Hochschule Bremerhaven

Masterarbeit im Studiengang
Anwendungsorientierte Informatik

The Evolution of the World Wide Web: The Role of Bidirectional Linking

Markus Klie
(Matrikelnr. 32380)

Erstprüferin: Prof. Dr. Ulrike Erb
Zweitprüfer: Alfred Schmidt

Bremerhaven, den 21. Dezember 2015

“ *future generations will curse our names for assuming that we understood *anything* about link semantics ‘back in ’95’* ”

—Craig Hubley, 1995 [HTML-WG-95]

Erklärung an Eides statt

Hiermit erkläre ich an Eides statt, dass ich diese Arbeit
ohne fremde Hilfe selbständig verfasst und nur die
angegebenen Quellen und Hilfsmittel benutzt habe.

Bremerhaven, den 21. Dezember 2015

(Markus Klie)

Abstract

At the end of the 1990s, the World Wide Web had proven to be globally scalable and searchable and it was rewarded by a truly world-wide adoption. Having solved the major technical problems, a human issue became apparent: Web search—so it turned out—was highly susceptible to commercial manipulation to a degree, that search results were rendered completely useless. The manipulation of Web sites in favour of a seemingly high relevance in the search engines of that time—a technique dubbed *spamdexing*—triggered many discussions and research on the special nature of the Web and its possible future.

The findings of one academic research would eventually result in the world's most popular search engine: Google. The makers of Google succeeded in developing a search algorithm that proved to be *insusceptible* to commercial manipulation by determining the relevance of a Web page independently of its *own* content but merely based on references from *other* sites. In that way Google found a working analogy for the real world, in that a work is not to be considered important, if its authors say it is, but when *others* do.

A more fundamental criticism of that time was, that the Web was merely a collection of documents interlinked with meaningless links. In order to advance the Web, the next evolutionary step would be to promote the Web of documents to a Web of data, in which data elements could be interlinked in a meaningful way. Doing this in a standardized fashion would allow automatic software agents to perform tasks on the Web on behalf of their human users. Since the freedom of creating Web links on the Web of documents was considered key to its success, the idea to expand this freedom to a Web of data was promising.

Motivated by the still pending success of the Web of data, this thesis researches the evolution of both: the Web of documents and the Web of data. The linking capabilities of both Web trends are analysed over time and contrasted with each other. As a result, the Web of data is found to be highly susceptible to manipulation. If the Web of data will be one day globally adopted, commercial manipulation will become an issue again. However, because Web linking and the constraints under which it takes place are very different in both Web trends, the approach found by Google for the Web of documents will not work on the Web of data.

To make the Web of data less prone to manipulation, this thesis discusses three approaches: bidirectional linking for affirming facts, a same-origin policy for linking, and rules for changing the evaluation context. Bidirectional linking is found to have always played an important role in the evolution of the Web, also prior to the implementation in its present form. Bidirectional linking is consequently discussed against a wider background: from information theory and visionaries of global information management, its role in the evolution of the Web of documents and data to examples of bidirectional linking on today's Web.

Contents

Chapter 1

Introduction

1.1 Goal and Background

Goal

The World Wide Web has thus far evolved along two trends: the Web of documents and the Web of data. While Web linking—among documents or data—lies at the core of both trends, it is only the Web of documents that has become truly world wide. By contrast, the Web of data—when it comes to applications outside the academic world—is still in its infancy. In this thesis, the evolution of both the Web of documents and of the Web of data is analysed with regard to the respective linking capabilities. This analysis aims at identifying key features that have been crucial to the success of the Web of documents, but that are still lacking on the Web of data. If such a feature can be identified, approaches are discussed to bring that feature to the Web of data.

Evolution of the World Wide Web

Analysing the evolution of the World Wide Web is an endeavour, that can only be undertaken in a tentative way. While it may be beyond dispute *that* the World Wide Web both exists and evolves, it is very disputable *what* the Web is and *how* its architecture traversed over time. As a matter of fact, the mere question whether or not the Web has an architecture at all is a debatable one. Brian E. Carpenter raised that question on a larger scale, namely for the whole Internet, of which the World Wide Web is only one application. In [RFC1958] titled "Architectural Principles of the Internet" Carpenter writes that "[t]he principle of constant change is perhaps the only principle of the Internet that should survive indefinitely" (ibid.), and that "[m]any members of the Internet community would argue that there is no architecture, but only a tradition, which was not written down for the first 25 years" (ibid.).

What applies to the Internet, certainly holds true for the World Wide Web. Tim Berners-Lee, who wrote "Web Architecture from 50,000 feet" in an attempt to provide a high-level overview of the World Wide Web's architecture in 1998, is so cautious to speak

of "architecture, then, in the sense of how things hopefully will fit together." [Ber98a]. Two years later, Roy Fielding in his doctoral thesis "Architectural Styles and the Design of Network-based Software Architectures" for the first time names and defines architectural constraints, against which networked based software such as the World Wide Web can be evaluated (cf. [Fie00]). Roy Fielding is also among the authors of the "Architecture of the World Wide Web" [AWWW], that was published as late as 2004 and that has the merit of organizing the Web's core design components into the main areas of *identification*, *interaction*, and *data formats*. Despite all those efforts, the question of what the Web is, remains a question to this day (cf. [Not14b]).

Bidirectional Linking

From a conceptual point of view, today's World Wide Web is an example of a hypertext system with unidirectional links, i.e. a link from Web page A to page B does not necessarily imply a reverse link from page B to page A. The possibility of placing unidirectional—or one-way—links comes with a great degree of freedom, for it allows everyone to link to anyone else's page without requiring a reverse link to establish the linkage. There are, however, hypertext systems that only support bidirectional links. In those systems a linkage between page A and B can only be established by placing a link from A to B *and* a reverse link from B to A. While this makes data management easier, for the removal of one page on either side automatically invalidates the link, it imposes significant constraints on the creation of links. One could argue that the world-wide adoption of the Web is also to be sought in its support for one-way links. In that way, we may place a valid and functional link from our own Web page to our favourite newspaper, TV series or pizza delivery service without having to wait for a validation of our link by means of a reverse link from the respective page back to our page. Even though today's Web links are one-way by design, the implementation of a bidirectional hypertext system was considered, though discarded in the early days of the Web (cf. [Ber90a]). However, by examining the history of the Web of documents and the foundations of the Web of data, the author will show, that bidirectional linking—albeit in other forms of appearances—has always played an important role on the Web and will be crucial for the Web's next evolutionary step.

1.2 Methodology

Based on an analysis of relevant documents and discussions of the W3C community, the evolution of the Web is analysed along two trends: the Web of documents and the Web of data. By analysing and contrasting both trends, the author aims at identifying differences that explain why both trends developed with varying degrees of success. While those trends are not necessarily without overlap, they are a helpful differentiation between two distinct *foci* of the World Wide Web. The former trend aims primarily at a Web on which documents can be interlinked on a global scale by human users. The latter trend aims

at increasing the granularity of the Web by inter-linking not merely on a document level, but on a "data" level. That finer granularity is intended for enabling automatic software agents to read and understand the Web on a human user's behalf. The evolution of those two Web trends is analysed mainly by looking at the history of two *data models*: HTML for the Web of documents, and RDF for the Web of data. Since Web linking is at the core of both trends, questions pertaining to *identification* and *interaction* are discussed to the extent to which they are relevant to inter-linking documents or data. Analysed documents include especially normative sources. Those are recommendations by the World Wide Web Consortium (W3C) and Requests for Comments (RFC) by the Internet Engineering Task Force (IETF). Of the numerous W3C working groups and their collaboration tools, the publicly available mailing lists and archives of the W3C Technical Architecture Group (TAG), the Semantic Web Interest Group (SW IG) and the HTML working group are another source for the analysis. The public mailing list of the Hypertext Application Technology Working Group (WHATWG), who maintains HTML as a "living standard" both in competition to and in cooperation with the W3C, has also been considered. An additional source are the personal notes of Tim Berners-Lee, namely his collection of "Design Issues"[1], that the inventor of the World Wide Web shares with the public.

1.3 Structure

Chapter 2

Chapter 2 describes the conceptual foundations of linking information by introducing Paul Otlet's thoughts on this matter as described in his work *Traité de documentation*. In doing so, the author of this thesis wants to highlight the fact that theories regarding linking global information pre-date the present-day *Web* linking. Paul Otlet's itemization of "inventions to be made" to achieve a Web of global knowledge as well as his thoughts on the granularity of information still provide a high-level overview of the challenges of global information management. Likewise Vannevar Bush and Ted Nelson are presented as visionaries of today's World Wide Web.

Chapter 3

Chapter 3 describes Tim Berners-Lee's motivation to invent what later would become the current world-wide Web of documents. His ideas are presented as they were explained in his original proposal, followed by an analysis of the implementation of those ideas as they evolved in the different versions of HTML. Furthermore the author explains in which way the World Wide Web is different from traditional information systems, why traditional approaches of indexing the Web for search purposes failed, and how Google interpreted and utilized the networked nature of the Web to provide a solution.

[1] `http://www.w3.org/DesignIssues/` (last update 2015)

Chapter 4

The Web of data, that tries to advance the Web from a collection of interlinked documents to a collection of interlinked data, is analysed in detail in chapter 4: from Tim Berners-Lee's initial ideas to the manifests of the "Semantic Web" to its present-day implementation in the form of "Linked Data". Different ways of publishing data or *facts* on the Web and identifying things *not* on the Web are discussed with a focus on RDF, since this is the currently preferred data model for Semantic Web applications. Current approaches to harvest and consolidate facts from the Web are presented as well.

Chapter 5

The evolution of the Web of documents is assessed in chapter 5. The advent of Google and its successful interpretation and utilisation of the Web's structure is presented as a key event in that evolution. An explanation of how Google's approach touches upon the very foundations of the Web's nature is given. Afterwards current problems with the Web of data are contrasted with that of the Web of documents in the late 1990s. It is shown why Google's solution for the Web of documents is not applicable to the Web of data.

Chapter 6

Approaches for advancing the Web of data are discussed in chapter 6. Based on the analysis in chapter 3 and 4, and the evaluation in chapter 5, bidirectional linking is discussed as a possibility of affirming facts. To that end, the importance of the HTML `REV` attribute is emphasized. Furthermore a same-origin policy for linking on the Web of data and rules for changing the evaluation context are discussed as possible solutions.

Chapter 7

Examples of current Web applications that utilize bidirectional linking are presented in chapter 7 and compared with the ideas put forward in chapter 6. Since the discussed examples are taken from private Web applications, proposals are discussed how they can be made available to everyone on an open Web. Ted Nelson's recent comments on the re-utilisation of bidirectional linking in other domains than the World Wide Web are presented as well.

Chapter 8

The results of this thesis are summarized and discussed in chapter 8. The feature to be searchable, with a measure of relevance applied to the search results, that is insusceptible to manipulation, is presented as a key feature in the evolution of the Web of documents. The Web of data has been found to lack such a feature. Since bidirectional linking plays a crucial role throughout the evolution of the Web, its importance in the respective evolutionary steps of the Web is summarized.

Chapter 2

A Theory of Linking Information

2.1 Inventions to be Made

With the rather recent advent of the World Wide Web in the 1990s one might think that theories regarding global information storage and retrieval do not date back to much earlier than that. However, questions pertaining to global information management were addressed already in the 1890s. In 1895 Paul Otlet together with Henri La Fontaine founded the *Institut International de Bibliographie* (IIB), that by means of the *Répertoire Bibliographique Universel* (RBU) strived for nothing else but the creation of a worldwide index of documents to answer two questions on a global scale: what was written by a given author, and what was written on a given subject [1]. Deeming it a prerequisite for the latter, Paul Otlet developed the Universal Decimal Classification (UDC), that would allow to navigate the world's knowledge by subject area in a strict hierarchical manner. Paul Otlet, who used index cards for his endeavour, was very well aware of the technological limitations of his time, and he made a list of "inventions to be made" (*inventions à faire*) that would be needed to reach the aims of the IIB more effectively ([Otl34], p. 390, loosely translated):

3. *Photographie* a pocket device for photocopying texts and images of a book automatically [...]
5. *Lecture* a device for reading and extracting the content of documents automatically [...]
8. *Télélecture* a device for reading text at a distance
9. *Téléscription* a device for writing text at a distance [...]
13. *Machine à traduire* a device for translating languages automatically

Those devices, rather than operating separately from each other, would have to be connected to a single unit to perform the following actions automatically (ibid., p. 391, loosely translated):

[1] *Quels sont les ouvrages écrits par tel auteur*, and *qu'a-t-on écrit sur tel sujet* [Bib05]

1. Transforming sound into writing
2. Duplicating the resulting text as many times as needed
3. Organising the documents in such a way, that every piece of information (*donnée*) has its own identity and maintains its relations to the totality, so that the totality can be recalled if necessary
4. Describing every piece of information and attaching the description to the respective document
5. Describing the documents and organising them accordingly
6. Automated retrieval of those documents for review and presentation purposes—be that by the human eye or by a machine—for additional annotations
7. Automatic and arbitrary manipulation of all stored pieces of information in order to obtain new combinations of facts, new correlations of ideas (*rapports d'idées*), new operations with the help of numbers

Who reads the above *inventions à faire* of Otlet's *Traité de documentation* (published in 1934) at present time cannot help but thinking of today's search engines on the World Wide Web. In fact, a journalist of *Le Monde* dubbed the Mundaneum, that preserves the heritage of the IIB, the *Google de papier* [Dji09].

2.2 Books, Documents and Data

Paul Otlet referred to his own discipline *documentation* or *bibliologie*. At the beginning of his work he defines what he means by the terms book, document and *documentation* ([Otl34], p. 9, loosely translated):

> Book ... is the conventional term used here to refer to all kind of documents. It does not only comprise the book in the literal sense, a manuscript or print work, but it comprises reviews, journals, writings and graphic reproductions of any kind, drawings, engravings, maps, plans, diagrams, photographs, etc. *Documentation* in a larger sense of the word comprises: book, elements to indicate or reproduce a thought (*pensée envisagée*) in whatever form.

Following this definition, Otlet writes about the necessity of the *bibliologie*, a uniform science and a general technique of the document (ibid., loosely translated):

> There is a common language, a common logique, common mathematics. One must create a common *bibliologie*: the art of exposing, publishing and disseminating the data (*les données*) of science.

Otlet's redefinition of the term "book" to what we would call today "unit of information" or simply "data" necessarily raises the question of granularity. If a "book" in terms of information science no longer corresponds to a "book" in the physical world, how are

we going to determine what constitutes a "unit of information" that we consider independent and important enough to be treated separately? Looking at a journal, what do we consider a unit: the whole issue, an article, a paragraph of an article, or indeed every fact or thought (*pensée envisagée*)? Whatever the granularity, information scientists call that grain the "documentary reference unit (DRU)" and they distinguish it from its description and representation, that they call "documentary unit (DU)" ([SS13], p. 44). At the time of Otlet the DU of an article was an index card describing and representing that article. Nowadays, a typical DU is a database record doing the same thing. Following Otlet's above 'actions to be performed automatically' (cf. section 2.1) using those present-day technical terms, we get the following:

1. Define what constitutes a documentary reference unit (DRU)
2. For each DRU, create a documentary unit (DU) that describes and represents that DRU
3. Relate the DU to the DRU
4. Maintain the relationship between the DRU and the 'superior grain', in case of a journal: make sure that the article knows to which journal it belongs
5. Create a DU for the superior grain based on the DUs of its components; in case of a journal: make sure that the journal knows what its articles are about
6. Arbitrarily search all DUs in order to find relevant DRUs across sources
7. Derive new insights from the arbitrary manipulation (i.e. recombination) of DU

By publishing his *Traité de documentation* Otlet hoped to describe a standard and uniform way to perform the above steps on a global scale, which eventually would result in a globally search-able interlinked web of knowledge.

2.3 Visions of the Web

Paul Otlet's "inventions to be made" (cf. section 2.1) was not the only vision of what today reminds us of the present-day World Wide Web. In July 1945, Vannevar Bush, who participated in the second world war as director of the US American *Office of Scientific Research and Development* calls for the development of a pacific technology. In his article titled "As We May Think", Bush describes the state of the art of the science of his time and highlights the great opportunities for new inventions, among them a device called "memex":

> "Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory." [Bus45]

This device, that would "presumably be operated from a distance", would not only digitize "[b]ooks of all sorts, pictures, current periodicals, newspapers" as well as "[b]usiness correspondence" (ibid.) in order to make those items searchable and readable on screen, but it would also support the inter-linking of two arbitrarily chosen items by the user by employing a technique that Bush calls "associative indexing" or "building a trail":

> "Before him are the two items to be joined, projected onto adjacent viewing positions. [...] The user taps a single key, and the items are permanently joined [...]; and on each item [automatically inserted] dots by their positions designate the index number of the other item." [Bus45]

In 1965, Ted Nelson, then at Vassar College (N.Y.), refers to Bush's article from 1945 and states that "[t]wo decades later, this machine is still unavailable" [Nel65]. In support of the development of a machine with the features described by Bush, Nelson proposes a evolutionary file structure *ELF* consisting of three elements: entries, lists and links. About the link, Nelson writes:

> "A link is a connector, designated by the user, between two particular entries which are in different lists; [...] An entry in one list may be linked to only one entry in another list." [Nel65]

For the next 40 years Ted Nelson would work on the realization of an inter-linked information system as it had been originally proposed by Bush. In that same period he would also remain a critical contemporary of the actual developments in the field of computer science, that—as we know today—were significant, but according to Nelson lacked a unified and holistic approach. In 1981, Nelson summarizes his criticism of the insufficiencies of the computer technology of that time in his work "Literary Machines". To his mind, a key problem when dealing with electronic texts is a fundamentally wrong interpretation of text as being something intrinsically sequential. He argues that the sequential notion of text is caused by the "sequentiality of language and the sequentiality of printing and binding" [Nel81]. Owing to the digitization of text, however, the *presentation* of electronic texts would not necessarily have to be sequential anymore. As an alternative, Nelson proposes a non-sequential presentation of text that would allow to "create different pathways for different readers, based upon background, taste and probably understanding" (ibid.). To that end, chunks of texts would be connected by means of links so that the reader may move through a text in a non-sequential way "by reading one chunk, then choosing the next" (ibid.). Nelson calls this sort of text "chunk style hypertext" (ibid.). Although Nelson coined the word hypertext, it was not be the hypertext system proposed by him, that would make that word famous. We will return, however, to Nelson's ideas in section 7.3.

Chapter 3

Linking on the Web of Documents

3.1 The Advent of the World Wide Web

Unlike Paul Otlet or Ted Nelson, Tim Berners-Lee did not want to develop a global web of knowledge, when he invented what later would become the World Wide Web. Instead, in his paper "Information Management: A Proposal" [Ber89] he addressed a local problem at the organisation CERN for that he worked at that time. According to Berners-Lee, the information management at CERN was insufficiently taken care of using documentation (!) systems that used a tree structure and keyword[1] search. As an example of a system based on a tree structure, he mentions among others the UUCP newsgroup system. In that system, every message and subsequent replies are published in exactly one (sub) category (such as `alt.hypertext`), even though the discussion might relate to many categories[2]. The problem with keywords according to Berners-Lee lies in the fact, that "two people never choose the same keyword" (idb.) and that knowing the application is a prerequisite for knowing the right keyword.

The key problem of CERN in terms of information management, according to Berners-Lee, was the high turnover of people at the organisation that resulted in a constant loss of information. He writes "If a CERN experiment were a static once-only development, all the information could be written in a big book" (ibid.). However, since CERN was a highly dynamic environment a "book" and related documentation systems were not appropriate tools for manage the information of the organisation: "The actual observed working structure of the organisation [CERN] is a multiply connected 'web' whose interconnections evolve with time" and "This is why a 'web' of notes [sic!] with links (like references) between them is far more useful than a fixed hierarchical system" (ibid.).

Even though he meant to solve a local problem at CERN, Berners-Lee forboded that CERN's problem could become a global one: "CERN is a model in miniature of the rest of world in a few years time" (ibid.).

[1] the term *keyword* is used here imprecisely to denote a *subject heading* taken from a controlled vocabulary, cf. section 3.5 for a disambiguation

[2] Cross-posting is (still) technically possible, but it is discouraged to avoid duplication

3.2 Nodes and Links

By proposing a system whose nodes and links would better represent the networked nature of CERN than traditional information systems using categories or keywords, Berners-Lee introduced the question of which kinds of nodes and links such a system would have to represent. In his original proposal he gave examples for both nodes and links:

- *Example nodes* [Ber89]
  - People
  - Software modules
  - Groups of people
  - Projects
  - Concepts
  - Documents
  - Types of hardware
  - Specific hardware objects
- *Example links* (ibid.)
  - depends on B
  - is part of B
  - made B
  - refers to B
  - uses B
  - is an example of B

In addition, he gave examples of how data interconnected with "typed links" could be used for automatic data analysis. Using the example of nodes representing people, he writes (ibid.):

> "In a complex place like CERN, it's not always obvious how to divide people into groups. [...] Perhaps a linked information system will allow us to see the real structure of the organisation in which we work." [Ber89]

Reading these lines in the original proposal of what would become the World Wide Web makes us wonder, whey it took applications such as the rather recent on-line social networks so long to emerge (cf. section 7.1).

3.3 Link Types

When Tim Berners-Lee itemized several link types in his original proposal for the WWW (cf. section 3.2), he did not mean to define them, he merely exemplified the concept of a link. Likewise [HTML1] only defined the tag `A` with the attribute `TYPE` and the definition:

> "An attribute TYPE may give the relationship described by the hyertext [sic!] link. The type is expressed by a string for extensibility. Strings for types with particular semantics will be registered by the W3 team. The default relationship if none other is given is void." [HTML1]

It took the WWW community some time to codify which different link types were to be supported. In a lengthy discussion of [HTML3.0] pertaining to this topic, Craig Hubley summed up the difficulties of the community to define link meanings by writing to the

HTML-WG mailing list: "future generations will curse our names for assuming that we understood *anything* about link semantics back in '95" [HTML-WG-95].

First proposed in [HTML3.2], [HTML4.01] codified a list of supported link types, that "User agents, search engines, etc. may interpret ... in a variety of ways" [HTML4.01-types]:

- Alternate
- Appendix
- Bookmark
- Chapter
- Contents
- Copyright
- Glossary
- Help
- Index
- Next
- Prev
- Section
- Start
- Stylesheet
- Subsection

Notably, those values were no longer defined for the `TYPE` attribute (since then used for content types), but they were declared valid for the `REL` (forward link types) and `REV` (reverse link types) attributes of the `A` and `LINK` elements. The link types of [HTML4.01] became among others the initial content of the registry of relation types for Web links defined in [RFC5988]. This registry seeks to register relation types independently of HTML (cf. section 6.4) and requires new relation types to be requested by sending an email to `link-relations@ietf.org`[3]. Link relation types are defined by many other RFC, preceding and succeeding [RFC5988]. A complete list with reference to the respective RFC and other sources is maintained in [IANA-link-types].

The [HTML5] recommendation defines supported link types as part of the HTML vocabulary in section [HTML5-links] (~~deletions~~ and **additions** with respect to [HTML4.01]):

- Alternate
- ~~Appendix~~
- **Author**
- Bookmark
- ~~Chapter~~
- ~~Contents~~
- ~~Copyright~~
- ~~Glossary~~
- Help
- **Icon**
- ~~Index~~
- **Licence**
- Next
- **Nofollow**
- **Noreferrer**
- **Prefetch**
- Prev
- **Search**
- ~~Section~~
- ~~Start~~
- Stylesheet
- ~~Subsection~~
- **Tag**

Interestingly, the removal of `SECTION` and `SUBSECTION` as valid values for the `REL` attributes coincides with the introduction of the `SECTION` element to the [HTML5] vocabulary. [HTML4.01] defined those relations as "Refers to a document serving as a [(sub)]section in a collection of documents". While the removal might have been motivated by a completely unrelated reason (possibly lack of adoption), it does illustrate the difficulty to define what constitutes a *document* on the Web (cf. section 2.2).

Likewise, the addition of value `AUTHOR` illustrates the unsettled relationship of link types to meta data: while [HTML5] supports `<META NAME="AUTHOR"` it explains the `META`

[3] see `http://www.ietf.org/mail-archive/web/link-relations/current/maillist.html` for link-relations archive

element as representing "various kinds of metadata that *cannot* be expressed using the title, base, link, style, and script elements." (ibid., emphasis by the author).

Unlike earlier version, [HTML5] does not support the `REV` attribute on `A` or `LINK` elements anymore, which was meant to express a reverse link type. Having `<LINK REL="author" href="book.htm">` as a forward link type, we would use `<LINK REV="made" href="author.htm">` as the corresponding reverse link type. Reverse links and their importance are discussed in section 6.2, the role of the `REV` attribute is revisited in section 6.3.

It is the value of the `REL` attribute, that defines whether the `A`, `AREA` or `LINK` elements create a *link to an external resource* or a *hyperlink*. [HTML5] provides the following definitions to clarify the difference:

> "Links to external resources: These are links to resources that are to be used to augment the current document, generally automatically processed by the user agent.
>
> Hyperlinks: These are links to other resources that are generally exposed to the user by the user agent so that the user can cause the user agent to navigate to those resources, e.g. to visit them in a browser or download them." [HTML5-links]

According to the specification, `A` and `AREA` elements *without* a `REL` attribute but with a `HREF` attribute must also create a hyperlink, the *implied hyperlink*. That is the default hyperlink of the form `<A HREF="">` that "has no special meaning" (ibid.).

Of the [HTML5] link types, only `ICON`, `PREFETCH` and `STYLESHEET` create *links to external resources*, `NOFOLLOW` and `NOREFERRER` do not create links but *annotate* links created by other values of the same `REL` attribute (i.e. the implied hyperlink, if there is no other value). All other link types create *hyperlinks*.

At the end of [HTML5-links], a reference is made to the microformats wiki page "HTML5 link type extensions" [MFREL] where extensions to the defined link types can be registered - by anyone! This wiki page currently defines dozens of link types. Most remarkably, it is referred to not only by the [HTML5] specification of the World Wide Web Consortium (W3C), but also by the HTML specification of the Hypertext Application Technology Working Group (WHATWG). The WHATWG was founded in 2004 out of dissatisfaction with the W3C and its alleged lack of interest in HTML; today the working group maintains HTML as a "living standard" (cf. [HTML-living-standard]) opposing the W3C's normative recommendations with "frozen" revisions (cf. [HTML5]). By way of reference to the microformats wiki page, that is a "living standard" as well, the competing W3C and WHATWG seem to be in agreement, at least as far as the need for non-normative sources in the area of link types is concerned.

3.4 The Networked Nature of the Web

Even though Tim Berners-Lee's motivation to propose a linked information system for information management at CERN was motivated by the alleged insufficiency of meta data (cf. section 3.1), the first attempts for information retrieval on the WWW were undertaken using this very method: meta data.

The advent of the Web, however, "brought into play millions of new users without any training in cataloguing, indexing and classification" [Vel01] that could publish documents and link to other documents ad libitum and that quickly outnumbered a "small elite" (ibid.) who was trained to work with the aforementioned techniques.

More importantly, the traditional methods of information science simply do not match the very nature of a distributed linked information system. First of all, what is to be considered a *documentary reference unit* (DRU) (cf. section 2.2) on the World Wide Web? Is it a Web site (www.example.com), a part of that Web site (www.example.com/part/) or an individual page within that part (www.example.com/part/page.htm)?

Even on a page level, we have not reached the smallest possible unit of reference, for that page can include other image files using the `IMG` element or link to external resources (cf. section 3.3), that for their part can do the same (ad infinitum). Thus even if the "small elite" manages to classify a site, a part or a page thereof, what would be the range of this classification?

By way of example, if a librarian assigned an author to a Web site using the `<META NAME="AUTHOR">` at one level, we have trouble to define the applicability (or transitivity) of this annotation to subordinated levels, for we know, that unlike a tree structure, on the Web there is no leaf node, where this transitivity could end. It is important to note that this problem persists independently of the chosen URL design, i.e. changing the above to www.example.com/part.htm and www.example.com/page.htm does not help.

The Open Archives Initiative's (OAI) standard "Object Reuse and Exchange" [OAI-ORE] addresses this difficulty of defining the level of reference as the *aggregation problem* (ibid.). Rather than classifying the URL of a document it promotes an aggregate as an independent concept on the web. This aggregate would define its range by listing all URLs it applies to, and meta data would be assigned only on the aggregate level. However, the problem of the undefined range of an individual URL (now listed as part of the aggregate) persists.

3.5 Relevance and Mixed Motives

In the early years of the Web, another challenge quickly became apparent. Even if one eventually succeeded in making the Web searchable, the list of matching Web sites would be way to long to be scanned by a human being for relevant hits. Put another way, the mere ability to search millions of Web sites would become useless, if this search returned a million unsorted results.

Now relevance is what cataloguing, indexing and classification are striving for by intellectually determining the "aboutness" (cf. [SS13], p. 519) of a document. Whereas the mere *occurrence of the word*[4] "Greece" in a given document does *not* imply that this document is about "Greece" (think of phrases such as "unlike Greece", "not Greece", or other irrelevant references), the assignment of the subject heading "Greece" (or a classification code, respectively) to this document does establish this aboutness.

Notwithstanding the problem of the undefined range of an URL (cf. section 3.4) and contrary to the original motivation of Berners-Lee (cf. section 3.1), the assignment of subject headings to a HTML document was supported since [HTML2.0] by means of the `<META NAME="keywords">` (meaning subject heading) tag. Early search engines looking for "Greece" considered a document containing `<META NAME="keywords" CONTENT="Greece">` significantly more relevant than a document containing `<BODY>...Greece...</BODY>`.

Knowing of this simple way to increase the relevance of one's own Web site, meta tags were quickly abused. Either by duplicating the relevant subject heading (`<META NAME="keywords" CONTENT="Greece, Greece, Greece, Greece, ...">`) or by deliberately using popular subject headings that one's own site had nothing to do with, but that would lure a great number of users to visit one's site. This abuse became a legal issue, when companies started to use the names of their respective (probably more popular) competitors as content of the `<META>` tag. How worse this practice dubbed "spamdexing" got, is described and substantiated with references to actual search engine results at that time by Ira S. Nathenson in 1998:

> "Suppose, for example, you use an Internet search engine to look for Web pages on the late Princess Diana. Instead, you may find get-rich schemes and pornography. Or you search for an attorney – by name – and instead get Internet service providers ("ISPs") and software companies. If you search for pages on "Monica Lewinsky," you might be shocked to find that the top listing from one search engine is "CityAuction," an Internet classified advertising site." [Nat98]

Nathenson discussed "spamdexing" from a legal perspective. Hence he was primarily concerned with the use of registered trademarks such as "Playboy" or "Playmate" in meta tags by others than the trademark owner. However, he also identified the invisibility of meta tags as a key reason for their abuse: "Because keyword meta tags are invisible to those browsing the Internet, many webmasters are tempted to use keywords that are irrelevant (or remotely related) to actual Web page content." (ibid.). Thus most users just ended up being disappointed with irrelevant Web sites a search engine provided them with, but they would not necessarily accuse the Web sites' owners of potentially illegal spamdexing; unless they were familiar with the particularities of HTML.

[4] the true sense of the term *keyword*

3.6 Relevance by Reference

In that same year (1998), another even more important reason for the failure of the meta tag was identified: the owner or author of a Web site is simply the wrong person to evaluate his own site by annotating it with meta tags of assumed relevance, for "Many web page authors would simply claim that their pages were all the best and most used on the web" [Pag+99]. Thus in order to determine the aboutness and relevance of a given Web site, one should not look at what that site says about itself, but what *other sites* say about it. A new search engine whose inner workings were motivated by this analysis was born: Google.

The analogy that Lawrence Page et al. used to explain their new approach is the practice of academic citation. The relevance of an academic at their (and our) time was measured by the number of publications citing that academic. Applied to the Web, "One can simply think of every link as being like an academic citation" ([Pag+99], p. 2). Accordingly, a Web site that many *other sites* link to would be considered more relevant than another one that was "cited" less.

In order to determine the aboutness of a site, Page et al. refrained from using that site's meta tags (*what the site says about itself*), but they used the anchor text of other sites' links to that site (*what other sites say about it*). Thus when site www.example.com places the hyperlink `<A HREF="http://www.example.org">Greece</A>` it does not only increase the relevance of www.example.org by "citing" it, but at the same time it establishes its aboutness as being "Greece".[5]In that way, it became also possible to say something *about* images that could not have been annotated with meta tags in the first place due to the lack of markup: `<A HREF="holiday.jpg">Greece</A>`.

At last, Google did not stop entirely to look at a Web site's own content, i.e. its full text. To improve search results, the proximity of keywords matching the search was taken into consideration. Thus among the search results for "greek food" a Web site containing `greek food` would be considered more relevant than a Web site containing `I bought some food before I went to Greek class` because of the higher proximity of words in the former site. Likewise "font information" [BP98] was used to assign typographically emphasized (for example bold) text a higher relevance than text with a normal font weight.

As plausible the combination of link count, anchor text, word proximity and font information might be to determine relevance and aboutness, as difficult is the implementation. First of all, where to begin? Again, the web is not a tree structure, so there is not a *root node* that one could start with to explore the structure of the Web. In addition, since all links of the Web are *forward* links (linking *from* one site *to* another but not back), how would one be able to calculate the total link count to a Web site?

Sergey Brin and Lawrence Page explain their proposal for exploring the Web intuitively by assuming a "random surfer" who starts browsing the Web on a randomly assigned Web page and who explores the Web from there by keeping clicking on links without ever going

[5]That's why *Don't use "click here" as link text*, `http://www.w3.org/QA/Tips/noClickHere`

back, until he starts over again at another randomly assigned Web page. "The probability that the random surfer visits a page is its PageRank" ([BP98], p. 110), a single number that indicates the position in "a global ranking of all web pages, regardless of their content, based solely on their location in the Web's graph structure" ([Pag+99], p. 15).

The "random surfer" is automated by means of Web crawlers who create an index of URLs as they crawl the Web. Each URL is converted into a unique integer that is stored in a database along with the Parent ID of the URL pointing to it. The PageRank of a each URL, i.e. its position of the global ranking of all web pages, is calculated iteratively until the process converges. Convergence is possible, for the PageRank calculation takes place in a downloaded finite sample of the Web (24 million pages at that time) and "dangling links", i.e. pages with not outgoing links, are disregarded until all the PageRanks are calculated (cf. [Pag+99], p. 6).

Contrary to the intuitive explanation of the relevance of a Web page as being established by its "citation count", PageRank is supposed to reflect the *probability* that the "random surfer" (implemented by a Web crawler) visits that Web page. Thus for a Web page u to gain a high relevance, it should suffice to have only *one* incoming link from another page v that in turn has many incoming links, for u's probability to be visited by the "random surfer" depends on the probability of the incoming page v to be visited. Since u may have many incoming pages v_1, v_2, v_n it is the *sum* of the averaged PageRanks of the incoming pages that determines the PageRank of u. Formalized and simplified, with B_u being the set of pages that point to u, F_v being the set of pages v points to (among them u), $N_v = |F_v|$, and c being a factor to ensure that the sum of all ranks of all Web pages is 1, the PageRank R of u is calculated as follows ([Pag+99], p. 3):

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

Applied to the example Web pages u_1, u_2, u_3 and v_1, v_2, v_3 shown in figure 3.1 we would calculate the page rank of Web page $R(u_3)$ as follows (ignoring c):

$$F(v_1) = \{u_1, u_2, u_3\}$$
$$N_{v_1} = |F_{v_1}| = 3$$
$$\textit{calculate } N_{v_2} \textit{ and } N_{v_3} \textit{ accordingly}$$

$$B(u_3) = \{v_1, v_2, v_3\}$$
$$R(u_3) = R(v_1)/N_{v_1} + R(v_2)/N_{v_2} + R(v_3)/N_{v_3}$$
$$R(u_3) = R(v_1)/3 + R(v_2)/1 + R(v_3)/2$$

Since Google's invention was motivated by the vulnerability of other search engines at that time to (commercial) manipulation (cf. section 3.5), the benefit of this calculation of

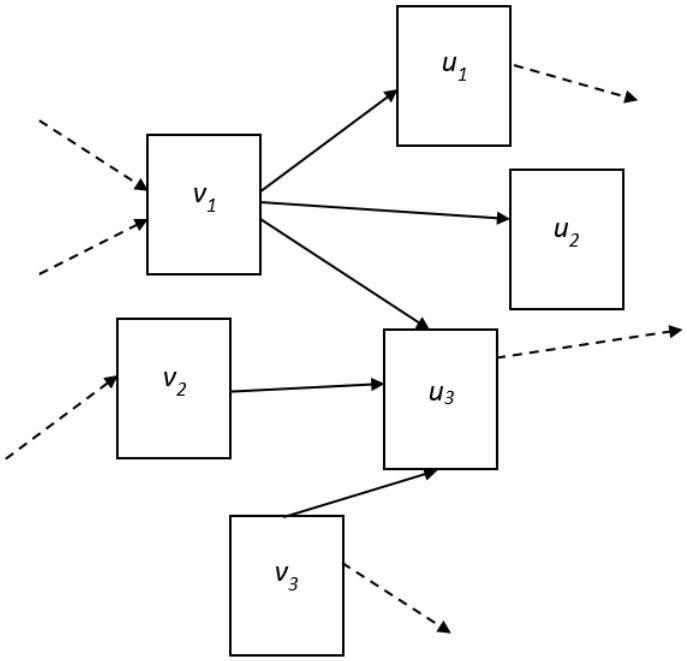

Figure 3.1: Web page u_3 and its position in the Web graph

relevance was seen in its immunity to manipulation, "[f]or a page to get a high PageRank, it must convince an important page, or a lot of non-important pages to link to it" (ibid., p. 12). Page et al. concluded their paper with the humble statement, that "the structure of the Web graph is very useful for a variety of information retrieval tasks" (ibid., p. 15).

Chapter 4

Linking on the Web of Data

4.1 Beyond a Web for Humans

When Google manages to return relevant results to *keyword*-based inquiries of its *human* users, this relevance does not arise from Google understanding the content of the Web or the inquiries of its users. Google is "just" good at matching man-made content (a *keyword* on `http://www.example.com`) with man-made keywords referring to that Web site (`<A HREF="http://www.example.com">`*`keyword`*`</A>`) with a search string (`https://www.google.de/#q=`*`keyword`*) entered by a human user.

How is it possible that Google finds Web sites whose *content* is likely to be considered *meaningful* by a human user? How is this achieved by sorting the list of matching Web sites according to their PageRank, that is calculated "regardless of their content" ([Pag+99], p. 15) based on counting links with "no special meaning" [HTML5-links]? Why does Google's keyword-based approach work so well in a networked information system whose invention was motivated by the insufficiency of keyword-based systems, that assumingly failed because "two people never choose the same keyword" [Ber89]?

The success of Google's approach is to be sought in its "humanness". Google embraces the fact that the Web is made and used by human beings that act human. If in the real world a human believes in the reputation of somebody else because many others refer to that somebody as being reputable, he will trust a Web site in the on-line world, that many others refer to. In addition, a human user is neither interested in finding all relevant pages on the Web nor capable of reading all matching results. If an average keyword search yields millions of Web sites, the exclusion of thousands of Web sites because of the user's keyword choice is simply not an issue. Furthermore, the human user is still doing a final evaluation of the search results Google presents him with, and he is ready to accept some irrelevant matches as long as there is one relevant match on the first page of results.

Tim Berners-Lee's vision of the web, however, is not confined to human users, but he envisions a "Semantic Web" with "a new form of Web content that is meaningful to computers" [BHL01] allowing software agents to perform tasks on the Web automatically on behalf of humans. The biggest hindrance on a way to the *Semantic Web* was early

identified as a "lack of semantic markup" [HF99]. Thereby one refers to the limited set of elements of the HTML vocabulary that is used to publish content on the "Web for humans". In order to make that content suitable for a "Web for agents", one would have to use a much richer mark-up than that of HTML.

In [HF99] Frank van Harmelen and Dieter Fensel review the different approaches discussed at that time to make Web content meaningful for computers. One approach uses the existing vocabulary of HTML with semantic values for either the `NAME` attribute of the `META` element – `<META NAME="author" CONTENT="Frank">`, or for the `CLASS` attribute of the `SPAN` element – `<SPAN CLASS="author">Frank</SPAN>`, `location` and `tel` being other examples. Another approach favours XML vocabularies to mark-up content on the Web semantically, which results in the definition of new tags such as `<AUTHOR>Frank</AUTHOR>`. A third approach uses a new data model called RDF to encode simple facts on the Web independent of the structure of the document, in the syntax of that time: `AUTHOR(http://www.cs.vu.nl/~frankh)` = Frank.

As unclear the question was at that time which approach—HTML, XML or RDF—would be chosen to mark-up Web content for computers, as clearly James Hendler pointed out in 2001 what would be the major challenge when creating a semantic web using any of those approaches:

> "However, for this vision to become a reality, a phenomenon similar to the Web's early days must occur. Web users will not mark up their Web pages unless they perceive value in doing so, and tools to demonstrate this value will not be developed unless Web resources are marked up." [Hen01], p. 31

4.2 Linked Data

Brin's and Page's proposal to use link counts as a measure for relevance in order to improve search results for human users on the existing Web, and Tim Berners-Lee's proposal to advance the Web to make it a better place for agents, both date back to the late 1990s. However, whereas in 2015, *googling* has become a synonym for searching on the Web, a *semantic* Web still seems to be futuristic. So where are we today?

The current state of affairs of semantic web efforts is often referred to as "linked data" [Ber06], and this catch-cry fittingly describes an approach targeting at a new level of granularity on the Web. While on the existing Web everyone may publish documents and may arbitrarily link from one document to another, *linked data* strives for doing the same on a data or *fact* level.

The idea is fascinating: if we agree on a standard to publish simple *facts*—rather than just documents—on the Web, and if we allow everyone to publish facts and arbitrarily link from one fact to another, we will get a *Web of data*—rather than just a *Web of documents*. Likewise, given that the standard way of publishing *facts* on the Web is *meaningful to computers*, automatic agents could harvest the Web for facts, rather than

for documents.

Let us consider a simple example: say Web page D_1 states the fact F_1: Olympia is located in Greece . *Another* Web page D_2 states the fact F_2: Olympia has-primary-deity Zeus. An agent crawling the Web of data looking for "Places in Greece with primary deity Zeus" and finding both Web pages could *link* both facts together and include Olympia in its results.

There are a couple of prerequisites for this to work. Let us consider them one by one. As mentioned earlier, we will have to agree on a standard way to encode *facts*. From the above examples we can derive the general structure of a fact consisting of *subject*, *predicate*, and *object*, in that order. While Frank van Harmelen and Dieter Fensel still discussed the then emerging RDF as a data model for metadata[1], in the course of time RDF developed into a data model that does exactly that: encoding facts using three elements in the given order.

RDF is often mistaken for a data *format*, but it is important to note that it is merely a data *model*. In mathematical terms, an RDF fact or statement is a finite sequence (an ordered list) of 3 elements (*subject*, *predicate*, *object*), also called a 3-tuple or *triple*. There are many *formats* that one can use to *serialize* the same RDF triple. The first RDF format standardized in [RDF/XML-1999] and registered as Internet media type `application/rdf+xml` [RFC3023] was an XML serialization. There are, however, other non-XML serializations that are often preferred to XML because of their better human-readability, among them the popular turtle format [RDF/TURTLE1.1]. Recently, a serialization in JSON [RDF/JSON1.1] has been proposed.

Unfortunately, there are many issues with the registration of non-XML RDF serializations as Internet media standards (cf. [Pru08]), which certainly contributed to a wide-spread confusion as to the difference between one data model (RDF) and its many serializations (e.g. RDF/XML, RDF/TURTLE, RDF/JSON). For the following sections it is important to bear in mind that, regardless of the RDF serialization or format used, the data model is always the same.

In order to combine the above facts F_1 and F_2, however, it is not enough to merely publish them in an RDF serialization on the Web. In addition, F_1 and F_2 have to make sure that at least one element in both facts can be recognized by an agent as being "the same" such that both facts can be linked to each other (hence *linked data*). Using the given example, one such element would be *Olympia*.

Now, sameness can be established for at least two different things: a *name* or a *concept*. It is tempting to assume that it suffices for F_1 and F_2 to use the same *name*, e.g. the English literal `"Olympia"`. However, this would deprive us of the possibility of expressing that one *concept* has many names, and of linking back from those many names to the same concept. Speaking of the *concept Olympia* (a place in Greece) we could be using names other than the English literal `"Oympia"`, such as the official Greek name "$Αρχαια\ Ολυμπια$".

To complicate matters, the same literal might be in use as a name for different concepts.

[1] RDF stands for *Resource Description Framework*, which still indicates the historical roots

Olympia, in addition to being a name of a geographic place in Greece, it is the name of the capital of Washington (the state) as well as a name of a global sports event. Hence it is considered favourable to establish an independent "name-less" concept that the above literals can be associated with as names or labels. Consequently, links among facts would be created among those "name-less" concepts rather than among names.

Given, we agree on a identifier for *Olympia* (a place in Greece) and given, both facts F_1 and F_2 use that identifier, we remain with the question how the linking of both facts would take place. Put another way, how would an agent harvesting RDF triples on the Web and finding F_1 on one Web page A and F_2 on another Web page B be able to recognize that both facts are referring to the same concept and hence can be linked together?

Obviously there is another prerequisite for the above inference to work: triples harvested from many different Web pages must be stored in one database where they can be linked together and queried for. We can think of such a database as a "bucket of triples" [SEMANTIC-WEB-14] harvested from the Web, and regardless of the technological implementation, we will refer to it as a *triple store*. Unlike an index of the textual web, such a triple store does strive for creating a full-text index of web pages, but it merely stores harvested facts in the format *subject*, *predicate*, *object*. Let us summarize what are the prerequisites for the query "Places in Greece with primary deity Zeus" to be answered by an agent:

1. Facts are published in machine-processable semantics (cf. section 4.3).
2. Facts use the same identifiers for the same concepts (cf. section 4.4, 4.5, 4.6).
3. Agents harvest, store and consolidate facts in a triple store (cf. section 4.7).

4.3 Publishing Facts on the Web

When Frank van Harmelen and Dieter Fensel compared different ways of publishing "machine-processable semantics" [HF99] or facts on the Web, they discussed HTML-based semantic markup, XML, and RDF (cf. section 4.1). At that time, XML had just been ratified as a W3C recommendation [XML1.0] and RDF had yet to become one. Nevertheless, in their conclusion they were very decided in their assessment of XML and RDF for knowledge representation on the Web:

> "One of the surprises to us when writing this paper was that the HTML `SPAN`-mechanism already provided much of the functionality now so loudly advertised for XML. Furthermore, it is rather disappointing to see that RDF ignores a few basic lessons in language design." [HF99]

Their lack of enthusiasm for the "so loudly advertised" (ibid.) XML stems from the similarity of the `<SPAN CLASS="author">Frank</SPAN>` (HTML) and the `<AUTHOR>Frank</AUTHOR>` (XML) approach when it comes to adding semantics to a Web page. Both approaches allow a user to define his own semantic labels (attributes in HTML,

Listing 4.1: Redundancy and Inconsistency between data and metadata [HF99]

```
<HEAD>
    <META NAME="AUTHOR" CONTENT="FRANK">
</HEAD>
<BODY>
    This page was written by Dieter.
</BODY>
```

Listing 4.2: HTML-based semantic markup using "HTML <SPAN>-mechanism" [HF99]

```
<body>
    This page is written by
    <span class="author">Frank van Harmelen</span>.
    <span class="location">
        His tel.nr. is <span class="tel">47731</span>,
        room nr. <span class="room">T3.57</span>
    </span>
</body>
```

elements in XML), and both result in the same data structure, namely a semantically labelled tree. Furthermore, in both HTML and XML semantic information is a "by-product of defining the structure of the document", such that "structure and semantics of document are interwoven" (ibid.).

RDF is different from both approaches insofar as it is *not* interwoven with the document structure, for RDF statements about a document are made independently of the document's data structure on its own meta level. According to the authors, this necessarily results in redundancy and potentially inconsistency. This criticism is not targeted at RDF in particular, but at meta-data in general. To illustrate their point, the authors give the example in listing 4.1. The same information—author of the page—is codified twice (redundancy) in a conflicting manner (inconsistency). Put another way, RDF gains its advantage of decoupling the structure of meta-data from the structure of the document at the price of *duplicating* the same information.

HTML-based semantic markup

Sixteen years after Frank van Harmelen and Dieter Fensel reviewed the different ways for knowledge representation on the Web, one notices that all approaches still exist. The "HTML <SPAN>-mechanism" has been implemented by several initiatives in only slightly different ways. Compare van Harmelen's and Fensel's original example (cf. listing 4.2) with the same semantics expressed in [microformats2] (cf. listing 4.3), [microdata] (cf. listing 4.4), and [RDFa] (cf. listing 4.5).

The present-day approaches of embedding semantics into HTML (cf. listing 4.3–4.5) are similar in that they distinguish between the *type* of a described object and its *properties*. *microformats2* uses prefixes of `class` values for this purpose, whereas microdata and RDFa define their own attributes to achieve the same (cf. table 4.1). There is, however, a significant difference between microformats2 on the one hand side, and microdata and RDFa on the other. *microformats2* does not only define a format, but in addition it defines

Listing 4.3: HTML-based semantic markup using microformats2

```
<body class="h-entry">
    This page is written by
    <div class="p-author␣h-card">
        <span class="p-name">Frank van Harmelen</span>.
        His tel.nr. is <span class="p-tel">47731</span>,
        room nr. <span class="p-extended-address">T3.57</span>
    </div>
</body>
```

Listing 4.4: HTML-based semantic markup using microdata

```
<body itemprop="author">
    This page is written by
    <div itemtype="http://schema.org/Person">
        <span itemprop="name">Frank van Harmelen</span>.
        His tel.nr. is <span itemprop="telephone">47731</span>,
        <!-- room is not a registered property of type Person -->
        room nr. <span itemprop="room">T3.57</span>
    </div>
</body>
```

Listing 4.5: HTML-based semantic markup using RDFa

```
<body prefix="schema:␣http://schema.org/">
    This page is written by
    <div property="schema:author" typeof="schema:Person">
        <span property="schema:name">Frank van Harmelen</span>.
        His tel.nr. is <span property="schema:telephone">47731</span>,
        <!-- room is not a registered property of type Person -->
        room nr. <span property="schema:room">T3.57</span>
    </div>
</body>
```

	Type indicator	Property indicator
microformat2	`h-` prefix for `class` values	`p-` prefix for `class` values
microdata	`itemtype` attribute	`itemprop` attribute
RDFa	`typeof` attribute	`property` attribute

Table 4.1: Comparison of Type and Property Indicators

Listing 4.6: Semantics in RDF/TURTLE — long version

```
<http://www.cs.vu.nl/~frankh> <http://schema.org/author> _:frank .
_:frank a <http://schema.org/Person> .
_:frank <http://schema.org/name> "Frank␣van␣Harmelen" .
_:frank <http://schema.org/telephone> "47731" .
# 'room' is not a registered property of schema.org
_:frank <http://schema.org/room> "T3.57" .
```

its own vocabulary in the form of predefined string values for the `class` attribute. Hence by using microformats2, one is automatically limited to the vocabulary that comes "hard-wired" with the language. On the contrary, microdata and RDFa do not define their own vocabulary, but they refer to vocabularies defined independently of the respective standard. In our example, both listing 4.4 and 4.5 make use of the same [schema.org2.0] vocabulary, that defines the type `Person` and the properties `name` and `telephone` (cf. section 4.6 for more on vocabularies). [schema.org2.0] does not define a `room` property, but it has been proposed [SCHEMAORG-I-545].

RDF triples

Since listing 4.4 and 4.5 use the same vocabulary, we can express the semantics embedded in both with the same RDF statements. For readability reasons, we use the RDF/TURTLE serialization (cf. section 4.2), that comes in a long (cf. listing 4.6) and compact version (cf. listing 4.7).

As discussed in section 4.2, RDF is a data model to represent statements in the form of *subject*, *predicate*, and *object*. Formally, "[a]sserting an RDF triple says that some relationship, indicated by the predicate, holds between the resources denoted by the subject and object." [RDF1.1], whereas *subject* and *object* can be denoted by three means:

- IRI, i.e. an URL,
- literal, i.e. a string, or
- blank node, i.e. a local variable.

Listing 4.7: Semantics in RDF/TURTLE — compact version

```
@prefix schema: <http://schema.org/> .

<http://www.cs.vu.nl/~frankh> schema:author [ a schema:Person;
    schema:name "Frank␣van␣Harmelen";
    schema:telephone "47731";
    # 'room' is not a registered property of schema.org
    schema:room "T3.57" ] .
```

Considering listing 4.6, line 1, we see that the subject is denoted by an URL (`<http://www.cs.vu.nl/~frankh>`), and the object is denoted by a local variable (`_:frank`). The predicate, connecting subject and predicate, is denoted by an URL (`<http://schema.org/author>`) too. An RDF predicate can also be referred to as *property*, and it is because of the explicit denotion of a property in microformats2, microdata and RDFa (cf. table 4.1) that RDF statements can be derived from them. The subject of property `<http://schema.org/author>` is not explicitly stated in listing 4.4 or 4.5, and hence it is assumed to be the identifier of the web page (the URL), that the semantic markup is published on. Likewise, the object of the property is not equipped with its own identity, but it is merely declared to be of type `<http://schema.org/Person>` (listing 4.6, line 2) with the properties `<http://schema.org/name>` (line 3), `<http://schema.org/telephone>` (line 4) and `<http://schema.org/room>` (line 6).

Because of the lack of an identity for the person with those properties, a *local identifier* in the form of a *blank node* is used to represent that person. We have arbitrarily decided to call this blank node `_:frank`, a software programme extracting listing 4.6 automatically from listing 4.4 or 4.5 will choose a less mnemonic, but likewise unique name. In both cases, it is important to understand, that blank node identifiers are "always locally scoped to the file or RDF store, and are not persistent or portable identifiers" [RDF1.1]. Hence they cannot be referred to from another RDF file or RDF store.

It is the local variable `_:frank` that glues together the independent RDF statements in listing 4.6 to form a single *RDF graph* rather than 5 disconnected ones. *RDF graphs* are by definition "sets of subject-predicate-object triples" [RDF1.1], which is different from the mathematical definition of a graph. To be correct and to avoid confusion it is therefore important to use "RDF graph" as one term as it is defined in [RDF1.1] and not to speak of an RDF file such as listing 4.6 to be graph. Cf. [SEMANTIC-WEB-14] for a humorous and lucid description of the confusion that results from using the word "graph" without the "RDF" qualifier.

HTML-based semantic markup vs. RDF triples

Since RDF statements (cf. section 4.3) can be derived from HTML-based semantic markup (cf. section 4.3), thanks to the possibility of declaring predicates or *properties* directly in HTML (cf. table 4.1), it is tempting to believe that there is no difference between expressing semantics in the former or the latter. However, since HTML is primarily used to mark up natural language, that may significantly deviate from RDF's strict *subject-predicate-object* order, the possibility of embedding RDF-compatible semantics directly in HTML is limited. Listing 4.8 exemplifies how the word order in natural language may differ from that of RDF triples, a mismatch that prompted Tim Berners-Lee to conclude in the early days of RDF that "[human readable] documents are typically not ripe for RDF conversion anyway." [Ber98b].

Listing 4.8: Limitations of HTML-based semantic markup

```
<!-- note how natural language may deviate from the subject, predicate, object
    order of RDF triples -->
<html>
<body>
    <p>This page is written by Frank and Dieter.</p>
    <p>In case you have any comments, please call telephone 47731 (Frank) or 6921 (
    Dieter).</p>
</body>
</html>
```

4.4 Identifying Things (not) on the Web

One argument of van Harmelen and Fensel in favour of HTML-based semantic markup was that unlike a separate RDF representation it "does at least not necessitate . . . redundancy" [HF99]. However, by embedding semantics into a Web page's HTML one might very well create redundancy, namely when the same semantics apply to many pages.

Consider again our statement `<body>This page is written by Frank van Harmelen. His tel. nr. is 47731</body>`. Even though there is a way to use HTML-based semantics to encode this HTML in a way compatible with RDF (cf. listings 4.4 and 4.5), we would have to repeat the same HTML-based semantics on every page for that these semantics hold true.

In addition, how would we express the fact that two different web pages, possibly residing on different hosts, are written by the *same* Frank van Harmelen? We are touching on the subject of equipping RDF subjects and objects with their own identity and sharing the same identifiers across different Web pages.

Say we wanted to equip the author of this thesis with *one* identity on the Web so that we can refer to that identity from *many* Web pages. In a Web context, it seems obvious to use a URL for that purpose. However, using a URL to unambiguously identify a *person*—a thing that does not exist on the Web—is on principle different from identifying a *description of that person* in the form of a HTML page—a thing that exists on the Web.

Let us consider the URL `http://agnosis.de/r/Markus`. On principle, it can be used for at least four different things [Boo03]:

- The *name*, i.e. simply the string "http://agnosis.de/r/Markus" that happens to conform to the URL syntax
- A *person*, i.e. the human being who wrote this thesis
- A *Web location*, i.e. an endpoint on the Web from which we may retrieve one or many documents. Think of a Web location as being a folder on the Web that may contain one or many files
- A *document instance*, i.e. the result of a request to this URL

If we want to publish facts on the Web in a way meaningful to computers, we have to make sure that a computer can tell from a URL for which of the four purposes above it is used. David Booth discusses two different approaches to allow for this distinction:

	Thing on the Web	Thing not on the Web
Name	`http://agnosis.de/r/Markus`	`thing:http://agnosis.de/r/Markus`
Context	`http://agnosis.de/r/Markus`	`(http://agnosis.de/r/Markus)`

Table 4.2: Distinction of Use of URLs, loosely based on [Boo03]

"Use different names to refer to different things; or [u]se different context to distinguish the different uses, while using the same name." [Boo03].

For example, by using different names, we could distinguish *a thing that does not exist on the Web* (`thing:http://agnosis.de/r/Markus`) from *a thing that exists on the Web* (`http://agnosis.de/r/Markus`). Likewise, we could use the same name for both things, but make the distinction by the syntactic *context*. Cf. table 4.2 for a comparison.

As sensible both approaches are, as problematic the use of URLs for *things that do not exist on the Web* remains, for URLs are one of the few least common denominators that the Web technology stack is based on. Literally every request and response relies on a common understanding of the use of an URL, its syntax and parsing as well as associated operations. Accordingly, it is very difficult to change a pillar of the Web as important as the URL standard after the event of wide-spread adoption.

Nevertheless, as with other Web technologies, the definition of *Universal Resource Identifiers* (URI), of which *Uniform Resource Locators* (URL) are a subset (cf. [RFC3986], section 1.1.3.), evolved over time. Originally introduced in an informational RFC as "A Unifying Syntax for the Expression of Names and Addresses of *Objects on the Network*" ([RFC1630], emphasis by the author), URI were codified in [RFC2396]. The current version [RFC3986] clearly says about a resource (the *R* in URI):

> "A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources. Likewise, abstract concepts can be resources, such as the operators and operands of a mathematical equation, the types of a relationship (e.g., "parent" or "employee"), or numeric values (e.g., zero, one, and infinity)." [RFC3986]

The question of *how* to tell apart the different uses of an URL remains a controversial one. The related discussion is known in the Web community by the ticket numbers under which it has been logged in the ticket system of the W3C's Technical Architecture Group (TAG): [TAG-ISSUE-14] had been opened in 2002 and was re-closed (sic!) in 2012. It was superseded by ticket [TAG-ISSUE-57] that has been opened in 2007 and that is still open to date.

The TAG calls a Web object identified by a URL an *information resource*, as opposed to a real-world object or abstract concept identified by a URL, that is called *any kind of resource*. [RFC3986] defines the term *resource* for the purposes of a URI (see above). For the purposes of HTTP, it was defined in [RFC2616] since 1999 as

> "[A resource is a] network data object or service that can be identified by a

> URI, as defined in section 3.2. Resources may be available in multiple representations (e.g. multiple languages, data formats, size, and resolutions) or vary in other ways." [RFC2616]

Among other considerations, it was this distinction between *one* resource, that may be available in *many* representations, that led to a proposal, to use HTTP and its status codes to distinguish an *information resource* from *any kind of resource*. [TAG-ISSUE-14] was preliminarily closed with the following resolution in 2005

a) "If an "http" resource responds to a GET request with a 2xx response, then the resource identified by that URI is an information resource;
b) If an "http" resource responds to a GET request with a 303 (See Other) response, then the resource identified by that URI could be any resource; [...]" [WWW-TAG-05]

HTTP status code 303 (See Other) was originally not defined with this use in mind. At the time of this proposal the intended usage was explained with "This method exists primarily to allow the output of a POST-activated script to redirect the user agent to a selected resource." [RFC2616] along with the note that "Many pre-HTTP/1.1 user agents do not understand the 303 status" (ibid.).

Nevertheless, the proposal to use a HTTP status code to make the distinction between an *information resource* and *any kind of resource*, exempts the TAG from using different names for different things and from defining all foreseeable syntactic contexts (cf. table 4.2). However, this use of a status code exemplifies that RFC2616 "was written for the "old" Web, lots of situations weren't thought of, and needed clarification." [Not14a].

Consequently, in a 7-year long effort, RFC2616 was completely rewritten and superseded by six new RFC, not in order to change the original RFC, but to clarify its applicability on the present-day web. Surprisingly, the corresponding RFC replacing RFC2616 in the matter of status codes still speaks of the same primary usage. There are nonetheless several amendments to broaden its application but without explicit reference to its use on the Semantic Web:

> "A 303 response to a GET request indicates that the origin server does not have a representation of the target resource that can be transferred by the server over HTTP. However, the Location field value refers to a resource that is descriptive of the target resource, such that making a retrieval request on that other resource might result in a representation that is useful to recipients without implying that it represents the original target resource." [RFC7231]

Thus when doing a REQUEST to an URI that was set up to denote *any kind of resource*, the server returns a 303 response without any representation but with a Location header. That Location header contains again an URI, that—once requested by the client—is the actual representation. Since the data model for representing *any kind of resources* on the Web is RDF (cf. section 4.3), that representation should be in one of RDF's formats.

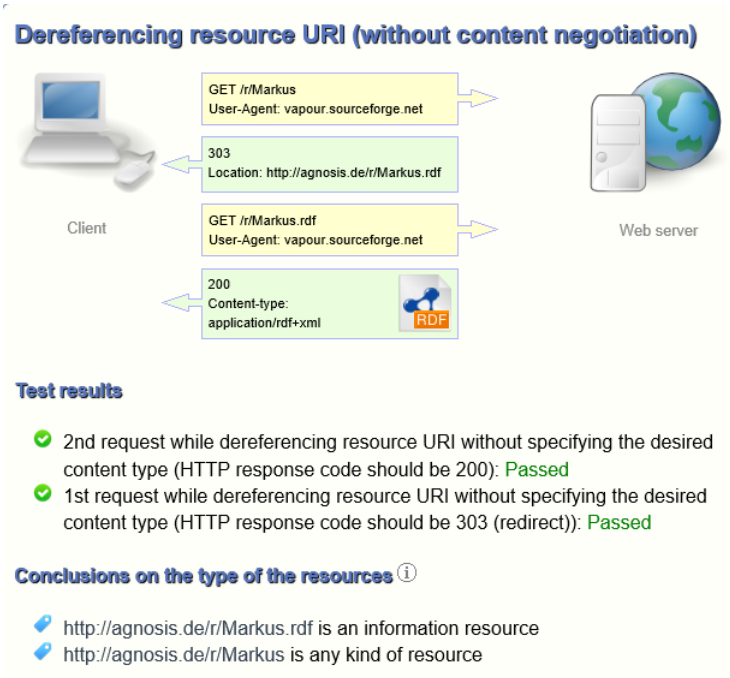

Figure 4.1: Dereferencing *any kind of resource* (1/2) [VAPOUR]

Listing 4.9: Resource `http://agnosis.de/r/Markus` (RDF/XML)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:schema="
    http://schema.org/">
    <rdf:Description rdf:about="http://agnosis.de/r/Markus">
        <rdf:type rdf:resource="http://schema.org/Person" />
        <schema:givenName>Markus</schema:givenName>
        <schema:familyName>Klie</schema:familyName>
        <schema:email>markus.klie@web.de</schema:email>
    </rdf:Description>
</rdf:RDF>
```

There are best practices for publishing *any kind of resource* on the Web, along with tools to test, whether they were published correctly. One such tool is [VAPOUR] and the test results for URI `http://agnosis.de/r/Markus` are shown in figure 4.1 (without content negotiation) and figure 4.2 (requesting RDF/XML).

Content negotiation (cf. [RFC7231]) allows us to request from the same URI many representations of different types. In that way, one can request from the same URI `http://agnosis.de/r/Markus` a representation in RDF/XML (for machines), in HTML (for human beings) or in PNG or JPG (for rubbernecks). The respective header field is called `Accept` [RFC7231]. Using [cURL], one can test the behaviour of the server depending on the `Accept` value provided: compare the server response to the request `curl -H "Accept: application/rdf+xml" http://agnosis.de/r/Markus` with that of the request `curl -H "Accept: text/html" http://agnosis.de/r/Markus`. The dereferenced location of the former request is shown in listing 4.9 (RDF/XML), the one of the latter in listing 4.10 (HTML).

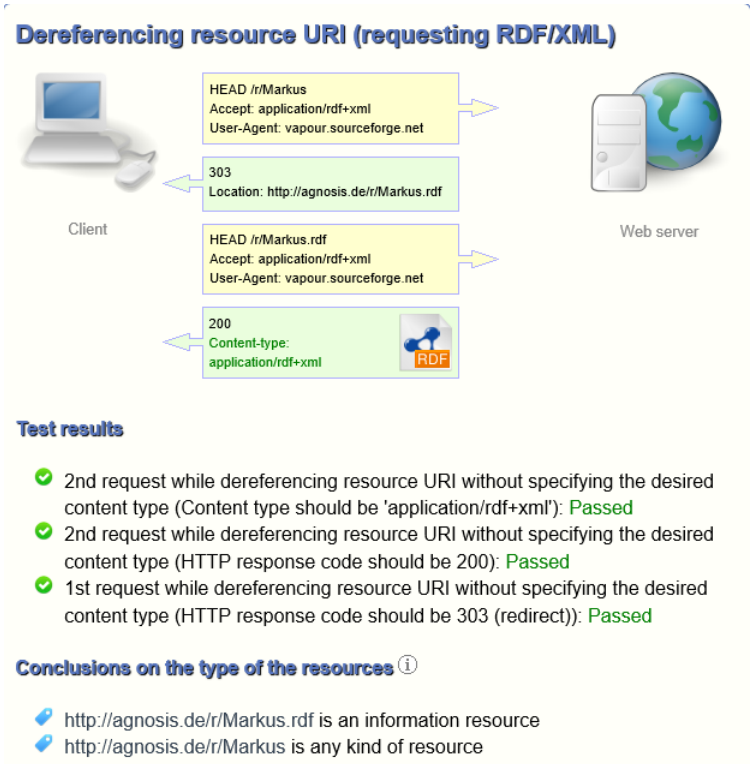

Figure 4.2: Dereferencing *any kind of resource* (2/2) [VAPOUR]

Listing 4.10: Resource `http://agnosis.de/r/Markus` (HTML)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>About me</title>
</head>
<body>
<h1>About me</h1>
<p>My name is Markus Klie an this is my <a href="mailto:markus.klie@web.de">e-mail
    address</a>.</p>
</body>
</html>
```

4.5 Interlinking Data on the Web

Equipping *things not on the Web* with their own Web identity is a prerequisite for the vision of *Linked Data* to become a reality. While on the *Web of documents* everyone may link to any other *information resource*, on the *Web of data* everyone may link to *any kind of resource*. Let us reconsider the example from section 4.2 that shows why the English literal "Olympia" is insufficient for identifying the locality in the country of Greece. Say we were to publish a fact on the Web pertaining to that Greek place, we would first have to look up a URI that unambiguously identifies it.

For disambiguation of the literal "Olympia", the free encyclopedia Wikipedia is a great place. It presents us with many different *things* that go by that name, among them `https://en.wikipedia.org/wiki/Olympia,_Greece` that is about the place in Greece. As a matter of course, this URI identifies one of many *documents* on Wikipedia *about* that place, it does not identify the place itself. However, by means of the [DBpedia3.7] vocabulary URI have been defined for concepts that Wikipedia documents are about, and the one for the locality Olympia in Greece is: `http://dbpedia.org/resource/Olympia,_Greece`.

[VAPOUR] shows that a request to URI `http://dbpedia.org/resource/Olympia,_Greece` returns HTTP status code 303 (See Other) that denotes the resource as *any kind of resource*. Depending on the `Accept` header of our request, it points us to a HTML representation located at `http://dbpedia.org/page/Olympia,_Greece`, or an RDF representation located at `http://dbpedia.org/data/Olympia,_Greece.xml`. Listing 4.11 shows a simplified excerpt of the latter: see how the resource is typed as `http://dbpedia.org/ontology/Place` (line 6), and see how the facts stated use both literals (lines 7–12) and other resources (line 13) as an object (cf. section 4.3).

Now, the *Web of data* allows us to add our own facts by using existing resources published by others. Not stated in listing 4.11 is the fact, that the primary deity of that place is Zeus. Let us claim that fact and add it to the Web of data by publishing the statement shown in listing 4.12. Note that none of the resources involved—neither subject

Listing 4.11: RDF/XML representation for resource `http://dbpedia.org/resource/Olympia,_Greece` (simplified excerpt), published on Web page D_1

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <rdf:RDF
3      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4      xmlns:dbpprop="http://dbpedia.org/property/">
5      <rdf:Description rdf:about="http://dbpedia.org/resource/Olympia,_Greece">
6          <rdf:type rdf:resource="http://dbpedia.org/ontology/Place" />
7          <dbpprop:latDeg>37.638</dbpprop:latDeg>
8          <dbpprop:lonDeg>21.63</dbpprop:lonDeg>
9          <dbpprop:name xml:lang="en">Ancient Olympia</dbpprop:name>
10         <dbpprop:popCommunity>972</dbpprop:popCommunity>
11         <dbpprop:popMunicipality>13409</dbpprop:popMunicipality>
12         <dbpprop:postalCode>27025</dbpprop:postalCode>
13         <dbpprop:stateParty rdf:resource="http://dbpedia.org/resource/Greece" />
14         <!-- [...] -->
15     </rdf:Description>
16 </rdf:RDF>
```

Listing 4.12: Publishing *additional* facts for resource `http://dbpedia.org/resource/Olympia,_Greece` on Web page D_2

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dbpprop="http://dbpedia.org/property/">
    <rdf:Description rdf:about="http://dbpedia.org/resource/Olympia,_Greece">
        <dbpprop:primaryDeityGod rdf:resource="http://dbpedia.org/resource/Zeus" />
    </rdf:Description>
</rdf:RDF>
```

Listing 4.13: Publishing *conflicting* facts for resource `http://dbpedia.org/resource/Olympia,_Greece` on Web page D_3

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dbpprop="http://dbpedia.org/property/">
    <rdf:Description rdf:about="http://dbpedia.org/resource/Olympia,_Greece">
        <dbpprop:stateParty rdf:resource="http://dbpedia.org/resource/Roman_Empire"
     />
    </rdf:Description>
</rdf:RDF>
```

`Olympia,_Greece`, nor the property `primaryDeityGod`, nor the object `Zeus`—is published by us. We merely create a new link among the already existing resources.

Likewise, we may claim facts contradicting the facts published by others. This does not necessarily mean that one publisher is wrong, but the contradiction might stem from a different context. Consider listing 4.13 (line 6) for a fact contracting that of listing 4.12 (line 12).

If we want to say something about the Greek place Olympia on the Web of data, re-using an existing resource of [DBpedia3.7] for that place is an obvious choice. However, it is not the only one. [GeoNames3.1] published a resource for the same place available under `http://sws.geonames.org/264637/`. Requesting an RDF representation from that URI results in listing 4.14. By comparing listing 4.14 with listing 4.11, we—as human beings—can tell that both of them speak of the same thing (the locality of Olympia in Greece) and the same properties (names, geographic coordinates, and population). However, if the idea of *Linked Data* is about "a new form of Web content that is meaningful to computers" [BHL01], we have to look into methods to tell computers that `http://dbpedia.org/page/Olympia,_Greece` is the same as `http://sws.geonames.org/264637/`.

Listing 4.14: RDF/XML representation for resource `http://sws.geonames.org/264637/` (simplified excerpt), published on Web page D_4

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF
    xmlns:gn="http://www.geonames.org/ontology#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:wgs84_pos="http://www.w3.org/2003/01/geo/wgs84_pos#">
    <gn:Feature rdf:about="http://sws.geonames.org/264637/">
        <gn:name>Olympia</gn:name>
        <gn:officialName xml:lang="el">Αρχαια Ολυμπια</gn:officialName>
        <gn:countryCode>GR</gn:countryCode>
        <gn:population>1208</gn:population>
        <wgs84_pos:lat>37.64788</wgs84_pos:lat>
        <wgs84_pos:long>21.6271</wgs84_pos:long>
        <!-- [...] -->
    </gn:Feature>
<rdf:RDF>
```

4.6 Expressing *Sameness* on the Web

Thus far we have referred to RDF as a data model to encode facts as a finite sequence of 3 elements: subject, predicate, and object (cf. section 4.2). By giving examples of RDF triples we have tacitly used three vocabularies: [schema.org2.0], [DBpedia3.7], and [GeoNames3.1]. Those vocabularies are defined using a semantic extension of RDF called RDF Schema [RDFS1.1]. By defining a vocabulary using RDFS one codifies which elements of the vocabulary can act as a predicate—those elements are referred to as *properties*, and which elements can act as a subject or object—those elements are called *classes*.

By convention, the names of vocabulary items that are classes start with an upper-case letter, e.g. `<http://schema.org/Person>`, whereas the names of properties start with a lower-case case, e.g. `<http://schema.org/givenName>`, `<http://schema.org/firstName>`. This reminds us of object-oriented programming languages (OOP), that also distinguish between classes and properties (or *attributes*). However, RDFS is different from OOP insofar as it allows not only classes to be extendible, but also properties. Furthermore, RDFS enforces class-property constraints not by class definition, but by property definition. Thus the definition of the property `<http://schema.org/givenName>` codifies which are valid classes on the subject-side (the *domain*), and which are valid classes on the object-side (the *range*). Cf. table 4.3 for a comparison between OOP and RDFS.

Since both a *property*—an element acting as predicate—and a *class*—an element acting as either subject or object—are extendible, they are in fact both of the same type, namely

	OOP	RDFS
Extendible classes	Yes	Yes
Extendible properties	No	Yes
Class-property constraints	Defined by class	Defined by property

Table 4.3: Comparison of OOP and RDFS

Listing 4.15: RDF(S): Everything is a Resource [RDFS1.1]

```
1  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
2  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
3
4  rdfs:Resource a rdfs:Class ;
5      rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
6      rdfs:label "Resource" ;
7      rdfs:comment "The␣class␣resource,␣everything." .
8
9  rdfs:Class a rdfs:Class ;
10     rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
11     rdfs:label "Class" ;
12     rdfs:comment "The␣class␣of␣classes." ;
13     rdfs:subClassOf rdfs:Resource .
14
15 rdf:Property a rdfs:Class ;
16     rdfs:isDefinedBy <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ;
17     rdfs:label "Property" ;
18     rdfs:comment "The␣class␣of␣RDF␣properties." ;
19     rdfs:subClassOf rdfs:Resource .
```

`rdfs:Class`, and they are both instances of the class `rdfs:Resource`, that in turn is an instance of `rdfs:Class`[2]. Put another way, everything in RDF(S) is of type `rdfs:Class` and every class is an instance of class `rdfs:Resource`. In that way, every element of any vocabulary defined using RDFS can be traced back to one class: "The class resource, everything." (cf. listing 4.15, line 7).

While the use of RDFS for defining RDF vocabularies ensures, that a computer programme reading RDF statements codified in different vocabularies (cf. listing 4.11 and listing 4.14) may successfully recognize *classes* and *properties* and trace them back to common super-types, RDFS does not have a mechanism to express *sameness* between 2 independently defined concepts. Where RDFS fails, a richer language has to be used: The Web Ontology Language (OWL[3]) [OWL2], of which RDFS is only a subset.

Unlike RDFS, OWL is a fully-fledged knowledge representation language, that supports a great variety of language constructs to eventually allow for automatic reasoning. Contrary to RDFS, where one vocabulary item may be a subclass of one class and an instance of another, OWL is more rigid when it comes to instances, classes and statements using them. Apart from being more rigid, OWL is in many ways richer than RDF. To name just one example, it implements features of set theory, by means of which one may define a class as being an intersection or a union of other classes.

However, we are primarily interested in the language construct which allows us to express *sameness* between 2 concepts: `owl:sameAs`. It forms the basis of a mapping tool of the same name: [sameAs] maps one URI of the Web of data to another. Looking up `http://dbpedia.org/resource/Olympia,_Greece` returns—among many others—`http://sws.geonames.org/264637/`. Thus a computer programme encountering the statement in listing 4.16 will be able to understand that the statements in listing 4.11 and the statements in listing 4.14 are, despite the use of different resource URI, in fact

[2] In RDFS the same entity can be both: instance and class.
[3] Why OWL and not WOL? See [WEBONT-WG-01].

Listing 4.16: *Sameness* of resources `http://dbpedia.org/resource/Olympia,_Greece` and `http://sws.geonames.org/264637/`, published on Web page D_5

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .

<http://dbpedia.org/resource/Olympia,_Greece> owl:sameAs
<http://sws.geonames.org/264637/>.
```

about *the same thing*.

4.7 Consolidating and Querying Facts on the Web

Let us assume that each fact pertaining to the locality of Olympia in Greece was published on a different Web page (cf. table 4.4), and let us further assume that an agent crawling the Web for facts finds all pages, harvests the published facts and stores them in one *triple store* (cf. 4.2). The resulting collection of stored facts harvested from Web pages D_1–D_5 is shown in 4.18. As a matter of course, in a real-world triple store, thousands or millions of triples would be stored, and not just the triples involving one particular concept like the locality of Olympia.

Having stored all facts in one place is a prerequisite for being able to query all harvested properties and objects of a given resource. For querying a triple store in a standard way, the W3C published the SPARQL Query Language [SPARQL1.1Query]. Similar to SQL, a SPARQL query consists of a `SELECT` clause and a `WHERE` clause. By means of the WHERE clause one provides a pattern of the form *subject*, *predicate*, *object* that is to be matched against the triple store. For each position in the `WHERE` clause pattern, either a known IRI, literal or variable (denoted by a `?`) can be provided. In the `SELECT` clause one can refer to the variables provided in the `WHERE` clause as "columns" of the result set.

In section 4.2 the prerequisites were outlined that would have to be met, in order for a machine to answer the sample query "Places in Greece with primary deity Zeus" based on facts published on separate Web pages. We have discussed the prerequisites one by one: Publishing facts (section 4.3), identifying *things* (section 4.4), interlinking data (section 4.5) and expressing *sameness* (section 4.6) on the Web. It is now that we can translate the sample query of section 4.2 into SPARQL in order to query our triple store in listing 4.18 for an answer. Cf. listing 4.17 for the SPARQL version of query "Places in Greece with primary deity Zeus". The result of this query—and by now this will not come as a surprise

Listing	Web page	URL
4.11	D_1	`http://dbpedia.org/data/Olympia,_Greece.rdf`
4.12	D_2	`http://www.example.com/d2.rdf`
4.13	D_3	`http://www.example.com/d3.rdf`
4.14	D_4	`http://sws.geonames.org/264637/about.rdf`
4.16	D_5	`http://www.example.com/d5.rdf`

Table 4.4: Facts and their place of publication

Listing 4.17: Retrieving places in Greece with primary deity Zeus (SPARQL Query)

```
SELECT ?subject WHERE
{
    ?subject dbpprop:primaryDeityGod <http://dbpedia.org/resource/Zeus>;
    dbpprop:stateParty <http://dbpedia.org/resource/Greece>
}
```

Listing 4.18: Linked data harvested from Web pages D_1–D_5 (RDF/TURTLE)

```
@base <http://dbpedia.org/> .

@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dbpprop: <http://dbpedia.org/property/> .
@prefix gn: <http://www.geonames.org/ontology#> .
@prefix wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#> .

# Web page D1
<Olympia,_Greece> a <ontology/Place> .
<resource/Olympia,_Greece> dbpprop:latDeg "37.638" .
<resource/Olympia,_Greece> dbpprop:lonDeg "21.63" .
<resource/Olympia,_Greece> dbpprop:name "Ancient␣Olympia"@en .
<resource/Olympia,_Greece> dbpprop:popCommunity "972" .
<resource/Olympia,_Greece> dbpprop:popMunicipality "13409" .
<resource/Olympia,_Greece> dbpprop:postalCode "27025" .
<resource/Olympia,_Greece> dbpprop:stateParty <resource/Greece> .

# Web page D2
<resource/Olympia,_Greece> dbpprop:primaryDeityGod <resource/Zeus> .

# Web page D3
<resource/Olympia,_Greece> dbpprop:stateParty <resource/Roman_Empire> .

# Web page D4
<http://sws.geonames.org/264637/> a gn:Feature .
<http://sws.geonames.org/264637/> gn:name "Olympia" .
<http://sws.geonames.org/264637/> gn:officialName "Αρχαια Ολυμπια" .
<http://sws.geonames.org/264637/> gn:countryCode "GR" .
<http://sws.geonames.org/264637/> gn:population "1208" .
<http://sws.geonames.org/264637/> wgs84_pos:lat "37.64788" .
<http://sws.geonames.org/264637/> wgs84_pos:long "21.6271" .

# Web page D5
<resource/Olympia,_Greece> owl:sameAs <http://sws.geonames.org/264637/> .
```

to the reader—is `subject:` `http://dbpedia.org/resource/Olympia,_Greece`.

Chapter 5

The Missing Reverse Link

5.1 The Repaired Web of Documents

In the early 1990s, Tim Berners-Lee was still musing on the pros and cons of implementing the Web as a bidirectional or unidirectional hypertext system. He saw the benefits of bidirectional linking in the reversibility of relationships and the facilitation of data management (cf. [Ber90b]); but at the same time he pointed out that any enforcement of bidirectional relationships "might constrain the author of a hypertext" (ibid.), since the author would have to create for every forward link a corresponding reverse link. He added, that "a system in which different parts of the web have different capabilities cannot insist on bidirectional links" [Ber90a], and gives the following example:

> "Imagine, for example the publisher of a large and famous book to which many people refer but who has no interest in maintaining his end of their links or indeed in knowing who has refered to the book. In this case the link may be only of use to the person who made it." (ibid.)

That is why he considered the *automatic* creation of reverse and, in that way, bidirectional links on a Web, that itself is unidirectional. One way to create reverse links automatically would be by means of a background process that reads Web pages of a particular domain, assembles a database of links, and uses that database to create the reverse link automatically. As an alternative, he refers to the idea of Phillip Hallam-Baker, who proposed to use what later would become the HTTP referrer to achieve the same.

At the end of the 1990s searching the Web yielded a list of supposedly matching Web pages in no particular order. As a result, a user had to scan a great number of results in order to find a relevant match. The makers of Google pointed out what was needed to provide more useful search results: both relevance and aboutness of a page had to be established in a way that was independent of the page's content and hence insusceptible to manipulation. They found the solution for both by exploiting the linked structure of the Web: relevance of a page is established by the number of other pages pointing to it, and aboutness by the anchor texts of other pages referring to it. Intuitively, Google's approach

can be interpreted as having successfully mapped the foundations of reputation in human societies to the Web: someone is not important if he says about himself that he is, but if others do. In a way, Google succeeded by understanding and exploiting the humanness of the Web of documents.

Is strikes us as remarkable that the PageRank algorithm that Google uses to determine a Web page's relevance (cf. section 3.6) can be seen as an implementation of the above background process that originally was proposed to automatically create bidirectional links. Hence the success of Google's algorithm is not only to be sought in finding an on-line equivalent for social reputation, but also in its ability to calculate the missing reverse and thus bidirectional links on an intrinsically unidirectional Web. Even though Google cannot use the derived reverse links to make the actual Web a bidirectional one, the insusceptibility of its algorithm is based on those reverse links.

5.2 The Unrepaired Web of Data

The Web of data is driven by the motivation to make the Web work for automatic software agents so that a human being—or again another software programme—may instruct an agent to search on his behalf on the Web and to come back with a precise answer, and not with a list of Web pages. We have discussed the prerequisites for this to work in detail in the previous sections. However, even when looking at our few and admittedly simplified examples, a problem of today's Web of data quickly becomes apparent. Say we wanted to answer the query "In which state is Olympia located" with our triple store in listing 4.18, what would be the answer? Would it be "Greece" (line 16) or "Roman Empire" (line 22)? Likewise, when inquiring about population of the same place: is the answer "972" (line 13) or "1208" (line 29)? Unfortunately the respective answer is: both, since running a SPARQL query on that triple store will return two matching facts for each query *in no particular order*.[1] This reminds us very much of the problem of the Web of documents in the late 1990s. Thus when exploiting the structure of the Web provided a solution for the Web of documents, can it do the same for the Web of data?

At least it seems obvious to search for a solution in the exploitation of the graph structure, since the underlying concept of the Web of data is similar to that of the Web of documents. While the latter allows anybody to place a link to any other Web page, the former allows anyone to place a link to any other piece of data. Thus the same way we can represent the Web of documents as a graph, we can also represent the Web of data as a graph. Consider figure 5.1 for a representation of all triples in listing 4.18 as RDF graph, and note how resource `http://dbpedia.org/resource/Olympia,_Greece` has two identical properties `dbpprop:stateParty` that are only different in their values. It is important to understand, however, that the graph representation of the *facts* harvested from Web pages D_1–D_5 is different from the graph representation of the *Web pages* on

[1] In this case "no particular order" means "not applying any metric of relevance". As a matter of course, SPARQL supports an `ORDER BY` clause.

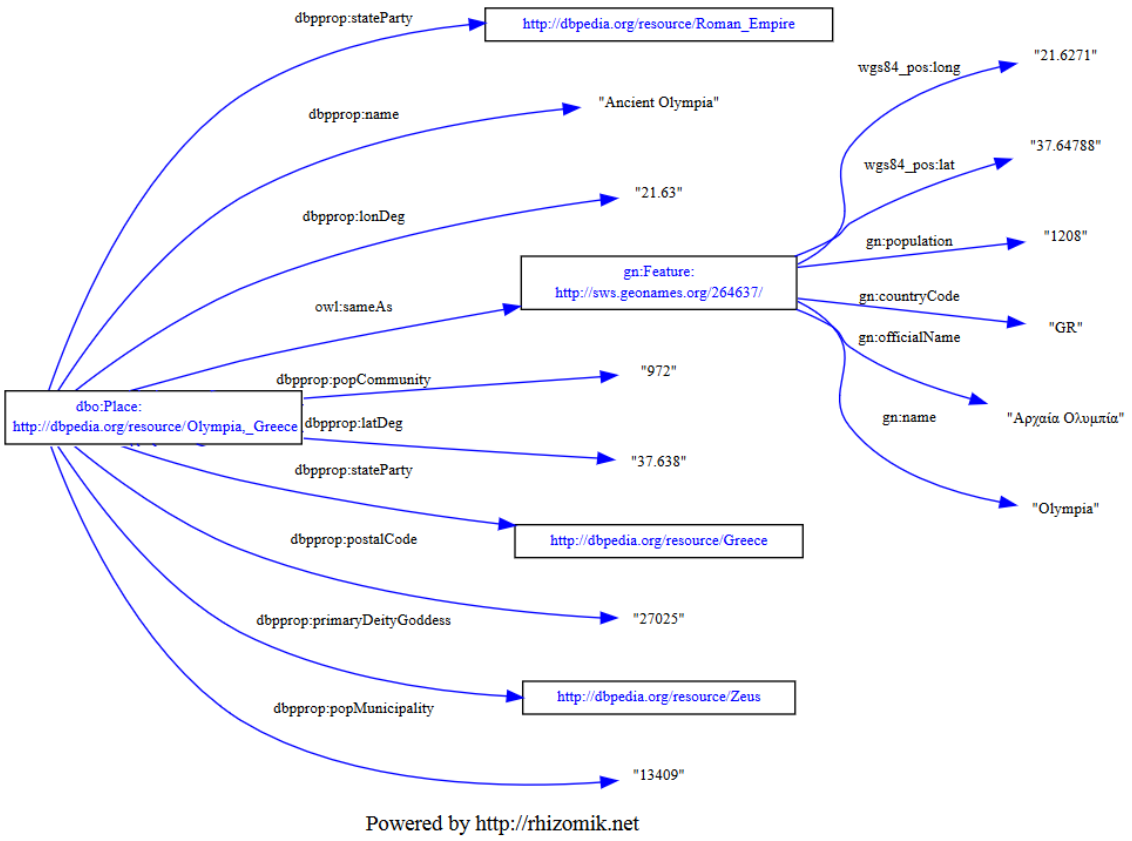

Figure 5.1: Linked data harvested from Web pages D_1–D_5 [Rhizomik] (RDF Graph)

which those facts have been published. Consider figure 5.2 for a (hypothetical) graph of the Web pages D_1–D_5.

As a matter of course, the difference of the two graphs can be explained by the two different types of linking. While on the Web of documents a link is placed *from one page to another* using the HTML `<A>` tag, linking on the Web of data means binding together a subject and an object by means of a predicate whereby the resulting fact may be published *on the same page*. As self-evident this difference might be, as subtle are the implications. On the Web of documents, the only Web pages from that I can link to another page are pages on that I may publish content. While since the advent of "Web 2.0" this does not necessarily imply ownership of those Web pages any more, this constraint still significantly reduces the number of pages *from* that I may link to another. In contrast, on the Web of data there are no constraints as to who may link from which piece of data to another. Any published class may be used by anyone as a subject or object, any published property as a predicate to formulate any new fact. If I were to make an untruthful statement about the population or the state of Olympia, nobody stops me from doing that. Again, that's the same on the Web of documents where I may likewise claim anything I want. However, on the Web of documents—thanks to the PageRank algorithm—my personal home page is considered significantly less relevant by Google than for example the *The New York Times*, for far more *other* Web pages refer to *The New York Times* than to my personal home

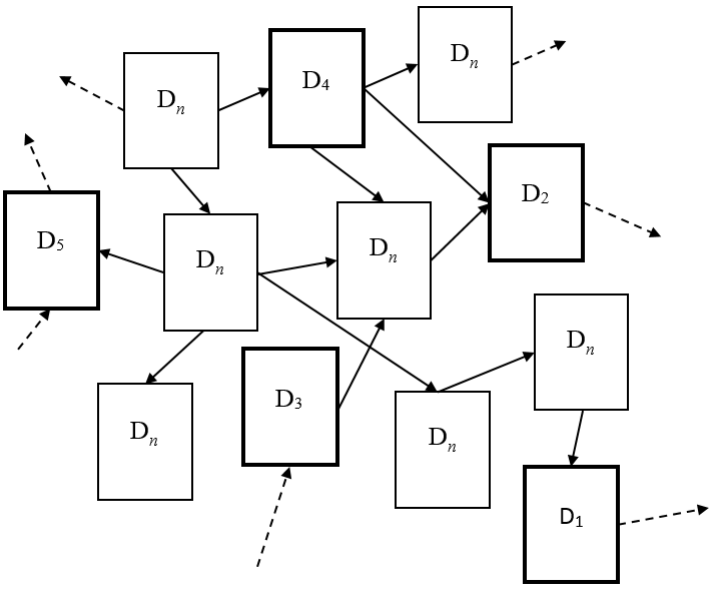

Figure 5.2: Web pages D_1–D_5 as part of the Web of documents

page. Since on the Web of data anyone may refer from anyone's subject to anyone's object using anyone's predicate, Google's approach to measure relevance by reference cannot be applied.

Chapter 6

Advancing the Web of Data

6.1 A Same-origin Policy for Linking

If we consider it a useful constraint on the Web of documents, that one may only place links from one's own Web page—and not from someone else's Web page—to other pages, we will have to look into possibilities of introducing such constraints on the Web of data. Listings 4.12, 4.13 and 4.16 are examples to show the opposite: the publishers of the facts have no authority over the resources involved (cf. table 4.4), nevertheless, they are all perfectly legal ways to publish facts on the Web of data. There are, however, other ways to publish the same fact that would impose the intended constraint on the publisher. Against that background, let us re-consider the HTML link types, that were introduced in section 3.3. Remember that Tim Berners-Lee's mentioned link types in his proposal that would eventually kick off the World Wide Web, and that they have been part—even if not always in a codified way—of every HTML version ever since. In section 3.3 the author presents the currently accepted link types as a list of valid values for the HTML `REL` attribute of the HTML elements `LINK`, `A`, or `AREA`, and also makes a reference to the "microformats wiki existing rel-values page" [MFREL], a "living standard" that allows anyone to register additional `REL` values. In addition to the `REL` values of HTML or the microformats wiki, "The remaining [REL] values must be accepted as valid if they are absolute URLs containing US-ASCII characters only and rejected otherwise" [HTML5-links]. Put another way, absolute URLs are perfectly valid `REL` values and do not require any further registration. We remain with the question how typed links help us to impose the constraint on authors to only publish facts that pertain to resources they have some authority over.

For that purpose, let us consider the example of authorship, a relation or *link* that binds together a person (the author) and a work created by that author. On the Web of data it is perfectly possible to interlink an author with a work without having authority over the author or the work. By way of example, let us claim in listing 6.1 that "Romeo and Juliet" was written by Christopher Marlowe.

Again, what strikes us as rather strange is that we may publish this fact although

Listing 6.1: Claim: Romeo and Juliet was written by Marlowe (RDF/XML)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dcterms="http://purl.org/dc/terms/">
    <rdf:Description rdf:about="http://shakespeare.mit.edu/romeo_juliet/index.html"
    >
        <dcterms:creator rdf:resource="http://dbpedia.org/resource/
    Christopher_Marlowe" />
    </rdf:Description>
</rdf:RDF>
```

Listing 6.2: Claim: Romeo and Juliet was written by Marlowe (HTML)

```
<!-- assumed document URL: http://shakespeare.mit.edu/romeo_juliet/index.html -->
<html>
<head>
<title>Romeo and Juliet: List of Scenes</title>
<link rel="http://purl.org/dc/terms/creator" href="http://dbpedia.org/resource/
    Christopher_Marlowe">
</head>
<body>
<!-- [...] -->
</body>
</html>
```

we have neither published its subject "Romeo and Juliet" nor its object "Christopher Marlowe". Even though the Web of documents allows us to make the same claim in plain text, it does not allow us to place a link from the Web page "Romeo and Juliet" to the Web page "Christopher Marlowe" unless we own the page "Romeo and Juliet". However, if we were the publisher of Web page "Romeo and Juliet" we could use typed links to express the same relationship by using the absolute URL of the `dcterms:creator` property as a value of the `REL` attribute as shown in listing 6.2[1].

Contrary to listing 6.1 that explicitly specifies subject (value of `rdf:about`), predicate (`dcterms:creator`), and object (value of `rdf:resource`), listing 6.2 merely specifies predicate (value of `REL`) and object (value of `HREF`). As with the use of HTML `A` element, the subject of the `LINK` element is implicitly assumed to be the *document's URL*, i.e. the URL of the Web page on that listing 6.2 is published. Thus by utilising the possibility of providing a predicate's absolute URL as a value of the `REL` attribute, and the object's URL as a value of the `HREF` attribute, we successfully impose the constraint on the fact's author to only publish facts pertaining to *subjects* or *objects* he published himself. Consequently, even though both listing 6.1 and listing 6.2 have the exact same RDF/TURTLE representation (cf. listing 6.3), we would only accept listing 6.2 as a valid *origin* of that triple, for it abides by the desired constraint. Nevertheless, even when limited to subjects or objects published by himself, of his own subjects and objects an author may still claim anything, i.e. he may still place a link from them to any resources on the Web. Thus we need an additional mechanism to *affirm* the fact in listing 6.3, and that affirmation must

[1] In this case, we could also use the short form `dcterms:creator` since it has been registered in [MFREL].

Listing 6.3: Claim: Romeo and Juliet was written by Marlowe (RDF/TURTLE representation) of listing 6.2 and 6.1)

```
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://shakespeare.mit.edu/romeo_juliet/index.html> dcterms:creator <http://
    dbpedia.org/resource/Christopher_Marlowe> .
```

not come from the publisher of this fact's subject, but it most come from the publisher of the fact's object. What we need is a *bidirectional link*.

6.2 Affirming Facts using Bidirectional Linking

In order to discuss the role of bidirectional linking on the Semantic Web, let us reconsider listing 6.3. To affirm its claim *Romeo and Juliet was written by Marlowe*, we would need a second fact that confirms that *Marlowe wrote Romeo and Juliet*. However, for that claim it does not suffice to merely let subject and object—*Romeo and Juliet* or *Marlowe*, respectively—change their position. We also need a new predicate, because the predicate `dcterms:creator` always expects a Creator as its *range* (the object-side). In order to be complimentary to the fact in listing 6.3, the new predicate needs to have the *reverse* meaning of *dcterms:creator*. Searching in the well-established [DCTERMS] vocabulary that we have used tacitly so far, we see that this vocabulary does not come with a predicate that accepts the creator as a *domain* (the subject-side). By way of example, let `ex:made` be the reverse predicate of `dcterms:creator`, then we would express *Marlowe wrote Romeo and Juliet*—the reverse fact of listing 6.3—as shown in listing 6.4.

An agent that harvests the facts of listing 6.3 and 6.4 could derive a bidirectional link between `http://shakespeare.mit.edu/romeo_juliet/` and `http://dbpedia.org/resource/Christopher_Marlowe` merely based on the observation that both facts contain the same classes in inverted position. While that bidirectional link would affirm a two-way relationship between the two classes, it would not affirm the *type* or meaning of this relationship. We postulated that `ex:made` has the reverse meaning of `dcterms:creator`, but without formally expressing that one property is the reverse of the other, the agent would just derive two unidirectional links pointing in the reverse direction (see figure 6.1).

In order to formally express, that one property is the reverse of another property, we have to utilize the features of a higher ontology language with greater ca-

Listing 6.4: Confirming listing 6.3: Marlowe made Romeo and Juliet

```
# Assumed place of publication: http://dbpedia.org/resource/Christopher_Marlowe
@prefix ex: <http://example.org/vocab/> .

<http://dbpedia.org/resource/Christopher_Marlowe>
ex:made
<http://shakespeare.mit.edu/romeo_juliet/index.html> .
```

Listing 6.5: Expressing that `ex:made` is the inverse property of `dcterms:creator`

```
@prefix ex: <http://example.org/vocab/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .

ex:made owl:inverseOf dcterms:creator .
```

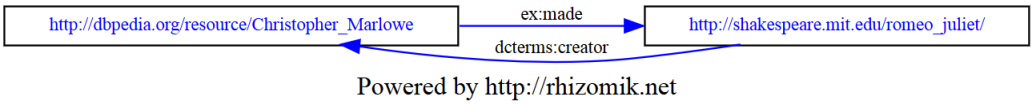

Figure 6.1: Affirming a link without affirming the semantics [Rhizomik]

pabilities than RDF. In section 4.6 OWL2 was shortly introduced as such a language. In addition to the `owl:sameAs` property discussed in that section, OWL2 also features an `owl:inverseOf` property (cf. [OWL2QuickGuide]) that is meant to express that one property is the *inverse* of another. Listing 6.5 demonstrates a formal way to express that `ex:made` is the inverse property of `dcterms:creator`. An agent having harvested the facts of listing 6.3, 6.4 and 6.5, in addition to creating a syntactic bidirectional link between `http://shakespeare.mit.edu/romeo_juliet/` and `http://dbpedia.org/resource/Christopher_Marlowe` could also affirm the semantics expressed by those links (cf. figure 6.2).

6.3 Reviving the HTML `REV` Attribute

Even though the approach in section 6.2 yields the intended result, it necessitates the declaration of a new property—in our example `ex:made`—and a formal expression of that new property being the inverse of another. Rather than defining one's own vocabulary, it would be easier if there was a generic way to use an existing property such as `dcterms:creator` in *reverse* meaning. This reminds us of the `REV` attribute of the HTML `A` or `LINK` element that was already described in section 3.3. In order to show how a *forward* link of the `REL` attribute is complement to the *reverse* link of the `REV` attribute, the following example is given in the HTML 4.01 recommendation (quotation on the next page).

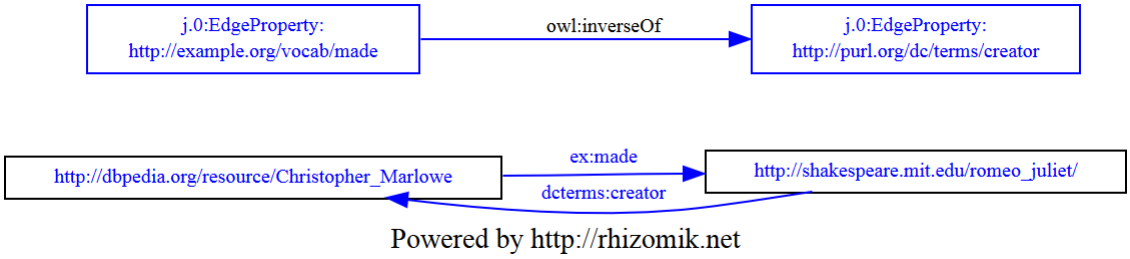

Figure 6.2: Affirming link *and* semantics using two complementary properties [Rhizomik]

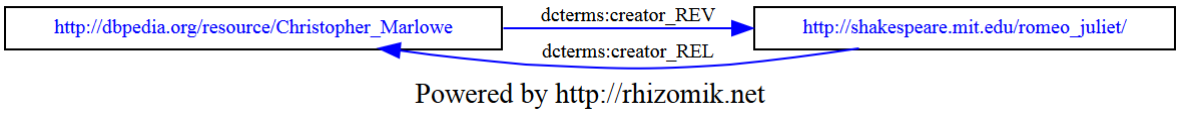

Figure 6.3: Affirming link *and* semantics using the same property as value of the `REL` and `REV` attribute [Rhizomik]

> "Consider two documents A and B.
> `Document A:` `<LINK href="docB" rel="foo">`
> Has exactly the same meaning as:
> `Document B:` `<LINK href="docA" rev="foo">`" [HTML4.01-links]

Applied to our case and using the shorthand notation of [MFREL], we would get:

> Consider two documents A "Romeo and Juliet" and B "Marlowe".
> `Document A:` `<LINK href="docB" rel="dcterms:creator">`
> Has exactly the same meaning as:
> `Document B:` `<LINK href="docA" rev="dcterms:creator">`"

This use of the `REL` and `REV` attribute allows us to express bidirectional semantics using only one already existing property as shown in figure 6.3. In that way, we have found a way to affirm the claim *Romeo and Juliet was written by Marlowe* with a complementary claim *Marlowe wrote Romeo and Juliet*. The `REV` attribute comes in handy for affirming facts on the Web, but it is no longer available in the most recent version of HTML (cf. [HTML5-links]). Ian Hickson, who as member of the WHATWG maintains HTML as a "living standard" (cf. [HTML-living-standard]), explained its removal on the public WHATWG mailing list:

> "Basically, nearly nobody was using it [the `REV` attribute], and almost all those that were using it incorrectly. This indicates that there is a problem." [WHATWG-06a]

This reasoning was not left without opposition. Charles Iliya Krempeaux replied to the above message:

> "[I]f I wanted to claim that I'm a member of [the Microformats.org group], then I could put a Microformats badge on my blog, and link it to the Microformats.org Web site.
>
> `<a rev="member" href="http://microformats.org/"> [...]</a>`
>
> [...], but anyone can put a badge like that on their site. How do you know you can trust that? Well, what if the Microformats.org Web site had a members page, and linked to all their members, including my blog with the following:

> `<a rel="member" href="http://changelog.ca/">Charles Iliya Krempeaux</a>`
>
> Well then we'd know that we trust this relation, because we have the same relation being asserted both ways." [WHATWG-06b]

The author of this thesis strongly supports Krempeaux' argumentation to continue the support of the HTML `REV` attribute, for this is required in order to affirm an already existing relation without having to declare a complementary property.

6.4 Bidirectional Linking for Non-HTML Objects

In section 6.2 the author of this thesis gave reasons why bidirectional linking should be used in order to affirm facts published on the Web. The importance of the HTML `REV` attribute for this purpose was discussed in section 6.3. While bidirectional links as shown in figure 6.3 provide a conceivable solution for interlinked Web pages in HTML, the question arises, how the same approach could be used for objects that are not HTML pages. By way of example, if a HTML page of Web site A links to an image of Web site B, how could the image of Web site B place a *reverse* link back to the HTML page of Web site A? Clearly, the image—of JPEG or PNG format or the like—cannot utilize any HTML markup to achieve that.

Against this background, it is worthwhile to revisit RFC5988 that was introduced in section 3.3 as a registry for link relations, i.e. valid values for the `REL` attribute. The application of the link relations defined in that registry is explicitly *not* confined to HTML, but also declared a valid part of the HTTP `Link` header field that is "semantically equivalent to the <LINK >element in HTML" [RFC5988]. Thus by adding a HTTP `Link` header along with the respective relation type to an image published on the Web, it is very well possible to place a link from that image to a Web page. Consider listing 6.6 (line 6) for an example of a link from a HTML Web page to a JPEG image and listing 6.7 (line 11) for a link back from the JPEG image to the HTML Web page. In both listings we use the relation type `rel="alternate"` to simply express the meaning that the respective target features an alternate version of the same content.

However, if we want to add a link to the image that points to its creator, we face the

Listing 6.6: Linking from a HTML page to a JPEG image using HTML `LINK`

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>About me</title>
6  <link rel="alternate" href="Markus.jpg">
7  </head>
8  <body>
9  <!-- [...] -->
10 </html>
```

Listing 6.7: Linking from a JPEG image to a HTML page using HTTP `Link` [cURL]

```
$ curl -I http://www.agnosis.de/r/Markus.jpg

HTTP/1.1 200 OK
Date: Sat, 6 Jun 2015 22:39:53 GMT
Content-Type: image/jpeg
Content-Length: 112267
Connection: keep-alive
Server: Apache
Last-Modified: Sat, 6 Jun 2015 21:29:48 GMT
Accept-Ranges: bytes
Link: <http://www.agnosis.de/r/Markus.htm>;rel=alternate
```

exact same problem as described in 6.3. Using the [DCTERMS] vocabulary, we could not use `dcterms:creator` or its fully qualified version `http://purl.org/dc/terms/creator`, because that property by definition only accepts a Creator as range (the object side). Hence it would be preferable if there was a way to express the that the `dcterms:creator` property is used in *reverse* meaning. However, in the same way that [WHATWG-06a] gave reasons for the removal of the `REV` attribute from HTML, RFC5988 says about the `REV` attribute:

> "The 'rev' parameter has been used in the past to indicate that the semantics of the relationship are in the reverse direction. [...] 'rev' is deprecated by this specification because it often confuses authors and readers; in most cases, using a separate relation type is preferable." [RFC5988]

In favour of the affirmation of links between HTML pages and images—as well as other static files—by means of bidirectional linking, the author of this thesis advocates the re-introduction of the `rel` attribute as a valid attribute for the HTTP `link` header field.

6.5 Rules for Changing the Evaluation Context

In section 4.4 the distinction between an *information resource* and *any kind of resource* was discussed along with the controversy around it. The bottom line is: if we want *many* information resources—e. g. a Web page, an image, a PDF document—to be recognized by automatic software agents as being representations of *one* concept—e. g. a person—, the concept has to be equipped with its own identity and its representations should provide pointers to the concept to establish the linkage between representations and concept. Now, if we use the `REL` attribute for semantic linking—either in a HTML context (cf. section 6.3) or in a HTTP context (cf. section 6.4), we explicitly specify the predicate—the `REV` attribute's value—and the object—the URL of the `HREF` attribute. We do *not* specify the URL of the subject. In section 6.1 the benefit of this approach was presented as successfully imposing the favourable constraint on a fact's author to only publish facts pertaining to *subjects* he published himself. However, since the subject's URL is implicitly the URL of

the document that the `REL` attribute is part of, that URL is likely to be the URL of an *information resource*, i.e. the URL of a representation, but not of the concept.

As an example, consider again listing 6.2. The fact's predicate and object are specified by means of the `LINK` element. The implicit subject is the Web page's URL, in this case: `http://shakespeare.mit.edu/romeo_juliet/index.html`. This is clearly the URL of an *information resource*, namely a HTML representation of the work "Romeo and Juliet". It is *not* the URL of the concept of "Romeo and Juliet"—*any kind of resource*—for which there are other representations in different formats. If we wanted to establish the linkage between the *concept* of "Romeo and Juliet" and the *concept* of Marlowe by using the `LINK` element, how would we do that? Put another way, how can we make sure that the subject's URL points a potential agent to the concept by continuing to use the `LINK` approach and its advantage of adhearing to the same-origin policy for linking (cf. 6.1)?

We are touching on a topic, that is addressed by the RDFa recommendation (cf. section 4.3) under the name "Changing the Evaluation Context" [RDFa]: if we want a subject's URL to be different from the Web page's URL, we have to change the *context of the evaluation*. RDFa comes with different approaches to do just that:

- HTML `BASE` element: cf. listing 6.8 (line 4)
- RDFa `ABOUT` attribute: cf. listing 6.9 (line 5)
- RDFa `RESOURCE` attribute: cf. listing 6.10 (line 3)

Unlike the HTML page in listing 6.2 that—once parsed—resulted in the RDF triple in listing 6.3 with the subject `http://shakespeare.mit.edu/romeo_juliet/index.html`, the HTML pages in listing 6.8, 6.9, and 6.10 result in the RDF triple shown in listing 6.11 with the subject `http://shakespeare.mit.edu/romeo_juliet/`. The advantage of the latter subject's URL over the former one is, that using the methods described in section 4.4 an agent interested in the concept may query that URL by requesting an `application/rdf+xml` response—describing the concept—and a Web browser may query the same URL by requesting a `text/html` response—describing the representation. While all three approaches using the HTML `BASE` element or the RDFa `ABOUT` or `RESOURCE` attributes in the given examples yield the same result, the author of this thesis proposes to use the `BASE` element for this purpose. Of the three approaches only the `BASE` element ensures that the same-origin policy described in section 6.1 is adhered to, for it does not only changes the *semantic* evaluation context—i. e. the subject of our HTML-based facts—but also the *syntactic* evaluation purpose inasmuch as the `BASE` element "specif[ies] the document base URL for the purposes of resolving relative URLs" [HTML5]. Consequently, by changing the `BASE` element's value to something else than its document's URL has an effect on resolving hyperlinks with relative URL such as `<a href="about">...</a>`. Since an author of a Web page has no interest to invalidate his syntactic hyperlinks by changing the evaluation context for semantic purposes, we successfully impose the constraint on him, to only publish facts about subjects he published himself while providing him with means to distinguish between concept and representation.

Listing 6.8: Changing the evaluation context using the HTML `BASE` element

```
<!-- assumed document URL: http://shakespeare.mit.edu/romeo_juliet/index.html -->
<html>
<head>
<base href="http://shakespeare.mit.edu/romeo_juliet/">
<title>Romeo and Juliet: List of Scenes</title>
<link rel="http://purl.org/dc/terms/creator" href="http://dbpedia.org/resource/
    Christopher_Marlowe">
</head>
<body>
<!-- [...] --->
</body>
</html>
```

Listing 6.9: Changing the evaluation context using the RDFa `ABOUT` attribute

```
<!-- assumed document URL: http://shakespeare.mit.edu/romeo_juliet/index.html -->
<html>
<head>
<title>Romeo and Juliet: List of Scenes</title>
<link about="http://shakespeare.mit.edu/romeo_juliet/" rel="http://purl.org/dc/
    terms/creator" href="http://dbpedia.org/resource/Christopher_Marlowe">
</head>
<body>
<!-- [...] --->
</body>
</html>
```

Listing 6.10: Changing the evaluation context using the RDFa `RESOURCE` attribute

```
<!-- assumed document URL: http://shakespeare.mit.edu/romeo_juliet/index.html -->
<html>
<head resource="http://shakespeare.mit.edu/romeo_juliet/">
<title>Romeo and Juliet: List of Scenes</title>
<link rel="http://purl.org/dc/terms/creator" href="http://dbpedia.org/resource/
    Christopher_Marlowe">
</head>
<body>
<!-- [...] --->
</body>
</html>
```

Listing 6.11: Claim: Romeo and Juliet was written by Marlowe (RDF/TURTLE represention of listing 6.8, 6.9, and 6.10)

```
@prefix dcterms: <http://purl.org/dc/terms/> .

<http://shakespeare.mit.edu/romeo_juliet/> dcterms:creator <http://dbpedia.org/
    resource/Christopher_Marlowe> .
```

Chapter 7

Bidirectional Linking Today

7.1 The Social Web

The idea to let Web links represent relationships among people has already been part of Tim Berners-Lee's original proposal of what would become the World Wide Web (cf. section 3.2). Later Berners-Lee underlined the social dimension of the Web by saying:

> "The web is more a social creation than a technical one. I designed it for a social effect—to help people work together—and not as a technical toy. The ultimate goal of the Web is to support and improve our weblike existence in the world. We clump into families, associations, and companies. We develop trust across the miles and distrust around the corner." [BF99], cited by [XFN1.1] (intro)

Today "social networks", such as Facebook, are considered an integral component of the current Web. Yet, it is important to observe that Facebook—while it uses Web technology—is at its core not part of the Web. Unlike the open Web, that Tim Berners-Lee meant to create and that is based on open standards and governed by the Web community, participation in Facebook requires the creation of a proprietary user account and adherence to the rules of a private company.

Notwithstanding its private nature, Facebook and other social network applications are a great example of bidirectional linking in current Web applications. Let us consider one basic linking capability, that every social network provides its users with in one way or another: the possibility of *connecting* two users of the social network with each other as *friends*. To this end, one user A would send a user B a friendship request. Once accepted by user B, that would make users A and B *friends* on the respective social network.

It would strike us as rather strange, if the mere sending of a friendship request by user A would create that *friendship* connection between the two users. However, when being looked at from a linking perspective, that is not different from the possibility on the current Web of data to place a meaningful one-way link from page A to page B without the necessity of creating a reverse link (cf. section 5.2). Consider figure 7.1 for a graph

Figure 7.1: Unconfirmed friendship relation [Rhizomik]

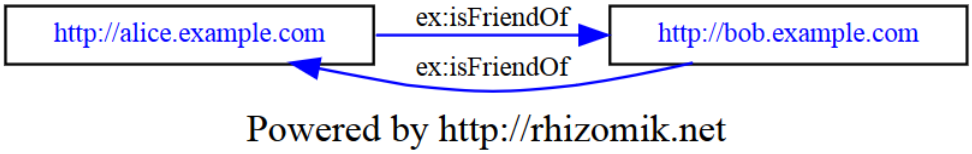

Figure 7.2: Confirmed friendship relation [Rhizomik]

representation of an unconfirmed friend relationship, and figure 7.2 for that of a confirmed friend relationship.

In figure 7.2 we are affirming a fact using bidirectional linking (cf. section 6.2), but we do not require a complementary property or the possibility of expressing a reverse relationships (cf. section 6.3), for the *friendship* relation—once confirmed—is a symmetric relationship. Formally: user A and B are friends if A is a friend of B *and* B is a friend of A. Thus in this case, fact affirmation can be done by repeating the fact with subject and object in inverted position, but by using the same predicate.

Again, Facebook is using Web technology. However, it is not on the Web, but it is merely a private subnet of the Web. Having said that, there are approaches to open up Facebook to the Web. By means of an RDFa based vocabulary called "The Open Graph protocol" [OGP] any page on the open Web, i.e. outside Facebook, may become a node in Facebook's social graph. While this creates the possibility of integrating Web pages on the open Web into Facebook, it does not create the possibility of linking from the open Web to a node on Facebook. Put another, there is no possibility of connecting to someone on Facebook without being on Facebook.

The "XHTML Friends Network" [XFN1.1] is one example of an approach, to bring the features of private social networks such as Facebook to the open Web. To that end, the HTML `REL` attribute is used to represent human relationships, of which *friend* is only one. XFN defines the following values for the `REL` attribute to express friendship, physical, professional, geographical, family or romantic relationships [XFN1.1]:

- contact
- acquaintance
- friend
- met
- co-worker
- colleague
- co-resident
- neighbor
- child*
- parent*
- sibling
- spouse
- kin
- muse*
- crush*
- date
- sweetheart
- me

Listing 7.1: Bidirectional social links using symmetric XFN `REL` values

```
<!-- assumed document URL: http://alice.example.com -->
<a href="http://bob.example.com" rel="friend">Bob</a>

<!-- assumed document URL: http://bob.example.com -->
<a href="http://alice.example.com" rel="friend">Alice</a>
```

Most XFN `REL` values are symmetric, i.e. if Alice wanted to express that she is a friend of Bob, and Bob wanted to express that he is a friend of Alice, both would place a `<A REL="friend">` link to their respective Web sites as shown in listing 7.1. The graph representation of those links is the same as the one depicted in figure 7.2. Unlike Facebook, XFN explicitly supports one-way friend relationships, because "[w]hether or not you consider yourself to be a friend of someone else is something under your control" ([XFN1.1], 'background').

In the itemization of valid XFN `REL` values, the ones indicated with an asterisk (*) are *not* symmetric. Consider listing 7.2 as an example of a bidirectional social link using `REL` values with inverse meanings, in this case `child` and `parent`. As discussed in section 6.3, creating a bidirectional semantic link from predicates with inverse meanings requires knowledge of which the vocabulary and its definition of which predicates are to be considered to be of the inverse of others.

In section 4.6, ways were discussed to express *sameness* on the Web. In the context of human relationships the capability of expressing *sameness* is required, in order to express that two different Web pages are representing the same person. XFN provides the special `REL` value `me` for this purpose. It is the only `REL` value that needs to be expressed bidirectionally to be considered valid. Consider listing 7.3, in which Alice says of two different Web sites that both represent her.

XFN is a surprisingly simple and straightforward implementation of bidirectional linking on the Web. Its simplicity can partly be sought in its predominantly symmetric relationships, that exempt us from the necessity of inverse properties (cf. section 6.2) or the `REV` attribute (cf. section 6.3). Notwithstanding the mostly symmetric nature of its relationships, XFN neither defines a complex RDF(S) vocabulary (cf. section 4.6), nor does it touch on the particularities of URI semantics (cf. section 4.4).

It is easy to point out the weaknesses resulting from those omissions: lack of extensibility in case of the former, ambiguity in case of the latter. However, with the inclusion of XFN into the list of "existing rel values" [MFREL], an itemization of `REL` values accepted by both the WHATWG's and the W3C's HTML (cf. [HTML-living-standard] and [HTML5]), it is perfectly legal to use those `REL` values in one's own HTML markup as a first and easy step towards bidirectional linking on the Web.

Listing 7.2: Bidirectional social links using inverse XFN `REL` values

```
<!-- assumed document URL: http://alice.example.com -->
<a href="http://bob.example.com" rel="child">Bob</a>

<!-- assumed document URL: http://bob.example.com -->
<a href="http://alice.example.com" rel="parent">Alice</a>
```

Listing 7.3: Expressing *sameness* using special XFN `REL` value `me` (required symmetric)

```
<!-- assumed document URL: http://alice.example.com -->
<a href="http://alice.example.org" rel="me">Alice's second Web site</a>

<!-- assumed document URL: http://alice.example.org -->
<a href="http://alice.example.com" rel="me">Alice's first Web site</a>
```

7.2 The Bibliographic Web

In section 7.1 bidirectional linking is introduced as a core feature of private social networks, and XFN is described as one way to bring that feature from "closed" Web applications to the "open" Web. Unlike Facebook, Google claims to develop applications for the open Web rather than the closed Web. When Marissa Mayer, then vice president at Google, was asked in January 2011, why Google did not co-operate with Facebook in the field of the social networks, she replied: "We think someone has to support the open Web" [Kir11].

In that same year (2011), Google launched its own social network dubbed "Google+" (cf. [Gun11]). Like Facebook, Google+ provides the possibility of creating a *profile* page on the Web that may serve as a user's identity on the Web; and like Facebook participation in Google+ requires the creation of a proprietary user account and adherence to the rules of a private company. In the context of this thesis, however, we look at an interesting interplay of the Google profile with XFN's `<A REL="me">` and HTML's `<A REL="author">` link to enable "authorship markup", a feature announced by Google also in 2011 [Han11].

Google's utilization of XFN's `<A REL="me">` link is the same as the one shown in listing 7.3: a bidirectional `<A REL="me">` link between two Web pages expresses the meaning that both pages represent the same person. In the context of Google's application, one of the two Web pages is a user's Google profile page, the other Web page is another profile page of the same user on the open Web, e.g. an "about me" page of a personal blog.

Google proposes to place a HTML `<A REL="author">` link from an author's content page such as an on-line article to the Google profile page—or another profile page that has been interlinked with the Google profile—to declare authorship of that article. In order to confirm this declaration of authorship, a link from the Google profile *back* to the author's content page has to be placed. Let us consider the following example:

- Content page: `http://alice.example.org/post/`
- Author page `http://alice.example.org/about/`
- Google profile: `http://plus.google.com/alice/`

Figure 7.3: Google's proposal to declare authorship (1/2) [Rhizomik]

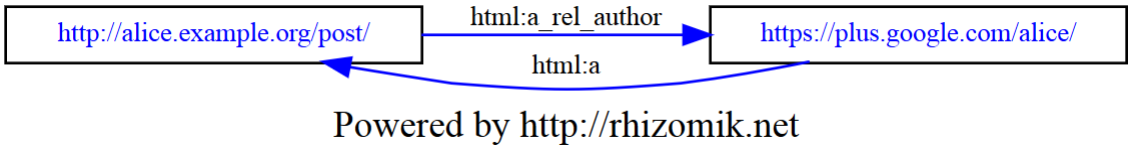

Figure 7.4: Google's proposal to declare authorship (2/2) [Rhizomik]

Following Google's recommendations (cf. [Goo11]), Alice would declare authorship of the content page by placing the links as shown in figure 7.3. Alternatively, Alice could directly link from the content page to her Google profile and back from her Google profile to her content page as shown in figure 7.4.

Both figure 7.3 and 7.4 are examples of using bidirectional linking to affirm facts. The fact, that `http://alice.example.com/about/` is representing the same person as `https://plus.google.com/alice/` is claimed *and* confirmed using a `<A REL="me">` link. The fact, that the blog post `http://alice.example.org/post/` has the author `https://plus.google.com/alice/` is claimed using the `<A REL="author">` link and it is confirmed by placing an `<A>` link back from the profile page to the blog post.

In this example, the difference between symmetric and asymmetric relationships becomes apparent again. While we may use `<A REL="me">` in both directions to confirm the identity (cf. section 7.1), we cannot use the `<A REL="author">` link to confirm the authorship. Google uses an untyped `<A>` link without any `REL` attribute—consequently having "no special meaning" [HTML5-links]—to confirm the authorship relation. The semantics that the back link has the inverse meaning of the authorship relation are not expressed explicitly, but they are implicitly assumed by Google's own *usage* of the authorship declaration.

When Google's authorship markup was announced, it was made clear that Google would be "experimenting with using this data to help people find content from great authors in our search results" [Han11]. Consequenty, the presentation of authorship in Google's search results varied greatly over time. In addition to the mere appearance of the author's name or photo in Google's search results, experiments included a "more by this author" feature, that would list all online publication of a give author on the Web (cf. [Sma14]. In 2014, however, three years after its introduction, Google announced that it put an end to the authorship experiment by not showing authorship in the search results anymore, because "[we at Google] observed that this information isn't as useful to our *users* as we'd hoped, and can even distract from those results" [Mue14], emphasis by

the author.

By saying that the authorship markup did not turn out to be as useful for the *users* as intended, Google seems to look at this markup solely from the perspective of the *Web of documents*, in that human users are browsing and searching the Web. In section 6.2 the author of this thesis has given reasons, why bidirectional linking is crucial for affirming facts on the *Web of data*, in that automatic software agents are searching the Web on a human user's behalf.

Few bits and pieces are lacking to bring Google's authorship markup to the Web of data: possibilities of confirming relationships that are not symmetric, such as the authorship relationship, were presented in section 6.3 and 6.3. Using the approaches discussed in section 4.4, any Web profile, and not only a Google or Facebook profile, could be used to host an identity on the Web. In that way authorship could be both declared *and* confirmed using only the (sematic) technology of the open Web.

There are rumours, that Google might continue to support authorship markup (cf. [Sch15]). The author of this thesis would strongly support this continuation along with the changes above to free this markup and its discovery from proprietary standards. If Google is serious in its support of an open Web and if Google considers the Web of agents to be as equally important as the Web of documents, continuation and extension of the authorship markup would only be a consistent move.

7.3 The Office Web

In section 2.3 Ted Nelson was introduced as the person who coined the word *hypertext* to describe a non-sequential representation of text by means of inter-linked chunks of text, through which a reader may arbitrarily move. While we may see in today's World Wide Web an implementation of Nelson's hypertext, we must not forget that Nelson visions have been never confined to the idea of an inter-linked Web of documents. Part of his holistic and unified approach to information and computer science have been elaborated ideas on *version control* and *transclusion*. The latter describes a property of a hypertext system, that allows one document to directly embed a "chunk of text" of *another* document. In that way, a new virtual document can be created that merely consists of embedded content of other documents without copying this content. It is because of those advanced features, that stable links between two chunks of texts were always of a special concern to Nelson, for in a hypertext system that supports the embedding of pieces of text, version control of both documents and links are a mandatory feature. Consequently, Nelson protests against statements that describe his hypertext system—later referred to by Nelson as *Xandadu*—as the failed World Wide Web.

In 1999, Nelson writes:

> "Project Xanadu, the original hypertext project, is often misunderstood as an attempt to create the World Wide Web. [...] The Web trivialized this original Xanadu model, vastly but incorrectly simplifying these problems to a world of fragile ever-breaking oneway links, with no recognition of change or copyright, and no support for multiple versions or principled re-use. Fonts and glitz, rather than content connective structure, prevail." [Nel99]

With the early ideas on the implementation of a Bush-like information system in the early 1960s, Ted Nelson has now been working more than 50 years on his project. In 2014, *The Guardian* launched an article about Project Xanadu titled "World's most delayed software released after 54 years of development" [Her14]. The occasion was a presentation of Ted Nelson of a working model of his hypertext system earlier that year at the Chapman University. The working model now called *OpenXanadu* consists of a single document only, that demonstrates the feature of transclusion. Alex Hern of *The Guardian* quotes Nelson as saying:

> "We screwed up in the 1980s, and missed our chance to be world wide hypertext (the web got that niche). However, we can still compete with PDF, which simulates paper, by showing text connections." Nelson cited by [Her14]

To date, it seems, Ted Nelson has not given up on his project, but he sees its potential no longer as a candidate for the World Wide Web, but in the domain of office management. Ironically, that is one of the field of applications that Bush had in mind for his hypothetical *memex* device (cf. section 2.3). Maybe a domain with a scope smaller than that of a worldwide Web offers in fact applications for Nelson's ideas of version control, transclusion and bidirectional linking.

Chapter 8

Results and Discussion

8.1 The Evolution of the World Wide Web

Key to the successful evolution of the Web of documents is its feature to be searchable, with a measure of relevance applied to the search results, that is insusceptible to manipulation (cf. section 3.5 and 5.1). While the Web of data is searchable (cf. section 4.7), it does not have a measure of relevance that could be applied to the search results (cf. section 5.2), nor is it immune against manipulation (cf. section 6.1). Confer table 8.1 for an overview of the Web of data's shortcomings with respect to the Web of documents.

The algorithm that determines the relevance of a page on the Web of documents in a way that is insusceptible to manipulation is based on the calculation of a Web graph, that represents intrinsically meaningless links among Web pages (cf. section 3.6). The calculation of that graph assumes certain constraints under which links can be created on the Web of documents. Neither does the Web of data impose the same constraints on the creation of links, nor does it utilize the same Web graph (cf. section 5.2). Hence other measures of relevance that are insusceptible to manipulation have to be found.

In order to make the Web of data less prone to manipulation, the author of this thesis discusses the following approaches: a same-origin policy for linking on the Web of data (cf. section 4.6), the utilization of bidirectional linking for affirming facts (cf. section 6.2) as well as for linking non-HTML objects (cf. section 6.4), and rules for changing the evaluation context (cf. section 6.5). With respect to the application of bidirectional linking, current approaches are presented along with proposals how they can be detached from a commercial context to become truly open Web technologies (cf. section 7.1 and 7.2).

While this thesis discusses approaches to make the Web of data less insusceptible to manipulation, it does not discuss any approaches for a measure of relevance that could be applied to valid—i.e. non-manipulated—search results. However, any advances in finding valid measurements of relevance for searching on the Web of data will have to be assessed against the question whether or not they are insusceptible to manipulation.

	Web of documents	Web of data
searchable	Yes	Yes
measure of relevance	Yes	No
immunity to manipulation	Yes	No

Table 8.1: The shortcomings of the Web of data

8.2 The Role of Bidirectional Linking

The Role in Visions of the Web

Bidirectional linking is among the approaches considered in this thesis for making the Web of data less prone to manipulation (cf. section 6.2 and 6.4). Irrespective of this particular application, bidirectional linking is a recurring theme throughout the evolution of the World Wide Web. In Paul Otlet's work, that is referred to in this thesis as one theoretical foundation of linking information, bidirectional linking is proposed to maintain a connection between data of different granularity. In a traditional information space such as a library, Paul Otlet considers it necessary to establish a linkage between the part and the whole, for example by means of a link between a journal issue and its articles as well a link between those articles and the issue (cf. section 2.2). Applied to today's information space of the World Wide Web, the same requirement can be expressed by saying that not only every Web page should know of which pieces of data it consists, but also every piece of data should know to which Web page it belongs. Likewise for Ted Nelson bidirectional linking is an integral part of a hypertext system (cf. section 2.3). His ideas of version control and *transclusion* are indeed not feasible without a mandatory bidirectional link. Having accepted a World Wide Web without those features, Nelson still sees a possible application of his ideas in more confined domains (cf. section 7.3). However, the observation that the two trends, *Web of documents* and *Web of data*, constitute two different Web graphs, demonstrates that today's Web does not meet the requirements of Paul Otlet nor the requirements of Ted Nelson. Its fundamental shortcomings necessitate different approaches against manipulation on both Webs (cf. section 5.2). Against this background, the approaches discussed in this thesis merely address the symptoms of a fundamental deficit, rather than the deficit itself.

The Role on the Web of documents

Even though the Web of documents is a famous example of a unidirectional hypertext system, it is unidirectional only in the sense that it *supports* unidirectional linking, but not in the sense that it disallows bidirectional ones. Already the original link relations proposed by Tim Berners-Lee suggest a bidirectional application (cf. section 3.2). An explicit evidence for *reverse* linking on the Web of documents is the existence of the HTML `REV` attribute, that was supported by different versions of the HTML standard until it was dropped in HTML5 (cf. section 3.3). In this thesis both the reasons for its

abolishment are presented (cf. section 6.3), as well as the arguments for its continuation (ibid. and section 6.4). Tim Berners-Lee himself explained why a Web of "different capabilities cannot insist on bidirectional links" (section 5.1): given that parties of different importance are represented on the same Web, the requirement of both a forward *and* a reverse link to establish a linkage would significantly hinder a rapid growth of the Web. He also presented ways to automatically derive reverse links, rather than creating them manually (ibid.). Having analysed the evolution of the Web of documents in this thesis, it is not without irony to observe that the susceptibility of the early Web to manipulation (cf. section 3.5), has been addressed by an approach that automatically calculates the back links to a Web page (cf. section 3.6). That is not bidirectional linking in the truest sense of the word, however, it underlines the necessity of corrections to the unidirectional design of the Web employing methods that are similar to bidirectional linking.

The Role on the Web of data

Hitherto the evolution of the Web of data has been dominated by *technical* questions regarding the representation of machine-readable semantics (cf. section 4.3), the identification of *resources* both on and off the Web (cf. section 4.4), ways of interlinking data (cf. section 4.5) and approaches to express *sameness* between resources (confer section 4.6). Considering the aim of the Web of data to make the Web meaningful to machines (cf. 4.1), it is not surprising that the presented answers to those questions have been evaluated against their suitability for automatic agents that *read* the Web. However, they have *not* been assessed with respect to their manipulability by humans who *build* the Web and who eventually will use the output that is generated by those automatic agents. Technical issues have also dominated the early years of the Web of documents. It was not before the Web of documents reached a certain size and adoption that the human issues of an application as ubiquitous as the World Wide Web became apparent. Even if the Web of data is built for machines, there are no reasons to believe that the human issues concerning its application will be any different than the ones that became apparent in the evolution of its predecessor. As to the bidirectional linking capabilities of the Web of data, one can observe that while they exist, their implementation if often unnecessarily complex (cf. section 6.2) and their importance is—with the few exceptions presented (cf. section 6.3)—not recognized for sparing the Web of data a similar fate as the Web of documents experienced in the late 1990s.

The Role for the Advancement of the Web of data

Bidirectional linking is discussed in this thesis as one key approach to address the Web of data's current susceptibility to manipulation. The presented approaches (cf. section 6.2) along with examples of their application (cf. section 6.3 and 6.4) are meant to overcome a shortcoming of the Web of data, that could become a major hindrance on the Web of data's way to world wide adoption. However, the very same approaches have the potential

to become a hindrance for the Web of data's future evolution themselves, for the same reasons that prompted Tim Berners-Lee to decide *against* a bidirectional link *by design* (cf. 5.1). The Web of data—as well as the Web of documents—is and will be a Web of different capabilities. Bidirectional linking therefore must not be the only way to establish a valid linkage. Having said that, it is clear that a confirmed fact will have to be treated differently—namely with a higher relevance—than an unconfirmed one. Which measure of relevance is to be used, in order to rank confirmed facts higher than unconfirmed ones, and—more importantly—to rank a fact within a set of confirmed or unconfirmed facts, respectively, has not been discussed in this thesis, but will be likewise crucial to the success of the Web of data.

The Role on the Current Web

The role of bidirectional linking is surprisingly widespread in current Web applications such as Facebook or like social networks (cf. section 7.1) or Google's (discarded) approach to build a bibliographic Web (cf. section 7.2). Both applications would not exist without the support *and* control of the private companies behind them. Nevertheless, they are great examples of applications in that bidirectional linking—albeit not by that name—is accepted as necessity to confirm certain meaningful relationships. It is common sense on the present-day social Web, that the mere receipt of a friendship request does not make the recipient of that request automatically a friend of the sender: it is the *confirmation* of the request that does. Likewise, the necessity to confirm a claimed authorship with a reverse link from a profile controlled by the author is accepted because otherwise the risk of wrongly claimed authorship by merely placing a one-way link would be way too high. In both applications meaningful or typed links, a feature that is usually attributed to the Web of data. In both cases, however, they are used on the Web of documents for the purposes of the human user. In case of the social Web applications they are in general considered useful, in case of the bibliographic Web the benefit has been assessed as not "as useful to our users as we'd [at Google] hoped" (cf. section 7.1), which led to its abolishment. From the viewpoint of an assumed automatic agent crawling the Web on behalf of a human user, both applications—even though they were built for the human Web—are perfectly working examples of a machine-understandable Web employing bidirectional linking for the purposes of confirming facts. As pointed out in the respective sections, ways have to be found to bring those applications from the privately run—and hence easily controllable—private subnets such as Facebook and Google+ to the open Web.

There seems to be a trade-off between the *necessary* control of linking to facilitate a meaningful Web and the *desirable* openness of the Web. In this thesis bidirectional linking, a same-origin policy for linking, and rules for changing the evaluation context were discussed to provide useful constraints for Web linking. The question remains, what degree of control is feasible on an open Web and whether this control necessarily involves private companies. This greater question is again not a technical but a human issue.

Listings, Figures, Tables

Listings

List of Figures

List of Tables

References

Bibliography

[Ber89] Tim Berners-Lee. "Information Management: A Proposal". HTML copy of the original proposal for a global hypertext system at CERN. Mar. 1989. URL: http://www.w3.org/History/1989/proposal.html.

[Ber90a] Tim Berners-Lee. "Building back-links". 1990. URL: http://www.w3.org/DesignIssues/BuildingBackLinks.html.

[Ber90b] Tim Berners-Lee. "Topology". 1990. URL: http://www.w3.org/DesignIssues/Topology.html.

[Ber98a] Tim Berners-Lee. "Web Architecture from 50,000 feet". Personal note. Last updated 2009-08-27. Sept. 1998. URL: http://www.w3.org/DesignIssues/Architecture.html.

[Ber98b] Tim Berners-Lee. "Why RDF model is different from the XML model". Personal note. Oct. 14, 1998. URL: http://www.w3.org/DesignIssues/RDF-XML.html.

[Ber06] Tim Berners-Lee. "Linked Data". Personal note. July 27, 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html.

[BF99] Tim Berners-Lee and Mark Fischetti. *Weaving the Web*. HarperSanFrancisco, 1999.

[BHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. "The Semantic Web. a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities". In: *Scientific American* 284 (5 2001), pp. 34–43.

[Bib05] Institut International de Bibliographie. *Manuel abrégé du repertoire bibliographique universel ...* Publication no 65. 1905.

[Boo03] David Booth. "Four Uses of a URL: Name, Concept, Web Location and Document Instance". Personal note. Jan. 28, 2003. URL: http://www.w3.org/2002/11/dbooth-names/dbooth-names_clean.htm.

[BP98] Sergey Brin and Lawrence Page. “The Anatomy of a Large-scale Hypertextual Web Search Engine”. In: *Proceedings of the Seventh International Conference on World Wide Web 7*. WWW7. Brisbane, Australia: Elsevier Science Publishers B. V., 1998, pp. 107–117. URL: http://dl.acm.org/citation.cfm?id=297805.297827.

[Bus45] Vannevar Bush. “As We May Think”. In: *The Atlantic Monthly* (July 1945). URL: http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/ (visited on 06/2015).

[Dji09] Jean-Michel Djian. “Le Mundaneum, Google de papier”. In: *Le Monde Magazine* (Dec. 19, 2009), pp. 46–51.

[Fie00] Roy Thomas Fielding. “Architectural Styles and the Design of Network-based Software Architectures”. Doctoral dissertation. University of California, 2000. URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

[Goo11] Google. “Authorship”. 2011. URL: https://support.google.com/webmasters/answer/1229920 (visited on 06/2015).

[Gun11] Vic Gundotra. “Introducing the Google+ project: Real-life sharing, rethought for the web”. Blog post. June 28, 2011. URL: https://googleblog.blogspot.de/2011/06/introducing-google-project-real-life.html.

[Han11] Othar Hansson. “Authorship markup and web search”. June 7, 2011. URL: http://googlewebmastercentral.blogspot.de/2011/06/authorship-markup-and-web-search.html.

[HF99] Frank van Harmelen and Dieter Fensel. “Practical Knowledge Representation for the Web”. In: *IJCAI Workshop on Intelligent Information Integration*. Ed. by Sixteenth International Joint Conference on Artificial Intelligence. 1999. URL: http://www.cs.vu.nl/~frankh/postscript/IJCAI99-III.pdf.

[Hen01] J. Hendler. “Agents and the Semantic Web”. In: *Intelligent Systems, IEEE* 16 (2 Mar. 2001), pp. 30–37. ISSN: 1541-1672. DOI: 10.1109/5254.920597. URL: http://dx.doi.org/10.1109/5254.920597.

[Her14] Alex Hern. “World’s most delayed software released after 54 years of development”. In: *The Guardian* (June 6, 2014). URL: http://gu.com/p/3pq2q/sbl.

[Kir11] David Kirkpatrick. “Google’s Mobile Matchmaker”. In: *The Daily Beast* (Jan. 7, 2011). URL: http://thebea.st/1uNEegK.

[Mue14] John Mueller. "[End of authorship in Google search results]". Personal blog. Aug. 28, 2014. URL: https://plus.google.com/+JohnMueller/posts/HZf3KDP1Dm8.

[Nat98] Ira S. Nathenson. "Internet Infoglut and Invisible Ink: Spamdexing Search Engines with Meta Tags". In: *Harvard Journal of Law & Technology* 12 (1 1998), pp. 44–146. URL: http://papers.ssrn.com/sol3/papers.cfm?abstract%5C_id=1469706.

[Nel65] Theodor Holm Nelson. "Complex Information Processing: A File Structure for the Complex, the Changing and the Indeterminate". In: *Proceedings of the 1965 20th National Conference*. ACM '65. ACM, 1965, pp. 84–100. URL: http://doi.acm.org/10.1145/800197.806036.

[Nel81] Theodor Holm Nelson. *Literary Machines*. Swarthmore, 1981.

[Nel99] Theodor Holm Nelson. "Xanalogical Structure, Needed Now More Than Ever: Parallel Documents, Deep Links to Content, Deep Versioning, and Deep Re-use". In: *ACM Comput. Surv.* 31 (Dec. 1999). ISSN: 0360-0300. URL: http://doi.acm.org/10.1145/345966.346033.

[Not14a] Mark Nottingham. "RFC2616 is Dead". Blog. June 7, 2014. URL: https://www.mnot.net/blog/2014/06/07/rfc2616_is_dead.

[Not14b] Mark Nottingham. "What is the Web?" Personal blog. Dec. 4, 2014. URL: https://www.mnot.net/blog/2014/12/04/what_is_the_web.

[Otl34] Paul Otlet. *Traité de documentation. Le livre sur le livre*. 1934. URL: http://lib.ugent.be/fulltxt/handle/1854/5612/Traite_de_documentation_ocr.pdf.

[Pag+99] Lawrence Page et al. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Stanford InfoLab, Nov. 1999. URL: http://ilpubs.stanford.edu:8090/422/.

[Pru08] Eric Prud'hommeaux. "Media Types Issues for Text RDF Formats". Jan. 2008. URL: http://www.w3.org/2008/01/rdf-media-types.

[Sch15] Barry Schwartz. "Google: Leave Your Authorship Markup On Your Page". Oct. 1, 2015. URL: http://searchengineland.com/google-leave-your-authorship-markup-on-your-page-232090.

[Sma14] Ann Smarty. "Google Authorship Search Snippet: Evolution + Changes". Jan. 31, 2014. URL: http://de.slideshare.net/seosmarty/google-authorship-search-snippet-evolution-changes.

[SS13] Wolfgang Stock and Mechtild Stock. *Handbook of information science*. Trans. by Paul Becker. Berlin: de Gruyter, 2013. ISBN: 978-3-11-023499-2.

[Vel01] Kim H. Veltman. "Syntactic and Semantic Interoperability: New Approaches to Knowledge and the Semantic Web". In: *The New Review of Information Networking* 7 (2001).

Normative and Semi-normative Sources

[AWWW] Ian Jacobs and Norman Walsh. *Architecture of the Web*. W3C recommendation. W3C, Dec. 15, 2004. URL: http://www.w3.org/TR/2004/REC-webarch-20041215/.

[DBpedia3.7] DBpedia community. "DBpedia Ontology 3.7". URL: http://wiki.dbpedia.org/services-resources/ontology (visited on 06/2015).

[DCTERMS] DCMI Usage Board. *DCMI Metadata Terms*. June 14, 2012. URL: http://dublincore.org/documents/2012/06/14/dcmi-terms/.

[GeoNames3.1] Bernard Vatant. *GeoNames Ontology v3.1*. GeoNames, Oct. 29, 2012. URL: http://www.geonames.org/ontology/documentation.html.

[HTML1] CERN. *The first version of HTML*. 1992. URL: http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/MarkUp.html.

[HTML2.0] T. Berners-Lee and D. Connolly. *Hypertext Markup Language - 2.0*. IETF, 1995. URL: http://www.ietf.org/rfc/rfc1866.txt.

[HTML3.0] Dave Raggett. *HTML 3.0 Draft*. W3C, 1995. URL: http://www.w3.org/MarkUp/html3/.

[HTML3.2] Dave Raggett. *HTML 3.2 Reference Specification*. W3C, Jan. 14, 1997. URL: http://www.w3.org/TR/REC-html32.html.

[HTML4.01] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. *HTML 4.01 Specification*. W3C, Dec. 24, 1999. URL: http://www.w3.org/TR/1999/REC-html401-19991224/.

[HTML4.01-links] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. *[HTML 4.01] Links*. W3C, Dec. 24, 1999. URL: http://www.w3.org/TR/1999/REC-html401-19991224/struct/links.html.

[HTML4.01-types] Dave Raggett, Arnaud Le Hors, and Ian Jacobs. *Basic HTML data types*. W3C, Dec. 24, 1999. URL: http://www.w3.org/TR/1999/REC-html401-19991224/types.html.

[HTML5] Ian Hickson et al. *HTML5*. W3C, Oct. 28, 2014. URL: http://www.w3.org/TR/2014/REC-html5-20141028/.

[HTML5-links] Ian Hickson et al. *HTML5 - Links*. W3C, Oct. 28, 2014. URL: http://www.w3.org/TR/html5/links.html.

[IANA-link-types] Mark Nottingham, Julian Reschke, and Jan Algermissen. *IANA Link Relations*. Jan. 21, 2015. URL: http://www.iana.org/assignments/link-relations/link-relations.xhtml.

[MFREL] microformats. *HTML5 link type extensions*. Mar. 26, 2015. URL: http://microformats.org/wiki/index.php?title=existing-rel-values&oldid=64898.

[microdata] Ian Hickson. *HTML Microdata*. Working Group Note. W3C, Oct. 29, 2013. URL: http://www.w3.org/TR/2013/NOTE-microdata-20131029/.

[microformats2] microformats community. *microformats 2*. June 3, 2015. URL: http://microformats.org/wiki/index.php?title=microformats2&oldid=65040.

[OAI-ORE] Open Archives Initiative. *Objects Reuse and Exchange*. Aug. 14, 2014. URL: http://www.openarchives.org/ore/1.0/.

[OGP] The Open Graph community. *The Open Graph protocol*. Oct. 20, 2014. URL: http://ogp.me.

[OWL2] W3C OWL Working Group. *OWL 2 Web Ontology Language*. Second edition. W3C, Dec. 11, 2012. URL: http://www.w3.org/TR/2012/REC-owl2-overview-20121211/.

[OWL2QuickGuide] Jie Bao et al. *OWL 2 Web Ontology Language Quick Reference Guide (Second Edition)*. Second edition. W3C, Dec. 11, 2012. URL: http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/.

[RDF1.1] Richard Cyganiak, David Wood, and Markus Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. Version 1.1. World Wide Web Consortium. Feb. 25, 2014. URL: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[RDFa] Ben Adida et al. *RDFa Core 1.1 - Third Edition*. Recommendation. W3C, Mar. 17, 2005. URL: http://www.w3.org/TR/2015/REC-rdfa-core-20150317/.

[RDF/JSON1.1] Ian Davis, Thomas Steiner, and Arnaud J Le Hors. "RDF 1.1 JSON Alternate Serialization (RDF/JSON)". W3C working group note. Nov. 7, 2013. URL: http://www.w3.org/TR/2013/NOTE-rdf-json-20131107/.

[RDFS1.1] Dan Brickley and R.V. Guha. *RDF Schema 1.1*. W3C Recommendation. Version 1.1. W3C, Feb. 25, 2014. URL: http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[RDF/TURTLE1.1] David Beckett et al. *RDF 1.1 Turtle*. W3C, Feb. 15, 2014. URL: http://www.w3.org/TR/2014/REC-turtle-20140225/.

[RDF/XML-1999] Ora Lassila and Ralph R. Swick. *Resource Description Framework (RDF) Model and Syntax Specification*. W3C, Feb. 22, 1999. URL: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

[RFC1630] T. Berners-Lee. *Universal Resource Identifiers in WWW*. IETF, June 1994. URL: https://tools.ietf.org/html/rfc1630.

[RFC1958] B. Carpenter. *Archtectural Principles of the Internet*. June 1996. URL: https://tools.ietf.org/html/rfc1958.

[RFC2396] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifiers (URI): Generic Syntax*. IETF, Aug. 1998. URL: https://tools.ietf.org/html/rfc2396.

[RFC2616] R Fielding et al. *Hypertext Transfer Protocol – HTTP/1.1*. IETF, June 1999. URL: https://tools.ietf.org/html/rfc2616.

[RFC3023] M. Murata, S. St. Laurent, and D. Kohn. *XML Media Types*. IETF. Jan. 2001. URL: http://tools.ietf.org/html/rfc3023.

[RFC3986] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifier (URI): Generic Syntax*. IETF, Jan. 2005. URL: https://tools.ietf.org/html/rfc3986.

[RFC5988] Mark Nottingham. *Web Linking*. IETF. Oct. 2010. URL: https://tools.ietf.org/html/rfc5988.

[RFC7231] R. Fielding and J. Reschke. *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. IETF, June 2014. URL: http://tools.ietf.org/html/rfc7231.

[schema.org2.0] schema.org community. *Schema.org version 2.0*. May 13, 2015. URL: https://schema.org/version/2.0/.

[SPARQL1.1Query] Steve Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. W3C, Mar. 21, 2013. URL: http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[XFN1.1] Tantek Celik, Matthew Mullenweg, and Eric Meyer. *XFN 1.1 relationships meta data profile*. URL: http://gmpg.org/xfn/11 (visited on 06/2015).

[XML1.0] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen. *Extensible Markup Language (XML) 1.0*. W3C, Feb. 10, 1998. URL: http://www.w3.org/TR/1998/REC-xml-19980210.

Mailing Lists and Ticket Systems

[HTML-living-standard] WHATWG. *HTML Living Standard*. 2015. URL: https://html.spec.whatwg.org/multipage/ (visited on 06/2015).

[HTML-WG-95] Craig Hubley. "Re: REL and REV attributes (Was: More comments on HTML 3.0)". E-Mail to HTML-WG. Apr. 27, 1995. URL: https://listserv.heanet.ie/cgi-bin/wa?A2=html-wg;dp01Bw;199504271623220400.

[SCHEMAORG-I-545] vholland. "Issue 545 - Need a way to express room number in Event location". GitHub issue tracker. May 26, 2015. URL: https://github.com/schemaorg/schemaorg/issues/545.

[SEMANTIC-WEB-14] Hugh Glaser. "Re: RDF Graphs". Mail to: semantic-web@w3.org. Oct. 28, 2014. URL: https://lists.w3.org/Archives/Public/semantic-web/2014Oct/0268.html.

[TAG-ISSUE-14] W3C. "What is the range of the HTTP dereference function?" W3C Technical Architecture Group Issue Tracking. Sept. 10, 2014. URL: http://www.w3.org/2001/tag/group/track/issues/14.

[TAG-ISSUE-57] W3C. "Mechanisms for obtaining information about the meaning of a given URI". Sept. 10, 2014. URL: http://www.w3.org/2001/tag/group/track/issues/57.

[WEBONT-WG-01] Tim Finin. "Re: NAME: SWOL versus WOL". Mail to www-webont-wg@w3.org. Dec. 27, 2001. URL: https://lists.w3.org/Archives/Public/www-webont-wg/2001Dec/0169.html.

[WHATWG-06a] Ian Hickson. "Where did the "rev" attribute go?" July 5, 2016. URL: https://lists.w3.org/Archives/Public/public-whatwg-archive/2006Jul/0036.html.

[WHATWG-06b] Charles Iliya Krempeaux. "Where did the "rev" attribute go?" July 5, 2016. URL: https://lists.w3.org/Archives/Public/public-whatwg-archive/2006Jul/0037.html.

[WWW-TAG-05] Roy Fielding. "[httpRange-14] Resolved". Mail to www-tag@w3.org. June 18, 2005. URL: https://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html.

Tools

[cURL] cURL community. *cURL*. URL: http://curl.haxx.se/ (visited on 06/2015).

[Rhizomik] Grup de Recerca en Interacció Persona Ordinador i Integració de Dades. *RDF to SVG Form*. Universitat de Lleida. URL: http://rhizomik.net/html/redefer/rdf2svg-form/ (visited on 06/2015).

[sameAs] Hugh Glaser. *sameAs. interlinking the Web of Data*. URL: http://sameas.org/ (visited on 06/2015).

[VAPOUR] Parque Científico Tecnológico de Gijón. *VAPOUR. a Linked Data validator*. URL: http://validator.linkeddata.org/vapour (visited on 06/2015).