FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Bildung einer Kunstdatenbank – der Ort wo Web Technologien und Rechtsregulativen aufeinandertreffen

Wissenschaftliche Arbeit

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

(Mag.rer.soc.oec.)

im Rahmen des Studiums

Masterstudium Informatikmanagement

eingereicht von

Vladimir Živankić

Matrikelnummer 0509757

An der Fakultät für Informatik der Technischen Universität Wien

Betreuer: ao. Prof. Dr. Jürgen Dorn

Wien, 09.04.2014 _____ _____

(Unterschrift Verfasser) (Unterschrift Betreuer)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Creating an artwork database – a place where Web technologies and legal regulations meet

Diploma Thesis

For the acquisition of the academic degree

Magister rerum socialium oeconomicarumque (Mag.rer.soc.oec.)

Within the course of study

Management in Computer Science

Submitted by

Vladimir Živankić

Matriculation number 0509757

At the Faculty of Computer Sciences at the Vienna University of Technology

Mentor: ao. Prof. Dr. Jürgen Dorn

Vienna, 09.04.2014 _____ _____

(Student's signature) (Mentor's signature)

Erklärung zur Verfassung der Arbeit

Vladimir Živankić, Rechte Wienzeile 21/23, 1040 Wien

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen – , die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe."

Wien, 09.04.2014.

Unterschrift:

Table of Content

Abstract

The amount of artwork material available on the Web is constantly increasing. This work proposes a way for collecting artwork material whilst complying with the legal regulations, by using existing open source Web data mining technologies – the easy wGet. A manually written wrapper is developed to allow data extraction from the collected material. For this purpose a general method is developed depicting the process of material collection and data extraction. This method is implemented on a basis of one artwork source – the Olga's gallery. By doing this, an artwork database was created by using artwork material from the Web. The legal regulation relevant for this process are examined and summarized – the copyright law and the database directives in Austria, the EU and the USA. The potential artwork sources and Web data mining technologies are presented.

Immer mehr Kunstmaterial ist im Internet verfügbar. Die vorliegende Arbeit beschreibt einen Ansatz, der es ermöglicht mit Hilfe von Open Source Data Mining Technologien, easy wGet, Kunstmaterial zu sammeln. Dabei werden auch rechtliche Rahmenbedingungen berücksichtigt. Eine manuell entwickelte Wrapper Software erlaubt es Daten aus dem gesammelten Material zu extrahieren. Im Rahmen dieser Arbeit wird eine generellere Methode entwickelt, die den Prozess der Materialsammlung und der Datenextrahierung beschreibt. Die Methode wird auf Basis einer Ressource, der "Olga's gallery", implementiert. Daraus ergibt sich die Entwicklung einer Kunstdatenbank. Die wichtigen rechtlichen Rahmenbedingungen für diesen Prozess werden geprüft und zusammengefasst: das Urheberrecht und die Datenschutzrichtlinien in Österreich, der EU und in den USA. Die potentiellen Quellen und die Web Data Mining Technologien werden dabei dokumentiert.

1 Introduction

1.1 Initial situation with the presentation of a problem

The World Wide Web is a vast source of information in a form of interlinked collections of HTML formatted documents. The continuously growing amount of information has become an obstacle for information retrieval [3]. Web Scraping, also known as web data extraction or web harvesting is a technique of automatic web data extraction used to extract data from HTML formatted documents [1]. Web scraping can be seen as a computational analysis of a web page having the extraction of some data for its goal. A web crawler is a program used for methodical and automated browsing of the World Wide Web. Web crawlers are primarily used to create a copy of all visited pages that will be used for later processing [2]. Gathering data in this manner is associated with legal frames defining how the harvested data may or may not be used.

The goal of this work is to analyze how it would be possible to achieve the fetching and the extraction of desired data using existing open source technologies, and if needed developing some new modules or adjusting the existing ones, with the aim of building a local artwork database. In this work the easy wGet is used for data fetching and a manually written wrapper is developed for data extraction from the HTML. The goal of this work is not to gather as much data as possible, but to develop a general method for collecting and extraction of data, which represents the abstracted solution of this work. The general method is implemented on the basis of one source containing the artwork material – in this work the Olga's gallery is used as the artwork data source. The goal of this work is not to build a GUI for the collected artwork data.

The available artwork material is in a form of collections on the Web. It implies that using this material is regulated by law. Law regulations differ among different regions and in this work the law regulations in Austria, The European Union and The United States of America are examined. Two distinct legal areas are examined: the database law, that regulates collections made available publicly, and the copyright law, that regulates certain rights of the artwork, which are in the exclusive possession of the copyrights holder. This is important as

the data found in databases is regulated by the Copyright Law and databases as collections are regulated by the Database Law.

Further, it is shown how Creative Commons License is used for distribution of copyrighted works [4]. It means that the copyrighted material, which underlies the copyright law, is distributed with additional rights granted by the copyright owner. The licence itself is also machine readable, so the search engines and web robots can read it and understand it.

As the sources of artwork are in the centre of interest of this work, a survey on available sources was performed. It was important to investigate the content of the available material, the forms of presentation, the terms and conditions stated by the source owners, and also to investigate the qualitative nature of the available artwork: available epochs and descriptive information available for the artwork.

Europeana, Europe's multimedia online library, museum and archive is a project reflecting partially the idea of this work. It comprises books, maps, recordings, photographs, archival documents, paintings, and film material given by national galleries and cultural institutions from Europe's twenty-seven member states. It was to be explored which material is available and how the material is presented.

1.2 **Methodical Approach**

The World Wide Web is a vast media for information distribution. Due to immense and constantly growing amount of information available on the World Wide Web, information gathering, screening and delivering systems have become a necessity [5]. Significant effort has been made in the area of web crawlers, programs primarily used to make a copy of visited web pages for later processing. In recent years some new crawling techniques have been in the focus of the academic interest, like intelligent crawling, cooperative crawling, crawling based on semantic web etc. [2]. The methodical approach encompasses several steps, the first being a literature survey on web crawler's technology, data mining, Web archiving, national and international law regarding database law regulations and artwork copyright regulations. At the end of this phase a general method is created which represents the abstracted solution of this work: collecting artwork on the Web and extracting desired data and storing it locally.

The second phase involves the identification and analysis of web platforms containing artwork material, especially including museums and social media sites. The following step is to analyze some of the open-source web crawler technologies or similar tools already available and to identify the best one to collect the artwork from one source. The final step is to write a manually written wrapper for data extraction. These technologies are used to implement the general method on a basis of one source – the Olga's gallery. By following this strategy it is be possible to deliver desired prototype for creating an artwork database by harvesting specific information from the Web without law infringement.

1.3 **Expected Result**

A pilot project is to be developed by adapting the best suitable existing open-source tool for data collection, which will be used for artwork data collection from available sources in the World Wide Web. Desired data is to be extracted from the collected material. The extracted information should encompass the artwork title, the artist, the original link, the jpeg and possibly other useful information. The aim is to build a local artwork database without copyright infringement. Further goals are to explore the possible sources of artworks available in the World Wide Web and how they deal with copyrights of available artwork material, to provide an overview of national and international copyright directives regarding artwork, to explore the available epochs, and to define the difference between using artworks that are in public domain and artworks that are copyrighted and distributed under the Creative Commons License.

2 Theoretical part on national and international law regarding database law regulations and artwork copyright regulations

In this chapter only the relevant parts of the legal regulations are presented that are of interest for this work. The scope of detailed presentation of each of the following Directives is of the scope of this work.

2.1 Database directive in EU

Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases [39] is one the European Union Directives in the field of copyright law. It defines the treatment of databases under the copyright law. It defines also new sui genesis rights for the creators of databases.

This Directive defines the legal protection of databases in any form 1(1). The database is defined as a 'collection of independent works, data or other materials arranged in a systematic or methodical way and individually accessible by electronic or other means' 1(2). This doesn't include the software used to make the database accessible 1(3).

The article 3(1) states that compilations of content that reflect the intellectual creations and individuality of the author are to be protected by the copyright. This is a sufficient criterion. The article 3(2) states that the copyright protection of databases can't be applied to the content itself. It means that the content can be copyrighted and the restrictions following the copyright are valid regardless of the database copy right.

Article 4 defines the database authorship. The author can be the natural person or a group of natural persons or the legal person designated as the right holder, where the legislation of the Member States so permits 4(1). If a group of persons holds the copyright, and as such is recognized by the jurisdiction, the economic right are also collective 4(2).

Article 5 defines the restricted acts. The copyright owner of a database is in a possession of the right to perform alone or allow someone the following acts:

(a) Reproduction of the database;

(b) Different alterations;

(c) ‘any form of distribution to the public of the database or of copies thereof. The first sale in the Community of a copy of the database by the rightholder or with his consent shall exhaust the right to control resale of that copy within the Community’;

(d) Different public presentations;

(e) ‘any reproduction, distribution, communication, display or performance to the public of the results of the acts referred to in’ (b).

The Article 5 shall not permit the lawful use by lawful users 6(1). Member States may define any or all of the following limitations (Article 6.2):

(a) If non-digital compilations are reproduced for private purposes;

(b) If the content is used for educational purposes, scientific research and teaching. The usage has to be non commercial. The copyright owner has to be cited always, according the to the standards;

(c) If a juridical procedure and any other administrative procedure requires it or when the public security calls for it;

(d) If other exceptions to copyright are involved that usually are protected by the national law.

Duration of the copyright is defined in the Council Directive 93/98/EEC and will be presented within the section on the Copyright Directive.

Chapter 3 of the Directive defines a sui genesis form of protection for the investment (financial, human resources, effort and energy of any kind) of the content of databases, as a supplement to copyright. The Article 7(1) states that ‘Member States shall provide for a right for the maker of a database which shows that there has been qualitatively and/or quantitatively a substantial investment in either the obtaining, verification or presentation of the contents to prevent extraction and/or re-utilization of the whole or of a substantial part, evaluated qualitatively and/or quantitatively, of the contents of that database’. The lawful user of a database that was made available to the public may freely extract and/or re-utilize insubstantial parts of the database content (Art. 8(1)). However, a lawful user of a database which is made public in whatever manner shall not perform acts which differ from the normal

exploitation of the database or unreasonably prejudice the legitimate interests or the author (Art. 8(2)). Further, the lawful user 'may not cause prejudice to the holder of a copyright or related right in respect of the works or subject matter contained in the database' (Art 8(3)).

The Article 9 states that Member States may define any or all of the following limitations for lawful users of a database which is made available to the public in whatever manner. The lawful users may, take, use and re-use substantial parts of the content, without an explicit permission of the copyright holder:
(a)If the extraction is for private purposes and a database is not digital;
(b) If the content is used for educational purposes, scientific research and teaching. The usage has to be non commercial. The copyright owner has to be cited always, according the to the standards;
(c) If a juridical procedure and any other administrative procedure requires it or when the public security calls for it.

Sui genesis database right lasts for fifteen years from the date of completion of the making of the database. The Article 10 states that 'any substantial change, evaluated qualitatively or quantitatively, to the contents of a database, including any substantial change resulting from the accumulation of successive additions, deletions or alterations, which would result in the database being considered to be a substantial new investment, evaluated qualitatively or quantitatively, shall qualify the database resulting from that investment for its own term of protection'.

2.2 **Database directive - implementation in Austria**

Implementation in Austria is in force from January the 1st 1998, as the addition to the copyright law UrhG, under the name BGBI. I Nr. 25/1998. Article II (BGBI I Nr. 25/1998) defines the relation of the extension of the Austrian Copyright Law to the Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases. It states that with the BGBI. I 25/1998 the Austrian Copyright Law (UrhG) is adjusted to the Directive 96/9/EC [40].

2.3 **Database Law in the USA**

Databases or ‘compilations’ have been protected by the copyright since 1790, the year when the first USA Copyright was enacted. The 1976 Copyright Act extended the definition of ‘complications’ stating that ‘compilations’ require original selection, coordination or arrangement [41]. By this, databases are not protected by default by the Copyright Law in the USA. The protection applies only if there is a substantial creativity in selection, arrangement or coordination.

The restrictiveness of the copyright law has imposed on the database creators and owners a need to find the ways to protect themselves. They have developed three important procedures: ‘enhancing copyright protection by altering the structure or content of their databases to incorporate greater creativity, increasing reliance on contracts; and employing technological safeguards to prevent unauthorized access and use.’[41]

2.4 **Copyright law in the European Union**

In this section only the relevant parts of the EU Copyright Law will be presented.

Directive 2001/29/EC of the European Parliament defines the rights of the copyright holder:

- Reproduction right: ‘exclusive right to authorize or prohibit direct or indirect, temporary or permanent reproduction by any means and in any form’ (Art. 2)
- Right public presentation of the works and right of making available to the public other subject-matter (Art. 3)
- Distribution right (Art. 4) [43]

In the Article 1 of the Directive 93/98/EEC the duration of author’s right of a literary or artistic work within the meaning of Article 2 of the Berne Convention is valid throughout the artist’s life and 70 years after his death, regardless when the work is published [42].

Protection of photographs is defined in the Article 6 and states that the ‘photographs which are original in the sense that they are the author's own intellectual creation shall be protected in accordance with Article 1. No other criteria shall be applied to determine their eligibility for protection.’ [42]

2.5 Copyright law – implementation in Austria

Implementation in Austria is covered by the Urheberrechtsgesetznovelle 1996, BGBI. Nr. 151/1996. It represents national implementation of the Directive 2001/29/EC of the European Parliament.

2.6 Copyright law in the USA

The copyright owner is in a possession of the exclusive rights to perform and to authorize any of the following actions:
(1) To make copies of the artwork;
(2) To change and modify the work;
(3) To sell copies of the work, or to transfer the ownership, or to rent, lease or lend it;
(4) 'perform the copyrighted work publicly in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works;
(5) to display the work publicly in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work;
(6) to perform the copyrighted work publicly by means of a digital audio transmission, in the case of sound recordings' [46].

In general, copyright in work lasts for a period consisting of the life of the author and 70 years after the author's death [47].

Subject matter for the copyright are, among others, "pictorial, graphic, and sculptural works" that include two-dimensional and three-dimensional works of fine, graphic, and applied art, photographs, prints and art reproductions, maps, globes, charts, diagrams, models, and technical drawings, including architectural plans [48].

2.7 **Creative Commons Copyright Licences**

The Creative Commons copyright licenses and tools allow sharing of otherwise copyrighted work. They represent a balance inside the traditional "all rights reserved" setting that copyright law creates. These licences and tools give individual creators equally as large companies and institutions a simple, standardized way to grant copyright permissions to their creative work [49].

All Creative Common licences share many important features. Every licence allows creator to retain copyright and at the same time allows other to copy, distribute, and make use of creator's work, at least non-commercially. With this in place, the creators retain the credit for their work. On top of this baseline features, creators can choose to grant additional rights when deciding how to make their work usable for others [49].

The public copyright licences are made out of three layers: the legal code layer, the Commons Deed, and the machine readable layer. The legal code layer is written using highly specialized legal terminology and is not understandable for the most of the people. The Commons Deed is the simplified version, understandable to everyone. The machine readable version is recognized by software, from search engines to office software products [49].

Additional rights that creators can grant when deciding how to make their work usable for other are incorporated in the following licences:

1. Attribution CC BY: allows others to distribute, remix, tweak, and build upon the original work, even commercially, as long as they credit the creator for the original creation.
2. Attribution – NoDerivs CC BY-ND: allows others the redistribution, commercial and non-commercial, as long as it is passed along unchanged and in whole, with credit to the creator.
3. Attribution-NonCommercial-ShareAlike CC BY-NC-SA: allows others to remix, tweak, and build upon the original work non-commercially, as long as they credit the creator and license their new creations under the identical terms.

4. Attribution-ShareAlike CC BY-SA: allows others to remix, tweak, and build upon the original work even for commercial purposes, as long as they credit the creator and license their new creations under the identical terms.
5. Attribution-NonCommercial CC BY-NC: allows others to remix, tweak, and build upon the original work non-commercially, and although their new works must also acknowledge the creator of the original work and be non-commercial, they don't have to license their derivative works on the same terms.
6. Attribution-NonCommercial-NoDerivs CC BY-NC-ND: allows others to download the original works and share them with others as long as they credit the creator, but they can't change the works in any way or use them commercially [49].

3 State-of-the-art

3.1 Theoretical part on existing technology for information extraction

3.1.1 Web crawlers

The technology used for automatic download of web pages is known under the terms web crawlers, spiders or robots. Web crawler is software designed to visit a huge number of web pages having as a goal a collection of information. The collected information is examined during the crawl process. This happens in two possible scenarios: the information is first stored and then analyzed or the analysis of the information is done during the download process. The nature of the Web is dynamic, new pages are generated constantly and therefore the size of the Web increases rapidly. In the scenario where the Web is seen a passive set of web pages, and where the number of web pages remains constant over the time, the web crawler would stop the execution after all the pages are fetched. The nature of the Web determines the design of the web crawler. The web pages get deleted or modified, or moved to the other location and new web pages are published constantly, so the web crawler should deal with this. Web crawlers represent the basic underlying technology for many business applications. For example, there are many business intelligence applications based on the web crawler technology what are primarily used to gather information about organisation's rivals and potential business partners. Many applications use web crawlers to watchdog the appearance of specific information and when it happens, a notification is sent to an individual or a group of people subscribed to receive the notification. Crawlers are also used for malicious applications, for example, the web crawler is designed to look for and store email addresses that are later used by spam applications or to extract useful information about individuals that can be used later by phishing applications. Nevertheless the crawler technology has found its most important role in the support of the search engines, where it constitutes the underlying technology. Among all other web applications and technologies, the web crawlers happen to be on the first place when it comes to question the consumption of the internet bandwidth. The aim of the search engines is to create indexes of the web pages and the web crawler is a module that collects web pages for this purpose. Some of the most

important search engines nowadays like Google and Yahoo operate on web crawlers designed to cluster web pages insensitive to their content, which are regarded to as universal crawlers. If the content type determines if the web page is collected by the web crawler, the crawlers are called preferential. [10]

The Google uses several distributed crawlers for downloading web pages. The process starts when the server hosting the URL lists communicate to the crawler the links that are to be gathered. The collected pages are, in the following step, sent to the server dedicated for storing the web pages. In the final step the compressed web pages are saved. Web crawlers are to be seen as the most sophisticated components of a search engine. [11]

Further in this chapter the main concepts, data structures and algorithms for the web crawlers will be explained. After addressing the implementation general for all crawlers, different types of crawlers will be addressed: recorder, universal, topical, focused and incremental. [10]

Basic crawler algorithm

In the simplest form the crawler begins with a URL list called seed pages. The entries in the URL list determine which initial pages are to be collected and used to extract the links. The links found in these pages are used to retrieve new web pages and the whole process iterates until some regulating event occurs. This could be the maximum number of web pages to be fetched during a crawl run, for example. The described model is to be seen as a simplified, general functionality of the web crawlers. It doesn't take into account many known problems that exist in the area of HTML parsing and connection bandwidth being some of them. [10]

Figure 1 depicts the design of the basic sequential crawler. The main characteristic of this type of crawlers is that a single web page is collected at a time, leading to the crawler's resources being idle the most of the time. The module responsible for holding a track of the unvisited links is addressed to as a frontier. The frontier is initially populated with a basic set of links weather manually by an individual or by another application. The following steps are performed by the crawler within the iteration of the main loop:

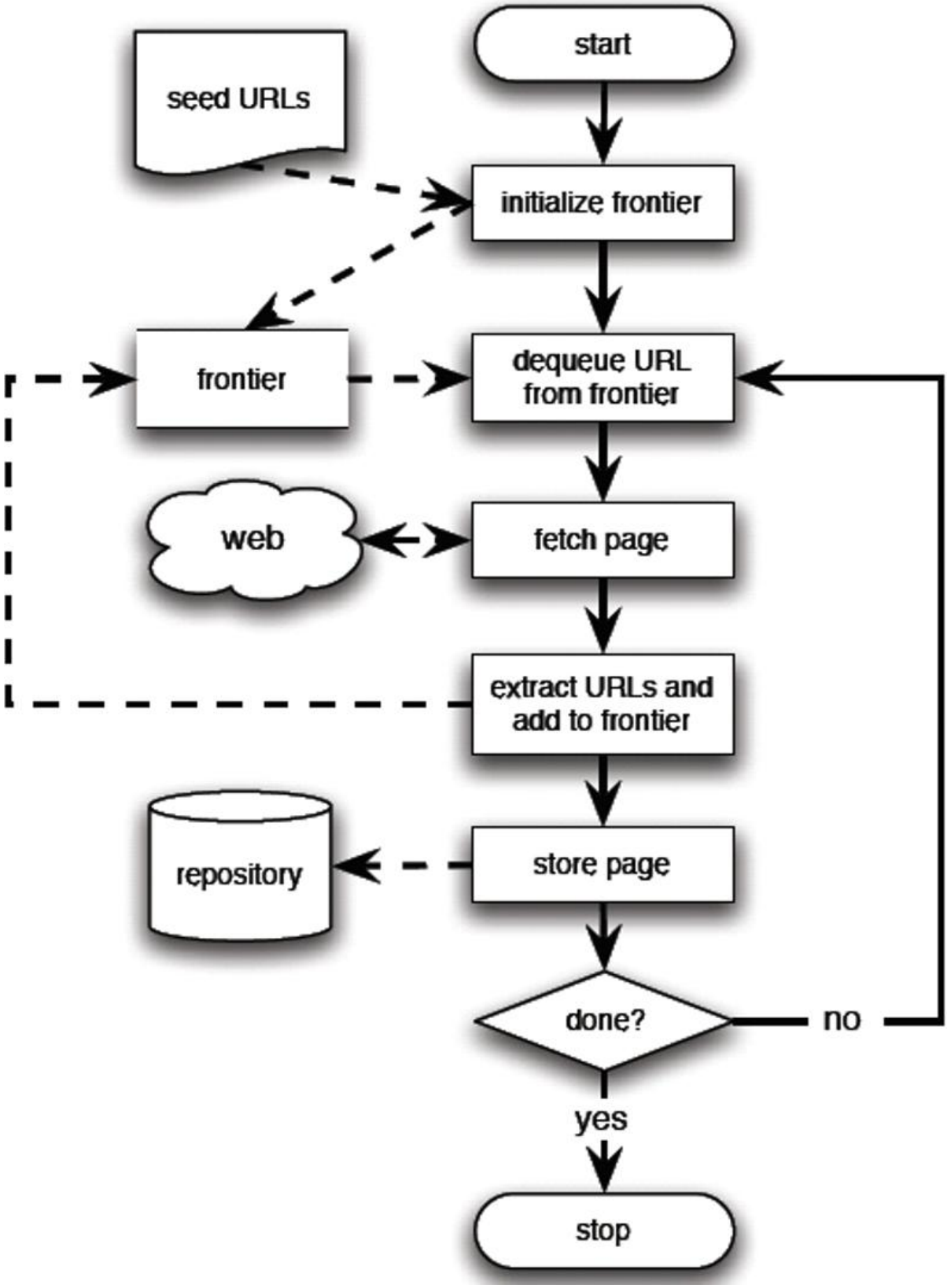
1. Choose the link of the web page to be gathered next
2. Download the page using the chosen link

3. Parse and store links from the collected web page
4. Populate the frontier with new links
5. Save the collected web page locally.

The crawler terminates when some predefined condition is reached. It also terminates when the frontier list becomes empty, which is rarely the case. This means that the crawler is basically a graph search algorithm. The crawler begins with few initial links that can be seen as nodes of a graph search. It then uses extracted links from the web pages form the initial set to get to the other nodes in the search graph. [10]

As the frontier holds the list of unvisited pages, it is to be seen as the key part of the web crawler technology and in order to increase its performance it is usually kept in the main memory. [10]

Figure 1: sequential crawler algorithm



(Source: [10], p.313)

Breadth-first crawler

This type of crawler technology represents a specialization of the general model and the frontier is populated in a FIFO manner. This means that the link that is to be visited next is the one that has been the longest in the queue. The links that are newly collected are positioned at the end of the queue. In this way it is insured that the links that have been in the frontier the longest are visited first. The problem occurs when the frontier gets full. From this point on the breadth-first crawler populates the frontier with only one extracted link from every page, losing all the remaining links [10].

In [12] the authors find that the web pages gathered at the beginning of a crawl run have a great quality. The quality of the fetched pages drops in the later phases of the crawl run, when the frontier gets populated and the only a one link per page is used after the link extraction to populate the frontier. Search engines aim to index the most of the pages available and fetching the high quality pages at the beginning of a crawl run is a sufficient for this.

The frontier contains only the links that are to be visited. In order to maintain this, the crawler needs to utilize some data structure. For this purpose each and every link visited is stored in a separated list together with the exact time. This list serves the crawler to crosscheck if a certain link has already been visited and when. It can also be used to calculate some statistical indicators after the crawl run. Another important detail is the prevention of doubling the URLs added to the frontier and avoiding that the same page is fetched twice [10].

Preferential crawlers

Preferential crawlers have the frontier implemented as a priority queue and the links are ranked using some criteria. The higher ranked links are visited first. The computation of the importance of the links is based on some features, like the type of the content [10].

Preferential crawlers are made to target areas of the internet which are of relevance for the triggering topic. The selection criteria to determine the relevance and importance of the web pages and links to be used for further crawling can be, for example, user entries in the comment section of the web page, or key words relevant to a certain topic. Nowadays

preferential crawlers became the basis for many specialized services like business intelligence applications of different types, where competitors and potential partners can be monitored. [14]

In the following section some of the most important implementation issues are presented.

Fetching

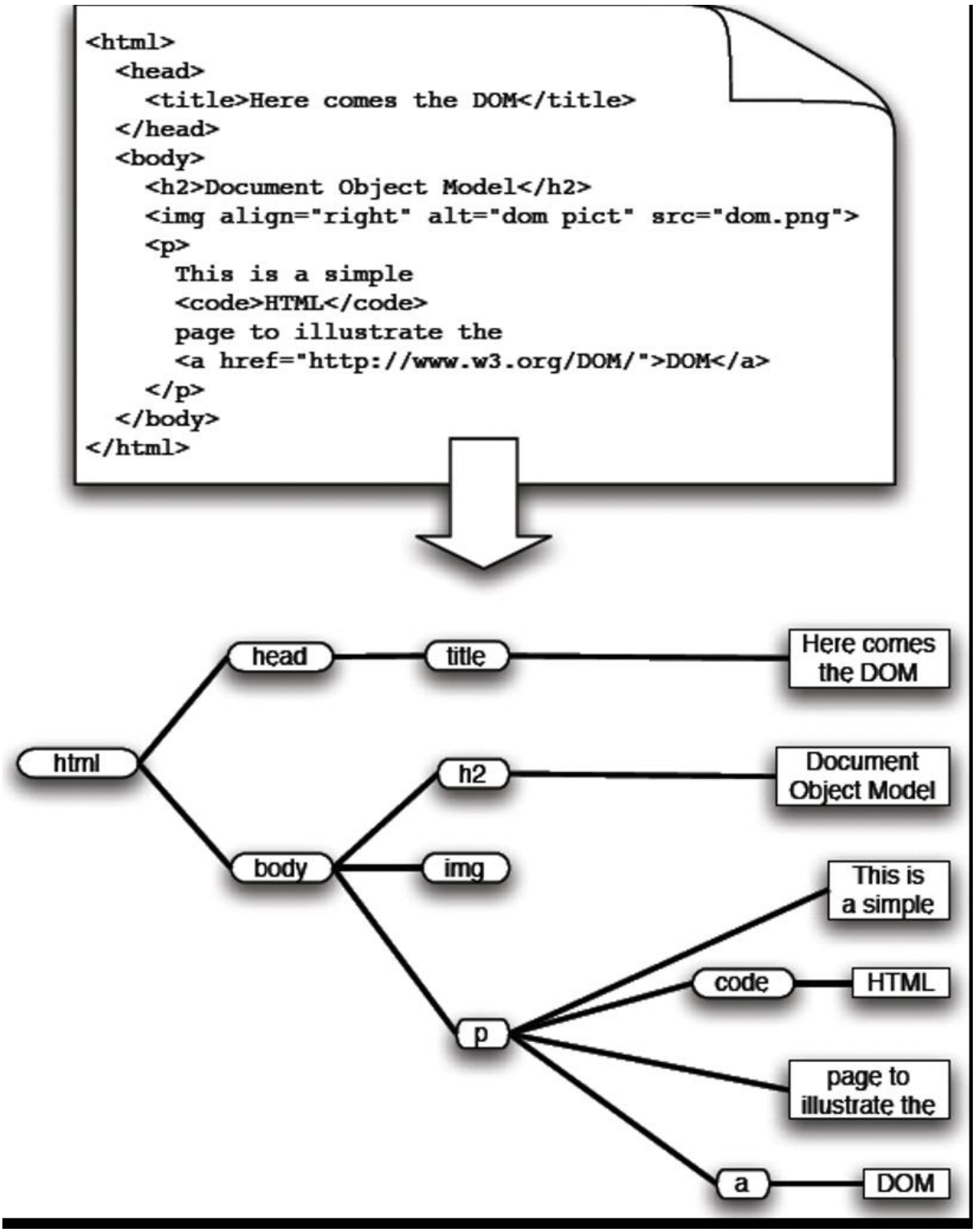
Fetching is the first part of the crawl process. It depicts the typical web client behaviour by sending a request to the server where the web page to be fetched resides, and analysis the response from the server. The crawlers have a build in mechanisms to time out connections to the busy servers, or when the response from the server is too big. Normally, only up to the first 100 KB are downloaded. The headers of the messages received by the server are analyzed and the useful information is extracted. Error-checking together with exception handling is of a great importance while fetching a page. Java and Perl among other programming languages offer ready to use modules that allow downloading web pages from the internet [10].

Parsing

The content of the fetched page is parsed whether while the page is being downloaded or after it was downloaded. The extracted information is used by two application levels: the crawler uses it to find the new links to be visited and the application on the top of the crawler uses it for its own purpose. HTML parsing in its basic form represents a link collecting from hyperlinks, but sometimes it involves much more serious examination of the HTML file. The Figure 2 shows the document object model (DOM) of an HTML code [10].

Correctness of the HTML pages is weakly enforced by web browsers. Very often the web pages contain missing tags, or the structure doesn't follow the proper nesting. These are just some of the many problems that make the HTML parsing difficult. This imposes large complexity issues on the web crawler's HTML parser. Similar as the web browsers, the crawlers need to tolerate these cases [10].

Figure 2: DOM tree built from a simple HTML page



(Source: [10], p.317)

Link extraction

During the fetching the web page is downloaded. This step is follow by the link extraction which is done by a certain module dedicated to this purpose. Here the web page is parsed and the new links are extracted. This step is inevitable in the crawl process and it happens in both scenarios: when the web page is saved in a repository and when it is deleted after the link extraction is finished. [13]

The HTML parser analyzes the HTML code of the downloaded page by making use of the tags and attribute and their values. For example, if a link is to be extracted, the anchor tag and a corresponding href attribute are to be found. After this step, it is necessary to process the collected links. This might include exclusion of the file types that are not to be crawled, like video or pdf files or exclusion of the dynamic web pages that that would involve interaction with some other applications or databases. The filtering criteria are basically defined by a crawler designer. If during the parsing process the extracted links are in the relative form, the conversion is needed before the frontier gets populated. Canonicalization stands for a conversion of the extracted links into their canonical form. Again, the canonicalization criteria can be very different and it is to be decided upon them in the design phase of the crawler. This implies that different crawler implementations specify different criteria. For example, some always specify the port number and others only when different from 80 [10].

Spider traps

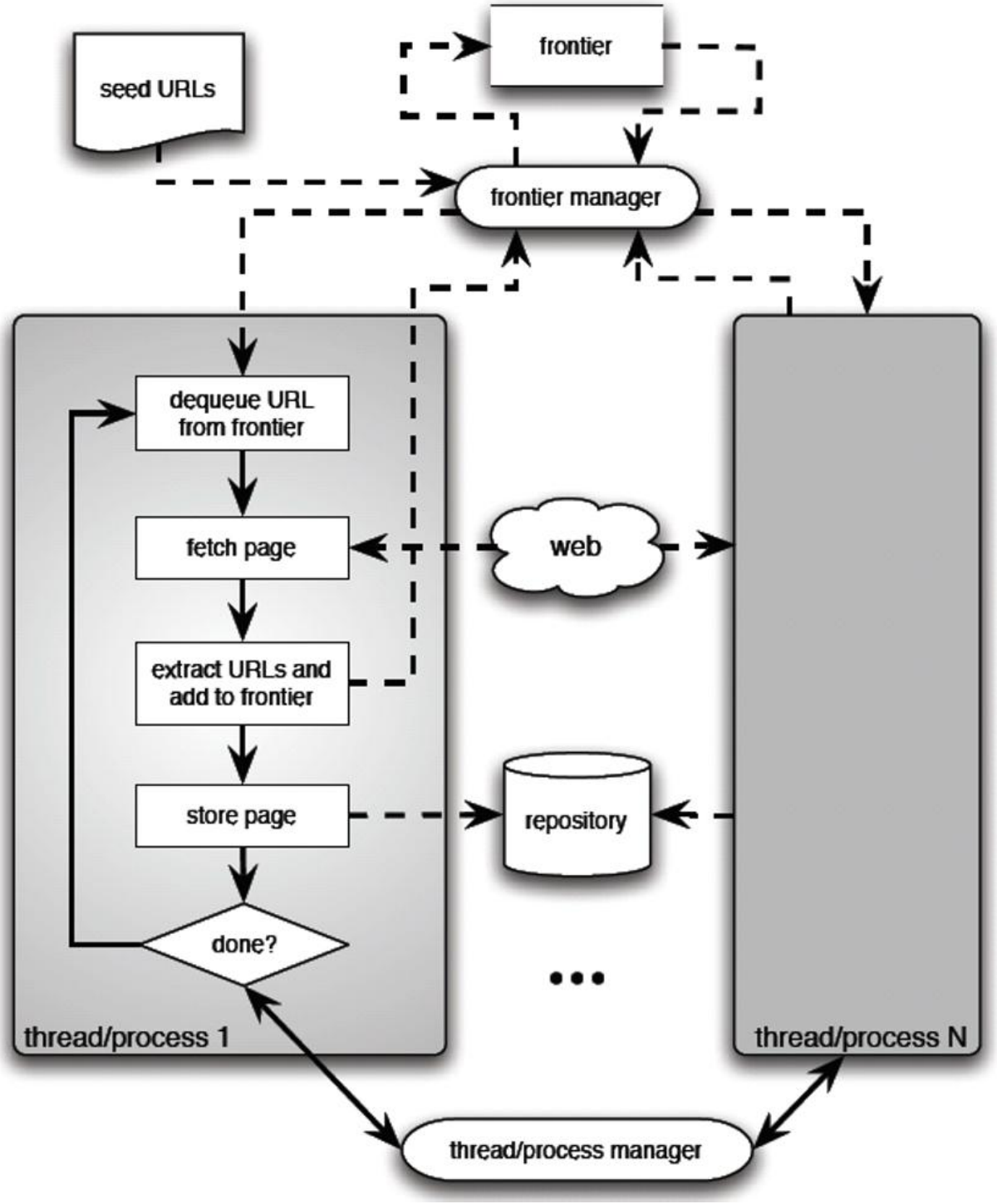
This term is used to define the script which form a part of the link and can evoke the dynamic creation of URLs that all point to the same page. Many different implementations exist nowadays. One of the most serious is when the links consists of a part that is a query. It means that a part of the link is dynamically changed and with the static part of the link it produces new links all the time and the crawler uses the new links to populate the frontier. This ends in a loop that consumes the resources. Among many other examples, the one that is often found is a calendar page, where the links are generated dynamically by the script within the URL, where newly generated links all point to the same page, but to the different dates of the calendar [15].

Spider traps not only consume the resources of the crawler, but also keep the server busy. If a script imbedded in the link contains a database query to populate database records for example, a crawler which is in a loop may populate the database with meaningless entries and consume the resources of the server [10].

Concurrency

The crawler uses resources to perform its task. The network is limited by the network bandwidth, the disk is limited by the seek time and transfer time and the central processing unit by its speed. These three resources are the most important when speaking about crawler performance. In the case of a sequential crawler only one of these resources is used at any given time, which puts the productiveness of the crawler on a very low level. Concurrency may be used on a process or a thread level to increase the overall performance of the crawler, whereas both possibilities have their disadvantages. In the case of a multiprocessing the operating system is in charge of the creation and destruction of the processes, which may have an impact on the reduction of the overall performance. Multithreading seem to be more difficult to implement. Figure 3 presents the concurrent crawler architecture independent if the crawler was implemented using threads or processes. Weather the concurrency is implemented on a process or a thread level, each of them represents a separated crawler. As in any concurrent environment, the shared data structures, in this case the frontier and the storage have to be synchronized. Concurrency allows a performance boost up to 10 times. [10]

Figure 3: Architecture of a concurrent crawler



(Source: [10], p.323)

Traditional (universal) crawlers

Search engines usually use universal crawlers to index the Web, as they are not interested in some particular topics, but rather the all available web pages on the Web [20]. Web crawlers face challenging tasks regarding the performance and reliability issues. For search engines, crawlers are the most fragile applications since they send http requests to a vast number of servers. This communication cannot be controlled by the search engines [11]. To point out the main difference between the large-scale universal and the concurrent breadth first crawlers, the following two criteria should be examined:

1. Achievement: in order to increase the performance and collect more pages per time unit by scaling up the system, it is necessary to refine the design of the crawler in several areas.
2. Policy: The goal is to collect the most important pages available on the Web and to keep them up to date.

Obviously these aims are opposed to each other. The implementation should meet the best possible trade-off between their objectives [10].
In order to meet these requirements, several issues need to be discussed: scalability, coverage, freshness and importance.

Scalability

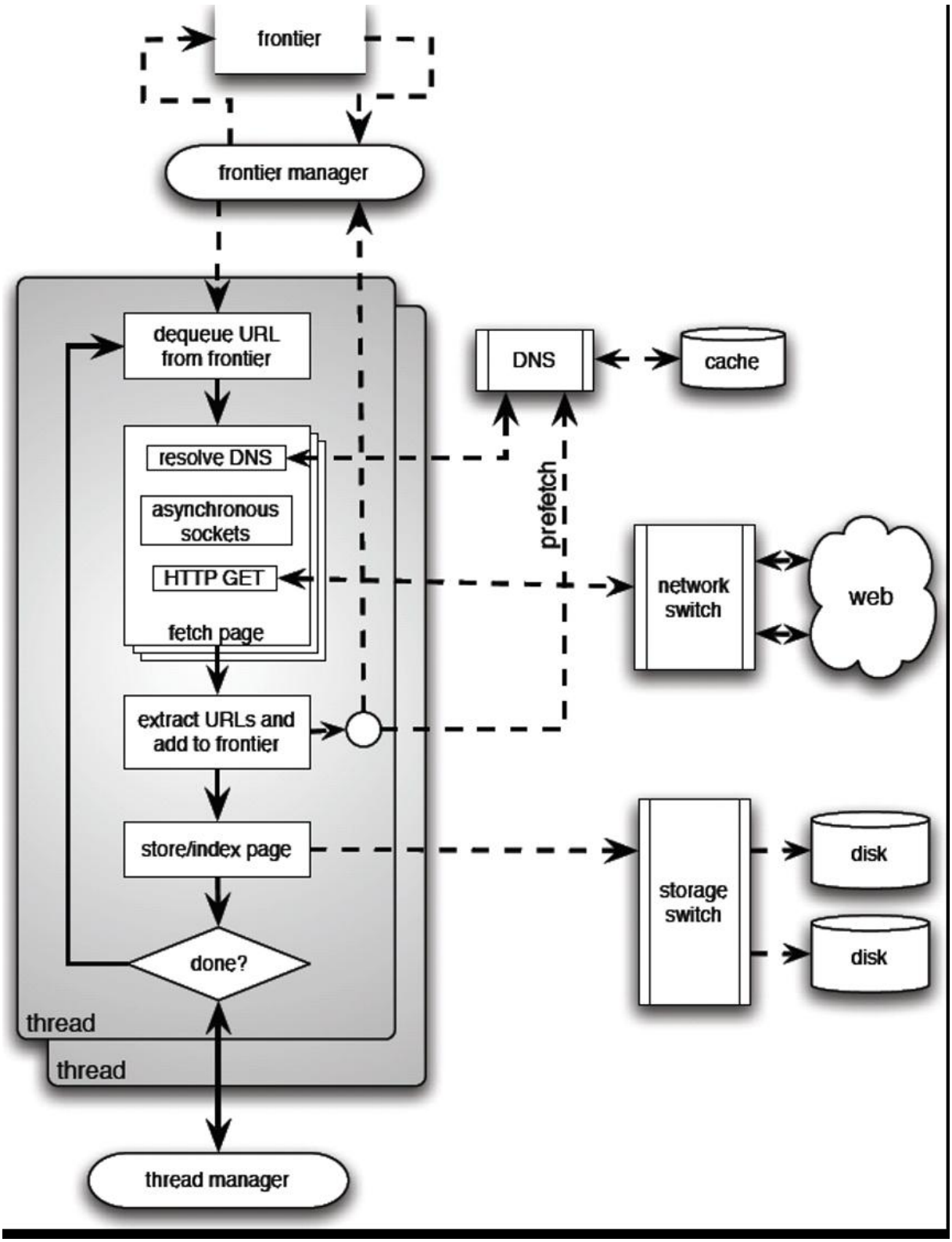
Figure 4 depicts the design of the large-scale crawler. The key difference when compared with the concurrent crawler depicted on Figure 3 is the use of asynchronous sockets instead of the synchronous, as they are not blocking. This opens the possibility for processes and threads to, at any given point, hold a vast number of connections opened and utilize the data bandwidth much better. If done in this way, the involvement of the operating system in the management of the processes and threads as well as the shared data structures is kept on its minimum. If additional routers are introduced to the system, the overall performance increases [10].

On the other hand, multithreading together with synchronous sockets involve much less complexity in the design of the crawler. The thread management is done by the operating

system instead of being implemented within the crawler. In this way it is easier to follow the track of the tasks performed by the threads [16].

If a managing module for the frontier is implemented the performance could increase. This managing module manages several queues containing links to a single server, which means that per server several queues would exist so when the TCP connection gets established, many pages could be collected during this single connection. In this way the number of TCP connections and related overhead are kept minimal [10].

Figure 4: Architecture of a scalable universal crawler



(Source: [10], p.325)

Coverage, freshness and importance

Search engines are commonly set into relation based on the number of the web pages indexed. This is a false criterion as there is an unlimited number of web pages on the Web. New pages get published every day, the size of the Web increases constantly and this phenomenon imposes some important challenges. Some of the key questions are: can the entire Web be downloaded? If not, which percentage of the Web should be enough? What is the quality of the missed web pages? Crawler coverage guarantee tells us which percentage of the good quality web pages are collected when the crawl run terminates. This information is also to be seen with a reserve, as its calculation needs to know the quality level of the pages that have not been fetched. Different statistical models are trying to deal with this challenge. Therefore, as in the case of the search engines, it is important to retrieve the most important web pages at the beginning of the crawl run. This would mean that the most important pages are downloaded once the crawler terminates [17].

The need to cover as many pages as possible stands in odds to the need to maintain fresh indexes. As the Web grows continuously, new pages are added, existing pages get modified or deleted constantly, the crawler needs to revisit pages already indexed [10].

Recorder

This type of a crawler is very much user driven. The user defines each and every step. This allows a very specialized and targeted crawling. The user populated the list with the links to be fetched, if there is a form to be filled, it provides data for it. As the initialization phase can be quite hard, there are already graphical user interfaces available to support the user while performing these tasks. If a user has to use a recorder without a GUI mask, the initialization process gets more complicated: the user has to know the HTML code in detail, it has to write a script defining each and every step to be performed, it should reference the obligatory fields in the forms, it should define how the forms are to be populated and much more. Regardless of the initialization process, the user initializes the form fields [18].

Focused crawlers

This type of crawler is designed to collect web pages whose content is of a certain topic. The basic functionality is same as in the case of the universal crawlers. The difference is that the collected page is examined if it belongs to some predefined topic. By doing this only the links to the web pages relevant for the crawl are given to the frontier. If the page doesn't belong to the desired topic, it gets stamped as being not relevant and the frontier doesn't get populated with the extracted links. The content based markers are used to classify the retrieved web pages. These markers are commonly a set of words characteristic for a certain topic that are to be found on the page itself, or in the links to the other pages. If the content based markers are found in the links contained within the page, it makes it possible to classify the page before the page is actually retrieved. There are also other different classification criteria for collected web pages. For example, the page structure can be used as a criterion or the actual location of the web page on the server. It is important to notice that the focused crawlers are automated; where the recorders require full user assistance, the focused crawlers function almost completely independently. [18]

The key problem with this type of crawlers is the correct classification and ranking of the pages. If this is achieved, the high quality pages are collected during the crawl run. A lack of reliable credit assignment strategy results in limited ability for focused crawlers to avoid the collection of the documents that are not relevant and omit the final goal, which is collecting the pages dealing with a certain topic. This means that the quality of the collected pages would be low and the total performance bad. This is where context-focused crawlers perform better [18].

The context focused crawler presented in [18] is an example how the functionality of search engines can be reduced in order to search only for the pages that refer to a specific topic, in this case to a specific document. The idea is to use the information about the document to form a set of criteria which will be used to analyze the web pages that are connected to the document through the links. The number of the links necessary to reach the document is predefined. In this way the maximal number of links that need to be followed to reach the document defines the set of web pages that are in the certain distance to the document. A module that distinguishes if a web page is in the range of the document is trained with this set

of criteria. This module is called a classifier. They are used to foresee during a crawl run how many links need to be followed to get from the current to the target link [18]. Sometimes it is possible to collect the pages of interest if it is known what kinds of links lead to them. [19].

Topical crawlers

The focused crawlers are to be thought before the crawl run begins with good pages and bad pages, so the crawler knows how to distinguish between them. This is rather an ideal scenario, as in the reality such a set is seldom available. In fact, commonly only a small set of initial pages is available together with the description of the theme. The theme is in a form of a short query and several initial pages relevant for the theme. If only this information is available at the beginning of a crawl run, these crawlers are addressed to as topical crawlers. What characterizes the topical crawlers is that they don't use any textual classifiers to control the crawl run. Nevertheless the success rate of this type of crawlers can be very high while traversing the web based on some preferences and collecting web pages that seem related to a certain topic. If a web page is collected depends if the content found on the web page relates to the topic specification [10].

Universal crawlers used for search engines face the already mentioned scalability problems. The topical crawlers represent a possibility to successfully address this issue, as they divide the process of collecting web pages over users or search queries and by doing so they decentralize the crawling process. The queries constructed referring to a certain topic or similar criteria drive the topical crawlers to collect only the relevant web pages. Also user profiles can be used as criteria to drive this type of crawlers and to determine which links are to be stored in the frontier. This is seen as an additional benefit [22].

Nevertheless there are known problems regarding this type of crawlers. The most important is how to recognize the exceptional characteristics in the web environment and react on them and recognize the high quality links that should be traversed. One of the exceptional environmental signals is the rich content. This, together with the information on the already visited web pages represents a good source of information. There are many different ways to use the available information, from very cautious to very open. Of course, the main application determines significantly the context the crawling is performed in. For example if

only the popular pages on the Web are to be collected, the implementation of the crawler will differ significantly from the implementation where only the new pages are to be collected. The crawling process can also have other potential limitations, where the number of collected pages is limited by, whether the number of pages itself or the size of collected pages on the local drive. Therefore a crawling process is to be seen as a search issue having multiple goals and many limitations and this performed in an unknown environment is a challenging assignment [10].

Crawler ethics

The server can get busy if the crawler keeps on sending HTTP requests without a sufficient amount of time in between. This can lower the digital bandwidth and consume more than normally other resources of the server. Unwritten rule is that the crawler should not send more than 10 HTTP requests within a second. Otherwise the server could become occupied by trying to answer to the crawler's requests and by doing so even limit its normal functioning, like interaction with a human individual. To prevent this, it is necessary for crawlers to put in place measures for request distribution across many servers and to enable any single server to receive requests at some reasonable frequency, like one request every few seconds.
Occupying all resources of a server represents one of many policies whose compliance is a must for every ethical crawler. These policies are known as crawler etiquette. It is also of a great importance to expose the crawler identity. Every HTTP request contains a header with a name, a version and a link to the information about the crawler. Commonly a separated web page exists containing all the relevant information about a crawler and it is sufficient id the link to this page is sent to the server within the request's header. The crawler etiquette should comply with the so called robot exclusion protocol, where the restricted area for the crawler is explicitly defined. File called robots.txt commonly contains entries like this. [10]

Analysis of some of the open source technologies for Web data mining

An overview of selected technologies for Web data mining is given in this chapter. There are many tolls available nowadays and it is out of the scope of this work to give an overview of available solutions, but rather present few selected examples.

Heritrix

In 2002 the Internet Archive, which is a non-profit organisation, wanted to make a publicly available digital library. The organisation wanted to crawl for its own purposes, but also to cooperate with other institution which also wanted to archive the Web. A large scale universal crawler was needed so the Internet Archive analyzed the solutions available on the market and concluded that at that time there was nothing which would satisfy the need. The Java was chosen to implement the crawler. Further, the software was made publicly available as the organisation wanted to improve the cooperation among different institution having the same goal: to archive the Web. [24]

The key features include:

- Recursive content collection
- The operator provides the seed and directs the crawl run
- Breadth-first crawling
- Scaling achieved by modulation
- Highly configurable, some of the options available are: settable locations for output log files, maximum download size, download limitation by the number of files, crawl time limitation, multithreading, politeness configuration to set the minimum and maximum time between requests, configurable filtering mechanisms and much more.

Web-based user interface allows the configuration of the crawls, starting, stopping and pausing the crawls and is also utilized to get an insight in the post crawl statistics. [24]

WebSPHINX

WebSPHINX (Website-Specific Processors for HTML INformation eXtraction) represents a Java development platform for crawler development. Crawler Workbench and the WebSPHINX class library are the two parts of the WebSphinks.

Crawler Workbench

The Crawler Workbench is a GUI that allows the configuration and supervision of a web crawler, which can be assembled out of different modules depending on the user's needs. The functionality involves:

- Web pages are represented in a graph three
- Store web pages for off-line processing
- Representation of multiple pages within one single document to ease the viewing
- Text extraction according to a certain criterion
- Implement a customizable web crawler using Java and define how the pages are to be processed. [25]

WebSPHINX class library

The WebSPHINX class library offers many modules that can be used while implementing a web crawler, like:

- Multithreading for web page collection
- Data structure representing web pages and extracted links
- Content classifiers
- HTML parsing
- Robots.txt
- Rich content matching
- THML processing modules [25]

Web Harvest

Web-Harvest is a software application implemented using Java programming language. IT is published as open source software. Web-harvest is a crawler which allows fetching of target web pages and extraction of target data. This is done by applying many known methods for text manipulation and HTML parsing. The most of the published web pages are HTML and XML based, so the Web-harvest in implemented to primarily address the web pages of this type.

Data extraction out of the web pages is known as web scraping or web data mining. The internet is the biggest collection of files currently known and it is challenging to extract exactly the data we want, because the web pages often, among the useful code containing valuable information contains as well some style code. This benefits the users as it makes everything human readable, but on the other hand, it makes it more difficult for automated agents like crawlers to read them. Programmers intend to clearly distinguish between the content and style following guidelines and using some tools. HTML files get computed on the server side and then delivered to the clients. [26]

The goal is to find out how to extract data from web pages, knowing that all the pages are computed following the same logic. Web-harvest is configurable by populating configuration files, where the flow of the data collection and extraction can be defined. [26]

The following steps are performed in order to fetch web pages and extract data:

1. Fetch the web page
2. Clean the HTML code
3. Search for the links and produce the list containing them.

Web-harvest can be extended with many different modules for data computation, iterations, conditional execution, HTML parsing and many others. [26]

Easy-wGet

Easy wGet is a free network utility designed for retrieving files from the internet using HTTP and FTP. Easy wGet is designed to work in the background unattended. It allows a user to download web sites and to define the depth level for the links, so basically it is possible to download the whole content of a server. It can be used as a crawler and it supports the robots.txt. [29]

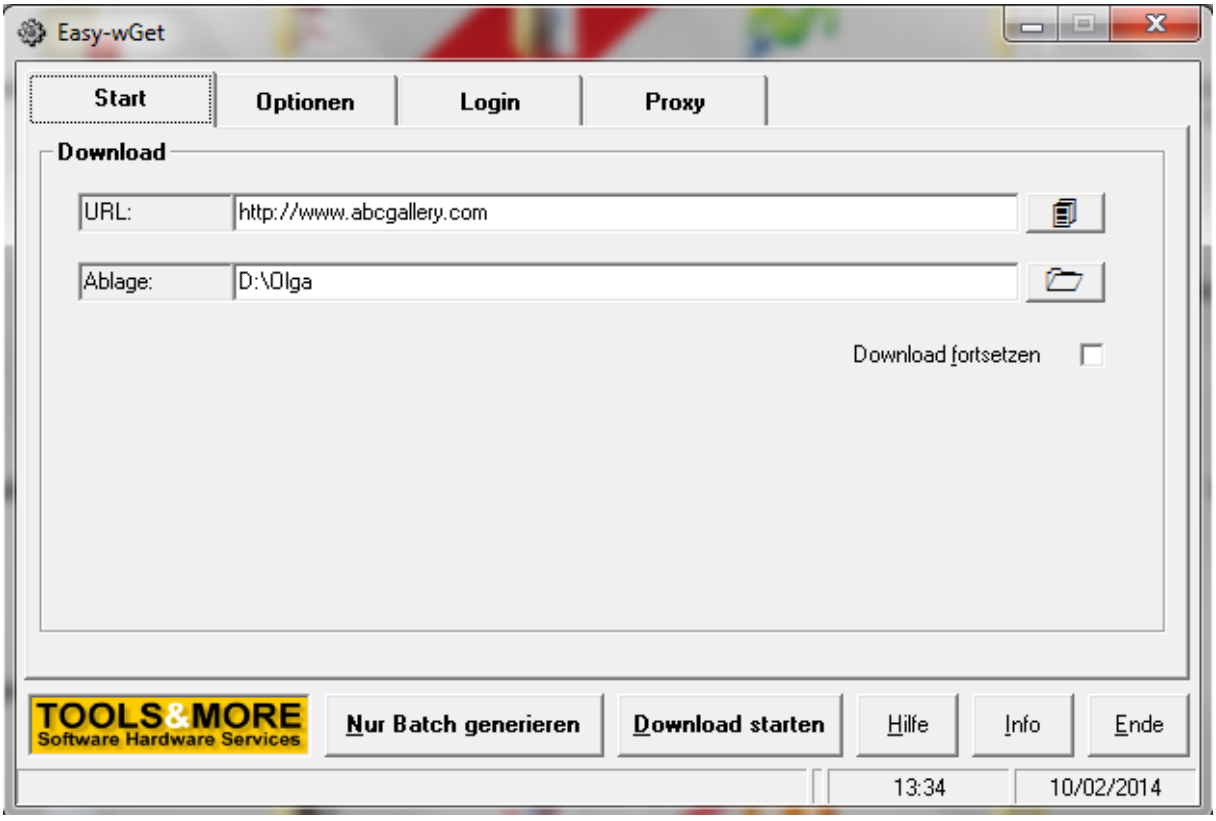
Using FTP or HTTP the Easy-wGet can be applied to fetch a single web site, or even the whole content of a server, creating a copy that can be used for offline browing. To achieve this neither a browser nor an FTP client is needed, the wGet allows it all with a single call of the console. The wGet is a console tool widely used and well known under the Linux users.

Easy wGet is a shell for the wGet. WGet is a tool that is initialized by a row of command line parameters [27].

The Easy wGet deals with these parameters in a user friendly way, having all the functions distributed in different tabs. The first tab called ‘Start’ is where the address from where to download data is to be defined, the desired URL, and the path on the local drive where the data is to be stored (Figure 5). After setting the parameters whether the download can be started immediately or the batch file can be generated to populate the wGet. The second option is useful if the download should be repeated frequently [27].

Easy Wget can be successfully used even if the stability and the speed of the connection are quite low. It is designed to continue the download until the entire web page is fetched and it can timestamp the fetched files so it knows when the download is needed. Easy wGet can utilize proxy servers, so the whole process of fetching gets faster and the bandwidth freed and the access behind firewalls achieved. All this functionalities are user defined, whether through the configuration file or through the GUI mask. [29].

Figure 5: easy wGet



GNU Wget can be configured to fetch big documents or even the whole content from a server by population a configuration mask. Some of the most important parameters are:

- Continue interrupted downloads
- Wildcards for file names
- The level of links to be followed
- Support for many languages
- Converting absolute documents to relative and allowing offline browsing
- Runs in Unix and Windows environment
- Utilizes HTTP cookies and proxies
- Utilizes continuous HTTP connections
- Background mode
- Timestamps files, so it knows if a file was already downloaded, if not a download is performed in the next run.
- Published under the GNU General Public License [28].

3.1.2 Web archiving

Cultural artefacts of the past have always been seen as crucial for the comprehension of the current state of the human society and for making reasonable predictions of its development. The World Wide Web is a medium where modern culture finds its natural form of expression. For this reason Web preservation is developmental must of the human society. Web preservation has been questioned and has not been accepted by all. Arguments that stand in odds with the necessity of the Web preservation could be divided into three groups: the first one comprises all who question the quality of the content available on the Web, the second all who believe that the Web would somehow preserve itself and the third one all who think that it is impossible to preserve the Web. [23].

The first group of individuals believe that there is no way of defining a standard for evaluating the quality of the content of the Web. In this group especially the representatives of the publishing industry are very energetic about it. The common arguments are based on the problem imposed by the amount of information available on the Web on one side and restricted knowledge about how it would be possible to archive the Web and what the cost would be to do it. The second group of individuals argues that the Web would preserve itself

by default. It means that if the content hosted on servers is useful and possesses the necessary quality to be further stored and offered on the servers, it won't be removed by the administrator. This category is mainly supported by people in the computer science world. The third group of individuals are questioning the possibility of preserving the Web although they realistically see the need for doing it. Some of the areas of problems they point out are the size of the Web, the legal side of the archiving process, as the content available on the Web underlies various national and international law regulations [23].

Web characterization in relation to preservation

The Web archiving has to be aware of some of the most crucial features of the internet. The most important characteristics will be reviewed in this section. The most important one is called a cardinality of the internet, which represents the number of identical instances of its content. The pages are published every day on the internet, which is the second most important feature of the internet. Third one is that the Web represents a cultural artefact of the human kind. [23]

On one side museums usually possess only one piece of each artwork, which is not true in some cases, for example there could be several examples of a photograph. Books and other printed media that are hold by the libraries represent a non unique artwork, where there are many identical replicas of the same piece. The Web is completely different form this. If we think that web pages are stored on a server, it would imply that there is only one example available. In this case the life of the web page directly depends on the server. Although the web page resides on the server and represents a unique source of content, there can be unlimited number of links and replicas of the same page, what happens when the page is collected by a web crawler. Even the link can't be used as unique identifier of the web page as the content of the web page can be changed. So it is important to notice that, for Web archiving purposes, all web pages on the Web depend on the servers and the content can be modifies on the servers, so a new resource is created containing the same URL for every alteration of the content [23].

Archiving methods

Many cultural institutions like museums and galleries have found ways of preserving the artwork and that has been very important for preserving the world's heritage. These routines can be used to deduce practices that can be used for the Web archiving. Nevertheless, taking into account the features of the Web discussed previously, the adaptation of these practices would be necessary, especially because they are mainly dealing with physical artwork. In the parts of this work that follow, it will be described what is needed to preserve the Web: the acquisition and organisation and storage [23].

Borgman (2000) in a discussion on the definition of global digital library, she explains the difference between the revolutionary and evolutionary aspects of the digital world. From the revolutionary point of view, the digital libraries are seen as a complex set of interconnected databases with special functionalities that will all together in the future compensate the traditional libraries. On the other hand, from the evolutionary perspective they are seen as a set of organisations that have been and will continue to serve the humanity in many different ways. So they are basically seen as an elongation of the standardized libraries. The author is rather for a moderate determination of the term digital libraries. According to her, they are to be seen as a vast heterogeneous and complex group of IT systems offering different functionalities on one side and other groups and institutions offering a vast range of information on the other side. The important question when talking about digital libraries is how the information can be fetched and how, after it is fetched it can be applied. [37].

Acquisition

The term acquisition is used to describe various methods for bringing the content into the digital library. It comprises the fetching of the content online and the storing of the fetched content in an off-line mode. How it is decided if the content is important and is of a high quality as well as the way to store the value of metadata attributes are not questioned here. The whole process of acquiring material online is very complex. The explanation for that is that there are many different ways to put the content online in technical means, so to take into account all of the possible methods is rather a very complex task. Further, the methods used to capture the content of the Web would need to deal with the limitation of the HTTP protocol

that fails to deliver a bulk copy of the content found on the server. In fact the delivery is possible only one file at a time and only if their unique identifiers are sent via HTTP request. So it leads to one of the key problems in the archiving of the Web being the identification of each individual source on the server.

There are three types of acquisition methods to capture the content online:

- Taking upon the role of an client
- Waiting whatever comes from the server side
- directly accessing the whole content kept on the server

Here we see one more application of the web crawler technology which is used to implement the first method [23].

The whole chapter is dedicated to Web crawler technology in this work. Here only the overview of this technology will be presented which will help to decide when to use it. The last two methods are still untapped and very difficult to implement as they would in all means depend upon the collaboration of the server owner. The fist one is called the client side archiving and is based on a crawler technology. [23]

The second method follows the following principle: it is important which content the user uses from the server so every interaction of the user is therefore recorded as it represents the selection criterion for the content based on the server. If some of the content hasn't been accessed, it won't be archived. It is therefore called 'transactional archiving'. The third method is based on storing the content directly on the side of the producer and is called 'server-side archiving'. These two last are very complex and face implementation limitations. They involve actively the producers of the content, the server owners, and therefore could vary for every single server. Nevertheless the overhead could be worth if the content of the server couldn't be fetched with the first method and possesses high quality content. [23]

Client-side archiving

Originally the crawler technology was developed for indexing purposes [10]. Application of the crawler technologies to Web archiving purposes implies some changes. The first one is that for the archiving purposes the crawler needs to collect all content and in its original form without any alteration, in order to store the original version of the content, whereas typically

web crawlers collect only the web pages they can address. One of the further problems is the way the web crawlers collect the web pages: they send an HTTP request to the server up to 10 times a second, as it was stated earlier, in order not to use too many resources from the server. For the archiving purposes this can be problematic as the resource content can be changed while the crawler is trying to send its requests politely. This is a big issue for the archiving crawlers as they are in charge of finding and collecting the content in its original form. Commonly the web crawlers just extract the links from the collected web pages, but the content remains on the server and if it gets changed, the links would still point to the content and the newest version would be accessed. This problem still doesn't have an adequate solution. Many alternatives exist, some of them involving some specialized crawlers, but none of them addresses the problem sufficiently enough. [23]

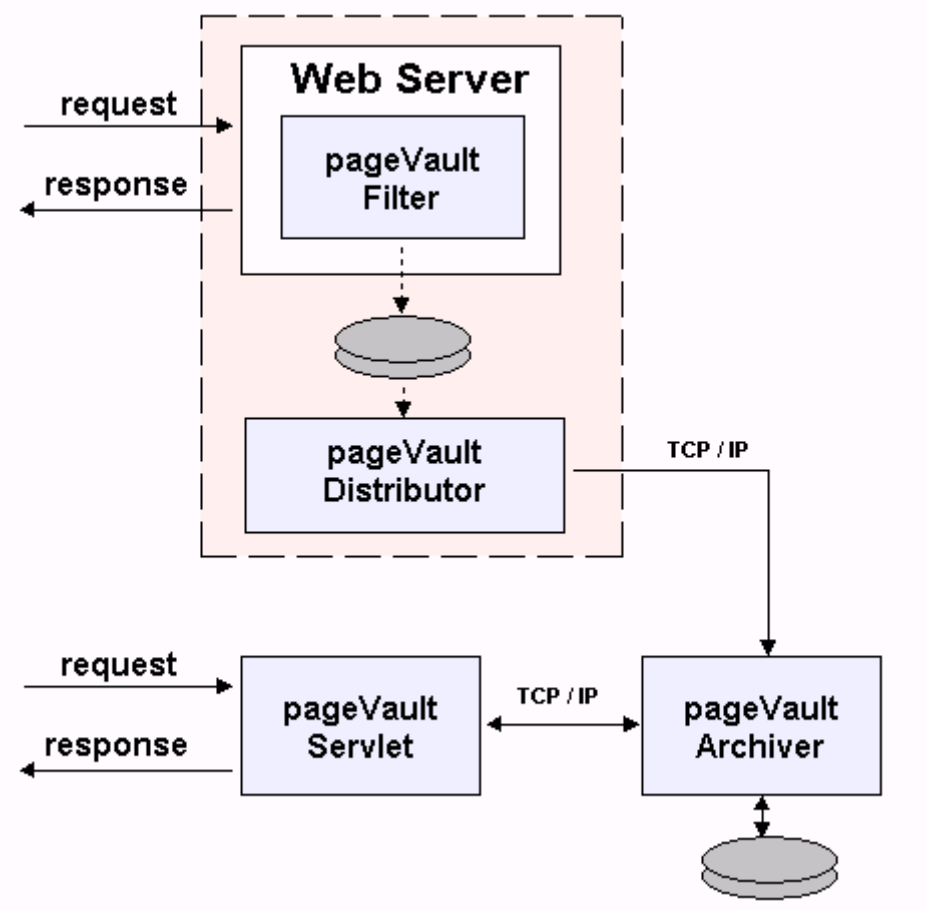
Transactions archiving

Web servers play a critical role in corporate communication nowadays; the documents published on the Web underlie the same law regulations as the printed media. So it is of a great importance to be aware of what exactly is put online as well as it is important to know what exactly was published in the printed media. There are many mechanisms that are trying to follow the lifecycle of the content and keep track of the resource alteration, like versioning for example. But the nature of the Web is very dynamic, the content is often generated real-time by combining many different sources, ever users in different communities are producing a new content in a form of forums, review pages etc. Even more, the generated content types are ranging from simple plain text files to complex digital files containing sound and pictures. Taking this into account, it is very difficult to state with a sufficient certainty how was the exact state of the content of the server at any given point of time, and which interactions happened between the server and the other content users. Transactions archiving represents a method for storing the request-respond pairs together with the transmitted content produced by the server, without taking into account the type of the transmitted files of how they were generated [23].

The PageVault design addresses this issue (Figure 6) where the Web server's solitary request/response pairs are saved and put into the archive, and by doing so the content viewed on the Web page is permanently saved. PageVault is to be seen as a filter for the

request/response pairs. This archiving method can be useful for tracking and recording all instantiations of the content, where the content never viewed will not be archived. The content available on the Web, which is never accessed, is called hidden or deep Web. The main problem with this method is that it implies that the possessor of the server would need to give its explicit permit. This method can be useful predominantly in the sphere of internal content storing. It keeps track of that was seen and when, in an exact manner [23].

Figure 6: PageVault



(Source: 37, page 14)

Server-side archiving

Server side archiving method is based on a direct copying of the server's content. In order to achieve this, the agreement of the server's possessor is necessary. Even if the method looks very simple to implement, the fact is that it is seriously complex and difficult once the content

is stored to make it accessible. If we simply observe the alteration the simple HTML files undergo, where the absolute links can't anymore be used to navigate through the files, it becomes clear why this method has its limitations. Even more, if the archived content is dynamic in its nature and requires many different resources to be assembled, the problem gets even more serious. Every running environment has its own environmental variables; the operating system often imposes some restrictions on the file presentation, so the same file could look differently on different platforms. If a script designed to work on a certain database is archived, it will be difficult to make this information usable, especially if the referenced database is external. All this make it clear that this method is very challenging when it comes to the implementation. When the content that is generated from many different resources is stored in its final form, the implementation is much easier. But in reality, for example in different blogs, this is not easy to achieve. One blog entry can contain links to the other pages or externally imbedded resources and this can continue endlessly. If all these web pages are later not present in the archived version, the stored document will differ from its original [23].

Organisation and storage

As it was shown previously, to copy of a website is a complex assignment. It is necessary to regenerate an information system where the information is accessible for users. In the best case scenario, the archive would have the same hierarchical structure, format, naming and linking as the original, but for practical reasons it is almost never the case. As shown previously, sometimes it is necessary to transform the files before they can be used. Usually, complex information systems are a unique combination depending on the operating system used, server configuration, and other environmental characteristics and in most cases, they would be difficult to recreate even for their original designers. That is why transformation strategies are necessary. These transformations could address linking and addressing mechanisms, formats, templates, and other features of the original information systems. There are three strategies introduced for the organisation of the stored Web content. The first one simply follows the idea that if a copy of all the material that resides on the server is copied; it will finally be possible to navigate through the material in the same manner as it was possible online. The second one invokes a web server. The web server would receive the requests of the user clients and fetch the archived material depending on the request and make it available

for the client. The last strategy is based on the reorganisation of the documents according to the specific naming and access paths. [23].

Local file system served archives

Local file system based archives make use of the fact that the unique resource identifiers definitions are able to apply certain logic on the prefix of the local file system and access files locally from the initial web page [23].

An example of this is given as follows:

http://www.abcgallery.com/A/aivazovsky/aivazovsky1.html
file:///D:/Olga/www.abcgallery.com/A/aivazovsky/aivazovsky1.html

By this one could navigate through the files saved into the local file system. This approach has also its advantages. The initial website can be stored locally and later on it will be possible to navigate through the file system to reach all the content of the initial webpage. This is much simpler than instantiating a web server to serve the archive. However in order to achieve this initial files need to get altered in a certain way. Transformation of the archived content is needed on two levels. The first is due to different naming conventions of the unique resource identifier on one side and the local file system on the other side. Further the conversion of the absolute into relative links is unavoidable. This implies the necessary transformation of the original. The main shortcomings of this approach are imposed by the file system itself, where the website's organisation has to fit into the organization of the file system. Mapping this organisation is connected with the changes and choices. Other limitation is the amount of data the archive has to serve. This challenges the file system capacity. [23]

Web Served Archives

This strategy is more demanding, whereas it offers much higher similarity to the initial web pages. It is free from the file system size limitations. Local files system served archives are based on file archiving; where Web served archives are based on response saving. Replies from the web server are saved in their original form in Web archive file container which

allows later access by an HTTP server. The main disadvantage of this procedure is that it is not possible to directly access the saved files. It is only possible to access then via an HTTP server [23].

Non-Web archives

Within this strategy the documents residing on the servers are extracted from the Web envoronment and reorganized, so the resulting documents would have different access logic and format. This happens when for example the documents that had link based access logic are reorganized to have catalogue based access logic. This is also the case when a whole webpage is transformed into the pdf format. This approach is useful for documents that were originally created and organized independently from the Web, like books, music and videos being some of them.

3.1.3 Data mining

The internet contains a constantly growing amount of Web pages with dynamic and frequently updated material. Data on the Web are usually imbedded in HTML pages and not always correspond to a know schema. Whilst the human user is able to understand data without problems, it is impossible to do so by a machine. It implies that extracting data from Web pages requires knowledge of both the data structure and data content. Basically, there are three methodologies dealing with this problem of data extraction from web pages:

- The first methodology is based on natural language processing (NLP).
- The second methodology tries to associate a Web page with some semantic markers (tags) in the process of Web page creation. One may use personalized markers. The limitation of this approach is well known: because the markers are personalized, it is very difficult to generalize them.
- The third approach is based on idea that as the original data are structured in a non-unique ways, it is more suitable to restructure them according to a common schema, which is independent of the original source. It implies that the extraction and combining data from different sources would be easier and more reliable [51].

Data mining is also known under the term knowledge discovery in databases. It is seen as a action of detecting arrangements of knowledge using different sources of information like databases, audio files and many more. Data mining is a complex research area and can be seen as unification of many different disciplines, some of them being areas of mathematics and computer science. Some of the most common data mining tasks are supervised and unsupervised learning. Supervised learning is also known as classification, as it uses predefined criteria to rank the content. Before the data mining process begins, data miners should analyze the application environment and select the source and the target data. Further, the data mining follows the following structure:

- Pre-processing: Out of various reasons the initial data is commonly difficult to mine and it may need to be cleaned. In the cleaning process the irregularities are cleaned. The available information can possibly be very big or contain content that is not important. In this case it is obligatory to cut down the size of the data and clean the unimportant content.
- Data mining: in the following step the knowledge discovery happens by applying a data mining algorithm to the available data.
- Post-processing: All not useful knowledge is disregarded in this step, so only the valuable knowledge arrangements are preserved.

This procedure, that is almost always iterative, is addressed to as a data mining. There is a traditional, web and text data mining. The previous utilizes data records from, for example databases. As the Web environment has gained on importance, the latter two data mining technique are becoming more and more interesting. Here we will focus on the web mining. [52]

Web mining

Web mining, also called Web harvesting or Web scraping has for goal discovering information of interest from the content of the web page, its link or usage data. The data that is to be found on the internet is very different in its presentation. The file formats differ, the structure of the files of the same file format is very different. The web data mining therefore extends the methods used in the traditional data mining to address the complexity imposed by

the web environment. There are some algorithms introduced in the past years. We will examine three:

- Web structure mining: this group of tasks discover useful information solely from the links. Links address the Web. If we analyze the links using some predefined criteria, depending on the criteria chosen we can find some important web pages, which is widely used in the search engine technology. It is possible as well to detect user groups connected by similar preoccupations.
- Web content mining: this group of tasks extract useful information out of the content of the web page. Nevertheless is possible to automatically cluster and classify Web pages depending on the topics they are dealing with. Mentioned practices are alike to the ones from the traditional data discovery, however the possibilities in the web environment are much better. The users for example leave reviews on a daily basis and this information can be used to achieve a ranking of the categories reviewed.
- Web usage mining: this group of tasks analysis the log file where every access to the server is noted. The goal is to deliver some predictability of the user access.

Both the traditional and web data mining are alike; commonly the difference comes out in the way the data is obtained. The traditional way the first step is commonly the harvesting of the data and storing it into the DWH. If we look closer the web data mining, we will see that the collection of information represents an important part of the process itself, by using web crawlers to address the vast amount of web pages of interest. When information collection is done, following three steps are same as for data mining: data pre-processing, Web data mining and post-processing. [52]

Structured data extraction

The extraction of the information from the Web aims to extract the data of interest from the web pages. Two problems are to be examined. This action faces two problems at the data extraction form the text written using the natural language significantly differs from the extraction of the data from the web pages. We focus here on the extraction of the structured data. Wrapper is a software module commonly implemented to address this issue. Depicting how to extract the information from the text written using natural language is out of the scope of this work. Structured data that is to be found on the internet are almost always records from

databases which are made available online using some kind of a mask to address the database fields. The extraction of such a data records is very useful and important, as it supports the collection and integration of heterogeneous data sources into much more powerful knowledge systems. This can be used for example when building an online shop or a page that ranks the prices of the products that are to be found on the internet. The number of business participants on the internet, form small companies to big enterprises, is growing every day. The number of data records grows every day and the importance of dealing with this circumstance is very high. Many firms are investing significant value in the development of applications that use the structured data extraction systems to feed the top business application with useful data. [52]

Smaller and bigger firms and researchers have done a significant work on the data extraction problem and there are three main approaches since the middle of 1990:

1. Manual approach: An individual examines the HTML code and tries to discover some knowledge schemes. After that is done, the human programmer codes the wrapper to achieve the data extraction. In this way it is very difficult to apply the same code to different pages, and even the modification can be complicated.
2. Semi-automatic wrapper approach: Also called a semi-automatic, supervised learning method. The research in this area started in 1995-1996. Within this approach, a set of non automatic labelled initial web pages and records is used to deduce a set of extraction rules. By applying these rules to comparable data sources, it is possible to extract the desired information.
3. Automatic extraction: in theory, for a given single or multiple pages, the rules are automatically found and the target data is extracted. [52]

A wrapper is basically a procedure designed for extracting information from initial dataset and producing the desired data in a predefined presentational form. When talking about rational databases, the wrapper module in charge for conversion of the records between different databases to allow communication and combination of different databases. Nevertheless when talking about the Web, the wrapper module should parse and extract information from the HTML and deliver the extracted information in a predefined manner. It is to be noted here that the wrapper deals with one source at a time, which means that if the wrapper is written to extract certain data from the HTML, this wrapper can be used only to process pages of the same kind. As soon as the logic and the structure of the page changes, the

wrapper needs to be adjusted or newly written. It also imposes that a wrapper library needs to be developed to compute files from different sources [53]. Further, the whole manual coding that is needed to develop the wrapper is seen as the biggest issues in the whole process. In addition to this, wrappers are always adapted to one certain source, they are difficult to maintain as they have to be changed for every change of the source. [54]

The first step in the wrapper generation begins with a detection of the initial set of the HTML pages. It is to be assumed that the pages are similar. The programmer reads the code and tries to derive a set of rules, possibly in a form of the regular grammar expressions and then to apply these rules to the initial set of pages to parse the files and to generate compute the desired result. It is not an easy task to find these rules. Actually, developing a regular grammar is a very serious problem described thoroughly in [55]. These rules can't be accurately deduced if the initial set of web pages consists only correct examples of the HTML files. Also if some negative examples are available within the initial set of pages, the reliable way to sufficiently good deduce the rules doesn't exist. This means that the wrapper can always deal with a certain amount of pages and for every new type of the HTML page it will be necessary to extend the wrapper [56]. Following the results presented in [55] a lot of research has been done to develop a way to deduce the rules for data extraction that would correctly work if a new type of a page is added. These proposals share some key features:

- This is the most intuitive approach as the initial set of positive examples is used to deduce the rules that would work on a more extended set of pages and possibly compute some new HTML pages successfully.
- This proposal is based on a software that is populated with the features the wrapper should posses. That is, rules are described within this software and the wrapper is then generated after the software execution. In some example it is possible to handle the nested data, but it is necessary to know in advance which attributes are to be extracted and how they are nested.
- The wrapper is generated for a single HTML page. [54]

For the manual generation of the wrapper, the programmer needs invest time and energy to analyze the HTML document. The wrapper is coded in the following step. Although it is easier to write the wrapper code for semi-structured documents, where it is easier to deduce the extraction rules, it still represents a challenging work. Of course if compared to the

extraction form the text written in natural language, the advantage is obvious. It is well know that manual programming faces known issues: it is easy to make errors and it takes time. Some solutions for automatic generation of wrappers exist. Basically these solutions are used to describe the rules needed for extraction. After all the rules are described, the wrapper is computed. The wrapper can be seen as a specialized HTML parser. Also, specifying the grammars using these tools is not a trivial task; one has to possess a deep knowledge about the software and invest significant amount of time. The fact is that many wrappers are hand written nowadays. Manually written wrappers cannot adapt automatically to Web page changes and need to be manually modified for every change of the domain. It is connected with high maintenance costs. On the Web, the number of potential sources is very big and is constantly growing; the structure and content of different sources can vary significantly. Even more, web pages are published every day, new files formats appear constantly. For this reason the technology and mechanisms used to construct wrappers are crucial if we want to achieve unsupervised data extraction from the Web. [53]

Semiautomatic wrapper generation means that there are some tools used to compute the wrapper. Some proposals offer GUIs where the user teaches the tool which data to extract from the THML files. This results in a much easier wrapper generation, as the user doesn't need to possess a lot of knowledge about the system, and as it uses predefined mask to enter the rules, it is more difficult to make errors. Nevertheless, whenever the web page changes, the rules have to be updated. Whenever a new web page occurs, new rules need to be deduced. [53]

Wrapper induction can be seen as a process to generate wrappers without supervision. Different algorithms are used to learn the regularities for data extraction. It is necessary to manually mark the data of interest. The wrapper induction system then discovers and learns the patterns and deduces rules to generate the wrapper. If more web pages are available for the training phase, the quality of the deduced rules is higher. The wrapper computed in this manner can be used to extract data from the web pages which have similar structure. [53]

3.2 Identification and analysis of web platforms containing artwork material

Further goals are to identify and analyze possible sources of artworks available in the World Wide Web and how they deal with the copyrights of available artwork material, the available epochs, the copyright law on national and international level and to yield a set of legal regulations and to gain an insight how Wikimedia deals with copyrights and how Creative Commons License is used for distribution of copyrighted works [4]

3.2.1 Olga’s gallery

Olga’s gallery was started as a family project to help the kids to achieve better results in the arts history, literature and religion classes. The "Olga's Gallery" was published in the 1999 and by a mistake it was launched under the false domain name "abcgallery.com". In June 2007 in the updated version of the owner’s description of the project was stated that the collection featured close to 300 painters and more than 12,000 artwork peaces. The page received over 30,000 visitors and 1,000,000 page views daily [30].

The Olga’s gallery offers many possibilities for obtaining the content: the artist index, the country index, the movement index, the name index, alphabetical ordering, the word/phrase search and the search artist by name. Further, the list with 20 most popular artists is available as well [30].

The available epochs are listed as follows: Gothic Art, Byzantine Art, Renaissance, Early Renaissance, High Renaissance, Northern Renaissance, Mannerism, Baroque Art, Rococo, Neoclassicism, Romanticism, Hudson River School, Pre-Raphaelite Brotherhood, Victorian Classicism, Symbolism, Realism, Impressionism, Pointillism, Post-Impressionism, Primitivism, Les Nabis, Fauvism, Art Nouveau, Analytic Art, Cubism, Expressionism, Dada and Surrealism [30].

In the terms of use section of Olga’s gallery is stated that all written materials belong to the owner and if the material is to be reused or reproduced, the explicit written permission is needed from the owner. All available material may be used with regards to the Fair Use Doctrine defined in the US copyright law. The governing idea is to advertise the comprehension and love for art. The project is non-profit. The displayed artwork material uses small resolution so any harmful for-profit use is excluded. All artwork can be used for Fair

Use with regard to the US copyright law. Whenever the artwork is used, it is mandatory to cite the origin and the citation should state the link “ww.abcgallery.com." [30]

3.2.2 Wikimedia Commons

Wikimedia Commons is web based database that consists of media files, like sound, images etc, which can be freely used. Wikimedia Foundation owns this project. Data form Wikimedia Commons is utilized in all projects owned by Wikimedia, including Wikipedia, Wikibooks, Wikivoyage, and many more. They can also be downloaded for offsite use. The collection of media files contains more that 20 million files. Wikimedia Commons aims to offer a web based database containing media files for free public use, especially for educational purposes. With educational purposes all possible ways of communicating knowledge are covered. Wikimedia Commons explicitly forbids upload of any content that is not free licensed. The following licences are accepted: the GNU Free Documentation License, the Creative Commons Attribution and Attribution/ShareAlike licenses. There are some other free licences regarding dealing with computer programmes that are accepted as well. The public domain is naturally accepted. [35]

Terms and conditions of reusing the information found on Wikimedia state, that the Wikimedia Foundation is not in position of almost none of the content that is to be found on Wikimedia sites. The ownership remains on the side of the individual creator. The most of the media files hosted on the Wikimedia Commons, although it may be used freely, their usage is still restricted in certain ways. One doesn’t need to get the written explicit permission to reuse the media file. This is needed only if the content is used under different terms than stated in the licence. Content under open content licenses may be reused without contacting the licensor, but:

- Sometimes it is obligatory to cite the creator
- Sometimes it is obligatory to explicitly state that the licence that regulated the use of the media file has to be applied on the reused media file as well. or even linked to it
- Sometimes if the media file is changed, the changed file or parts of the file need to be published under the same licence as the original file. [36]

Generally speaking as soon as one artwork becomes a public domain, it is not necessary to cite the creator anymore. However, some jurisdictions define some special regulations. However it is advised to always state the creator. Wikimedia Foundation believes that every media file hosted has the correct copyright and licensing. However, if this is not the case, the Foundation doesn't provide any warranty. Also other restrictions may apply, some of them being: trademarks, patents, personality rights, moral rights, privacy rights, or any of the many other legal causes which are independent of copyright and vary greatly by jurisdiction. [36]

3.2.3 The Web Gallery of Art

The Web Gallery of Art is virtual museum of European fine arts from 11th to 19th centuries. It has a form of a searchable database. The project was started in 1996 with the idea to present the Italian Renaissance of the 14th until the 16th century, but with the time it developed into a much larger project. The original compilation of artwork was enhanced to represent the evolution of Renaissance art as comprehensively as possible. The accent was put on the medieval roots of the Renaissance and how it developed to Baroque and Rococo. The 19th century art was also included, whereas the art of the 20th-century and contemporary art are not part of the collection [31].

The compilation possesses can be seen as an online gallery. Guided tours enhance the understanding of the connection between artwork and artists, taking into account historical situation. The user can play music while browsing typical for the epoch. At the same time various historical fact and artwork explanations are to be found along with the images. The compilation has a form of a database. [31].

As the site is in a form of a database, one can use the search mask to find images as a result of the search criteria. By populating the search mask, one can search after artists and make different groups of artists, depending on the search criteria. The alphabet presented in the middle of the page allows visitors to click on one of the letters. This results in a list with authors with that initial. Alternatively, one use search criteria to determine the result set. [31]

In November 2013 the collection had 33.337 images from 3.856 artists. The overview of the available epochs is not explicitly given. The database can be searched as described, but there is no clear separation by the epochs of the arts history.

In the terms of use it is stated that the Web Gallery of Art is protected under the copy right law as it represents a database. It is allowed to use images and documents downloaded from this site for scholarly and individual use. The use of images requires the written permission of the creator. [31]

3.2.4 Artsy.net

Artsy aims to allow the access to the artwork to anyone who browses the Web. It is a kind of a repository for artwork and information on artwork. It allows individuals to learn and enjoy the world's artworks. The constantly growing collection comprises more than 100,000 artworks by more than 18,000 artists from leading art fairs, galleries, museums, and art institutions. The contemporary art is due to the copy right rarely available online. This compilation consists of a great number of contemporary art pieces. Artsy partners with more than 1,500 leading galleries, as well as more than a couple of hundreds of museums other institutions across the globe. Artsy is supported by The Art Genome Project – defines the ways users with more or less expertise can discover new knowledge. The Art Genome Project tries to establish connections between artworks and artists across the globe. The criterion used to establish these connections is called a gen. On the platform there are currently more than 500 genes available, including movements in the arts history and other selection criteria. For example, Artsy would connect Warhol to Hirst using the pop culture gene. These connections allow endless opportunities for discovery and learning [34].

The displayed artwork is usually for sale, around 63% of the available material. The user can traverse the artwork in many different ways. The criteria include, among others, the price, medium (painting, photography) and many others. [34]

The terms and conditions section is very reach and only a brief summary of the parts of interest for this study will be given.

Proprietary Rights: All Content is owned by Artsy or by others who have licensed their Content to Artsy, and is protected by U.S. and international copyright laws, trademark laws and/or other proprietary rights and laws. The services are also protected under U.S. and international copyright laws as a compilation and/or collective work. Artsy owns and retains, solely and exclusively, all rights, title and interest in and to the Services, all Content that is created or made available to users through the services, and the look and feel, design and structure of the services, including but not limited to any and all copyrights, trademark rights, trade secret rights, patent rights, moral rights, database rights and other intellectual property and proprietary rights therein. The trademarks, service marks, logos and trade names that are displayed on or are in connection with the services are the registered and unregistered trademarks and service marks of Artsy or third parties in the United States and/or other countries. Use of the Services does not grant any ownership over any content, and except for the limited license and permission to access and use the services that are granted under this agreement. Use of the services does not grant any license or permission under any copyright, trademark or other intellectual property rights of Artsy or any third party, whether by implication, estoppel or otherwise. The rights not expressly granted in this Agreement remain reserved to Artsy [34].

Use of Content: Except as expressly permitted in this Agreement, it is prohibited to reproduce, distribute, adapt, modify, translate, create derivative works from, publish or otherwise use any Content for any purpose without express prior written permission from Artsy or the applicable rights holder. Any commercial use of any images or other content without express prior written permission from Artsy or the applicable rights holder is strictly prohibited [34].

3.2.5 Axisweb.org

Axisweb hosts the works of contemporary artists in the Great Britain. It allows artists to publish their work and to connect with each other and collaborate. The number of artists and other professionals from the art industry in the database exceeds 2.000. This makes it an interesting platform for everyone who is interested in the contemporary art of the UK. Arts Council of England founded the platform as a charity organisation. [32]

Originally founded in the 1999, the platform has changed its look very much with the launch of the new version in the 2013. Filtering and selection criteria are available on the site. When no filter is used, all the artwork is displayed to the user. The other option is to select one of the artists from the drop-down list, or to narrow the list of artists by selecting following criteria: location, region, occupation, activities and art form [31].

Axisweb is very interesting as it offers an insight into the contemporary art of the UK, whereas the contemporary art is copyrighted and copyright holders reserve rights defined in the copyright law.

The terms and conditions are strictly defined and among other things, it is stated that it is prohibited to copy, reproduce, republish, disassemble, decompile, reverse engineer, download, post, broadcast, transmit, make available to the public, or otherwise use Axisweb.org content in any way except for personal, non-commercial use. It is prohibited to adapt, alter or create a derivative work from any Axisweb.org content except for own personal, non-commercial use. It is prohibited to use any content in any way that is designed to create a separate content service or that replicates any part of Axisweb.org. It is prohibited to alter or remove, attempt to alter or remove any trademark, copyright or other proprietary or legal notices contained in, or appearing on, Axisweb.org, or any content appearing on the Platform [31].

3.2.6 Ibiblio – the Public's library and digital archive

Ibiblio.org is a project funded by several very important educational institutions in the USA. Ibiblio represents the biggest information database whose content is freely available. Everyone is welcome to supports the community by uploading material. Users can use many selection criteria to traverse through the database and explore the content. The information that can be found on the webpage is taken from many different sources. Arts and history, music and politics are just some of the areas that can be found on the platform. Around 15 million transactions daily characterize the Ibiblio [32].

Ibiblio offers many different topics, some of them being text, music and literature from all over the world, African American authors, American history, information on sports, and

philosophy of religion, computer programmes, artwork and many others. The users can view the compilation and leave a comment and even establish its own compilation. [32]

As the archive is very diverse in the content, the search options are quite powerful. One can access the recently added items or choose among areas like: social sciences, history, computer sciences and many more. In the arts and recreation there are many collections posted on the topic of arts, some of them being: Leonard de Vinci (as a link to an external source), the art and images of China (hosted directly by ibiblio.org) and many more. The advanced search mask is powerful and offers a vast number of criteria [32].

Access and use of the ibiblio.org internet server is the subject, among others, to the following terms and conditions: software, documentation, research data, and other materials submitted for installation on the ibiblio.org will be considered public domain, except for any express restrictions included the submitting party. University of North Carolina (UNC) is not responsible for providing notice of or enforcing any such restrictions. All parties submitting materials to the ibiblio.org represent and warrant to UNC that the submission, installation, copying, distribution, and use of such materials in connection with the ibiblio.org will not violate any other party's proprietary rights. UNC is not responsible for any errors created in or damage to the Materials as a result of their installation or maintenance on the internet server, or their use by anyone accessing the server. UNC disclaims all express warranties included in any materials, and further disclaims all implied warranties, including warranties of non-infringement of property rights [32].

3.2.7 Famouspainter.com

Famouspainter.com is an internet server hosting biographies and selected artwork material of some of the most famous painters. The compilation includes from Leonardo da Vinci and Michelangelo Buonarroti to Vincent Van Gogh, Pablo Picasso, Salvador Dali, Rene Margitte, Francisco de Goya, Frida Kahlo, Claude Monet, Henri Matisse, Rembrant van Rijn, Andy Warhol, Geogia O'Keeffe, Wassily Kandinsky, Edvard Munch and Gustav Klimt. Further, the collection includes a list of museums and other sources containing artwork material. There are no explicit terms and conditions stated on the homepage [33].

3.2.8 Museums

Many of the renowned museums offer some artwork material on their Web pages. It is out of the scope of this work to identify the nature of the offered information.

3.3 Selected projects of interest

This chapter gives an overview of selected projects which are dealing with similar topics as this work.

3.3.1 Europeana

Europeana represents a great number of artworks from many different cultural institutions in Europe. The published material includes paintings, sculptures, music, film, maps, photographs and many others. [44]

The users can perform the following actions with the available material: share, print, play, store locally and some more. Europeana is a project there many cultural institutions from Europe made an effort to offer publicly an insight into world's cultural heritage. [44]

Items on Europeana include:

- Pictorial work
- Written works
- Music
- Films and television.

Some material and topics represent world's cultural heritage, like Isaac Newton's book about the Laws of motion, Leonardo da Vinci's drawings, Johannes Vermeer's painting of the girl with a pearl earring and many more. [44].

3.3.2 dbpedia.org

DBpedia is a project that aims to extract information form Wikipedia and publish the extracted data on the Web. The information that is extracted from the Wikipedia is structured information. DBpedia makes it possible to combine information from Wikipedia with other sources and created combined collections. The idea is to discover new possibilities to apply the information available on Wikipedia. [45]

4 Practical part

4.1 Introduction

The idea of the work is to investigate the possibility of collecting artwork material available on the Web and store it locally. By doing so, the basis for an artwork database should be created. At the beginning of this work, when only the idea existed, but not the knowledge about the technologies and legal regulation that regulate the whole process, to the actual start of the work preceded the collection of information on Web mining technologies, law regulations and artwork material available on the Web.

The aim of this work is not to collect as much data as possible from various sources, but to investigate how it would be possible using existing technologies, and if needed, developing some new modules or adjusting the existing ones, to achieve the extraction of the desired data. Basically no constraints were set a priori only that the implementation should be simple and rely on existing technologies in order to avoid unnecessary coding. The Olga's gallery was used as a source for artwork material. Olga's gallery represent a big collection of artwork material, which is described in detail in a separate section.

Further, a method is to be developed, which would explain the process of collecting artwork material from various sources, extracting the information of interest and storing this information locally. The method should be applicable on all sources of artwork material and is to be seen as abstracted solution for the idea of this work, i.e. to collect available artwork material available on the Web and store it locally, by extracting only desired data from the potential sources. The implementation of the method on the basis of one source is the proof that the method implementation is possible. In the case of this work the Olga's gallery was used as a source for artwork material. It is to be explained, referring to the scientific findings represented in the theoretical section of the work, which part of the general method is implemented and which adaptations are necessary to make the implementation of the method applicable on other sources as well.

The available artwork material is in a form of collections of files and data on the Web. It implies that the usage of the material is possible in accordance to the law regulations. Law regulations differ among different regions and the idea of this work is to concentrate on the law regulations in Austria, The European Union and The United States of America. Basically, two distinct legal areas are to be examined: the database law, that regulates collections made available publicly, and the copyright law, that regulates certain rights on the artwork, which are in exclusive possession of the copyrights holder. This is important as the data found in databases is regulated by the Copyright Law and databases as collections are regulated by the Database Law.

Further, the Creative Common License is to be examined in this work. It is a digital copyright license that clearly defines the scope of copyright granted by the owners helping users and computational programs like Web crawlers to understand the scope of authority and gather digital contents from the World Wide Web without the infringement of the law. The licence grants additional rights to users, and the additional rights can be chosen by the copyright owner. It means that the copyrighted material, which underlies the copyright law, is distributed with additional rights granted by the copyright owner. The licence itself is also machine readable, so the search engines and crawlers can read it and understand it.

As the sources of artwork are in the centre of interest of this work, a survey on available sources was necessary. It was important to investigate the content of the available material, the form of presentation, the terms and conditions stated by the source owners, and also to investigate the qualitative nature of the available artwork: available epochs and descriptive information available for the artwork.

Further, Europeana, digital archive hosting Europe's artwork is a project reflecting partially the idea of this work. It comprises pictorial and written sources, maps, sound and video, and material given by national galleries and cultural institutions from Europe's twenty-seven member states. The available material was explored and how the material is presented.

4.2 **Requirement analysis**

The requirement analysis was necessary to answer the question on what was needed in the terms of technology, law and source artwork data, in order to design the solution of this work. Without sufficient knowledge about these areas, the extensive scientific literature survey was needed. This survey consisted of three parts: the Web data extraction technology survey, the legal regulations survey and the available artwork material survey. They will be presented in more detail.

4.2.1 Web data extraction and existing open source Web data mining technologies

This phase consisted of three distinct parts: the survey on the Web crawler technologies, the survey on the Web archiving and the survey on the Data mining.

Web crawler technologies seem to be the most important for this work. The survey comprised in depth analysis of the basic crawler algorithm, with two implementation approaches as breadth-first and preferential crawler, selected implementation issues general for crawler technologies like fetching, parsing, link extraction, spider traps and concurrency; traditional (universal) crawlers with scalability explanation as well as the coverage, freshness and importance, recorders, focused crawlers, and topical crawlers. Also, the crawler ethics is explained as an important part of the crawler technology.

Web archiving was another field that would potentially yield valuable input for the development of the solution of this work; therefore a survey had been conducted on this area. The Web archiving is introduced by explaining the general idea behind the Web archiving. Following, the Web archiving methods are presented: acquisition and organisation and storage. The term acquisition is used to describe different techniques that are used populate the archive with useful content. This embodies the content capturing on the Web and offline availability of the content. It neither includes the picking nor the process of saving the content with metadata generation [23]. The survey yielded that there are three main acquisition methods: the client-side archiving, that uses the Web crawler or Website copier adapted for the archiving purpose; the transactions archiving, that represents a method for recording and saving all different responds that are produced by a Web site,. It is not important what kind of content is present and how it is produced [37]; and the server-side archiving, that is based on a

direct copying of files from the server. The main finding is that the Web crawler technology represents the basis for Web archiving methods, as being the underlying technology of the client-side archiving. The methods for organisation and storage used for Web archiving are: local file systems served archives, Web served archives and non-Web served archives. The local file served archives are of importance for this work.

Data mining was the last field that would potentially yield valuable input for the development of the solution of this work; data mining is knows as a knowledge discovery from different data sources like databases and Web. Data mining begins with data analysis in order to understand the application domain and identify potential data set and select the target data. Afterwards, the actual data mining is done: Pre-processing, where the data cleaning is performed; data mining, where computation on the clean data is performed; and post-processing, where not useful data is disregarded. In this work we are interested in Web data mining.

Web mining, also called Web harvesting or Web scraping has as goal discovering information of interest from the links and page content as well from the usage data. Web data mining has inherited many techniques from the data mining. It is not only an application of traditional data mining because the information on the Web can be semi-structured, structured or unstructured. The survey showed that there are three basic Web data mining tasks: Web structure mining, which uses search engine technology for Web crawlers where the useful information is extracted from links; Web content mining where the target information is extracted from the content of the page; Web usage mining where information user interaction is extracted from the log files.

Web data mining can be seen as a part of broader discipline called data mining. We are here interested in Web data mining, wherefore the analysis of the Web data mining. Web information extraction represents the process of collecting target data from web pages. There are three approaches for information extraction from Web pages: Manual approach where the human programmer observes the HTML code and discovers some rules and codes a programme to extract the desired information; the semi-automated wrapper approach, where a set of manually marked files and data records is used to deduce a group of rules; and the automatic extraction, where for a given single or multiple pages, the rules are automatically found and the target data is extracted.

In the Web environment, wrappers should enable extraction of the data from the HTML documents and converting the extracted data into desired data structure. As it was shown in the corresponding theoretical part on structured data extraction, developing a wrapper generator for semi-automatic and automatic data extraction is a very difficult task and is out of the scope of this work. For this work, manual extraction will be accomplished by writing a data extraction algorithm, which would represent a manually written wrapper. This is supported by the scientific findings that many wrappers nowadays are hand written [53].

The survey on open source technologies for Web data mining included a number of solutions, where some of them are represented in a separate chapter. The presented technologies include: Heritrix, which represents a large scale crawler; WebSPHINX, which is a Java Class library and interactive development environment for Web crawler development; Web Harvest, which is an open source Web data extraction tool; and Easy wGet, which is a open source, free of charge application designed to retrieve files from the internet. Wget supports the usage of the robots.txt and is based on Web crawler technology [29].

For the ease of use the wGet is selected as a suitable open source technology for this project.

4.2.2 Legal regulations survey

As the researcher had no a priori knowledge on legal sciences, finding specific legal regulation that are of interest for this work was a challenging task. The aim was to discover legal regulations regarding database and copyright law. The regions of interest were specified at the beginning of the work and include Austria, the European Union and the United States of America. This information is used to judge the legal side of the solution presented in this work.

The Database directive of the European Union as well as the Copyright directive is implemented in Austria, which means that the national Law of Austria is adjusted according to Directives given by a higher legal instance, the European Parliament. For a detailed explanation it is to be referred to the corresponding chapter. It is to be said that the copyrightable databases gain protection of 70 years after the death of the last copyright

holder. In the USA there is nothing like Database Directive of the EU, which means that basically the databases are not copyrightable by default. Under certain criteria it is possible to copyright databases, but the process is rather unclear, as the criteria are not specified enough. The Copyright Law grants to the artwork, both in the USA and the EU, a period of 70 years of protection.

4.2.3 Available artwork survey

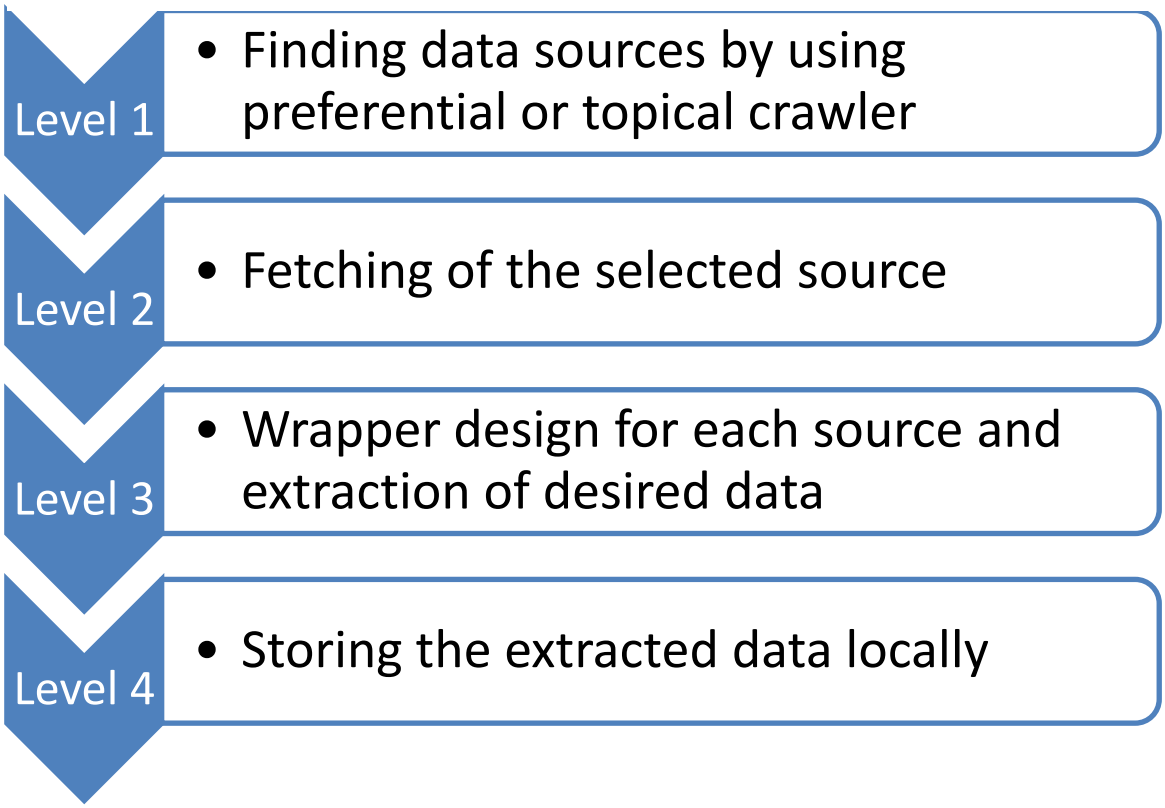
Within the requirement analysis it was necessary to give the answer on possible sources for artwork data mining. Therefore, a number of sources with artwork material were examined. It was necessary to understand the features of potential data sources, as that is the first step of the data mining process, as supported with the theoretical findings on data mining. The survey comprised many potential Web pages, some of them are described in the separated chapter. The summary of the findings is that the available artwork material on different Web sources differ very much in presentation, technology used for presentation, and kind of available material: some sources offer only images, other are mixture of images, sound and video. Some sources have underlying databases where the queries to the databases need to be executed in order to reach the artwork material. Some sources reside on servers in the local file systems. All resources have a unique positioning or nesting of the potential data. This finally stays in relation to the finding from the theoretical part on data mining, there wrappers, which are modules designed to enable extraction of data from the HTML are always written for one source, as data presentation in the HTML pages is always source unique.

4.3 General method design

The general method is presented in this chapter, which would explain the process of collecting artwork material from various sources, extracting the information of interest and storing this information locally. The method should be applicable on all sources of artwork material and is to be seen as the abstracted solution of the idea of this work, i.e. to collect available artwork material available on the Web and store it locally, by extracting only desired data from the potential sources. The general method is derived out of the findings presented in the

theoretical part of this work. The general method consists of four levels as shown on the figure 7.

Figure 7: The general method



4.3.1 Level 1 - Finding data sources by using preferential or topical crawlers

For this purpose focused crawlers or preferential crawlers should be used. Focused crawlers retrieve all pages that belong to a certain topic. Every retrieved page is analyzed if it belongs to a certain topic, usually by using a content-based Web page classifier. Only if the page belongs to the topic, the frontier is populated with its links. The focused crawler needs to be trained with labelled (positive and negative) examples of pages before the crawl starts. Such a set of pages is rarely available, as the focused crawler needs sufficient number of pages to be trained. Instead, usually a small initial set of web pages and a topic definition are available.

The topic can contain a single page or several pages and sometimes a short query. Preferential crawlers which use only this information are called topical crawlers. So, depending on the availability of the initial set of Web pages, whether preferential or topical crawler should be used.

4.3.2 Level 2 – Fetching of the selected sources

This level might be redundant if the crawler used in Level 1 allows fetching of the selected sources. If a crawler allows only the link extraction, this step is necessary. In that case, the easy wGet can be used to download the sources of interest by creating a batch file with entries for desired sources.

4.3.3 Level 3 – Wrapper design for each source and extraction of desired data

Wrappers are algorithms for extraction of the data from the HTML documents and converting the extracted data into desired data structures. The manual approach is used at this level, as the data sources differ very much in their structure and as the semi-automated or automated wrappers are extremely hard to implement, which is supported by the theoretical findings presented previously in this work. Despite the known weaknesses, the manually written wrappers are the most common wrappers present nowadays. The manual approach implies that the human programmer observes the HTML structure of a document discovers some rules and codes a wrapper to extract the desired data. Manually written wrappers are always written for one source only.

4.3.4 Level 4 – storing the extracted data locally

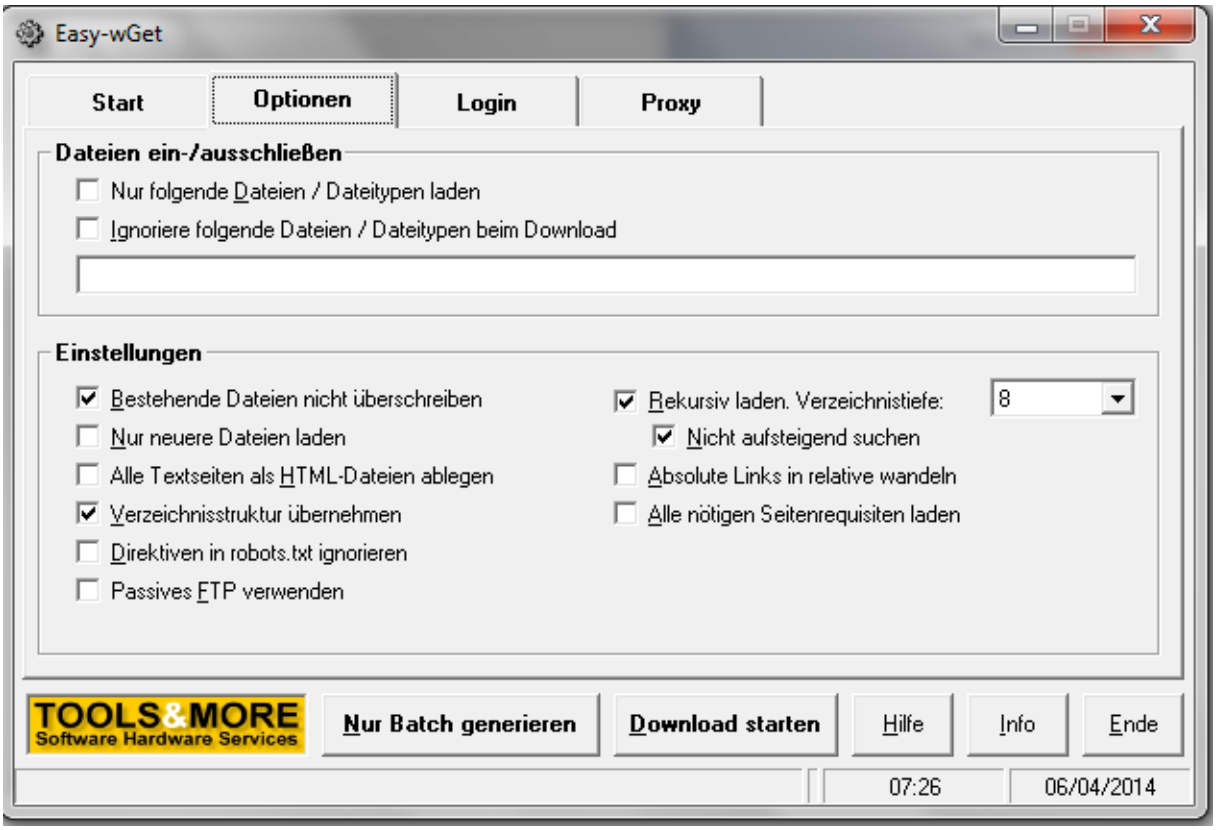
The data extracted in the Level 3 is stored in the local file system or database.

4.4 Implementation of the artwork database prototype

The general method represents a general solution where the data is collected from an arbitrary number of sources. After the data is collected, the desired information is extracted and stored locally. The implementation of the general method on the basis of one single source, the Olga's gallery, represents the proof that the general method can be implemented.

The implementation of the general method is done on the basis of the Olga's gallery. The easy wGet is used to retrieve the content form the server. All pages are fetched. This is achieved by setting the corresponding parameters in the easy wGet GUI. The parameters that were used to retrieve the Olga's gallery content are: the url, the destination folder on the local file system, the already downloaded pages are not overwritten if the download is repeated, the folder structure used on the source server is used also used in the destination local file system, and the depth level of the links is set to 8, as the smaller number seemed not to fetch all the data from the server. The GUI of the easy wGet showing the set parameters is to be seen on the following figure 8.

Figure 8: Set parameters on the easy wGet



After the content of the source server is fetched, the folder structure is analyzed by the programmer, in order to deduce the pattern for the data extraction. The folder structure contains alphabetically sorted folders, each of them containing all the artwork of a certain artist. The artist's folder contains pairs of html and jpeg files. The jpeg file represents the artwork itself, and the information about the artwork is to be found in the corresponding html file in its first paragraph.

The next step was the development of a manually written wrapper, which extracts the information from the html file and stores it in the database. The wrapper is developed using the Java programming language. Eclipse is used as a development environment. The corresponding jpeg file is stored in the local file system in the desired destination.

If the general method is to be implemented using some other artwork source, it is necessary to analyze the source and develop a wrapper to extract the data from the HTML and store the jpeg material.

For this purpose a database was needed. The WampServer is used. WampServer is a well known and widely used windows web development environment that allows creation of applications that are relying on Apache, PHP and MySQL databases [57]. The access to the database is possible also through the Web browser, which is shown on the following figure 9.

Figure 9: The artwork database

	author	link	path	title	infos	museum
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	View of the Coast Near St. Petersburg	1835. Oil on canvas. The Tretyakov Gallery, Mosco...	The Tretyakov Gallery, Moscow, Russia
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	View of Odessa by Moonlight	1846. Oil on canvas. The Russian Museum, St. Pete...	The Russian Museum, St. Petersburg, Russia
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	View of Constantinople by Moonlight	1846. Oil on canvas. The Russian Museum, St. Pete...	The Russian Museum, St. Petersburg, Russia
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	The Battle of Chesme	1848. Oil on canvas. The Aivazovsky Art Gallery, ...	The Aivazovsky Art Gallery, Feodosia, Ukraine
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	The Battle in the Chios Channel	1848. Oil on canvas. The Aivazovsky Art Gallery, ...	The Aivazovsky Art Gallery, Feodosia, Ukraine
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	Meeting of the Brig	Mercury with the Russian Squadron After the Defeat...	The Russian Museum, St. Petersburg, Russia
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	View of the Leander Tower in Constantinople	1848. Oil on canvas. The Tretyakov Gallery, Mosco...	The Tretyakov Gallery, Moscow, Russia
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2 \www.abcgallery.com\A\ai	D:\Olga\www.abcgallery.com\A\	Moonlit Night	1849. Oil on canvas. The Russian Museum, St. Pete...	The Russian Museum, St. Petersburg, Russia
Edit Copy Delete	Ivan Aivazovsky	D:\OlgaDestination2	D:\Olga\www.abcgallery.com\A\	The Tenth Wave	1850. Oil on canvas. The	The Russian Museum,

4.5 External libraries

In this project three external libraries were used: jsoup-1.7.1, mysql-connector-java-5.1.21-bin and commons-lang3-3.3.1.

Jsoup is a java library for working with HTML. It provides a very convenient API for data extraction and manipulation. Jsoup is designed to deal with all varieties of HTML found on the Web; from pristine and validating, to invalid tag HTML files, jsoup is able to create a sensible parse tree [58]. In this project it is used to parse the HTML files and extract the desired data from the HTML file.

Mysql-connector-java-5.1.21-bin is the official java database connectivity driver for MySQL [59]. It is used to connect the java application developed using Eclipse and the MySQL database of the WampServer.

Commons-lang3-3.3.1 provides a host of helper utilities for the java.lang API, String manipulation methods, basic numerical methods, object reflection, concurrency, creation and serialization and System properties [60]. In this project this library is used to deal with the HTML escape characters.

4.6 Main issues

Before the research on the legal frames which are relevant for this project was performed, the idea was to determine for each artwork what is stored in the database, if it complies with the legal regulations, and to store this information in a separated field in the database. With other words, it was to be stored in the database if it is legal to store the artwork in the database or not. After the legal regulations were examined, it would mean in the case of the Olga's gallery, that each of the records in the database would have one field with the positive value, as the entire content of the Olga's gallery is in the public domain, and the use of data from the Olga's gallery is free as long as it is used for educational purpose. For this reason this information is not stored in the separated field of the database record, as it would have no practical meaning.

The information about each artwork is stored in the HTML file in the first HTML paragraph. Olga's gallery is a project which started in 1999 and the structure of the HTMLs has not changed. The information about each artwork is in the form of a plain text, without a unique structure. Only the artist's name and the title of the artwork always appear on the same place and therefore it was possible to extract this information. The remaining information follows no predefined structure and it contains some or all of the following parts: the size of the artwork, the technique used, the museum where it is to be found, the owner, the year of creation or the supposed period of creation if the exact year is not known. These information form a plain text, separated by dots or commas. Therefore it was impossible to extract them and store as a separated fields in the database. This was however never the ultimate goal of the project. The goal was to determine which information could be successfully extracted and to extract and store that information. The idea was also to extract the information about the museum where the artwork is to be found on one side and to create a list containing some museums on the other side and then to compare this two values in order to determine if the artwork resides in the one of the museums from the list of museums, if not, a new museum would be added to the list. By doing so, the list of museums would be extended for all newly encountered museums and each database record would contain a separated field showing the museum where the artwork is to be found in. As the information about the museum is a part of unstructured plain text, the extraction of this information was not possible. The solution is to go through each HTML file and manually copy the name of the museum and populate a list of museums. That would mean that around 13000 files needed to be looked up in order to cover the available material from the Olga's gallery. A list containing some of the museums relevant for this work is created. To determine if the museum is in the list is done by a simple string compare method which checks if a string s2 is a substring of a string s1, like in the following example:

String 1: The Hermitage, St. Petersburg, Russia.
String 2: c.1653. Oil on wood, 71 x 59. The Hermitage, St. Petersburg, Russia.

Although technically very simple, this shows no level of automated data extraction, but rather a time consuming manual traverse through the data, which is not a goal of this project. This is due to unstructured data present in the HTML files. Therefore the information in the plain text is stored as one separated field in the database. The database column containing the museum

where the artwork is residing is populated only for the artworks whose corresponding museums are in the list of museums.

It was also very challenging to solve the problem of the correct text presentation, both in the java application and in the database. This was solved by using the commons.lang library and a proper collation for the columns in the database.

4.7 Quantitative analysis of the artwork prototype

The quantitative analysis of the solution encompasses the algorithm execution time, number of available artworks, number of invalid files, number of successfully extracted artworks and the success rate. The values are presented as follows:

Number of available files: 12673
Number of extracted files: 12589
Number of invalid files: 84
Success rate: 99.34 %
Execution time: 260 seconds

The invalid files are files where the artist's name or the artwork's title is missing. If one of the mentioned values is missing, the file is considered invalid. The implemented solution has a very high success rate of nearly 100%. The execution time differs slightly between successive executions and depends on the resource consumption by other applications and the operating system itself. It is to be said here, that this execution time refers only to the execution time of the java application that was written within this project. The time needed to download the server content of the Olga's gallery by using the easy wGet was around 7 hours.

4.8 Limitations of the work

The implementation of the Level 1 of the general method is a very complex task and is out of the scope of this work. It would involve a development of a focused or a preferential crawler, or adaptation of some of the available preferential or focused crawler technologies. The

development of a crawler is everything but a trivial task, which is explained in the theoretical part of this work. Focused crawlers could be used to achieve the retrieval of all pages that belong to a certain topic, by using a content-based web page classifier. Development of such content-based web page classifier is a complex task, as the sources on the web differ in the language used to describe the content and the key words used to describe it. So, the idea to traverse the web, detect the potential sources and perform the extraction of useful data can only be understood as an abstracted solution, whose implementation is rather impossible or the success rate would be very small, as there is no practical way for detection of all content of interest. The same applies for the preferential crawlers. Even if the small set of pages is available to train the crawler, it would be impossible to train the crawler sufficiently enough to recognize all the pages containing artwork material on the web. Further, crawling the web is a time consuming task and detecting the new artwork material would be just another limitation. The development of a wrapper that would extract the desired data from the sources discovered during the crawl is also limiting. As data representation varies among the sources, a wrapper should be developed which would address the differences among the Web pages and extract desired data. This is also just one more challenging task with known limitations. It is shown in the theoretical part that semi-automated or automated approach for wrapper generation is a very complex task, with known limitations.

It is also very difficult to compute the information if the artwork fetched from the source is in the public domain or still underlies the copyright law by automatically analyzing the HTML code. The most secure way would be to have in depth knowledge on the Copyright law and Database Regulation and analyze the potential sources and conclude if there are potential legal issues. It seems very difficult to exclude the human factor from this step.

5 Conclusion and future directions

It was shown in this work that it was possible to retrieve the artwork material form one source and perform highly successful data extraction, at the same time respecting the law regulations. The general method represented in this work for the artwork data gathering from the Web represents an abstracted solution of the initial idea of this work: to traverse the Web and collect the artwork material and build a basis for an online museum. In this work it was shown that this goal faces serious limitations: it would be very difficult to develop a sufficiently successful crawler to recognize and fetch the artwork material, to develop a semi-automated or automated wrapper for data extraction from the HTML files and to automatically deduce if the use of the artwork material is legal. Instead, the idea for the follow-up research would be to detect a manageable number of artwork sources, which preferably contain artwork material in similar form of presentation and develop a semi-automated wrapper or manually written wrapper for each of the sources. For each source it is to be proved if the contained artwork material and the database are protected by the Copyright law. By doing this a basis for an online museum could be created by using artwork from different sources without law infringement, by implementing the general method proposed in this work.

References

[1] Shan Lin, You-meng Li, Qing-cheng Li. Information mining system design and implementation based on web crawler. In: IEEE International Conference on System of Systems Engineering, 2008, 1-5

[2] Sanjay Kumar Malik , SAM Rizvi. Information Extraction using Web Usage Mining, Web Scrapping and Semantic Annotation. In: International Conference on Computational Intelligence and Communication Networks (CICN), 2011, 465-469

[3] Pooja Gupta, Kalpana Johari. Implementation of Web Crawler. In: 2nd International Conference on Emerging Trends in Engineering and Technology (ICETET), 2009, 838-843

[4] Chyan Yang, Hsien-Jyh Liao, Chung-Chen Chen. Implementing digital copyright on the internet through an enhanced creative common license protocol. In: Electronic Library, Vol. 27, Iss: 1, 2009, 20 - 30

[5] Shang-Hua Teng, Qi Lu, Matthias Eichstädt. Collaborative Web Crawling: Information Gathering/Processing over Internet. In: HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, vol.Track5, 1999, 12

[6] Sriram Raghavan, Hector Garsia-Molina. Crawling the hidden Web. In: Proceedings of the 27th International Conference on Very Large Data Bases, 2001, 129-138

[7] Lessig, L. Free Culture: How Big Media Use Technology and the Law to Lock Down Culture and Control Creativity, Penguin Press, New York, 2004

[8] Francis Crimmins. “Web Crawler Review”, In: Journal of Information Science, 2001

[9] Angelaki, G., et al., ATHENA: A Mechanism for Harvesting Europe's Museum Holdings into Europeana. In: J. Trant and D. Bearman (eds). Museums and the Web 2010: Proceedings, 2010, Toronto: Archives & Museum Informatics. http://www.archimuse.com/mw2010/papers/angelaki/angelaki.html

[10] Liu Bing. Web Data Mining: Exploring Hyperlinks, Contents and Usage Data. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011

[11] Brin S. and P. Lawrence. The anatomy of a large-scale hypertextual web search engine. In: Computer Networks, 30 (1-7), 1998, 107-117

[12] Marc Najork and Janet L. Wiener. Breath-first crawling yields high quality pages. In: Proceedings on the 10th international conference on World Wide Web, 2001, 114-118

[13] Christopher Olson and Marc Najork. Web Crawling. In: foundations and trends in information retriecal, Vol.4, No.3, 2010, 175-246

[14] Srinivasan, P., Menczer, F., and Pant, G. 2003. Defining evaluation methodologies for topical crawlers. In: SIGIR 2003 Workshop on Defining Evaluation Methodologies for Terabyte-ScaleCollections. 09.Aug.2013. http://dollar.biz.uiowa.edu/~gpant/Papers/crawl_framework_position.pdf.

[15] Mark Levene. An introduction to search engines and web navigation. Pearson education Limited, Harlow, Essex, England, 2006

[16] Allan Heydon and Marc Najork. Mercator: a scalable, extensible web crawler. In: World Wide Web, Vol. 2, No. 4, 1999, 219-229

[17] Cho Junghoo and Schonfeld Uri. Rankmass crawler: a crawler with high personalized pagerank coverage guarantee. In: Proceedings of the 33rd international conference on very large data bases, 2007, 375-386

[18] Inma Hernandez et al. A conceptual framework for efficient web crawling in virtual integration context. In: Web information systems and mining, Lecture notes in computer science, Vol. 6988, 2011, 282-291

[19] Xu Guandong, Zhang Yanchun, Li Lin. Web mining and social networking, techniques and applications. Springer, New York, Dordrecht, Heidelberg, London, 2011

[20] Alexander Graubner-Müller. Web mining in social media: use cases, business value and algorithmic approaches for corporate intelligence. Social media Verlag, 2011

[21] Diligenti M., F. Coetzee, S. Lawrence, C. Giles, and M. Gori. Focused crawling using context graphs. In: Proceedings of international conference on very large data bases, 2000, 527-534

[22] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers. In: ACM Transactions in Internet Technology, Volume 4, Issue 4, 2004, 378-419

[23] Julien Masanes. Web archiving. Springer, Berlin, Heidelberg, 2006

[24] G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic. Introduction to heritrix, an archival quality web crawler. In: Proceedings of the 4th International Web Archiving Workshop IWAW'04, Bath, UK, 2004

[25] http://www.cs.cmu.edu: WebSphinks, http://www.cs.cmu.edu/~rcm/websphinx/#about [01.02.2014]

[26] http://sourceforge.net/: Sourceforge, http://web-harvest.sourceforge.net/ [01.02.2014]

[27] http://toolsandmore.de/: Toolsandmore, http://toolsandmore.de/Central/Produkte/Software/Internet/Download/Easy-Wget/ [02.02.2014]

[28] http://www.gnu.org/: GNU, http://www.gnu.org/software/wget/ [07.02.2014]

[29] http://sourceforge.net/: Sourceforge, http://gnuwin32.sourceforge.net/packages/wget.htm [12.02.2014]

[30] http://www.abcgallery.com/: Olga’s gallery: Online Art Museum, http://www.abcgallery.com/ [11.02.2014]

[31] http://www.wga.hu/: Web Gallery of Art, http://www.wga.hu/ [11.02.2014]

[32] http://www.ibiblio.org, The Public's library and digital archive, http://www.ibiblio.org [11.02.2014]

[33] http://www.famouspainter.com/, Famouspainter, http://www.famouspainter.com/ [27.02.2014]

[34] https://artsy.net/, Artsy, https://artsy.net/ [27.02.2014]

[35] http://en.wikipedia.org, Wikimedia Commons, http://en.wikipedia.org/wiki/Wikimedia_Commons [28.02.2014]

[36] http://commons.wikimedia.org, Wikimedia Commons, http://commons.wikimedia.org/wiki/Commons:Reusing_content_outside_Wikimedia [28.02.2014]

[37] Borgman C.L. From Gutenberg to the global information infrastructure: access to information in the networked world. Cambridge, MA, 2000

[38] Fitch K. (2003). Web site archiving: an approach to recording every materially different response produced by a website. In: AusWeb 2003: The ninth Australian World Wide Web conference, Sanctuary Cove, Australia

[39] Directive 96/9/EC of the European Parliament and of the Council of the European Union of 11 March 1996 on the legal protection of databases, 1996

[40] BGBl. I Nr. 25/1998

[41] United States Copyright Office. Report on Legal Protection for Databases (August 1997)

[42] Council Directive 93/98/EEC of 29 October 1993 harmonizing the term of protection of copyright and certain related rights

[43] Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society

[44] http://www.europeana.eu, Europeana, http://www.europeana.eu/portal/aboutus.html [11.02.2014]

[45] http://dbpedia.org/, Dbpedia, http://dbpedia.org/About [02.03.2014]

[46] Title 17 of the U.S. Code §106 – Exclusive right in copyrighted works

[47] Title 17 of the U.S. Code §302 – Duration of Copyright: works created on or after January 1, 1978

[48] Title 17 of the U.S. Code §101 – Definitions

[49] http://creativecommons.org/, Creative Commons, http://creativecommons.org/licenses/?lang=en [11.03.2014]

[50] Muslea, Ion. Extraction patterns for information extraction tasks: A survey. In: The AAAI-99 Workshop on Machine Learning for Information Extraction, 1999, Vol. 2. No. 2.

[51] Snoussi, Hicham, Laurent Magnin, and Jian-Yun Nie. Toward an ontology-based Web data extraction. In: Proc. Workshop on Business Agents and the Semantic Web, 2002

[52] Bing Liu. Web Data Mining: Exploring hyperlinks, contents, and usage data, Springer Verlag, Berlin Heidelberg, 2007

[53] Priti S.S. and Rajendra A. Intelligent technologies for Web applications, Taylor&Francis Group, Boca Raton, 2012

[54] Valter Crescenzi , Giansalvatore Mecca , Paolo Merialdo. RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In: Proceedings of the 27th International Conference on Very Large Data Bases, 2001, p.109-118

[55] E. M. Gold. Language identification in the limit. Information and Control, 10(5), 1967

[56] E. M. Gold. Complexity of automaton identification from given data. Information and Control, 37(3),1978

[57] http://www.wampserver.com/, WampServer, http://www.wampserver.com/en/ [05.04.2014]

[58] http://jsoup.org/, Jsoup, http://jsoup.org/ [06.04.2014]

[59] https://mysql.com/, MySQL, https://dev.mysql.com/downloads/connector/j/ [06.04.2014]

[60] http://commons.apache.org/, Apache, http://commons.apache.org/proper/commons-lang/ [06.04.2014]