work Fuseki is used as a standalone server. At the time of development there are two versions of Fuseki, Fuseki1[15] and Fuseki2[16]. Although both versions are supported, Fuseki2 was chosen since it is the latest version. Detailed information about Fuseki2, how to configure and start it is available online at `https://jena.apache.org/documentation/fuseki2/`(last visited December 2016).

After configuring and starting a Fuseki server it is important to add or load the dataset, a PLE Triple Store. There are two way of doing that. The dataset can be loaded via the command line when starting the server by specifying the **- -loc** argument. Other, much convenient way is using the user interface as illustrated in the figure 4.3.

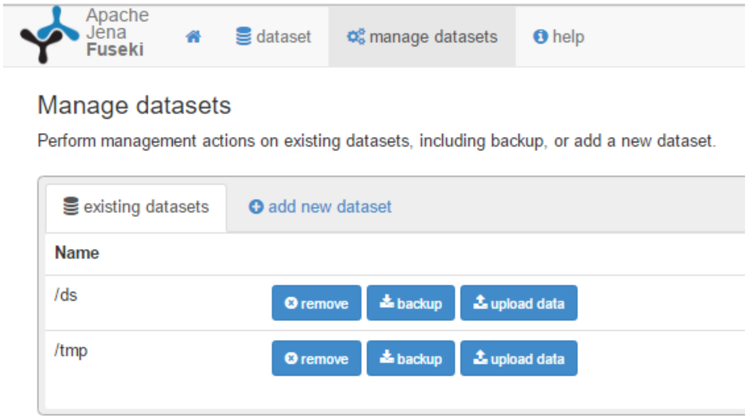
[12] Hypertext Transfer Protocol

[13] Representational state transfer

[14] `http://shiro.apache.org/` (last visited January 2017)

[15] `https://jena.apache.org/documentation/serving_data/` (last visited December 2016)

[16] `https://jena.apache.org/documentation/fuseki2/` (last visited December 2016)



Manage datasets

Perform management actions on existing datasets, including backup, or add a new dataset.

Name
/ds
/tmp

Figure 4.3: Managing Fuseki Datasets

Once the dataset is loaded the server provides some information about it, like the number of triples in all graphs.

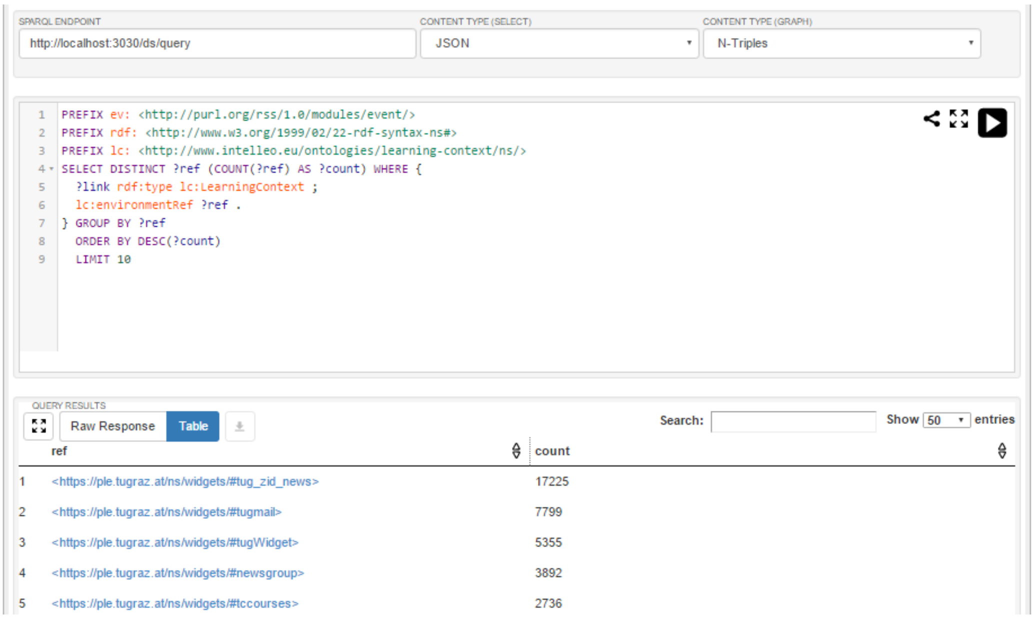
After starting the Fuseki server various services become available. List of the available services is listed in the table bellow.

Name	Link
File Upload:	http://localhost:3030/ds/upload
Graph Store Protocol:	http://localhost:3030/ds/data
Graph Store Protocol (Read):	http://localhost:3030/ds/get
HTTP Quads:	http://localhost:3030/ds/
SPARQL Query:	http://localhost:3030/ds/query
SPARQL Query:	http://localhost:3030/ds/sparql
SPARQL Update:	http://localhost:3030/ds/update

Table 4.1: Available services on Fuseki server

The Fuseki user interface enables running SPARQL queries via SPARQL Endpoint on the loaded dataset, the PLE Triple Store, even before implementing the dashboard client application. This enables testing and experimenting with the queries and results. The figure 4.4 illustrates the page from the Fuseki server user interface where simple query was executed on the PLE

Triple Store. At the bottom of the page the results are visible in the table view.



SPARQL ENDPOINT: http://localhost:3030/ds/query

CONTENT TYPE (SELECT): JSON

CONTENT TYPE (GRAPH): N-Triples

```
PREFIX ev: <http://purl.org/rss/1.0/modules/event/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX lc: <http://www.intelleo.eu/ontologies/learning-context/ns/>
SELECT DISTINCT ?ref (COUNT(?ref) AS ?count) WHERE {
  ?link rdf:type lc:LearningContext ;
  lc:environmentRef ?ref .
} GROUP BY ?ref
  ORDER BY DESC(?count)
  LIMIT 10
```

QUERY RESULTS

	ref	count
1	<https://ple.tugraz.at/ns/widgets/#tug_zid_news>	17225
2	<https://ple.tugraz.at/ns/widgets/#tugmail>	7799
3	<https://ple.tugraz.at/ns/widgets/#tugWidget>	5355
4	<https://ple.tugraz.at/ns/widgets/#newsgroup>	3892
5	<https://ple.tugraz.at/ns/widgets/#tccourses>	2736

Figure 4.4: Fuseki SPARQL Endpoint UI

After describing the configuration, starting and testing of the Fuseki server, the next section describes the client side implementation of the analytical dashboard.

4.5 The Analytical Dashboard Client

This section describes the client side implementation of the analytical dashboard. In contrary to the server where finding an adequate API was the main challenge, here implementing the client side application is required. A few general requirements for the client application are identified. These are:

- Capture user input
- Create and send requests to the server
- Receive and process the results from the server
- Visualize processed results

Next section presents the technologies used for implementing the client side application.

4.5.1 Used Technologies and Libraries

The client application is web based and therefore various web technologies and libraries for structuring, layout and communication with servers are used. Other than that finding the right library to visualize the results was important.

The following technologies and libraries are used:

- Hypertext Markup Language (HTML)
- Cascading Style Sheets (CSS)
- JavaScript
- jQuery
- Asyncronous JavaScript And XML (Ajax)
- D3.js

Three technologies, HTML, CSS and JavaScript are well known core technologies of the World Wide Web. HTML is the standard markup language for web pages and web applications. The presentation of HTML is described using a style sheet language called CSS. JavaScript is "a high-level, dynamic, untyped, and interpreted programming language" (Flanagan, 2011).

Other used technologies and libraries are described in more details in the following subsections.

jQuery

jQuery is a free, open-source JavaScript library designed to simplify client-side scripting of HTML. It provides an API which simplifies document navigation, selection of DOM[17] elements, creation of animations, handling of events and developing Ajax applications. jQuery library is small and all features mentioned above are contained in a single file which can be easily included.[18] It is supported by all relevant browsers. Initially it was released in August 2006. Currently the last stable version is 1.12.4, released in May 2016. In this work the version 1.11.3 is used. This version was compatible with visualization library d3.js, which is described later on. One of the huge advantages of jQuery is that it includes Ajax support and simplifies using it in applications. Ajax is described in next section.

Ajax

Ajax is short for Asynchronous JavaScript and XML. It is a set of techniques which enable creating asynchronous Web applications. The Data may be exchanged between the client and the server asynchronously. The presentation layer is decoupled from the data interchange layer which enables changing the content of the page dynamically without reloading the entire page. (Ullman and Dykes, 2007)

D3

D3, or d3.js[19], stands for Data-Driven Documents and is a JavaScript library for creating data visualizations (Murray, 2013). The original author of this

[17] Document Object Model

[18] https://jquery.com/ (last visited December 2016)

[19] https://d3js.org/ (last visited January 2017)

library is Mike Bostock[20] and the project is entirely open source and is freely available at https://github.com/mbostock/d3/ (last visited January 2017). It is released under BSD license and as such can be used and modified for commercial and noncommercial purposes free of cost.

It facilitates the DOM creation and manipulation and allows binding arbitrary data to it. This is done in four steps. These are:

- loading
- binding
- transforming
- transitioning

In the first step the data is loaded into the browsers memory. Then comes the binding step, where the data is linked to the existing DOM elements, or if necessary, new elements are created. Next, the transformation step, is the most important one. Here are the elements mapped to their visual properties. The structure for applying the transformation is provided by D3.js, but mapping rules are adjustable. In the last step the transitioning occurs. Based on the user input the elements transition between states (Murray, 2013).

[20] https://bost.ocks.org/mike/ (last visited January 2017)

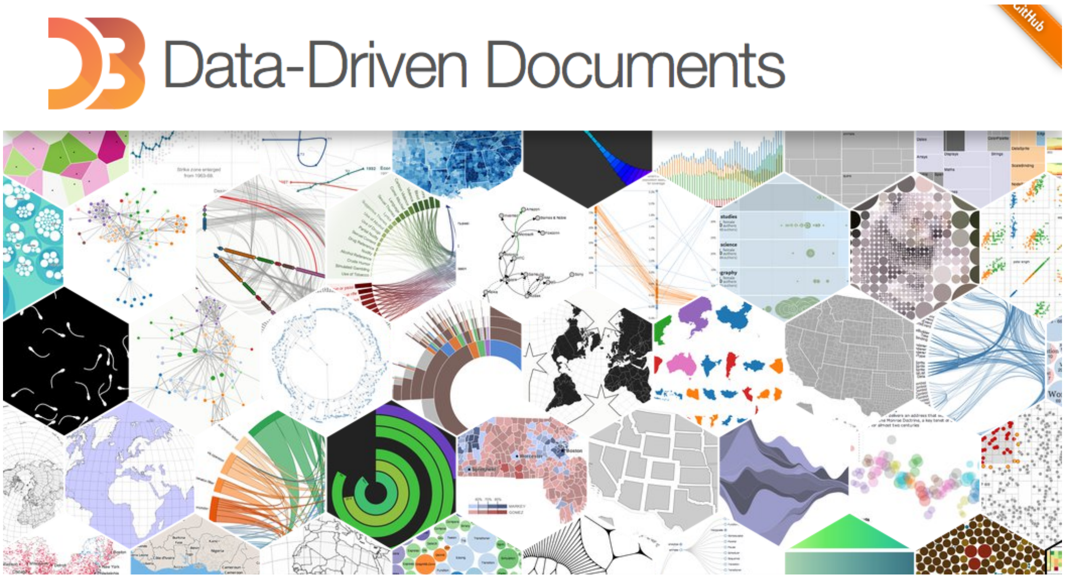


Figure 4.5: D3.js chart examples (https://d3js.org/ (last January 2016))

There are many reasons for choosing D3.js as a visualization library in our work. It is extremely fast, supports large datasets and most importantly supports dynamic behaviors for interaction and animation[21].

Figure 4.5 is a screen-shot from the d3.js website which gives an overview of small set of charts created using the d3.js library.

Additionally to d3.js another visualization library is used. It is called nvd3.js and enables building re-usable charts based on d3.js by keeping all power that d3.js provides[22]. Since using this library simplifies creating charts it is used along with d3.js.

[21] https://d3js.org/ (last visited December 2016)

[22] http://nvd3.org/ (last visited January 2017)

4.5.2 The Implementation

In the previous section the technologies and libraries, used for implementing the analytical dashboard client application, are described. On the one side it was important to find adequate technologies which facilitate communication with the server, and on the other side it was needed to create charts with a huge focus on user interaction.

JavaScript is the main language used. Together with the jQuery library which facilitates implementation it is used for client side scripting of HTML. jQuery with Ajax facilitates communication with the Fuseki server. Using Ajax HTTP requests SPARQL queries are created and sent to the server. During the creation of the request, JSON format is specified as the format for the query response. Since data exchange occurs asynchronously the client is not blocked. It means that the entire application can be started and the user can use the application while the data exchange occurs in the background. The response from the server automatically triggers visualizing received data.

The implementation of the application is based on the Model-View-Controller pattern. The illustration of this pattern in visible in the figure 4.6.

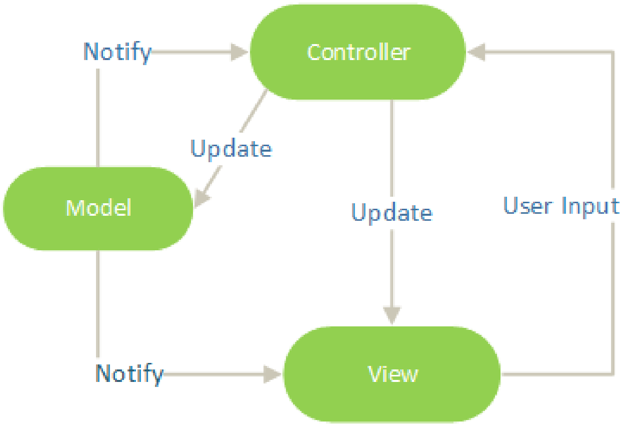


Figure 4.6: Model-View-Controller pattern)

It consists of three main components. These are:

- The Model
- The Controller
- The View

The Model is actually the domain layer which holds the data of the application. It parses and stores the data received from the server and notifies the view when the data changes.

The Controller is the main bond between the view and the model. It processes and responds to the events and invokes changes between them. The Controller is also responsible for the asynchronous data retrieval from the server. Once the data is retrieved the controller triggers the data update in the model.

The View is responsible for user interface and user interaction. It captures the user input and sends it to the controller. The user interface consists mostly of the charts which represent the data from the model.

The complete software design of the dashboard client application is illustrated in the figure 4.7.

After detailed description of used technologies and tools and detailed look into the technical part of the implementation, the next chapter presents the results visualized by the analytical dashboard application.

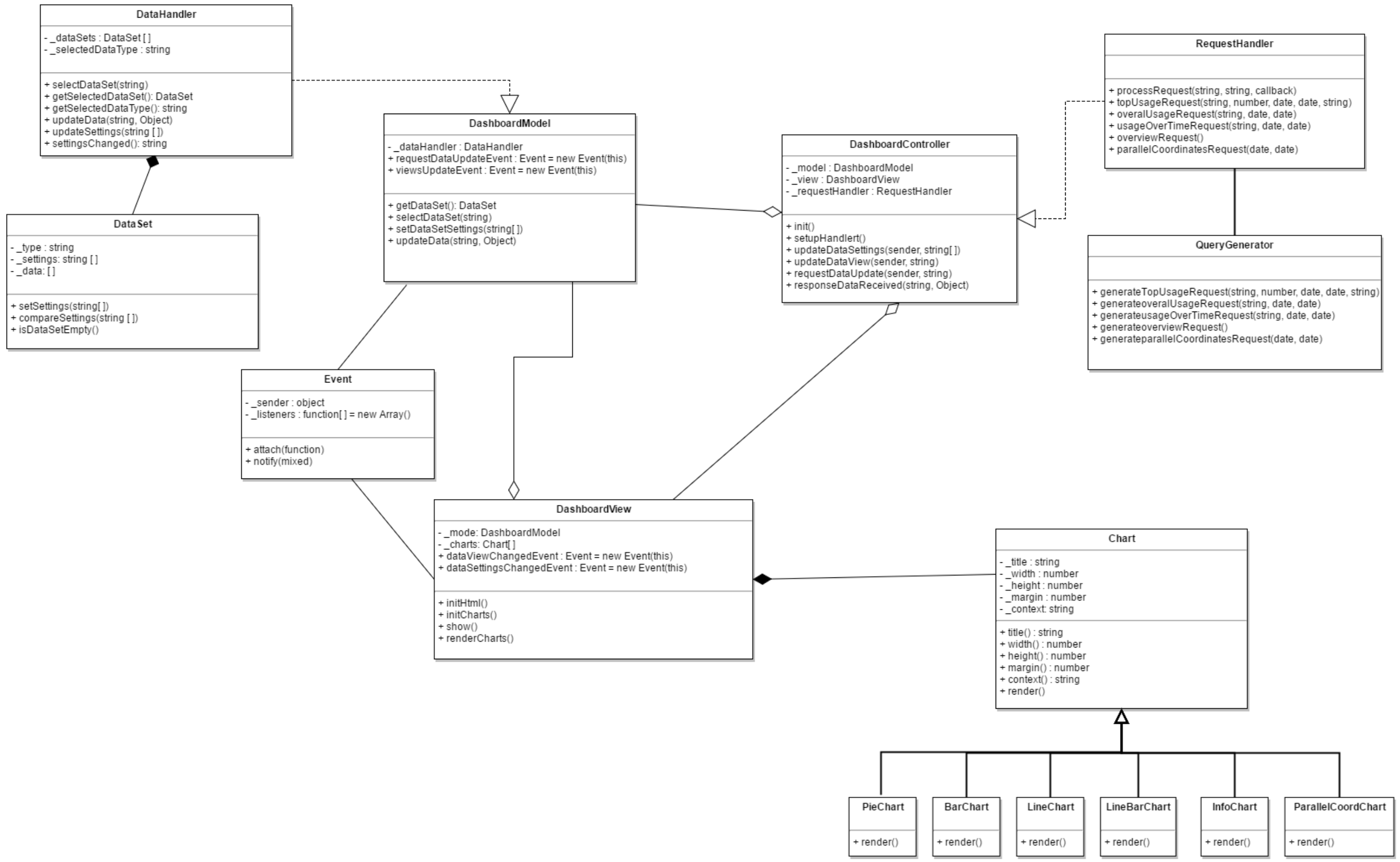

DataHandler
- _dataSets : DataSet [ ]
- _selectedDataType : string
+ selectDataSet(string)
+ getSelectedDataSet(): DataSet
+ getSelectedDataType(): string
+ updateData(string, Object)
+ updateSettings(string [ ])
+ settingsChanged(): string
DataSet
- _type : string
- _settings: string [ ]
- _data: [ ]
+ setSettings(string[ ])
+ compareSettings(string [ ])
+ isDataSetEmpty()
DashboardModel
- _dataHandler : DataHandler
+ requestDataUpdateEvent : Event = new Event(this)
+ viewsUpdateEvent : Event = new Event(this)
+ getDataSet(): DataSet
+ selectDataSet(string)
+ setDataSetSettings(string[ ])
+ updateData(string, Object)
DashboardController
- _model : DashboardModel
- _view : DashboardView
- _requestHandler : RequestHandler
+ init()
+ setupHandlert()
+ updateDataSettings(sender, string[ ])
+ updateDataView(sender, string)
+ requestDataUpdate(sender, string)
+ responseDataReceived(string, Object)
RequestHandler
+ processRequest(string, string, callback)
+ topUsageRequest(string, number, date, date, string)
+ overalUsageRequest(string, date, date)
+ usageOverTimeRequest(string, date, date)
+ overviewRequest()
+ parallelCoordinatesRequest(date, date)
QueryGenerator
+ generateTopUsageRequest(string, number, date, date, string)
+ generateoveralUsageRequest(string, date, date)
+ generateusageOverTimeRequest(string, date, date)
+ generateoverviewRequest()
+ generateparallelCoordinatesRequest(date, date)
Event
- _sender : object
- _listeners : function[ ] = new Array()
+ attach(function)
+ notify(mixed)
DashboardView
- _mode: DashboardModel
- _charts: Chart[ ]
+ dataViewChangedEvent : Event = new Event(this)
+ dataSettingsChangedEvent : Event = new Event(this)
+ initHtml()
+ initCharts()
+ show()
+ renderCharts()
Chart
- _title : string
- _width : number
- _height : number
- _margin : number
- _context: string
+ title() : string
+ width() : number
+ height() : number
+ margin() : number
+ context() : string
+ render()
PieChart
+ render()
BarChart
+ render()
LineChart
+ render()
LineBarChart
+ render()
InfoChart
+ render()
ParallelCoordChart
+ render()

Figure 4.7: UML Diagram of Dashboard Client Application based on MVC pattern

5 Results

The main focus of this chapter is to present and discuss the results of this work. The implemented application enables having deeper insight in the dataset captured between 2010 and 2015. The next section presents an overview of the features which enable the user to interact with the application and the results.

5.1 User Interaction

As described in chapter 2.3, user interaction is a must in each visual analytics approach. Therefore while implementing the application this was considered and the possibility for the user to interact with the presented results is provided. In this section these features are described.

User is enabled to have influence on the data and the visualized results. The user is able to specify:

- The date range
- The number of items displayed
- The overall period
- The view perspective

Since the current dataset is from five years period, choosing the start and end date of the analysis was the first logical feature to provide. It enables the user to specify the day, the month or the year he/she is interested in and only the data from that period is visualized. In order to keep the charts clear and visible the user has also the ability to influence the number of displayed items. Furthermore he/she is able to select the overall period. There are three options to choose, the day, the month and the year. This means that the user can see on which day/month/year the system is used the most or at least, based on his interest.

These three already mentioned user interaction methods can be set independent from other, the view perspective.

The view perspective is divided into five groups. These are:

- The Overview
- The Learner view
- The Activity view
- The Widget view and
- The Reflection view

These views (except the overview view) are decoupled from the first three interaction methods. This means, each of the methods can be applied to each of the views. So it is possible to set a date range for the Widget view, than only the data about widgets in the specified data range is visualized. Each of these views contains also reflection to time of its occurrence. So it is possible to look into detail at each of the views with the reflection to the time period. For example one can look into the overall widget usage over time and see at which days it is used the most.

In order to combine all three views together with the time of occurrence another view is added. This view is called the Reflection View. In this view the full reflection between the time, the learner, the widget and the activity is visualized. This fully represents the learning situation.

Besides features mentioned above, charts are also implemented in such manner that the user can interact with them by zooming in or out or by selecting which items to visualize.

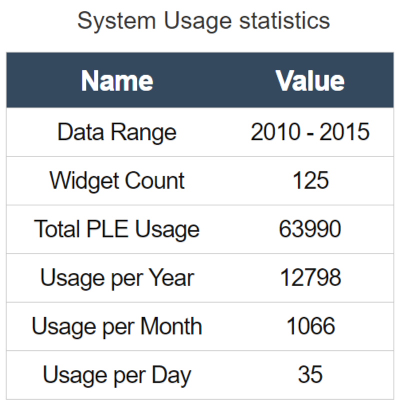
Next section presents results from each of the views.

5.2 The Results of Analysis

Although the application enables looking into a specific time period the results presented here are from the entire dataset. This is the first time that the entire dataset is fully analyzed and that is the main reason for presenting the results as such. The dataset contains data captured between October 2010 and September 2015. The results are presented and discussed in the following subsections.

5.2.1 The Overview page

The Overview page is the main page in the analytical dashboard. It provides some statistical information about the analyzed dataset as well as some general information and charts which enable quick overview of the dataset. Since the Widget is the most important item in our dataset and it is the central point of the PLE, most information presented here is based on it. The figure 5.1 shows some general information about the PLE usage. Looking at this first figure, some new information pops up, such as Widget count or total PLE usage. PLE has 125 widgets and they have been used almost 64.000 times in the last five years. The overall usage is also interesting. Each year PLE is used approximately 13.000 times.



System Usage statistics

Name	Value
Data Range	2010 - 2015
Widget Count	125
Total PLE Usage	63990
Usage per Year	12798
Usage per Month	1066
Usage per Day	35

Figure 5.1: The PLE Statistics

Another important chart from the Overview page is the bar chart which shows the widget usage over time. This chart is depicted in the figure 5.2. The main reason why this chart is placed on the main page, is to provide quick insight in overall usage of the entire dataset. Based on this chart the user shall get an idea where to focus and analyze further. Clearly the widget usage has been decreasing in the last three years. It seams that the PLE reached its peak at 2012. Further analysis shall provide more detailed information about PLE usage.

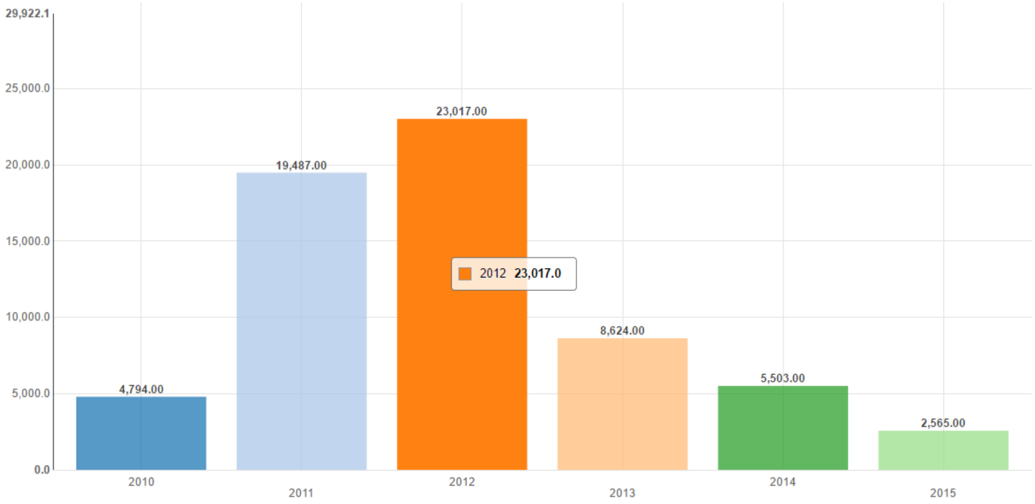


Figure 5.2: Widget Usage Over Years

5.2.2 The Learner View

In this section the results concerning the learners which used the PLE at Graz University of Technology are revealed. In order to protect the privacy of each learner, their names or any kind of personal information were not stored.

The figures 5.3 and 5.4 show the learner activeness over the time in form of a pie chart. Same information is shown in the figures 5.5 and 5.6 in form of a bar chart. Providing various charts for visualizing the same information is supported to increase the usability of the application. Each user has the possibility to use the preferred chart type. While analyzing the chart the users have the option of getting additional information about the displayed item in form of a tooltip as shown in the figure 5.3.

The pie and bar chart in figures 5.4 and 5.6 show the learner activeness over the entire week. The largest number of active learners is on Thursday where more than 12.000 learner were active. Learners were active almost equally on Friday and Saturday. Usage drops from Monday to Wednesday. The chart shows that compared to Thursday less than half usages occurred on Tuesday. Surprisingly the number of active users over the weekend, on Saturday and Sunday, remains stable. So it is clear that the learners are more active in the second half of the week (from Thursday to Sunday) than in the first half (from Monday to Wednesday).

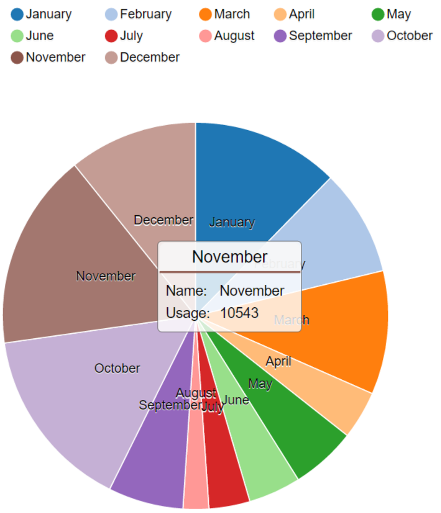


Figure 5.3: Active users over Months

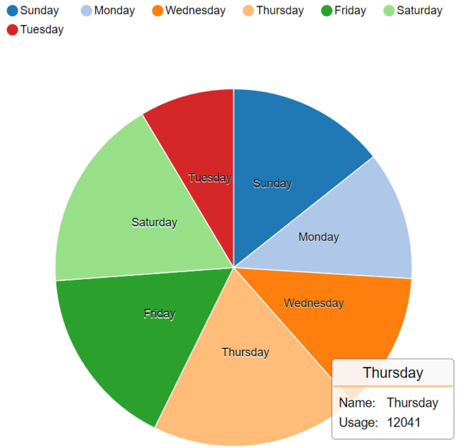


Figure 5.4: Active users over Days in a Week

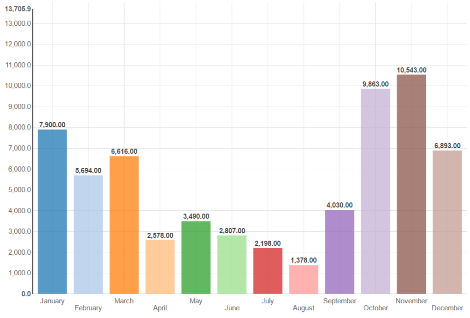


Figure 5.5: Active users over Months

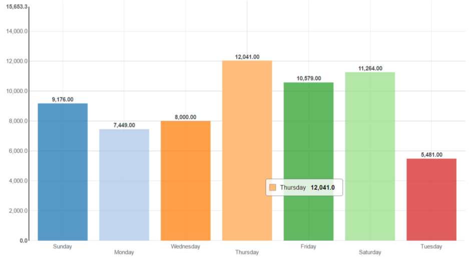


Figure 5.6: Active users over Days in a Week

The charts from figures 5.3 and 5.5 are about the learner activeness on a monthly basis. The chart clearly highlights that the learners are more active in the fall semester than in the spring semester. In fact, more than 50 percent of all usages occurred in the fall semester. The learners were mostly active at the beginning of the fall semester, the October and November, where they were engaged in more than 20.000 learning situations. Also significant increase of active users at the beginning of the spring semester in March compared to February (the winter holiday) is noticeable. But this seems to be temporary and the usage drops again for another 50 percent and does not change significantly until the beginning of the fall semester. During July and August the usage sinks bellow 2.000 usages, but that was expected since that is the time of the summer holidays. Generally the learners are active over the entire year but with huge variations between semesters.

5.2.3 The Activity View

This section presents the results concerning the activities in which the learners used the widgets. Based on the widget type, multiple activities can occur in a single widget, like for example searching, viewing and reading. The pie chart from the figure 5.7 shows the top 10 activities. Each slice of the pie chart represents a single activity. The learners used the widgets mostly for Reading. In fact, this is the case in more than 55.000 learning situations. The second largest activity is the Authoring activity with almost 19.000 occurrences. The third activity is the Search activity. This chart indicates that the learners used the widgets mostly for reading, authoring and searching.

The chart from the figure 5.8 shows the activity usage over time. X axis represents the time period, from 2010 to 2015. Y axis represents the activity count over time. Each activity is labeled and colored. Toggling among visible activities is possible. Therefore using this chart enables finding out which activities are used the most, or the least, and at which time. It also enables comparing various activities over time. This chart shows the usage of given features to compare the Authoring (blue line) and Game (green line) activity. Based on this it is easy to conclude that the learners used the widgets much more for Authoring than for Gaming over the entire period.

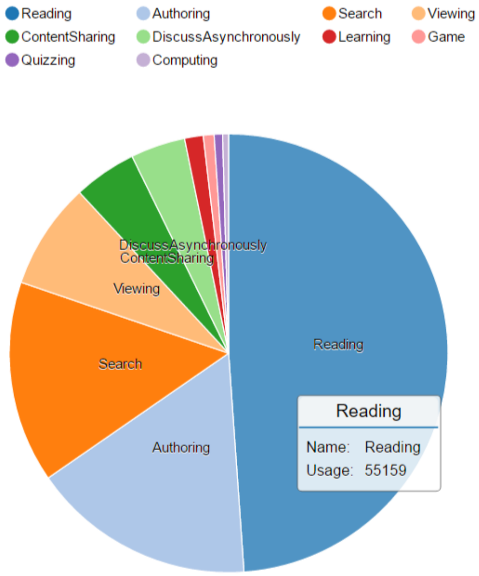


Figure 5.7: Top 10 Activities

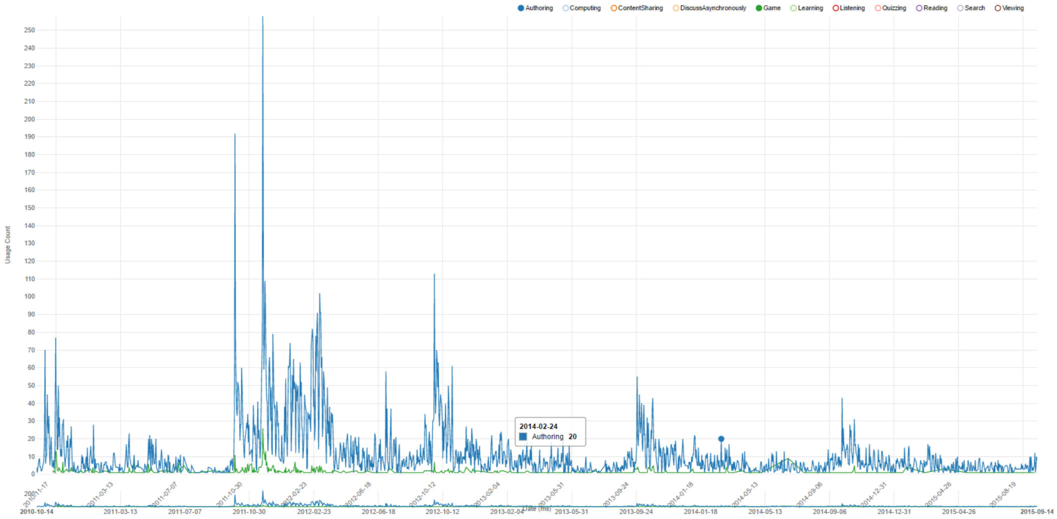


Figure 5.8: Activity usage over time

5.2.4 The Widget View

After looking into the results of learners and activities it is time to present the widget results. The results of top widgets are presented in form of a pie chart in the figure 5.9.

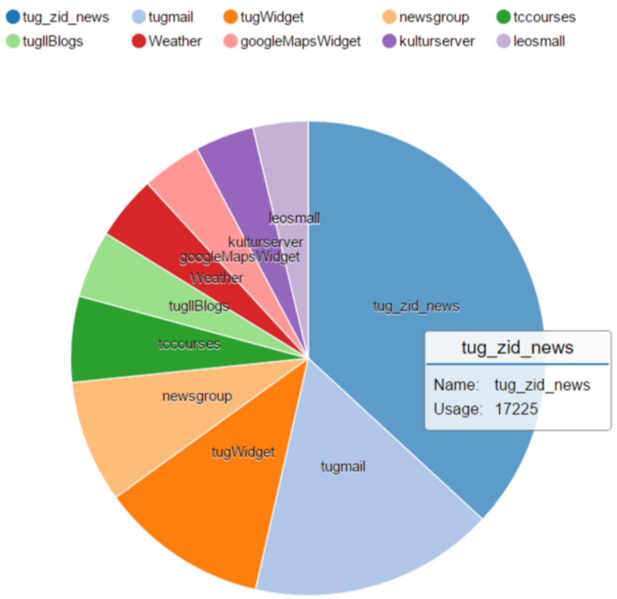


Figure 5.9: Top 10 Widgets

The pie chart shows the top 10 widgets. The widget called "tug zid news" has the largest number of usages and is used more than 17.000 times. That is a surprisingly high number for a single widget. It means that 27 percent of all PLE usages occurred in this widget. Second most used widget is "tugmail" used almost 8.000 times. It is followed by "tugWidget" used more than 5.000 and "newsgroup" widget used almost 4.000 times. These four widgets were used twice as much as all other 121 widgets in the PLE. Other widgets like "Weather", "googleMapsWidget", "kulturserver" are used much less compared to the top four. The learners mostly use widgets which integrate university services like e-mail, newsgroup etc. Other widgets which do not include university services are much less popular.

The following chart from the figure 5.10 represents the information regarding the widget usage over years and the total widget count in each year. This chart is similar to chart from figure 5.2 with additional information about the number of widgets. The bar chart (left y axis) represents the widget usage over the past five years and the line chart (right y axis) represents the widget count in each of the years. Looking at the line chart it is clear that the number of widgets has been increasing over the entire period. In the first year there were 20 widgets in a PLE. Already in the second year the number increased to 52. The growth continues also in the third year resulting in a remarkable 90 widgets in the year 2012. Since then more than 30 additional widgets were added to PLE resulting in a total 125 widgets. The bar chart shows widget usage over the entire period. In the first year the widgets were used almost 5.000 times. This number increased to astonishing 23.000 in the third year, 2012. Due to further increase of widgets it was expected the number of usages to increase, but surprisingly this was not the case. Already in next year the number of usages decreased more than half and resulted in less than 9.000 usages in that year. The usage of widgets continued to decrease in the following years and resulted in alarming 3000 usages in 2015. This gives an average daily usage count of 11 widgets.

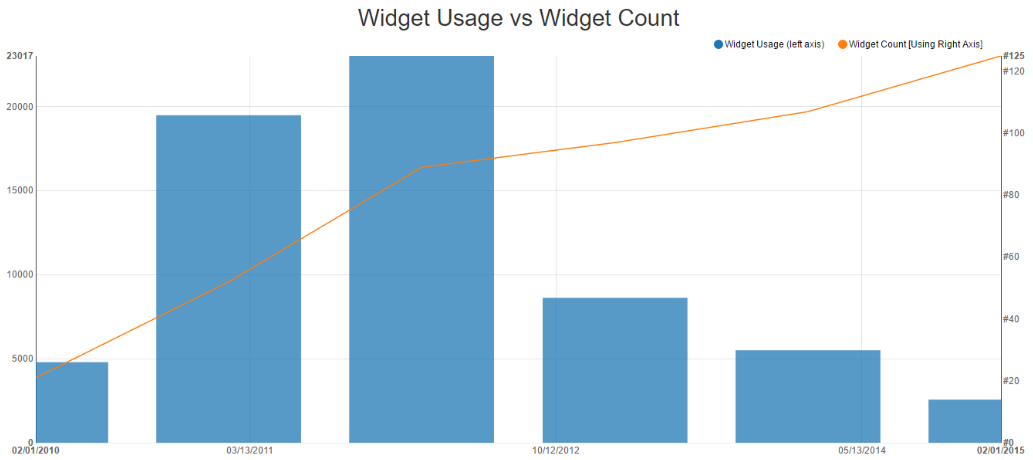


Figure 5.10: The Widget count vs Widget Usage over Time

The line chart from the figure 5.11 presents the results of widget usage on daily basis. To have a clearer view on the number of widgets used daily an additional filter was added which shows only widgets which have been used more than 10 times at least on one day over the entire period. This narrows down the widget count to only 30.

Additional feature provided by this chart is zooming to a specific time period. This allows showing the results where most variations are noticed and enable looking into the single semester like shown in figure 5.12. This line chart shows how many times was each of the widgets used on each day. The most used widget over the entire period is "tug zid news" which was used exactly 356 times on 24th of November in 2011.

Doubling the filter to 20 reduces the number of matches by half. This leads to the conclusion that only small amount of widgets are actively used.

Following section shows the results from the reflection view.

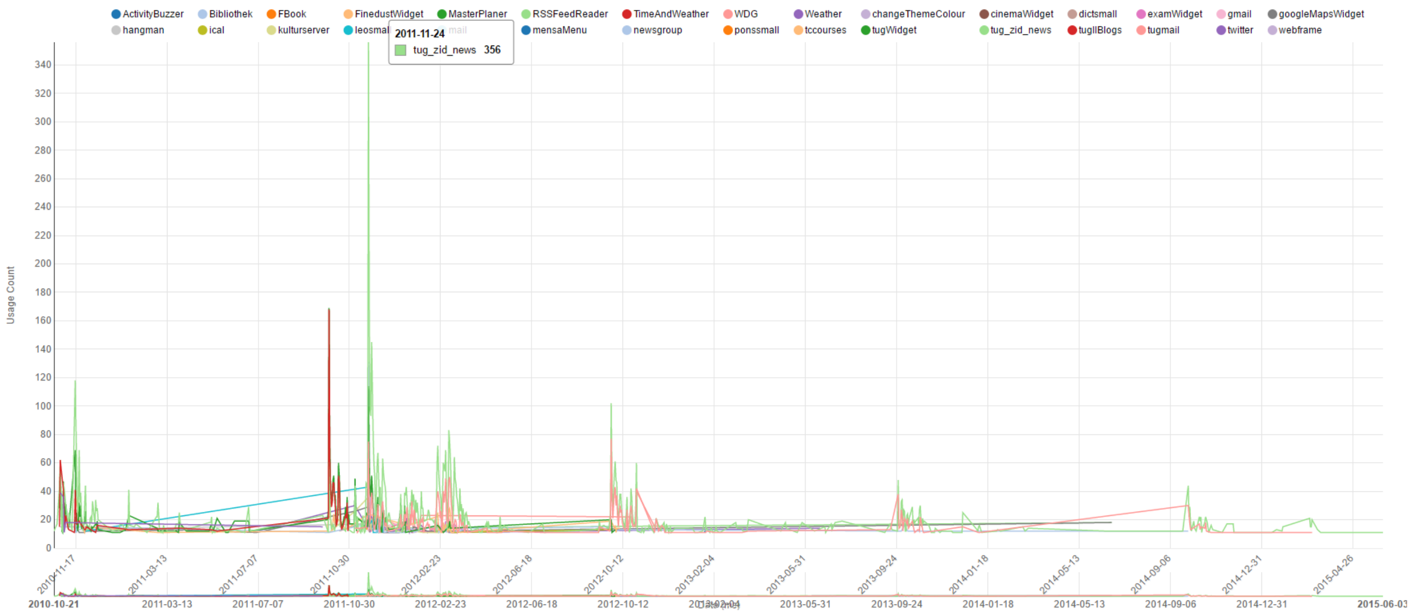


Figure 5.11: Widgets Over Time with at least 10 Usages

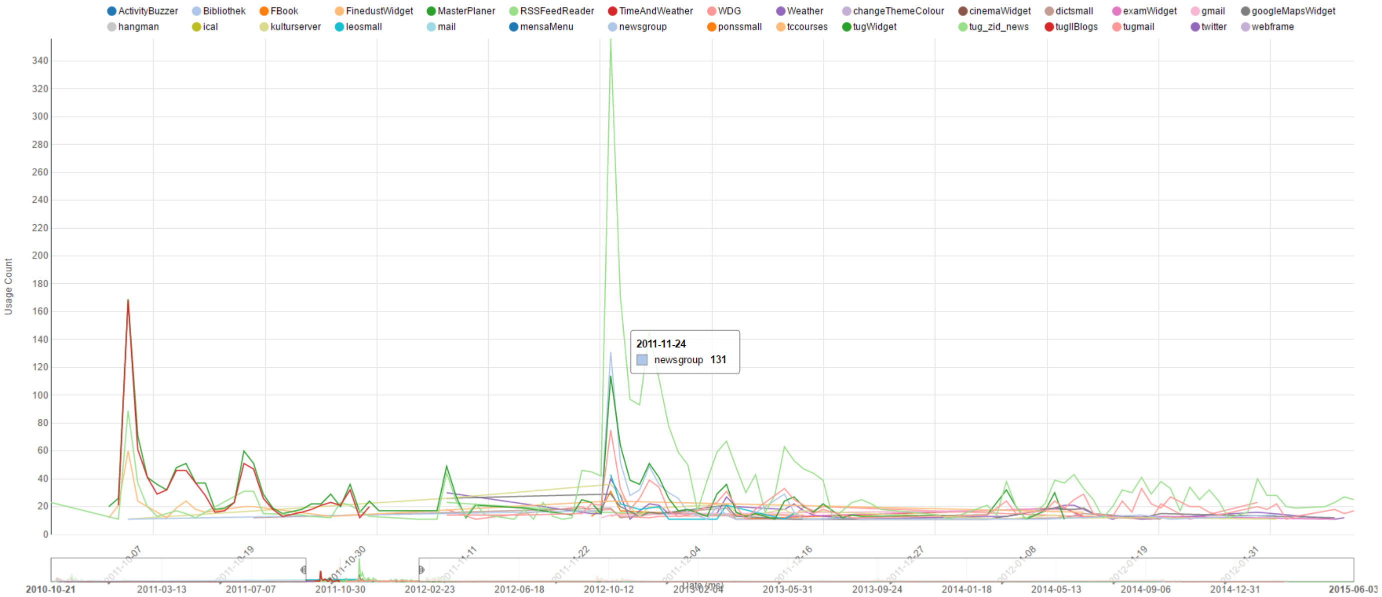


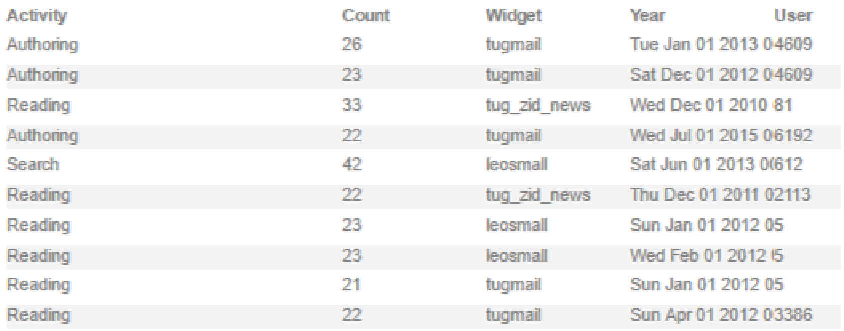
Figure 5.12: Widgets with at least 10 Usages in the fall semester 2011

5.2.5 The Reflection view

In this section the results from all three data dimensions including the time of their occurrence and the number of occurrences are presented. Visualizing all contained information together provides a reflection among the dimensions. In order to provide user interaction and to visualize all dimensions the best way, the parallel coordinates were used.

Using this type of chart PLE usage can be quickly compared by filtering along any of the provided dimensions. This chart helps identifying the patterns between the data dimensions. Figure 5.14 shows an example where the most used widgets, together with the count, are filtered. This allows focusing on specific widgets, in this case only on those which are used more than 10 times. Looking at the time of usage, it is clear that the usage drops over the years. The PLE was used more in 2011 and 2012 than it was used in 2013 and later. The count dimension indicates that the most of the widgets have less than 50 usages. Only a few have been used more than 50 times.

In order to get detailed information about the selected dimensions additionally the table with information about the relation among all dimensions is provided. Example of such table is illustrated in the figure 5.13. This gives detailed information about the learner, time of learning activity, widget, count of usages in that period, and the activity at which the usage occurred.



Activity	Count	Widget	Year	User
Authoring	26	tugmail	Tue Jan 01 2013	04609
Authoring	23	tugmail	Sat Dec 01 2012	04609
Reading	33	tug_zid_news	Wed Dec 01 2010	81
Authoring	22	tugmail	Wed Jul 01 2015	06192
Search	42	leosmall	Sat Jun 01 2013	0(612
Reading	22	tug_zid_news	Thu Dec 01 2011	02113
Reading	23	leosmall	Sun Jan 01 2012	05
Reading	23	leosmall	Wed Feb 01 2012	(5
Reading	21	tugmail	Sun Jan 01 2012	05
Reading	22	tugmail	Sun Apr 01 2012	03386

Figure 5.13: Table containing detailed information about all dimensions

Full parallel coordinates chart with all dimensions is shown in the figure 5.15.

This allows getting a full overview of the entire dataset. Looking at a year dimension, it is clear that learners were most active between 2011 and 2012, since most of the lines go towards the lower half of the chart. This also gives a pretty clear picture about the widget popularity. Widgets at the top are more popular than the widgets at the bottom. Also the popularity of activities is reflected in the last dimension. The count reflects the number of relations among the widget and the activity. This shows which widget is used in which activity and how often. The count of usages confirms previous findings. Only a few learners over the entire time period have used widgets more than 50 times in a single month. In fact most of the widgets were used only a few times in each month. This chart also shows that most of the PLE usages occurred in the same widgets, although the number of their usages is small. It is clear that there is a huge number of widgets which have been used only a few times over the entire period. The relation between the activity and the widget is also visible. The most used widgets are mainly used in same activities for reading, authoring and searching.

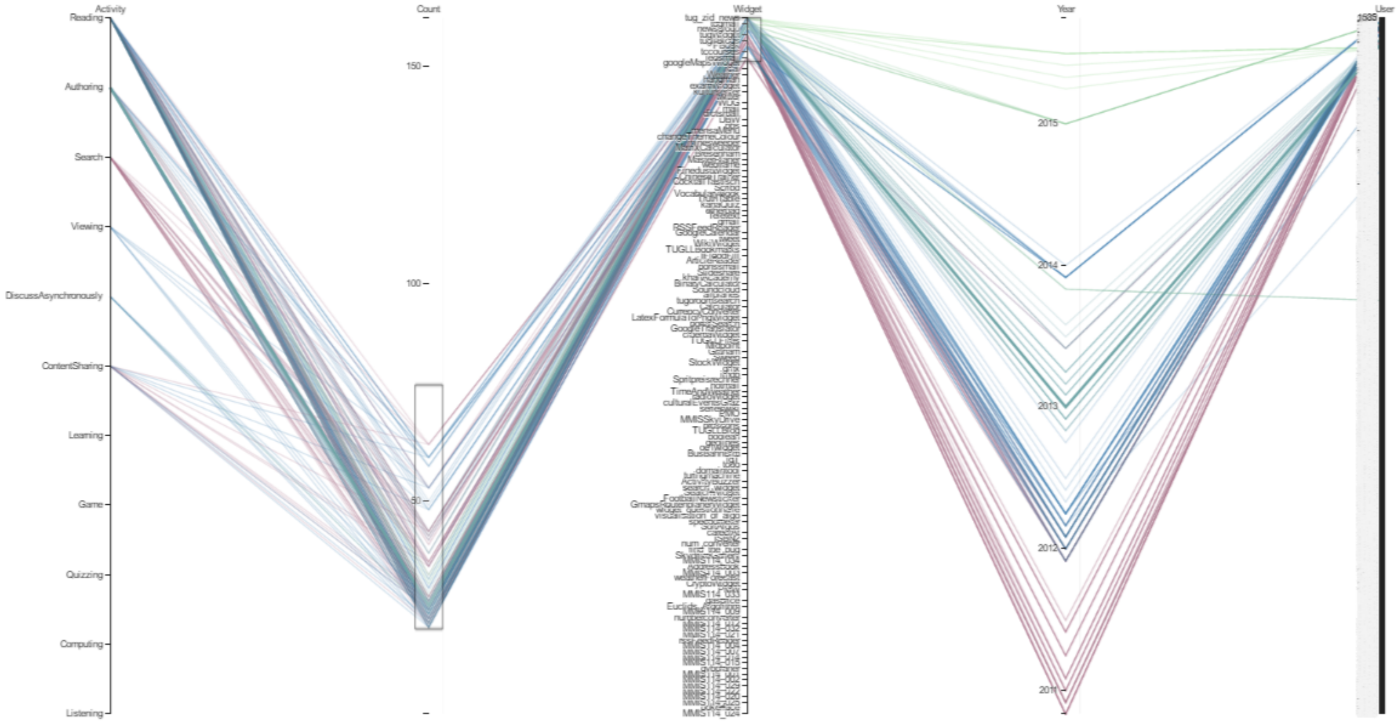


Figure 5.14: Parallel Coordinates filtering widget and count dimension

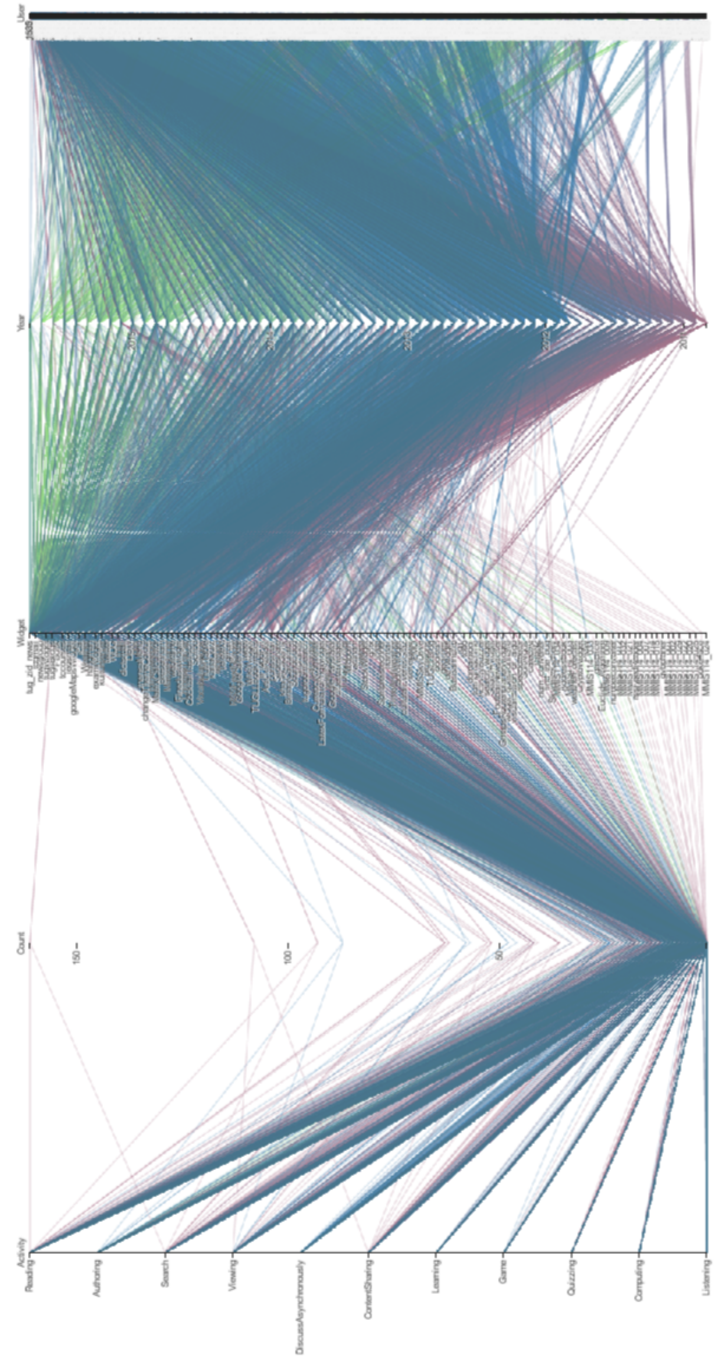


Figure 5.15: Parallel Coordinates Visualization of entire dataset

6 Conclusion and Future Work

This chapter summarizes the content of the thesis and answers the questions that lead to this work, see chapter 1. After presenting the related work and defining the concept, the architecture and the technical details of used approach were described. Additionally in the previous chapter a brief look into the results of analysis was presented.

The main focus of this thesis was to present a semantic web based approach for modeling the captured log data and to provide the analytical dashboard application for interactive data analysis and visualization.

The presented approach to reflect learner activity using semantic context brings several advantages and supports the improvement of the learning environment in general. Having structured and modeled data enables executing queries and such results can be visualized, as already presented in the previous chapters. But that is not the only use case. The exact same queries, which were used for the visualization of the top widgets, can also be used for instance as input for widget ranking in a widget store. This approach led to creation of a PLE triple store with the SPARQL endpoint, which made the PLE extensible and accessible using standardized interfaces. The data captured by the tracking module can now be accessed by other web tools and services.

Combining semantic web approach with the visual analytics led to the analytical dashboard which shows how combined approach can be used for data analysis and interactive visualization. This gave a clear insight in the PLE usage and will certainly contribute to understanding the students needs and trends.

After implementing the analytical dashboard, the initial questions (see chapter 1) that led to this thesis can be answered.

Can captured log data be used to measure the overall PLE usage and how?

- Captured log data contains enough information which enabled meaningful structuring and modeling of the captured dataset. This was achieved using the semantic based approach which led to creation of a PLE Triple Store, see section 4.2, which further brings manifold advantages.

How does the PLE compete against fast changing trends among students?

- Although the initial reports were very promising, the students trends changed very soon and a huge drop in the PLE usage is noticeable. According to the results presented in the section 5.2.4, figure 5.10, the number of students using PLE has been constantly sinking in the past three years. The turnover occurred in 2013 where the usage dropped for more than 60 percent. Average daily usage in the last year, 2015, was 11 usages. The fact that the students develop new widgets and that the total number of widgets increases in each year makes these numbers more alarming. Detailed analysis of the cause and drastic changes in the PLE are necessary in order to draw students attention and try to keep up with their needs.

When did the PLE reached the highest utilization degree?

- The answer to this question is related to the previous one, since keeping up with the students needs and trends enables increasing the utilization degree. Average number of students which studied at TU Graz and had the access to the PLE at TU Graz from 2010 to 2015 is higher than 12.000[1]. So if each student that has access to the PLE would use it only once a month, that would result in 144.000 usages in a year. This hypothesis gives the overall utilization degree for the entire period of 8 percent. The highest utilization degree was reached back in 2012 when it was 16 percent. The lowest was in 2015 when the utilization degree decreased to only 2 percent. Of course, since PLE is used for learning, the number of usages per student in each month

[1] `https://www.tugraz.at/en/tu-graz/university/tu-graz-at-a-glance/` (last visited January 2017)

> should be much higher. The PLE has a very high potential but as it currently looks, it is far from reaching it.

The results clearly indicate that the learners are interested in technologies for personalized learning to support them in courses and studies. The initial increase of learners using a PLE until 2012 clearly shows that. Although the PLE content constantly increased, that did not reflect positively on the learners. It is clear that the further focus of PLE shall be more on learners than on the content itself.

6.1 Future Work

The PLE triple store and SPARQL endpoint could be used for the widget recommendation in a PLE. The queries used here to query the top widgets could be integrated and used by the PLE itself to get the most used widgets and recommend them to other users. In that case the PLE would act as a client and would send queries to the server. Results would be presented to students in the PLE directly. Additionally the tracking module could be extended to provide user wise information of the PLE usage. PLE can provide these information about the users. Also the Learning context ontology already contains this option and can be easily adapted. This would enable the PLE store to narrow down the widget recommendation to a personal level. Widgets would be ranked higher and recommended based on the interests of each user.

Another idea for further work is the automated update of a PLE triple store. Currently, data captured by the tracking module is manually transformed and uploaded to the PLE triple store. Automated solution would enable tracking PLE usage and user behavior in real-time. This would be a great improvement which could help in detecting changes in system usage and user trends much earlier giving much more time to act.

Also the analytical dashboard client application could be extended to support mobile devices. Technologies used here allow such extensions and can provide mobile support.

Bibliography

Allen, Christopher (2004). *"Tracing the Evolution of Social Software"*. URL: http://www.lifewithalacrity.com/2004/10/tracing%5C_the%5C_evo.html (cit. on p. 40).

Anderson, T. (2006). *PLE's versus LMS: Are PLEs ready for Prime time?* http://terrya.edublogs.org/2006/01/09/ples-versus-lms-are-ples-ready-for-prime-time/. (Accessed on 01/07/2017) (cit. on pp. 40, 41).

Berners-Lee, T., R. Fielding, and L. Masinter (2005). *Uniform Resource Identifier (URI): Generic Syntax*. Tech. rep. DOI: 10.17487/rfc3986. URL: http://dx.doi.org/10.17487/RFC3986 (cit. on pp. 8–10).

Berners-Lee, Tim, James Hendler, and Ora Lassila (2001). "The semantic web." In: *Scientific American* 284.5, pp. 34–43 (cit. on p. 4).

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 0387310738 (cit. on p. 27).

Bizer, Christian, Tom Heath, and Tim Berners-Lee (2009). "Linked Data - The Story So Far." In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 5.3, pp. 1–22 (cit. on p. 7).

Brickley, Dan and Ramanathan Guha (2014). *RDF Schema 1.1*. W3C Recommendation. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/. W3C (cit. on pp. 15, 16).

Chatti, M. A. (2010). *LMS vs. PLE*. URL: http://mohamedaminechatti.blogspot.co.at/2010/03/lms-vs-ple.html (cit. on p. 41).

Downes, Stephen (2007). "Learning networks in practice." In: (cit. on p. 41).

Duerst, M. and M. Suignard (2005). *Internationalized Resource Identifiers (IRIs)*. Tech. rep. DOI: 10.17487/rfc3987. URL: http://dx.doi.org/10.17487/RFC3987 (cit. on p. 9).

Ebner, Martin, Nicolai Scerbakov, et al. (2010). "Teaching and Learning in Higher Education - An Integral Approach." In: *Proceedings of Society for Information Technology & Teacher Education International Conference 2010*. Ed. by David Gibson and Bernie Dodge. San Diego, CA, USA: AACE, pp. 428–436. URL: http://www.editlib.org/p/33375 (cit. on p. 42).

Ebner, Martin and Behnam Taraghi (2010). "Personal Learning Environment for Higher Education–A First Prototype." In: *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pp. 1158–1166 (cit. on p. 42).

Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth (1996). "Knowledge discovery and data mining : Towards a unifying framework." In: *Proceedings of the 2nd international conference on Knowledge Discovery and Data mining (KDD'96)*. AAAI Press, pp. 82–88 (cit. on p. 25).

Fayyad, Usama, Gregory Piatetsky-shapiro, and Padhraic Smyth (1996). "From Data Mining to Knowledge Discovery in Databases." In: *AI Magazine* 17, pp. 37–54 (cit. on pp. 24–26).

Flanagan, David (2011). *JavaScript : the definitive guide*. Cambridge: O'Reilly. ISBN: 978-0-596-80552-4. URL: http://opac.inria.fr/record=b1133669 (cit. on p. 64).

Keim, Daniel A. (2001). "Visual Exploration of Large Data Sets." In: *Commun. ACM* 44.8, pp. 38–44. ISSN: 0001-0782. DOI: 10.1145/381641.381656. URL: http://doi.acm.org/10.1145/381641.381656 (cit. on p. 30).

Keim, Daniel A., Joern Kohlhammer, et al., eds. (2010). *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics. URL: http://www.vismaster.eu/book/ (cit. on pp. 29–32, 34–37, 39).

Keim, Daniel A., Florian Mansmann, et al. (2008). "Visual Data Mining." In: ed. by Simeon J. Simoff, Michael H. Böhlen, and Arturas Mazeika. Berlin, Heidelberg: Springer-Verlag. Chap. Visual Analytics: Scope and Challenges, pp. 76–90. ISBN: 978-3-540-71079-0. DOI: 10.1007/978-3-540-71080-6_6. URL: http://dx.doi.org/10.1007/978-3-540-71080-6_6 (cit. on pp. 30, 33, 34, 37).

Krötzsch, Markus et al. (2012). *OWL 2 Web Ontology Language Primer (Second Edition)*. http://www.w3.org/TR/2012/REC-owl2-primer-20121211/. W3C (cit. on p. 17).

Lubensky, Ron (2006). "The present and future of Personal Learning Environments (PLE)." In: *Retrieved January* 4, p. 2010 (cit. on p. 40).

Lundblad, Patrik, Oskar Eurenius, and Tobias Heldring (2009). "Interactive Visualization of Weather and Ship Data." In: *13th International Conference on Information Visualisation, IV 2009, 15-17 July 2009, Barcelona, Spain*, pp. 379–386. DOI: 10.1109/IV.2009.38. URL: http://dx.doi.org/10.1109/IV.2009.38 (cit. on p. 38).

Martindale, Trey and Michael Dowdy (2010). "Personal learning environments." In: *Emerging technologies in distance education*, pp. 177–193 (cit. on pp. 40, 41).

Miller, Eric (1998). "An Introduction to the Resource Description Framework." In: *Bulletin of the American Society for Information Science and Technology* 25.1, pp. 15–19. ISSN: 1550-8366. DOI: 10.1002/bult.105. URL: http://dx.doi.org/10.1002/bult.105 (cit. on pp. 9–11).

Mott, Jonathan (2010). "Envisioning the post-LMS era: The open learning network." In: *Educause Quarterly* 33.1, pp. 1–9 (cit. on p. 41).

Murray, Scott (2013). *Interactive Data Visualization for the Web*. O'Reilly Media, Inc. ISBN: 1449339735, 9781449339739 (cit. on pp. 65, 66).

Olivier, Bill and Oleg Liber (2001). "Lifelong Learning: The Need for Portable Personal Learning Environments and Supporting Interoperability Standards." In: URL: http://wiki.cetis.ac.uk/images/6/67/Olivierandliber2001.doc (cit. on p. 40).

Piatetsky-Shapiro, G. (1991). "Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop." In: *AI Magazine* 11(5), pp. 68–70 (cit. on p. 24).

Prinsloo, Paul, Sharon Slade, and Fenella Galpin (2012). "Learning analytics: challenges, paradoxes and opportunities for mega open distance learning institutions." In: *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. LAK '12. Vancouver, British Columbia, Canada: ACM, pp. 130–133. ISBN: 978-1-4503-1111-3. DOI: 10.1145/2330601.2330635. URL: http://doi.acm.org/10.1145/2330601.2330635 (cit. on p. 47).

Renon, Flavia (2013). *Personal Learning Environments (PLEs): Ecologies for building student capability for lifelong learning* (cit. on p. 42).

Salkic, Senaid et al. (2015). "Linked Data Driven Visual Analytics for Tracking Learners in a PLE." In: *DeLFI 2015 - Die 13. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI), München, 1.-4. September 2015*, pp. 329–331. URL: http://subs.emis.de/LNI/Proceedings/Proceedings247/article27.html (cit. on p. 53).

Santos, Jose Luis et al. (2012). “Goal-oriented visualizations of activity tracking: a case study with engineering students.” In: *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*. LAK ’12. Vancouver, British Columbia, Canada: ACM, pp. 143–152. ISBN: 978-1-4503-1111-3. DOI: 10.1145/2330601.2330639. URL: http://doi.acm.org/10.1145/2330601.2330639 (cit. on p. 47).

Schaffert, Marco Kalz Sandra (2009). “Persönliche Lernumgebungen: Grundlagen, Möglichkeiten und Herausforderungen eines neuen Konzepts.” In: ed. by A. Hohenstein K. Wilbers. 5.16. Köln, Germany: Deutscher Wirtschaftsdienst (Wolters Kluwer Deutschland), pp. 1–24. URL: http://hdl.handle.net/1820/1573 (cit. on p. 41).

Schreiber, Guus and Yves Raimond (2014). *RDF 1.1 Primer*. W3C Note. http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/. W3C (cit. on pp. 9–12).

Shneiderman, Ben (1996). “The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.” In: *Proceedings of the 1996 IEEE Symposium on Visual Languages*. VL ’96. Washington, DC, USA: IEEE Computer Society, pp. 336–. ISBN: 0-8186-7508-X. URL: http://dl.acm.org/citation.cfm?id=832277.834354 (cit. on p. 37).

Softic, Selver et al. (2013). “Monitoring learning activities in PLE using semantic modelling of learner behaviour.” In: *Human Factors in Computing and Informatics*. Springer, pp. 74–90 (cit. on pp. 1, 45, 48).

Taraghi, Behnam, Martin Ebner, and Kröll Clemens (2012). “Personal Learning Environment – Generation 2.0.” In: *World Conference on Educational Multimedia, Hypermedia and Telecommunications*. AACE, pp. 1828–1835 (cit. on p. 42).

Taraghi, Behnam, Martin Ebner, and Sandra Schaffert (2009). “Personal learning environments for higher education: A mashup based widget concept.” In: *Proceedings of the Second International Workshop on Mashup Personal Learning Environments (MUPPLE09), Nice, France*, pp. 1613–0073 (cit. on p. 42).

Taraghi, Behnam, Martin Ebner, Gerald Till, et al. (2010). “Personal Learning Environment-a Conceptual Study.” In: *iJET, International journal of emerging technologies in learning* 5.S1, pp. 25–30 (cit. on pp. 42, 43).

Taraghi, Behnam, Selver Softic, et al. (2013). “Learning Activities in Personal Learning Environment.” In: *EdMedia: World Conference on Educational Media and Technology*. Vol. 2013. 1, pp. 2466–2475 (cit. on pp. 1, 44, 45).

Taraghi, Behnam, Christian Stickel, and Martin Ebner (2011). “Survival of the Fittest – Utilization of Natural Selection Mechanisms for Improving PLE.” In: *Proceedings of the first Workshop on Exploring the Fitness and Evolvability of Personal Learning Environments*, pp. 4–9 (cit. on pp. 1, 47).

Thomas, James J. and Kristin A. Cook (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr. ISBN: 0769523234 (cit. on pp. 30, 31).

Tim, O'Reilly (2005). “What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software.” In: URL: http://oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (cit. on p. 40).

Tukey, John W. (1977). *Exploratory Data Analysis*. Addison-Wesley (cit. on pp. 30, 34).

Ullman, Chris and Lucinda Dykes (2007). *Beginning Ajax*. Wrox. ISBN: 0470106751 (cit. on p. 65).

Van Harmelen, Mark (2006). “Personal Learning Environments.” In: *ICALT*. Vol. 6, pp. 815–816 (cit. on p. 41).

Ware, Colin (2004). *Information Visualization: Perception for Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1558608192 (cit. on p. 34).

Wilson, Scott et al. (2006). “Personal Learning Environments: Challenging the dominant design of educational systems.” In: *Proceedings of the first Joint International Workshop on Professional Learning, Competence Development and Knowledge Management - LOKMOL and L3NCD*. DOI: 10.1.1.107.3816. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.3816 (cit. on p. 41).

Wong, Pak Chung and Jim Thomas (2004). “Visual analytics.” In: *IEEE Computer Graphics and Applications* 24.5, pp. 20–21 (cit. on pp. 30, 31).