



Universidad
Carlos III de Madrid
www.uc3m.es

TESIS DOCTORAL

Conceptualization and Visualization of Tagging and Folksonomies

Autor:

Steffen Lohmann

Director/es:

**Prof. Dr. Maria Paloma Díaz Pérez
Prof. Dr.-Ing. Jürgen Ziegler**

DEPARTAMENTO DE INFORMÁTICA

Leganés (Madrid), Octubre 2013

Abstract

Tagging has become a popular indexing method for interactive systems in the past decade. It offers a simple yet effective way for users to organize an ever increasing amount of digital information for themselves and/or others. The linked user vocabulary resulting from tagging is known as folksonomy and provides a valuable source for the retrieval and exploration of digital resources.

Although several models and representations of tagging have been proposed, there is no coherent conceptualization that provides a comprehensive and precise description of the concepts and relationships in the domain. Furthermore, there is little systematic research in the area of folksonomy visualization, and so folksonomies are still mainly depicted as simple tag clouds. Both problems are related, as a well-defined conceptualization is an important prerequisite for the interoperable use and visualization of folksonomies.

The thesis addresses these shortcomings by developing a coherent conceptualization of tagging and visualizations for the interactive exploration of folksonomies. It gives an overview and comparison of tagging models and defines key concepts of the domain. After a comprehensive review of existing tagging ontologies, a unified and coherent conceptualization is presented that incorporates the best parts of the reviewed ontologies. The conceptualization is implemented in a standardized ontology language and enables scalable reasoning on folksonomies.

Based on the conceptualization, different tag cloud layouts and their ability to support users in information seeking tasks are investigated. Extensions to tag clouds are developed and combined into an analysis system, enhanced by sophisticated natural language processing and intuitive interaction techniques. Furthermore, existing work on visualizing folksonomies as graphs is analyzed and a new type of graph visualization tailored to the tag-based exploration of digital resources is introduced.

The work presented in this thesis has been successfully applied in different contexts. The developed conceptualization has been used in knowledge and requirements engineering, the tag cloud extensions have been applied in visual text analysis, and the novel graph visualization has proved useful in image retrieval and requirements analysis. These examples demonstrate the utility of the developed approaches beyond the topic of tagging and open up interesting questions for future research.

Resumen

El etiquetado se ha convertido en la última década en un método popular para la indexación de contenidos en sistemas interactivos. Ofrece una forma sencilla y eficaz para que los usuarios organicen una cantidad cada vez mayor de información para ser usada por ellos mismos y/o por terceros. El vocabulario de los usuarios resultante del etiquetado se conoce como folcsonomía y puede ser muy útil en la búsqueda y exploración de contenidos digitales.

Aunque se han propuesto varios modelos y representaciones para el etiquetado, todavía no hay una conceptualización coherente que describa completamente los conceptos y relaciones de la materia. Además, existe poca investigación sistemática en el área de visualización de folcsonomías, por lo que éstas se representan principalmente como simples nubes de palabras. Ambos problemas están relacionados, ya que una conceptualización precisa y bien definida es importante para el uso interoperable y la visualización de folcsonomías.

La tesis aborda estos dos problemas mediante el desarrollo de una conceptualización coherente de etiquetado y de visualizaciones para la exploración interactiva de folcsonomías. En primer lugar, se ofrece una visión general de los modelos de etiquetado y se definen los conceptos clave. Después de un análisis exhaustivo de las ontologías de etiquetado existentes, se presenta una conceptualización unificada que combina las mejores características de las ontologías analizadas. La conceptualización se implementa en un lenguaje de ontologías estandarizada y permite el uso de algoritmos de razonamiento eficientes.

Sobre la base de la conceptualización, se investigan diferentes formas de las nubes de etiquetas y las propiedades que ayudan a los usuarios en la búsqueda de información. Se desarrollan extensiones de las nubes de etiquetas y se las integra en un único sistema de análisis que utiliza técnicas avanzadas de interacción y de procesamiento del lenguaje natural. En un segundo paso, se analizan y se clasifican los enfoques de visualización gráfica de folcsonomías y, finalmente, se presenta un nuevo tipo de visualización gráfica que permite la exploración de contenidos digitales basada en etiquetas.

Los desarrollos de esta tesis se han aplicado con éxito en diferentes contextos. La conceptualización se ha utilizado tanto en la ingeniería del conocimiento y de requisitos. Las extensiones de las nubes de etiquetas se han integrado en un sistema de análisis del texto. Además, la visualización gráfica se ha aplicado tanto en la recuperación de imágenes como en el análisis de requisitos. Estas aplicaciones muestran los diversos usos de los desarrollos más allá de etiquetado y plantean preguntas interesantes para la investigación futura.

Zusammenfassung

Tagging ist im vergangenen Jahrzehnt zu einer beliebten Methode für die Indexierung von Inhalten in interaktiven Systemen geworden. Es bietet Nutzern eine einfache und effektive Möglichkeit, immer größere Mengen digitaler Informationen für sich selbst und/oder andere zu organisieren. Das durch Tagging entstehende Vokabular wird als Folksonomie bezeichnet und kann die Suche und Erschließung von digitalen Inhalten hilfreich unterstützen.

Obwohl mehrere Modelle und Repräsentationen für Tagging vorgeschlagen wurden, gibt es bislang keine umfassende Beschreibung der Konzepte und Beziehungen des Themengebiets. Zudem existiert wenig systematische Forschung im Bereich der Visualisierung von Folksonomien, so dass diese hauptsächlich in Form einfacher Wortwolken dargestellt werden. Beide Probleme bedingen sich gegenseitig, da eine präzise und konsistente Konzeptualisierung wichtige Voraussetzung für die interoperable Nutzung und Visualisierung von Folksonomien ist.

Diese Dissertation setzt sich mit den genannten Problemen auseinander, indem sie eine umfassende Konzeptualisierung für Tagging und Visualisierungen für die interaktive Analyse von Folksonomien entwickelt. Zunächst wird ein Überblick über Tagging-Modelle gegeben und werden wichtige Begriffe des Themengebiets definiert. Nach einer sorgfältigen Analyse vorhandener Tagging-Ontologien wird eine konsistente und wohldefinierte Konzeptualisierung präsentiert, die kompatible Teilbereiche der betrachteten Ontologien aufgreift und vereint. Die Konzeptualisierung ist in einer standardisierten Ontologiebeschreibungssprache implementiert und ermöglicht die Nutzung skalierbarer Reasoning-Verfahren auf Folksonomien. Ausgehend von der Konzeptualisierung untersucht die Dissertation anschließend verschiedene Darstellungsformen für Wortwolken und wie sie Nutzer bei der Informationssuche unterstützen. Es werden visuelle und interaktive Erweiterungen für Wortwolken entwickelt und zu einem Analysesystem integriert, das mit maschineller Sprachverarbeitung und intuitiven Interaktionstechniken ausgestattet ist. Darüber hinaus werden vorhandene Umsetzungen der Darstellung von Folksonomien als Graphen analysiert und schließlich eine neuartige Graphvisualisierung präsentiert, die die tag-basierte Erschließung von digitalen Inhalten ermöglicht.

Die entwickelten Ansätze wurden erfolgreich in verschiedenen Kontexten angewandt. Die Konzeptualisierung kam im Wissens- und Anforderungsmanagement zum Einsatz, die erweiterten Wortwolken halfen bei der visuellen Textanalyse und die neuartige Graphvisualisierung fand in der Bildersuche und Anforderungsanalyse Anwendung. Die Beispiele zeigen die vielfältigen Verwendungsmöglichkeiten der entwickelten Ansätze über das Themengebiet Tagging hinaus und führen zu interessanten Fragestellungen für zukünftige Forschung.

Acknowledgments

This thesis was conducted under a joint supervision (cotutelle) agreement between Universidad Carlos III de Madrid and Universität Duisburg-Essen. I would like to express my deep gratitude to my thesis supervisors Paloma Díaz and Jürgen Ziegler for making this possible and for their support, trust, and patience throughout the whole PhD process.

I am also very grateful to Thomas Ertl who gave me the opportunity and time to complete the thesis after starting work at the Institute for Visualization and Interactive Systems (VIS) of the University of Stuttgart. My grateful thanks are extended to Manuel Castro, Fabio Crestani, and Heinz Ulrich Hoppe who participated in the international PhD committee and examined the thesis and defense, as well as to Norbert Fuhr for his efficient support in coordinating the PhD committee and defense. Helpful feedback was also provided by the ‘prelectura’ committee at Universidad Carlos III de Madrid, consisting of Igacio Aedo, Eva Méndez, and Marco Romano.

Igacio Aedo and Paloma Díaz supported me very kindly during my time at the DEI Interactive Systems Group in Madrid, just as Andrea Bellucci, Alessio Malizia, and Teresa Onorati, which I much appreciated. For the great collaboration at the Interactive Systems Group in Duisburg, I would like to thank my former colleagues there, in particular Philipp Heim, Tim Hussein, Wolfgang Kaltz, Jörg Niesenhaus, Timo Stegemann, and Lena Tetzlaff. Likewise, I would like to thank the colleagues and students at VIS and VISUS in Stuttgart, where especially the discussions and joint work with Michael Burch, Florian Haag, Florian Heimerl, Steffen Koch, Simon Lange, Daniel Pompe, and Hansjörg Schmauder had some influence on the thesis. Special thanks go to my former colleague Martin Falk for kindly providing the \LaTeX template this thesis is based on.

I would also like to thank all other colleagues and friends from Duisburg, Madrid, and Stuttgart for the inspiring discussions and great time at all three locations. Last but not least, I owe my deepest gratitude to Nicole and my family for supporting and motivating me over the past few years.

Contents

1	Introduction	1
1.1	Problem Statement	2
1.2	Goals of the Thesis	3
1.3	Structure of the Thesis	4
1.4	Contributions	5
2	Tagging and Folksonomies	11
2.1	Basic Definitions	11
2.2	Advantages and Limitations	15
2.2.1	Advantages of Tagging and Folksonomies	15
2.2.2	Limitations of Tagging and Folksonomies	18
2.2.3	Semantifying Folksonomies	19
2.3	Core Elements	20
2.3.1	Resources	21
2.3.2	Users	22
2.3.3	Tags	23
2.4	Basic Models	27
2.4.1	Three-Layer Model of Tagging	28
2.4.2	Formal Folksonomy Representation	30
2.4.3	Beyond Basic Representations	32
3	Ontologies for Tagging and Folksonomies	35
3.1	Review of Tagging Ontologies	36
3.1.1	Newman’s Tag Ontology	36
3.1.2	Knerr’s Tagging Ontology	39
3.1.3	Ontology of Folksonomy	41
3.1.4	Social Semantic Cloud of Tags	42
3.1.5	Meaning of a Tag	45
3.1.6	Common Tag	47
3.1.7	Upper Tag Ontology	48
3.1.8	Tagora Tagging Ontology	51
3.1.9	NiceTag Ontology	53
3.2	Related Ontologies and Vocabularies	56
3.2.1	Annotea Schemas	57
3.2.2	Other Annotation Ontologies	58
3.3	Towards a Unified Tagging Ontology	61

4	The Modular Unified Tagging Ontology (MUTO)	65
4.1	Taggings	69
4.1.1	Tagging Comments	70
4.1.2	Tagging Source	70
4.1.3	Date and Time of Tagging	71
4.1.4	Private Tagging	72
4.1.5	Linking Taggings and Tags	73
4.2	Tags	73
4.2.1	Tag Label	74
4.2.2	Date and Time of Tags	75
4.2.3	Automatic Tags	76
4.2.4	Semantic Tags	77
4.2.5	Tag Relations	78
4.2.6	Tag Order	78
4.3	Users	79
4.3.1	Group Tagging	80
4.3.2	Access Control	81
4.4	Resources	81
4.5	Example Scenario	83
4.6	Discussion	85
4.7	Application	87
4.7.1	SWORE Ontology of the SoftWiki Project	88
4.7.2	Visual Notation for OWL (VOWL)	91
5	Tag Cloud Visualizations	95
5.1	Research on Tag Clouds	96
5.1.1	Types of Tag Clouds	96
5.1.2	Performance and Perception of Tag Clouds	97
5.2	Comparison of Tag Cloud Layouts	98
5.2.1	Experimental Design	99
5.2.2	Task-Layout-Performance	105
5.2.3	Analysis of Eye Tracking Data	107
5.2.4	Discussion	110
5.3	Prefix Tag Clouds	112
5.3.1	Prefix Tree Generation	113
5.3.2	Subtree Rendering	115
5.3.3	Tag Cloud Generation	116
5.3.4	Application Examples	118
5.3.5	Discussion	120

Contents

5.4	Considering Time and Tag Relations	121
5.4.1	Time-Varying Co-occurrence Highlighting	121
5.4.2	Application Example	123
5.4.3	Discussion	125
5.5	Interactive Analysis with Tag Clouds	126
5.5.1	Tag Cloud Explorer	127
5.5.2	Use Case	133
5.5.3	Qualitative User Study	137
5.5.4	Discussion	139
6	Folksonomy Graph Visualizations	141
6.1	Folksonomy Hypergraph Decomposition	141
6.2	Review of Existing Graph Visualizations	143
6.3	The ChainGraph Visualization	151
6.3.1	Creating a ChainGraph	152
6.3.2	Optimizing the Order in the Chains	153
6.3.3	Use Case	158
6.3.4	Evaluation of the ChainGraph	160
6.3.5	Discussion	167
7	Summary and Outlook	171
A	Referenced RDF Vocabularies	175
B	Items of the ChainGraph Evaluation	179
C	RDF/XML Serialization of MUTO	183
	Bibliography	195

List of Figures

Chapter 1

1.1	Illustration of the thesis contributions using the conceptual framework of Hevner et al. [127].	6
-----	---	---

Chapter 2

2.1	Web user interface to tag, comment, and save bookmarks with the social bookmarking service DELICIOUS.	13
2.2	Tag cloud of the 150 most often used tags on the photo sharing website FLICKR [36].	17
2.3	Three-layer model of tagging, illustrating the links that emerge between and within the layers.	29
2.4	Graph representations of tagging and folksonomies, (a) according to the three-layer model, (b) with relations between users and resources, and (c) with ternary relations connecting all three element sets.	30
2.5	Further graph representations of folksonomies, with (a) n -ary relations connecting more than one tag, (b) additional time information, and (c) links within the element sets.	32

Chapter 3

3.1	Tagging Ontology by Knerr [148].	40
3.2	Social Semantic Cloud of Tags by Kim et al. [144].	43
3.3	Local and global meanings represented with MOAT [144].	45
3.4	Basic structure of the CTAG ontology [18].	47
3.5	Concepts and relationships of the Upper Tag Ontology [78].	49
3.6	Tagora Tagging Ontology by Szomszor.	52
3.7	Tagging represented with the NiceTag ontology [161].	55
3.8	Annotea schemas for (a) annotations [136] and (b) bookmarks [152].	58
3.9	Annotation of an image with the Annotation Ontology [67].	59
3.10	Open Annotation Core Data Model [214].	60
3.11	Integration of TAGS, MOAT, and SCOT [144].	64

Chapter 4

4.1	Modular Unified Tagging Ontology (MUTO).	67
4.2	UML diagram of the MUTO ontology.	68
4.3	Example RDF graph depicting a private tagging.	83
4.4	Version 2.0 of the Semantic Web Ontology for Requirements Engineering (SWORE) [209].	89
4.5	Web platform for collaborative requirements engineering.	90
4.6	VOWL notation applied to the MUTO ontology [194].	92

Chapter 5

5.1	Examples of tag cloud layouts: (a) sequential layout with alphabetical sorting, showing the “most popular tags” on AMAZON.COM [37], (b) clustered tag cloud, displaying frequent tags of the famous speech “I Have a Dream” by M. L. King, Jr. [69].	97
5.2	Most user interfaces follow a grid layout that conceptually structures the interface by intersecting horizontal and vertical lines.	100
5.3	Tag cloud layouts for the corpus ‘France’: (a) sequential with alphabetical sorting, (b) circular with decreasing popularity, (c) clustered with thematic clusters, (d) reference with sequential, alphabetical sorting and no weighting of tags.	102
5.4	a) Paper demonstrator to explain tagging and tag clouds, b) Presentation of the tag clouds on a 17" TFT monitor with embedded eye tracking system.	104
5.5	Distribution of fixations in percent over the 5 x 5 subareas for the three tag cloud layouts. The coloring illustrates the pattern of the distribution.	108
5.6	a) Definition of areas of interest, b) Distribution of fixations over the quadrants.	108
5.7	Fixation density for the central-to-peripheral zones.	109
5.8	Distribution of fixation density in the center, first ring, and periphery zones for the three time periods.	109
5.9	Transforming a list of tags into a prefix tree: (a) Lexicographically ordered list of tags and their frequencies. (b) Node-link diagram illustrating the prefix tree structure, with tags shown in full length. (c) Node-link diagram of the vertically aligned prefix tree, with tags split into prefixes.	115

5.10	Alternative renderings of the same prefix subtree: (a) showing the complete tags, (b) showing only the prefixes.	116
5.11	Spiral placement of the prefix subtrees to create the circular tag cloud layout.	117
5.12	Differently shaped prefix tag clouds, each showing 150 words from publication titles indexed in the computer science bibliography DBLP: (a) most frequent words of all publication titles, b) most frequent words of publication titles containing the string “visual”.	119
5.13	Small tag cloud enhanced by (a) plain and (b) time-varying co-occurrence highlighting.	122
5.14	Visual interface showing a tag cloud where the time-varying co-occurrences of the tag ‘tennis’ are highlighted.	124
5.15	Tag Cloud Explorer consisting of the following components: (a) tag cloud visualization, (b) tag filter, (c) search box, (d) tag statistics panel, (e) info panel, (f) part-of-speech and named entity filters, (g) text viewer, (h) stopword editor, and (i) cloud control panel.	127
5.16	The tags <i>text</i> and <i>visualization</i> have been added to the tag filter and are highlighted in the resulting co-occurrence cloud.	131
5.17	Tag cloud for the text “The Hound of the Baskervilles”, with the tags colored according to their most frequent part-of-speech.	133
5.18	Initial tag cloud depicting one week of Reuters sports news in a sequential line-by-line layout with alphabetical sorting.	134
5.19	(a) Alphabetically ordered co-occurrence cloud for the tag <i>Olympic</i> , (b) Frequency ordered co-occurrence cloud for the tags <i>Olympic</i> and <i>champion</i> with highlighted person names.	135
5.20	(a) Frequency ordered co-occurrence cloud for the tags <i>Olympic</i> and <i>champion</i> filtered by person names, (b) Frequency ordered co-occurrence cloud for the tags <i>Olympic</i> , <i>champion</i> , and <i>Svetlana Masterkova</i>	136

Chapter 6

6.1	Complete decomposition of the folksonomy hypergraph (which is identical for the non-uniform and three-uniform variant due to Axiom 2).	142
-----	--	-----

6.2	Folksonomy subgraphs with (a) weighted edges instead of multi-edges, (b) edges filtered by weight, (c) edges restricted to a single user.	143
6.3	Visualizations of subgraph $G(T_R)$ by (a) Eaton [86] and (b) Halpin et al. [111].	146
6.4	Visualizations of subgraph $G(T_R)_u$ by (a) Zitvogel [257] and (b) Stefaner [229].	147
6.5	Visualizations of subgraphs $G(T_R)_u$ and $G(T_U)$ by (a) Michlmayr & Cayzer [187] and (b) Shaw [221].	149
6.6	Visualization of (a) subgraph $G(RT)$ by Oliphant [200] and (b) extended folksonomy graph by Kern et al. [140].	150
6.7	(a) Common graph and (b) ChainGraph visualization of the same folksonomy subgraph.	152
6.8	Resources are linked to tags which are bijectively mapped to chains.	153
6.9	An (a) optimal ChainGraph differs from a (b) random one in the ordering of the resources in the chains.	155
6.10	Using the ChainGraph to visually explore tag relations between images of Paris.	159
6.11	Using the ChainGraph to visualize shared metadata relations between software requirements.	167
6.12	Exploring a folksonomy with the ChainGraph visualization (a) on a powerwall and (b) on a multi-touch table.	168

List of Tables

Chapter 2

2.1 Common tag categories with tag examples from the domains of music (LAST.FM), photos (FLICKR), and bookmarking (DELICIOUS) [50].	25
---	----

Chapter 3

3.2 Reviewed tagging ontologies	37
3.4 Classification of tagging concepts.	62

Chapter 4

4.2 OWL sublanguages of the tagging ontologies.	87
---	----

Chapter 5

5.1 Graeco-Latin square design for the first task (presented to one group of participants): Combinations of tag cloud layouts (L) and tag corpora (C) for the four subgroups (SG) and four trials (Tr).	104
5.3 Performance values of the tag cloud layouts for the 12 participants (N) of each experimental group: Kruskal-Wallis (KW) mean rank, user votes, mean value, and standard deviation (in sec).	106
5.5 Findings on tag cloud perception and performance, contrasting related work and our own results (including both quantitative and qualitative observations).	111

Chapter 6

6.2 Reviewed folksonomy graph visualizations.	145
6.3 Connectivity matrices for the three graphs $G(RT)_i$ of the experiment with (a) 18, (b) 24, (c) 30 tag relations. The numbers in the table heads indicate the tags, the letters in the leftmost and rightmost columns indicate the resources.	162

Tables

6.4	Aggregated user ratings on the four evaluation dimensions. . .	165
6.6	Time and accuracy values for the three user tasks.	166
A.1	Names, namespace prefixes, and URIs of all referenced vocabularies.	177
B.1	Items of the ChainGraph evaluation grouped by category (negative items were positively coded), along with the ratings of the participants. The items were presented in random order. .	181

Introduction

Since its beginnings more than 20 years ago, the World Wide Web has experienced an impressive growth in the amount of information it contains. More and more digital resources are added to the web each day, making it not only for search and hosting services difficult to keep track and index all these resources. The *information explosion* [40] is also and especially a challenge for web users who try to efficiently retrieve and organize information that is relevant to them. Traditional organization principles based on taxonomies and folder hierarchies do not scale well with the massive amount of information that needs to be classified every day.

Driven by these developments, a new kind of manual indexing emerged in the last decade that is commonly known as *tagging*. In this indexing practice, users of an interactive system annotate digital resources with free-form text strings, so-called *tags*, in order to organize the resources for themselves and/or others. Having its roots in social bookmarking [113] and media sharing [139]¹, this way of indexing became very popular over the last couple of years and can now be found in many interactive systems, either as an additional feature or as a substitute for traditional forms of information management.

¹The social bookmarking service DELICIOUS [6] (formerly DEL.ICIO.US) was the first widely recognized software system starting to feature tagging in late 2003 [226, p. 162ff.], followed by the photo sharing website FLICKR [8] in 2004, adding much to its popularity [100, 240].

Tagging works in principle with all kinds of resources. What is considered a resource depends on the application context of the *tagging system*. It can be a web page bookmarked with a service like DELICIOUS [6], a photo or video on media sharing websites like FLICKR [8] or YOUTUBE [35], or an email in a webmail client like the one of GMAIL [10]. Even digital references of physical objects can be *tagged*, as long as they are uniquely addressable by the tagging system. For instance, books that are referenced in cataloging websites like LIBRARYTHING [12] or products in online shops such as AMAZON.COM [1] are also subject to tagging.

In contrast to other keyword-based indexing methods, the annotators are not professionals (e.g., authors, publishers, librarians, etc.) but common users. Furthermore, tagging breaks radically with most traditional forms of indexing by using neither a controlled vocabulary nor a hierarchical structure for classification. Instead, a tag can be any text string a user considers helpful in organizing a resource. Even though many tagging systems recommend tags, no terms are ‘forced’ onto users but they are free to use their own vocabulary. This vocabulary of the users along with the many links resulting from tagging is known as *folksonomy*.

1.1 Problem Statement

Tagging and folksonomies have become popular research topics in the last couple of years. They have been analyzed and utilized in a number of works, resulting in several interesting findings, for example, on tag use and distribution (see Chapter 2). However, tagging and folksonomies are still relatively young research topics that lack a shared understanding. Although several models and representations of tagging have been proposed, there is no coherent conceptualization that provides a precise and comprehensive description of the concepts and relationships in the domain.

A coherent conceptualization is not only important for a better domain understanding but also for the interoperable use of folksonomies. If the conceptualization is explicitly specified in an ontology, folksonomies can be accessed and shared in a uniform manner and validated against the ontology. This is especially useful for folksonomy visualizations, as it allows to develop them in a generic and reusable way, independently of individual tagging systems. In addition, the conceptualization describes the design space for folksonomy visualizations. It can help to identify limitations of existing approaches and to build new visualizations that address these limitations.

The lack of a proper conceptualization may be one reason why visualizations of folksonomies have so far mainly been limited to weighted tag lists, so called *tag clouds*. In fact, there is little systematic research in the area of folksonomy visualization that goes beyond simple tag clouds. However, a folksonomy is more than the sum of its tags. The ‘heart’ of folksonomies are the many links that evolve between the resources, tags, and users. These links are often used for recommendation, navigation, and filtering purposes, but their potential for the visualization of folksonomies is largely unexplored. Considering these links in folksonomy visualizations promises to reveal interesting patterns and relationships that are not visible in simple tag clouds. The close relation between the conceptualization and visualization of tagging and folksonomies is the main reason why both are jointly investigated in this thesis.

1.2 Goals of the Thesis

Based on the previous considerations, we identified a need for conceptualizations and visualizations that provide interoperable and reusable representations of tagging and folksonomies. Accordingly, the main goal of this thesis is twofold:

1. to develop a precise and coherent conceptualization of tagging;
2. to develop visualizations for the interactive exploration of folksonomies.

A major challenge in the development of the conceptualization and its implementation in an ontology is to find a good balance between comprehensiveness and comprehensibility. On the one hand, the ontology must be capable to represent different forms and aspects of tagging. On the other hand, it must not be too complex but remain easily understandable to the people who use it. This is why we aimed for a modular design consisting of a compact core ontology that can be extended when needed. Such a modular approach also helps to keep the conceptualization relatively stable with regard to future developments and advancements in the domain of tagging.

In addition, we built upon existing conceptualizations where appropriate. For this purpose, we looked at available tagging models and representations to identify parts that can be reused in the conceptualization. Since tagging combines different domains, we also looked at conceptualizations of related topics and examined their suitability to describe, for instance, the resources that are being tagged or the users who perform the tagging.

In contrast to the conceptualization, the goal of the visualizations is less to provide a comprehensive representation but rather to show the parts of the folksonomy that are of interest in a certain situation. Basically, we can distinguish two categories of folksonomy visualizations: (1) Those focusing on the elements of folksonomies (i.e., the resources, tags, users, etc.), and (2) those focusing on the relationships between these elements, i.e., the many links that emerge in tagging (see Section 2.4). An example of the first category are tag clouds, since they visualize the set of tags. Folksonomy graphs depicting relationships between tags, resources, users, and/or other elements belong to the second category.

Tag clouds and folksonomy graphs are also the two types of visualization addressed in this thesis. We will first have a closer look at tag clouds with the goal to learn more about how they perform and are perceived by users. We will then develop extensions for tag clouds that support the visual exploration and analysis of folksonomies and other data. Finally, we will look at folksonomy graphs with the goal to develop a new kind of graph visualization that supports the tag-based exploration of digital resources.

1.3 Structure of the Thesis

The thesis is structured as follows: Chapter 2 starts with an introduction into tagging and folksonomies. We define the key concepts and explain how they are related and used in tagging systems. We aim for short and precise definitions that provide a consistent terminological basis for the conceptualization. We also give an overview of the benefits and limitations of tagging and folksonomies in Chapter 2. Finally, we introduce basic models of tagging and discuss their strengths and weaknesses. We show that they describe only a small portion of the domain and argue for the need of a more comprehensive conceptualization.

In Chapter 3, we review conceptualizations of tagging available in the literature and on the web. We analyze and compare nine tagging ontologies in detail and also consider ontologies that have not been developed specifically with the goal to describe tagging but may be used for this purpose. Finally, we classify the found concepts and discuss the integration of the reviewed ontologies into a unified and coherent conceptualization.

Based on the review of Chapter 3, we developed the *Modular Unified Tagging Ontology (MUTO)* that is presented in Chapter 4. MUTO takes the ‘best’

parts of the reviewed ontologies and combines them into a coherent conceptualization. We describe MUTO in detail and discuss modeling alternatives. We provide an example scenario and report on two projects that make use of MUTO: the *Semantic Web Ontology for Requirements Engineering (SWORE)* and the *Visual Notation for OWL (VOWL)*.

The MUTO ontology provides the conceptual basis for the systematic investigation of folksonomy visualizations starting in Chapter 5. We first look into tag clouds and examine how different tag cloud layouts perform and are perceived by users. We then elaborate how tag clouds can be extended to better support the visual analysis of folksonomies and other data. We present *prefix tag clouds* and *time-varying co-occurrence highlighting* as two developments that increase the analytical power of tag clouds. The chapter concludes with a description of the *Tag Cloud Explorer*, an integrated analysis system that uses tag clouds as its main visualization method and equips them with advanced natural language processing and intuitive interaction techniques.

After looking into tag clouds in Chapter 5, we shift the focus to folksonomy graphs in Chapter 6. We first deal with the decomposition of folksonomy graphs into smaller subgraphs that can be reasonably visualized. Based on that, we classify existing work in this area and identify common limitations. To overcome these limitations, we developed the *ChainGraph*, a new kind of graph visualization that supports the visual exploration of folksonomies. We explain how a ChainGraph can be created and discuss two alternatives to optimize its layout. We evaluate the approach in a user study and discuss its strengths and limitations.

Chapter 7 concludes the thesis and provides an outlook on possible directions for future research. We summarize the presented work and discuss limitations, extensions, and application areas. We also point to open research challenges in the domain of tagging and beyond.

1.4 Contributions

In order to illustrate the main contributions of the thesis, we make use of the conceptual framework presented by Hevner et al. [127]. The framework helps to structure, understand, and evaluate research in information science, with strong references to design science and pragmatic research paradigms.

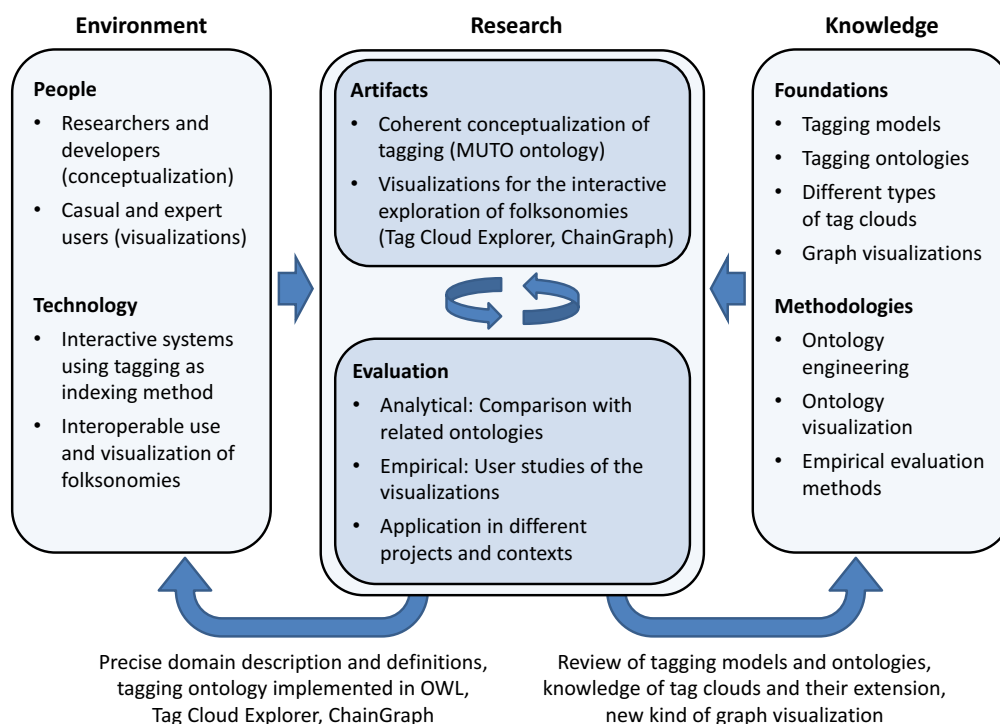


Figure 1.1 — Illustration of the thesis contributions using the conceptual framework of Hevner et al. [127].

We slightly adapted the framework to fit within the context of this work and applied it to the thesis contents as shown in Figure 1.1.

The framework is composed of three main components: 1) the *environment* defining the problem space and consisting of the *technology* and the *people* who use it, 2) the *research* that is conducted, and 3) the *knowledge base* providing the “raw materials from and through which [...] research is accomplished”, in particular “foundations and methodologies” [127].

The *research* component consists of two complementary activities: The development of *artifacts* and their *evaluation* with appropriate methods. The main artifacts of the thesis are the coherent conceptualization of tagging and visualizations for the interactive exploration of folksonomies. Depending on the type of artifact, we used different methods for the evaluation. The MUTO ontology implementing the conceptualization was analytically evaluated by comparing it with related ontologies and discussing its strengths and limitations. The Tag Cloud Explorer and the ChainGraph were empirically

evaluated in user studies. In addition, all three artifacts have been applied in different contexts and projects: The MUTO ontology has been used in knowledge and requirements engineering, the Tag Cloud Explorer in visual text analysis, and the ChainGraph in image retrieval and requirements analysis.

Regarding the *environment* component of the framework (see Figure 1.1), the research conducted in this thesis mainly addresses two groups of people: On the one hand, these are the researchers and developers who benefit from a precise and coherent conceptualization of tagging. The MUTO ontology does not only contribute to a better understanding of tagging but also allows for the interoperable use and visualization of folksonomies. On the other hand, there are the users who benefit from visualizations for the interactive exploration of folksonomies and other data. While the intuitive tag cloud extensions support both casual and expert users, the ChainGraph visualization requires some explanation and is therefore mainly intended for expert users.

The *foundational knowledge* of this work consists of tagging models and ontologies that need to be carefully reviewed in order to create a coherent conceptualization of tagging. In turn, the insights gained in the review and conceptualization extend the knowledge on tagging and related ontologies. Furthermore, knowledge on the visualization of folksonomies is needed to identify limitations and develop extensions and improvements. In particular, different types of tag clouds and graph visualizations are studied in the thesis. The results of the empirical study on tag cloud perception and performance extend the body of knowledge in the field. The ChainGraph as a new kind of graph visualization is of interest to the information visualization community and may inspire future work on visualizing graphs. Finally, knowledge on *methodologies* and best practices from the fields of ontology engineering and visualization is needed to develop and graphically represent the MUTO ontology. For the user studies conducted in the thesis, knowledge about empirical evaluation methods is required.

The conceptual framework does not only summarize the contributions of the thesis, but it also illustrates the interdisciplinary character of the work. While most of the contents belong to the field of computer science, knowledge and methods from other disciplines are incorporated in order to study the conceptualization of tagging and the visualization of folksonomies in a holistic way.

Parts of the work presented in this thesis have already been disseminated in the form of conference and workshop papers, journal articles, and book chapters. The relevant publications are referenced in footnotes at the beginning of each chapter. Additionally, all publications containing material of this thesis are listed in the following:

- S. Lohmann, J. Ziegler, L. Tetzlaff: Comparison of Tag Cloud Layouts: Task-related Performance and Visual Exploration. In: Proceedings of the 12th IFIP TC 13 Conference on Human-Computer Interaction (INTERACT '09), pp. 392–404. Springer, 2009.
- S. Lohmann, P. Heim, L. Tetzlaff, T. Ertl, J. Ziegler: Exploring Relationships between Annotated Images with the ChainGraph Visualization. In: Proceedings of the 4th International Conference on Semantic and Digital Media Technologies (SAMT '09), pp. 182–185. Springer, 2009.
(*Best paper award*)
- S. Lohmann, J. Niesenhaus, P. Heim, J. Ziegler: Fostering Knowledge Flow and Community Engagement in the Development of Interactive Entertainment. Journal of Universal Computer Science (J.UCS), 15(8), pp. 1722–1734, 2009.
- P. Heim, S. Lohmann: ChainGraph: A New Approach to Visualize Shared Properties in Resource Collections. In: Proceedings of the 9th International Conference on Knowledge Management and Knowledge Technologies (I-KNOW '09), pp. 106–114. TU Graz, 2009.
- S. Lohmann, T. Riechert: Adding Semantics to Social Software Engineering: (Re-)Using Ontologies in a Community-oriented Requirements Engineering Environment. In: Software Engineering 2010 – Workshopband, pp. 485–494. GI, 2010.
- S. Lohmann: Social Tagging im E-Learning: Einblick, Überblick, Ausblick. In: M. H. Breitner, F. Lehner, J. Staff, U. Winand (Eds.) E-Learning 2010, pp. 199–214. Springer, 2010.
- S. Lohmann: Learners as Metadata Source? A Three-Layer Model of Collaborative Content Annotation. In: Interdisciplinary Approaches to Technology-enhanced Learning, pp. 279–286. Waxmann, 2011.
- S. Lohmann, P. Díaz, I. Aedo: MUTO: The Modular Unified Tagging Ontology. In: Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS '11), pp. 95–104. ACM, 2011.

- S. Lohmann, P. Díaz: Representing and Visualizing Folksonomies as Graphs: A Reference Model. In: Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12), pp. 729–732. ACM, 2012.
- S. Lohmann, M. Burch, H. Schmauder, D. Weiskopf: Visual Analysis of Microblog Content Using Time-Varying Co-occurrence Highlighting in Tag Clouds. In: Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12), pp. 753–756. ACM, 2012.
- S. Lohmann: Semantic Models for Interactive Systems: The Case of Tagging and Folksonomies. In: T. Hussein, H. Paulheim, S. Lukosch, J. Ziegler, G. Calvary (Eds.) Semantic Models for Adaptive Interactive Systems, pp. 169–185. Springer, 2013.
- M. Burch, S. Lohmann, D. Pompe, D. Weiskopf: Prefix Tag Clouds. In: Proceedings of the 17th International Conference on Information Visualisation (IV '13), pp. 45–50. IEEE, 2013.
- S. Negru, F. Haag, S. Lohmann. Towards a Unified Visual Notation for OWL Ontologies: Insights from a Comparative User Study. In Proceedings of the 9th International Conference on Semantic Systems (I-SEMANTICS '13), pp. 73–80. ACM, 2013.
- F. Heimerl, S. Lohmann, S. Lange, T. Ertl: Word Cloud Explorer: Text Analytics based on Word Clouds. In: Proceedings of the 47th Hawaii International Conference on System Science (HICSS '14), pp. 1833–1842. IEEE, 2014.

Tagging and Folksonomies

As tagging and folksonomies are still relatively new topics to computer science, we will give some background knowledge in this chapter. After a definition of the key concepts, we will summarize the pros and cons of tagging and folksonomies. We will then take a closer look at the core elements of tagging and provide models that form the basis for the later conceptualization and visualizations.

2.1 Basic Definitions

Several definitions for concepts in the domain of tagging are used in the literature. The problem is that many of them are hardly compatible. They do not only use different terminology [205, p. 153ff.] but are partly not even based on the same understanding of tagging and folksonomies, as it has been criticized by Vander Wal [241].

This was the main reason why we decided to create own definitions in cases where we could not find conceptually and terminologically compatible ones. Although our understanding of the concepts can, of course, also differ from the viewpoints of others, this guarantees conceptual and terminological consistency, which is important for a coherent domain description. In addition, we aim to contribute to a better understanding and framing of tagging with

Parts of this chapter have already been published in [165], [166], and [168].

our definitions. We therefore published them under a creative commons license on the web to make them available to a broader audience and allow others to use, adapt, and extend them.¹

In the following, we provide short definitions and explanations for the most basic concepts. Further concepts are defined later in this chapter.

Definition 1. A *tag* is an arbitrary text label associated with a resource.

Definition 2. *Tagging* is the annotation of resources by users with tags.²

Definition 3. A *tagging system* is an interactive system that uses tagging for indexing.

The term *tagging system* must be taken with caution, as no system is purely made for tagging but always serves some other purpose (e.g., social bookmarking, photo sharing, etc.). However, we will use the term, as it has proven its worth to give this class of systems a name. It is thus comparable to terms like *distributed system* or *web-based system* that also emphasize one out of many aspects of a system but help to frame the work. Since we focus on tagging in this thesis, the term ‘tagging system’ is quite appropriate in our case.

A popular tagging system is the social bookmarking service DELICIOUS already mentioned in the introduction. It offers a website to save, share, organize, and search web resources (or more precisely bookmarks, i.e., the URIs of web resources). The bookmarks can be annotated with tags and are accessible from anywhere via the DELICIOUS website.

Figure 2.1 shows one of the web user interfaces to tag, comment, and save the URIs of web pages with DELICIOUS. It can be installed as an extension to the web browser. The title and URL of the web page to be bookmarked are automatically gathered by the browser extension and pre-populated into the corresponding text boxes. The only user input required for tagging the web page is a comma-separated list of tags entered in another text box.³

¹ The definitions are publicly available at <http://www.socialtagging.org>.

² The terms *tagging* and *tag* can have different meanings in computer science. In this work, we mainly refer to the meanings as given in Definition 1 and Definition 2, with only a few exceptions, such as in the term *part-of-speech tagging* we use in Chapter 5.

³ Other browser extensions and bookmarklets for DELICIOUS use spaces as tag delimiters or let the user enter every single tag in a separate text box.

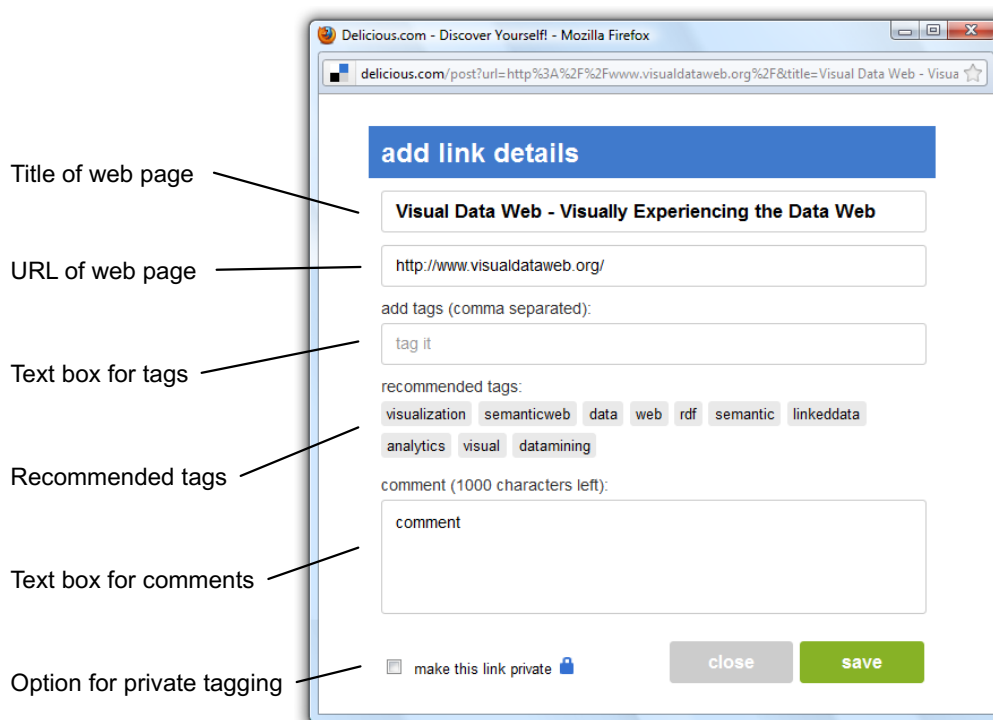


Figure 2.1 — Web user interface to tag, comment, and save bookmarks with the social bookmarking service DELICIOUS.

It is important to distinguish between the indexing method of tagging and its result. While a single annotation resulting from tagging is also called *tagging*, the sum of the annotations is called *folksonomy*.

Definition 4. A single annotation resulting from tagging is also called *tagging*. The sum of these taggings is the *folksonomy*.

The term folksonomy, a combination of “folk” and “taxonomy” [240], was first used by Vander Wal in 2004. Although there has been some debate about the adequacy of this portmanteau [113, 227, 205, p. 153ff.], it has become the de-facto term to denote the vocabulary and link structure that result from tagging. Therefore, we will also use it in this work. Vander Wal defines folksonomy as follows:

Definition 5. “Folksonomy is the result of personal free tagging of information [...] for one’s own retrieval. The tagging is done in a social environment

(usually shared and open to others) [...] by the person consuming the information.” [238, 240]

On the one hand, the definition emphasizes that tagging is usually performed by the person “consuming” the information for his or her “own retrieval”. On the other hand, it points out that tagging often happens in a “social environment”, which refers to open online environments like DELICIOUS in this context. Tagging in such environments has a ‘social’ component, as the tags of the users are typically also available to others. This powerful aspect of combining the tags from different users led to the coinage of terms like *social tagging* or *collaborative tagging*.⁴

Definition 6. *Social tagging* is tagging in an open online environment where the tags of the users are available to others.

In social tagging, tags are often used for recommendation. For instance, the DELICIOUS user interface shown in Figure 2.1 recommends a number of tags that have previously been entered by other DELICIOUS users who bookmarked the same web page.

In addition, many social tagging services provide an opportunity to mark those taggings as private that the user does not want to share with others. In Figure 2.1, this is possible by activating the corresponding check box at the very bottom of the user interface. Such *private tagging* can be regarded as the opposite of social tagging, as it is only visible to its creator.

Definition 7. *Private tagging* is tagging that is only visible to its creator (unless the creator has explicitly granted access to others).

There may be different forms of private tagging, such as a less strict variant where the friends of a user have also access to his or her private taggings, as it can, for instance, be configured on FLICKR. However, a comprehensive conceptualization of tagging must at least carefully distinguish between public and private taggings.

The sum of the taggings of a single user is sometimes called *personomy* [207, 129], independently of whether the taggings are private or public. Strictly

⁴Strictly speaking, tagging in open online environments is rather a *collective* than a *collaborative* activity (cp. [239]). Hence, we prefer terms like *collective tagging* or *social tagging* over the term *collaborative tagging*, which can nevertheless often be found in the literature.

speaking, a folksonomy can also denote purely the taggings of a single user, since it is simply “the result of personal free tagging” according to Definition 5, without any further quantification. Hence, a personomy is always a folksonomy (i.e., a personal folksonomy), while a folksonomy can contain one or more personomies (i.e., $personomy \supseteq folksonomy$). However, the term ‘personomy’ makes immediately clear that the taggings of a single user, not of many users, are meant.

Definition 8. *Personomy* is the sum of the taggings of a single user.

2.2 Advantages and Limitations

There has been much debate about the pros and cons of tagging and folksonomies (cp. [205, p. 247ff.]). They are often compared to classical indexing methods, such as controlled vocabularies or taxonomies, and it is sometimes overlooked that they are usually not a replacement but a complement to these indexing methods. In the following, we summarize some of the main advantages and disadvantages of tagging and folksonomies:

2.2.1 Advantages of Tagging and Folksonomies

Two main drivers for the rapid spread of tagging are apparently its simplicity and flexibility. They meet the needs of users who are looking for a quick way to manage an ever-growing amount of information. Tagging can also be a very fast indexing method, as illustrated by the user interface of DELICIOUS in Figure 2.1. In accordance with the literature [226, 205, 183], we can therefore summarize the main advantages for the users of tagging systems as follows:

- Tagging is simple: Users simply enter some arbitrary text labels, separated by spaces, commas, or others delimiters (depending on the design of the user interface).
- Tagging is fast: Entering some arbitrary text labels requires few interactions and little indexing time (e.g., compared to choosing an appropriate place in a folder hierarchy).
- Tagging is flexible: The text labels can consist of whatever character sequence a user considers helpful to organize resources.

These benefits are, of course, influenced by the usability of the tagging system and the tagging skills of the user, but they are usually true to a certain extent.

Folksonomies provide descriptive metadata that would otherwise often not exist. Especially for non-text-based resources, such as photos, videos, or music files, it is difficult to automatically extract descriptive metadata, in particular metadata that is based on inferred understanding. The tags added by the users partly contain such metadata. They can be used to retrieve, navigate, recommend, or explore digital resources. They reflect the language of the users, not of authorities, and provide linguistic variation [205, p. 212ff.]. Thus, folksonomies can help to bridge the semantic gap between different user types and reduce the *vocabulary problem* often reported in human-computer interaction: “users [...] use the wrong words and fail to get the actions or information they want” [97]. Furthermore, folksonomies are said to be sensitive to trends and developments, as they quickly adopt terminology that arises in a subject area [205, p. 213].

The main benefits of folksonomies for users and developers of tagging systems can therefore be summarized as follows, again in accordance with the literature [183, 226, 205, p. 212ff.]:

- Metadata source: Folksonomies provide metadata that would otherwise often not exist.
- User language: Folksonomies ‘speak’ the language of the users which broadens access to information.
- Dynamic vocabulary: The vocabulary of folksonomies is said to quickly adopt new terms and trends.

The linguistic variety of folksonomies is easily recognizable in many *tag cloud* visualizations. A typical tag cloud is given in Figure 2.2. It shows the “all time most popular tags” of the photo sharing website FLICKR [36], i.e., a selection of 150 tags that have most often been used to index photos on FLICKR (until August 2010, when this screenshot of the tag cloud was taken).

Definition 9. A *tag cloud* displays “a certain number of most often used tags [...]. A tag’s popularity is expressed by its font size (relative to the other tags) [...]. Sometimes, further visual properties, such as the font color, intensity, or weight, are manipulated [...]. Next to their visualization function,

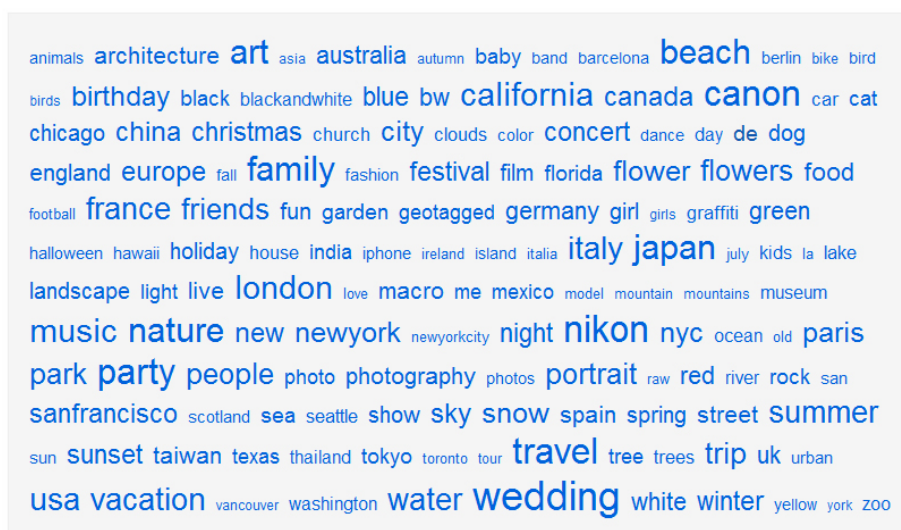


Figure 2.2 — Tag cloud of the 150 most often used tags on the photo sharing website FLICKR [36].

tag clouds are also navigation interfaces, as the tags are usually hyperlinks leading to a collection of items they are associated with.” [176]

There are several tags in the cloud that may denote the same or very similar things, such as ‘holiday’ and ‘vacation’ or ‘trip’ and ‘travel’ (see Figure 2.2). There are also cases of multilingualism, such as ‘italy’ which is listed with its English and Italian name (‘italia’), and inflected word forms, such as plural forms of terms that are also given in singular (e.g., ‘birds’, ‘flowers’, and ‘trees’). The tag ‘newyorkcity’ and its abbreviation ‘nyc’ both refer to the City of New York, whereas the U.S. state of New York is either written as one tag (‘newyork’) or as two (‘new’ and ‘york’). Likewise, the expression ‘black and white’ can be found in three different variations in the tag cloud, as one tag (‘blackandwhite’), as two (‘black’ and ‘white’), and abbreviated as ‘bw’.

Since the tag cloud displays only a small selection of tags, these are just a few of the many linguistic variations that can typically be found in folksonomies. Other common examples, such as misspellings (e.g., typing errors), spelling variants (e.g., British vs. American English), or personal style (e.g., ‘kidz’ instead of ‘kids’), are not visible in this highly filtered view but can often be found in folksonomies [110].

2.2.2 Limitations of Tagging and Folksonomies

On the one hand, the linguistic variety of folksonomies can broaden access to digital resources (e.g., both ‘holiday’ and ‘vacation’ may return relevant results in the FLICKR case of Figure 2.2). On the other hand, they cause problems that are usually to be avoided in indexing, for example, by using controlled vocabularies. For instance, synonym tags can negatively affect the *recall* of queries against the folksonomy, since only a part of the possibly relevant resources is returned (e.g., just photos tagged with ‘holiday’ and not those tagged with ‘vacation’).

Next to the problem of synonymy, ambiguity can demand a lot of guessing and filtering from the users. One such example in the FLICKR tag cloud is the tag ‘party’ that can have multiple meanings, among others it can refer to a ‘social gathering’ or ‘political organization’. Since the tag cloud is from a photo sharing website, it is likely that most of the photos tagged with ‘party’ are about social gatherings, leading to many irrelevant results for someone who is looking for photos of political parties. Thus, ambiguity lowers the *precision* of searches against the folksonomy, making it particularly difficult to search for a meaning of a tag that is not popular in a certain context.

In addition, the meaning of tags such as ‘bw’ or ‘macro’ is not clear and can only be guessed from the context. Typically, the user would need to select further thematically related tags to clarify the meaning of ambiguous tags. However, a multiple tag search is also restrictive, as it only returns resources that are indexed with all the tags entered by the user.

Generalizing from the above examples, the main limitations of folksonomies can be summarized as follows (cp. [183, 102, 79, 180]):

- *Synonymy*: Different tags have the same or a very similar meaning, as it may often be the case for the tags ‘holiday’ and ‘vacation’ from the FLICKR tag cloud of Figure 2.2. This includes abbreviations, such as the tag ‘bw’ that can also be found in its long form ‘blackandwhite’ in the tag cloud. Related to this are cases of multilingualism (‘italy’ vs. ‘italia’), spelling differences (e.g., British vs. American English), misspellings, or different treatments of multiword tags (‘newyork’ vs. ‘new york’).
- *Ambiguity*: Identically spelled tags have different meanings. This includes all forms of ambiguity that can typically be found in natural

language, such as homonyms, homographs, heteronyms, and polysemes, including ambiguous abbreviations.

Golder and Huberman [102] point to another limitation of folksonomies compared to controlled vocabularies that they call “basic level problem”. It denotes the fact that tags in a folksonomy “vary along a continuum of specificity ranging from very general to very specific” [102]. This results as “people may consider terms at different levels of specificity to be most useful or appropriate for describing” a resource. For instance, the tag ‘animals’ is more general than ‘birds’ in the FLICKR tag cloud of Figure 2.2, and there are also more specific tags for birds on FLICKR that are not visible in this highly aggregated view, such as ‘pigeon’, ‘dove’, or ‘citydove’. These varying degrees of specificity can also negatively affect indexing and retrieval, as it is well known from other application contexts, such as web search.

2.2.3 Semantifying Folksonomies

There are several approaches that try to overcome some of these limitations. Basically, they can be divided into two groups:

The first group of approaches aims to reduce the problems by automatically transforming folksonomies into more controlled vocabularies, for instance, by using statistical and/or lexical methods. Common techniques include the analysis of tag co-occurrences [74, 79], tag clustering [46], deriving hierarchies from folksonomies [128], and automatic mappings of tags to well-defined entities from lexical databases [74, 180], such as DBPEDIA [51] or WORDNET [191].

The second group of approaches involves the users to overcome some of the limitations. For instance, they let the users explicitly resolve ambiguity, either during the act of tagging [180, 202] or afterwards. The latter has been described with the image of *tag gardening* where the user ‘cares’ about his or her personomy like a gardener, i.e., he or she keeps it clean and well organized, for instance, by defining semantic relations between tags [103, 206]. The general ideas in these works are roughly the same as in the automatic approaches, with the difference that they are manually performed. However, one has to be careful with the application of such manual approaches, as they can change the nature of tagging and hence risk to lower the benefits of tagging described above (see Section 2.2.1).

The manual approaches are sometimes combined with techniques from the first group in a semi-automatic manner [74, 206, 202]. A popular example

is the disambiguation of tags that is often supported by an automatically generated list of suggestions. Yet, it is ultimately the user who selects the suggestion that matches the meaning he or she intended. This approach of linking tags to well-defined entities of the web [180, 202] is known as *semantic tagging* [180]. Since it is not the whole tagging that is disambiguated but the individual tags, we call these tags *semantic tags*. Note that a semantic tagging must not consist purely of semantic tags but can also include ‘ordinary’ ones, i.e., whenever a tagging is composed of at least one semantic tag, we call it a semantic tagging.

Definition 10. A *semantic tag* is a tag with an explicitly defined meaning (e.g., expressed by a link to a well-defined entity).

Definition 11. *Semantic tagging* is tagging with semantic tags.

Examples of systems that support semantic tagging are SEMKEY [180] or FAVIKI [7]. A comprehensive conceptualization of tagging must consider these approaches of ‘semantifying’ folksonomies. It must at least be capable to represent semantic tags and tag relations in order to be useful for these approaches.

2.3 Core Elements

Tagging consists of three sets of elements that form the basis for its conceptualization [188, 182, 111, 226, 128, 205, p. 157]:

1. *Resources* that are being tagged.
2. *Tags* that are associated with the resources.
3. *Users* who perform the tagging.

Though these elements are differently named in the literature, their semantics and relationships are always the same: One or more *users* (or people, actors, etc.) annotate *resources* (or objects, instances, etc.) with one or more *tags* (or keywords, labels, etc.).

Before we discuss the link structure of tagging in more detail, we will first take a look at each of the three element sets to get a better understanding of what need to be conceptualized.

2.3.1 Resources

The first set of elements are the resources that are being tagged by the users. As mentioned in the introduction, these resources can be anything, as long as they are uniquely addressable by the tagging system. This is why we prefer the term *resource* over other terms sometimes used in the literature, such as *content* [102, 190] or *document* [252]. Other authors use terms like *object* [108, 74, 128], *item* [56], or *instance* [188] instead, which are more appropriate alternatives. However, the term *resource* fits especially well with the terminology of the World Wide Web, which is the most popular application area of tagging and also focused in this thesis.⁵

Definition 12. A *resource* is anything that is uniquely addressable within a software system or computer network. In web contexts, it is broadly defined as “whatever might be identified by a URI. Familiar examples include an electronic document, an image, a source of information with a consistent purpose [...], a service [...], and a collection of other resources. A resource is not necessarily accessible via the Internet; e.g., human beings, corporations, and bound books in a library can also be resources.” [48, p. 5]

This definition emphasizes that anything can be resource, as long as it has a unique identifier, in web contexts a URI (or IRI [84]). Books, events, or places can all be subject to tagging, and even “human beings” are explicitly included in the definition. This is reflected in the idea of *people tagging* where tagging is used to “classify people by their skills and projects” [91]. Examples of systems for people tagging are, for instance, given in [92] and [57].

Ultimately, the tagging system determines what is considered a resource. Many tagging systems are specialized on a certain type of resource, such as photos (e.g., FLICKR), videos (e.g., YOUTUBE), slideshows (e.g., SLIDESHARE [22]), or books (e.g., LIBRARYTHING), while others, like most social bookmarking systems, can in principle be used to tag anything with a URI (though they are in fact also specialized, for instance, on web pages).

Some tagging systems allow to tag parts of resources, such as single slides from a slideshow [73] or selected scenes from a video [212, 33] – a tagging practice that is sometimes called *deep tagging* [43, 226, p. 185f.]. However, parts of a resources can be considered resources on their own, as long as they can be equipped with a unique identifier (e.g., a URI).

⁵The term *resource* has quite some history in web contexts, as outlined by Berners-Lee [47].

2.3.2 Users

The second set of elements are the users who perform the tagging. In this context, the term *user* denotes all people who use the tagging system to index or retrieve resources, independently of their role and motivation. Other terms sometimes used in the literature are *actor* [188, 74], *people* [226], or *tagger* [108].

In most cases, the users are information seekers who want to index resources for personal organization and later retrieval, such as in social bookmarking systems like DELICIOUS. However, a user can also be the author of a resource who tags the resource for himself and/or others, such as in media sharing websites like FLICKR. In some cases, users may even be professional indexers who tag resources to increase their findability, such as in e-commerce applications like AMAZON.COM. However, tagging by professionals, not common users, is the exception and would usually not be called tagging but *free indexing* or alike.

It is important to note the distinction between *user* and *user account*. While each tagging is associated with one user account, this does not mean that users can have only one account. Instead, users may create several accounts in a tagging system to separate, for instance, leisure- and work-related taggings.

A special case is *group tagging* where the indexing is done by a group of users. As an example, think of a team of people that search and index resources for a project. While each user performs these activities individually, the taggings of all team members are grouped in a shared project view.

Definition 13. *Group tagging* is tagging performed by a group of users.

Basically, three different forms of group tagging can be distinguished: The simplest is to agree on a tag that uniquely identifies the group. The tag can then be used to aggregate taggings from the individual group member accounts into a shared view. An alternative is to create a shared user account that all group members can log in to perform the tagging. The third and most advanced solution is to create a group account that the user accounts of all group members are linked to (e.g., as possible in FLICKR or BIBSONOMY [4]).

2.3.3 Tags

The third set of elements are the tags, i.e., arbitrary text labels associated with the resources (see Definition 1). Tags can be common words, slang, abbreviations, emoticons, ratings, or text strings that are only meaningful to the person who uses them. Theoretically, there are no restrictions on the number and kind of tags a resource can be annotated with. Practically, however, the possibilities are restricted by the tagging system, such as the maximum number of tags per tagging⁶ or the supported character encoding (ASCII, unicode, etc.). Some tagging systems (e.g., AMAZON.COM) filter objectionable words and many convert tags to lowercase for different reasons (e.g., aesthetics, aggregation, homogeneity, etc.). Sometimes, certain characters are reserved for special purposes and can thus not be used in ‘normal’ tags, such as spaces or commas if used as delimiters. Apart from that, no restrictions are usually made on what character string can be used as tag.

As the space character is often used as delimiter to separate tags, several workarounds have been invented by users of tagging systems to cope with multiple word tags (e.g., compounds such as ‘new york’). A very common practice is to simply omit the space and write the words as one tag (cp. the tag ‘newyork’ in the FLICKR tag cloud of Figure 2.2). Camel case (‘NewYork’) usually does not work, since the tags are converted to lowercase in many tagging systems. Other common workarounds are the use of abbreviations (‘nyc’) or separate tags (‘new’, ‘york’), as also visible in the FLICKR tag cloud of Figure 2.2. However, adding compounds as several tags (e.g., ‘new’ and ‘york’) can lead to ambiguities in the folksonomy, as discussed in Section 2.2.2.

Some systems try to solve this problem by using commas instead of spaces as tag delimiters. Others require to add quotes to compound tags or to enter each tag in a separate text box. However, such approaches run the risk of slowing down tagging speed and therefore reducing one of the main advantages of tagging (see Section 2.2.1). Furthermore, they may affect other components of the tagging system that have not been designed for multiple word tags. For instance, tags in a tag cloud are usually separated by whitespace so that it is not easily distinguishable whether a tag is a multiple word tag purely by looking at the tag cloud.

⁶ At time of writing, DELICIOUS allows a maximum of 50 tags per tagging (which is more than sufficient for common bookmarking), and FLICKR limits the number of tags per photo to 75 [9].

Although tags are freely chosen text labels, they are not free from convention. Some types of tags can often be found in tagging systems. For instance, the self-referential tag ‘me’ is frequently used in media sharing to refer to photos or videos that show the users themselves [183] (e.g., it is also included in the FLICKR tag cloud of Figure 2.2), whereas tags like ‘toread’ or ‘todo’ are popular in social bookmarking to assign a future task to resources (e.g., to read an interesting web blog article at a later time).

Tag Categories

In an analysis of more than 165,000 tags from the social bookmarking service DELICIOUS, Kipp and Campbell found that 16% of the tags are ‘time and task related’ [146]. Examples include the aforementioned tags ‘toread’ and ‘todo’, as well as similar ones, such as ‘tobuy’, ‘readme’, ‘gtd’ (for ‘getting things done’). These tags are characterized by pointing to some task in the future. As another class of tags that is frequently used, they identified *affective tags* “dwelling on a user’s emotional response” [146]. Examples of affective tags include ‘interesting’, ‘fun’, and ‘cool’. Related to this are tags used to rate resources via stars (‘***) or numbers (‘3/5’).

In a related analysis of tagging data from DELICIOUS, Golder and Huberman found seven main categories for tags [102]. Further categories are proposed by Sen et al. [220], Xu et al. [255], and Smith [226, p. 67], among others. Bischoff et al. [50] combined these categorization approaches into one classification and demonstrated its validity for different tagging systems. In particular, they used the categorization of Golder and Huberman and extended it by the classes ‘time’ and ‘location’. The resulting classification is given in Table 2.1, along with examples gathered from tagging systems in three different domains, namely music (LAST.FM [11]), photos (FLICKR), and bookmarking (DELICIOUS).

The category ‘topic’ contains tags that describe the contents of resources. Tags such as ‘bird’, ‘flower’, or ‘girl’ from the FLICKR tag cloud are examples for this category. Tags in the ‘time’ and ‘location’ categories add contextual information to the resource. For instance, the tags ‘winter’ or ‘july’ in the FLICKR tag cloud may provide information on *when* a photo was taken, while the tags ‘sanfrancisco’ or ‘italy’ describe *where* it was taken. Tags can also specify the ‘type’ of a resource, such as the file format (e.g., ‘jpeg’ in photo sharing) or media type (e.g., ‘blog’ in social bookmarking). This category can additionally include camera settings and photographic styles in photo sharing, such as tags like ‘portrait’ or ‘macro’ in the FLICKR tag

Category	Last.fm	Flickr	Delicious
Topic	love, revolution	people, flowers	webdesign, linux
Time	80s	2005, july	daily, current
Location	england, african	toronto, kingscross	slovakia, newcastle
Type	pop, acoustic	portrait, 50mm	movies, mp3, blogs
Author, owner	the beatles, wax trax	wright	wired, alanmoore
Opinions, qualities	great lyrics	scary, bright	annoying, funny
Usage context	workout, study	vacation, birthday, science	work, travelling
Self reference	albums i own, seen live	me, 100views	frommyrssfeeds

Table 2.1 — Common tag categories with tag examples from the domains of music (LAST.FM), photos (FLICKR), and bookmarking (DELICIOUS) [50].

cloud. The ‘author’ category contains tags that tell who created the resource (e.g., the name of the author of a blog article), while the ‘opinion’ category comprises the aforementioned affective tags and ratings (‘fun’, ‘cool’, ‘***’). Tags in the category ‘usage context’ describe the context or task associated with a resource, whereas ‘self referring’ tags are personal tags that are mainly of interest to the users themselves.

Bischoff et al. [50] emphasize that the categories are not equally useful for different tagging systems but that the distribution of tags among the categories depends on the resources that are being tagged. For instance, while more than 50% of the tags in FLICKR and DELICIOUS are content descriptions, most tags in LAST.FM describe the type of music [50].

Specific Types of Tags

It is often hard to decide which category a tag belongs to. A challenge is not only tag ambiguity, as introduced in Section 2.2.2, but also the often unclear function of a tag. Take, for instance, a photo annotated with the tag ‘alice’. Does it mean that the photo shows Alice or that it was taken by Alice? It may also mean that the creator of the tag thinks the photo

would be of interest to Alice? To overcome this problem, *triple tags* have been proposed as a way to add facets to tags.

Triple Tags Triple tags (also known as *machine* or *system tags*) are tags with a special syntax consisting of a namespace, a predicate, and a value [71, 9, 226, p.111]. The namespace defines the facet the tag belongs to (e.g., ‘system’), while the predicate is the name of a property for that namespace (e.g., ‘filetype’). Namespace and predicate are separated by a colon, predicates and values by an equals sign (e.g., ‘system:filetype=raw’). Triple tags allow for structured tagging and can help to reduce ambiguity. They can also refer to well-defined vocabularies like the *Dublin Core Metadata Element Set (DC)* [54]. For instance, ‘dc:subject=alice’ would state that the photo *shows* Alice, while ‘dc:creator=alice’ would mean that it was *taken by* Alice. However, triple tags do not resolve tag ambiguity in all cases. For instance, the tag ‘dc:subject=apple’ only tells that a resource is about ‘apple’ but it does not clarify the meaning of the word ‘apple’ itself. While such a complete disambiguation would only be possible with semantic tags (see Definition 10), the idea of triple tags is rather to express the category or function of tags and make it more machine processable.

Namespace collisions can be prevented by referencing URIs with the help of other triple tags (e.g., `xmlns:dc=http://purl.org/dc/elements/1.1/`). While this idea is adopted from XML namespaces [58], the triple structure is similar to *microformats* [13, 141] and the *Resource Description Framework (RDF)* [147]. The namespaces and predicates of triple tags are often not formally specified. They can in principle be freely chosen by the user. However, as for special tags like ‘toread’ or ‘me’, conventions have emerged over time. Some triple tags can even be automatically processed by tagging systems, especially in *geotagging*, which is a popular application area of triple tags.

Geotags *Geotags* are a special type of triple tags that can be used to annotate resources with geographical information. By convention, the geographic location of a resource is given by its latitude and longitude (in decimal degrees, as specified by the *World Geodetic System*) with the namespace ‘geo’ and the properties ‘lat’ and ‘lon’. For instance, geotags added to a photo showing the central square ‘Puerta del Sol’ in Madrid could look like this: `geo:lat=40.416618, geo:lon=-3.703748`. This pair of tags may be accompanied by a tag like ‘geotagged’ (depending on the convention) telling the tagging system that geotags are used. The tagging system might then

handle these tags differently, for instance, by processing the coordinates and presenting the associated resources on a geographic map.

There are several extensions to the basic geotagging convention that allow, for instance, to additionally encode the altitude, viewing direction, or place names. Geotags are usually not entered as decimal degrees by the users but are either added by selecting a location from a map or automatically attached to a digital resource (e.g., using data from the *Exchangeable Image File Format (EXIF)* in case of photo sharing). Since more and more devices (e.g., cameras, smartphones, etc.) are equipped with GPS units and network-based positioning systems nowadays, geographic information is increasingly available for tagging systems.

Automatic tags The method of automatically assigning tags to resources is known as *automatic tagging* [244, 41, 89]. Strictly speaking, it is not a form of tagging, as there is no user involved. However, as automatic tagging is an important concept in the domain of tagging, it must be taken into account in a comprehensive conceptualization.

Definition 14. An *automatic tag* is a tag that is automatically assigned to a resource (e.g., by the tagging system).

Definition 15. *Automatic tagging* is tagging with automatic tags.

2.4 Basic Models

A comprehensive conceptualization must not only consider the resources, tags, and users, but also the relationships between these elements created through tagging. In the following, we will examine basic models and representations that describe these relationships. They form the basis for the conceptualization of tagging and the visualization of folksonomies that are addressed later in this thesis.

To start with, we can define three finite and disjoint sets $R = \{r_1, r_2, \dots, r_k\}$, $T = \{t_1, t_2, \dots, t_l\}$, and $U = \{u_1, u_2, \dots, u_m\}$ that represent the resources, tags, and users (or more precisely user accounts).⁷ They are interconnected by taggings, i.e., a set of annotations $A = \{a_1, a_2, \dots, a_n\}$ that follow the

⁷ Though we stressed the importance of distinguishing between *users* and *user accounts* in Section 2.3.2, we will use these terms largely interchangeable in the following to simplify matters.

fundamental tagging principles. These principles must not be violated if common processing of the resulting folksonomy should be guaranteed (e.g., to generate tag clouds or to allow for *pivot browsing* [190]). Therefore, we defined them as axioms:

Axiom 1. Each tagging links exactly one resource with one user account and one or more tags.

Axiom 2. Each resource can be tagged at most once by each user account.

Axiom 3. Each tag can be assigned at most once to each resource by each user account.

Axiom 4. A tag consists of exactly one text label – otherwise it is not a tag.

These axioms are true for any folksonomy, i.e., if one of these principles is violated, it is not a valid folksonomy.

2.4.1 Three-Layer Model of Tagging

Tagging is often illustrated by means of a three-layer model similar to the one shown in Figure 2.3. It arranges the three element sets in layers that are connected by lines indicating the taggings. Similar illustrations can be found in several works on tagging and folksonomies [237, 182, 111, 128, 74, 229, p. 41].

The three-layer model illustrates the many links that emerge in tagging. Apart from the explicit links created through tagging, several implicit links result. These implicit links connect elements within the layers and are indicated by dashed lines in the model. For instance, two users are implicitly connected if they tag the same resource (as users u_2 and u_3 in Figure 2.3) or if they use the same tag (as users u_3 and u_4). Likewise, implicit links on the resource layer emerge if two resources are annotated with the same tag (r_1 and r_2) or if they are tagged by the same user. On the tag layer, implicit links result if two tags are used for the same resource (t_2 and t_3) or by the same user (t_3 and t_4).

The link weights between elements in a layer increase with the number of times the elements are used together. For instance, if the same two tags (t_4 and t_5) are associated with several resources (r_4 and r_5), the implicit link between these tags gets stronger. Similarly, links between two resources get

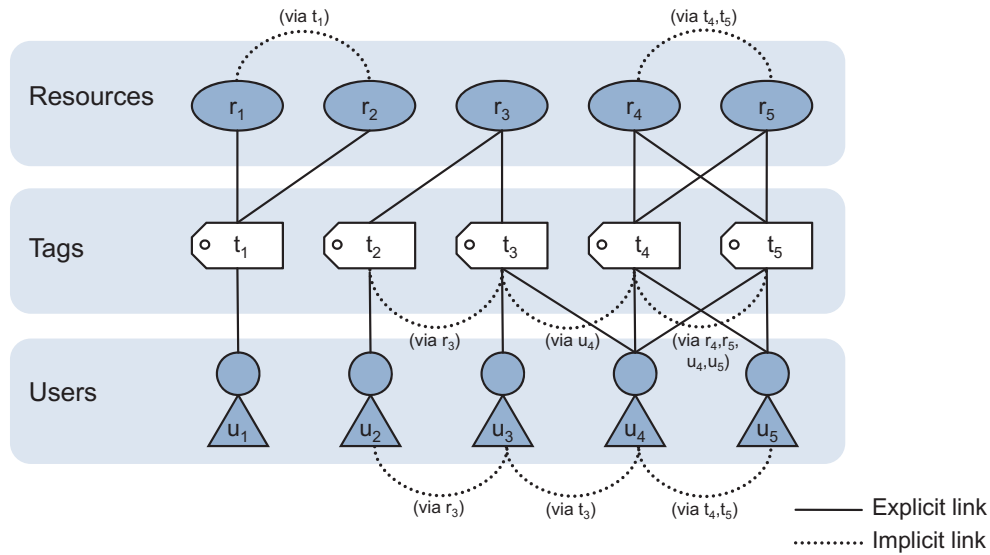


Figure 2.3 — Three-layer model of tagging, illustrating the links that emerge between and within the layers.

stronger if they are annotated with the same tag multiple times, or between two users if they repeatedly use the same tags.

The implicit links can, of course, also connect more than two elements on a layer. For instance, since users u_2 , u_3 , and u_4 annotate all the same resource r_3 in the example of Figure 2.3, they are implicitly connected. However, as users u_3 and u_4 use even the same tag (t_3) for indexing, the link between these two users is stronger than their links to user u_2 .

These implicit, weighted links open up interesting opportunities for social recommendation and navigation, such as *collaborative filtering* [235] or *peer finding* [91]. They are also useful in the visualization of folksonomies, as we will detail in Section 6.

Note that the model assumes that all tags with identical spelling are represented by the same element t_x , independently of their meaning. Such an aggregation is common to folksonomies and considered one of the four criteria that make up a folksonomy according to Smith [226, p. 84]. With tag clouds, we already introduced a popular visualization for aggregated tag vocabularies and discussed its limitations in Section 2.2.2. While tag aggregations are not considered by the basic models presented here, it is important to distinguish between tags and aggregated tags in a comprehensive conceptualization, as we will explain later (see Section 3.1).

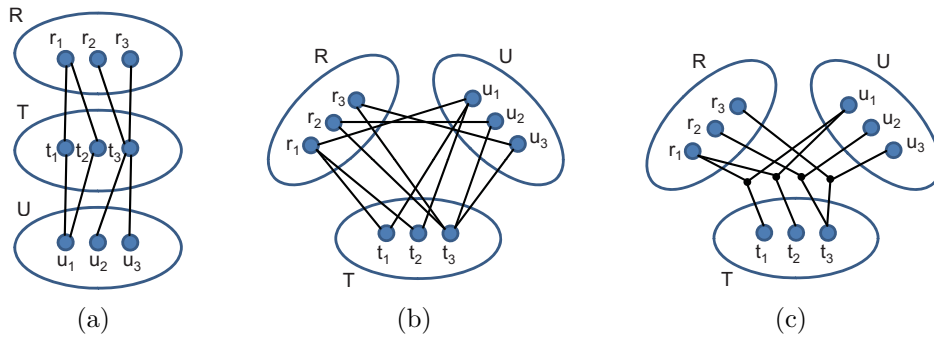


Figure 2.4 — Graph representations of tagging and folksonomies, (a) according to the three-layer model, (b) with relations between users and resources, and (c) with ternary relations connecting all three element sets.

2.4.2 Formal Folksonomy Representation

A simplified version of the three-layer model is given in Figure 2.4a. Compared to the illustration in Figure 2.3, the example has fewer elements and a different link structure for demonstration purposes. It is basically an undirected tripartite graph where each partition represents one model layer (i.e., one set of elements) and the layers are connected by binary relations. Each annotation consists of at least two binary relations that connect one resource with one user account via one or more tags, as given by Axiom 1. The set of annotations can thus be formally defined as $A_{2 \times 2} \subseteq (R \times T) \cup (T \times U)$.

Although the three-layer model provides an intuitive illustration of tagging, it is not suitable to represent folksonomies, as it misses information on which user annotated which resources. It is, for example, impossible to say from Figure 2.4a if resource r_3 has been annotated by user u_2 or user u_3 (or even by both). This is vital information that needs to be included in representations of folksonomies [240].

Three-Uniform Folksonomy Hypergraph

To include that information, we need to add connections between the resource and user layers, as illustrated in Figure 2.4b. The formal definition of the set of annotations is thereby extended to $A_{3 \times 2} \subseteq (R \times T) \cup (T \times U) \cup (U \times R)$. However, with the extended representation, it is still not possible to unambiguously describe which tags have been added by which user. To overcome this problem, we can replace each set of three binary relations by one ternary relation, resulting in a three-uniform hypergraph as

illustrated in Figure 2.4c. The set of all annotations can now be defined as $A_{1x3} \subseteq R \times T \times U$.

This hypergraph representation of folksonomies was first introduced by Mika [188]. Strictly speaking, it consists not only of the set of annotations A_{1x3} , i.e., the ternary relations between the users, tags, and resources, but also includes the three element sets themselves. The folksonomy hypergraph can thus be defined as $G(F) = (V, E)$ with $V = R \cup T \cup U$ and $E = \{\{r, t, u\} \mid (r, t, u) \in A_{1x3}\}$. Alternatively, it can be represented in tuple form, as proposed by Hotho et al. [129].

Definition 16. A *folksonomy* can be formally described as a quadruple $F = (R, T, U, A)$ where R , T , and U are the finite and disjoint sets of resources, tags, and users, and $A \subseteq R \times T \times U$ is the set of ternary relations resulting from tagging (cp. Definitions 4 and 5).

The folksonomy quadruple for the example in Figure 2.4c would then look as follows:

$$F_{ex} = (\{r_1, r_2, r_3\}, \{t_1, t_2, t_3\}, \{u_1, u_2, u_3\}, \{(r_1, t_1, u_1), (r_1, t_2, u_1), (r_2, t_3, u_2), (r_3, t_3, u_3)\}) \quad (2.1)$$

Given the triadic character of this representation, parts of the folksonomy can be derived by restricting the hypergraph or quadruple to a specific element or subset of elements. For instance, the personomy P_u of a user $u \in U$ is the restriction of F to u . In tuple form, it can be defined as $P_u = (R_u, T_u, A_u)$ where $A_u = \{(r, t) \mid (r, t, u) \in A_{1x3}\}$ is the user's set of annotations, $T_u = \{(t) \mid (r, t) \in A_u\}$ is the user's set of tags, and $R_u = \{(r) \mid (r, t) \in A_u\}$ is the set of annotated resources [178].

Non-Uniform Folksonomy Hypergraph

Although the three-uniform hypergraph and related quadruple representation are straightforward ways to describe folksonomies, they are conceptually imprecise, as all taggings are represented by ternary relations. Taggings that consist of more than one tag (which is the usual case [111]) must be split into several relations in these models. A conceptually more accurate representation would be a non-uniform hypergraph that allows for n -ary relations of variable n , as illustrated in Figure 2.5a. The folksonomy quadruple F_{ex} would then look as follows:

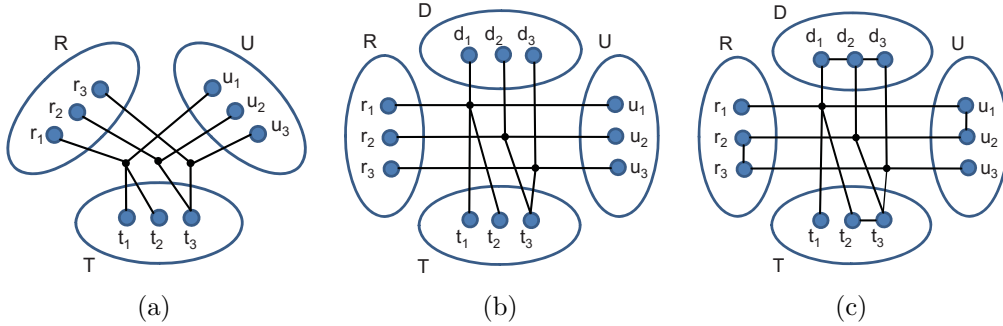


Figure 2.5 — Further graph representations of folksonomies, with (a) n -ary relations connecting more than one tag, (b) additional time information, and (c) links within the element sets.

$$F_{ex} = (\{r_1, r_2, r_3\}, \{t_1, t_2, t_3\}, \{u_1, u_2, u_3\}, \\ \{(r_1, t_1, t_2, u_1), (r_2, t_3, u_2), (r_3, t_3, u_3)\}) \quad (2.2)$$

A non-uniform hypergraph is not only conceptually more accurate than a three-uniform hypergraph, but it also allows for a more compact representation. On the other hand, it does not consist of triples but of n -ary relations, which may be seen as technical drawback (e.g., in RDF contexts, see Section 4). Ultimately, both hypergraph representations contain the same information and can be transformed into one another. This is possible due to Axiom 2, stating that each resource can be tagged at most once by each user account. So if we know the user and resource, we can always get the associated tags, also in the three-uniform hypergraph representation defined above.

2.4.3 Beyond Basic Representations

There is more than the linked resources, tags, and users that must be considered in a comprehensive conceptualization of tagging and folksonomies. Another important piece of information is the date and time of tagging. Many systems use this information to display taggings in reverse chronological order, while others enable users to specify time intervals when browsing folksonomies [160]. Time information can also be used for trend detection in folksonomies [130] or timeline visualizations of tag sets [26, 83, 186]. This is why some consider time as another core element of folksonomies [253, 226,

p. 101]. We illustrate that in the extended hypergraph representation of Figure 2.5b by adding set D containing date and time information.

Others emphasize the source of tagging as an important piece of information. It becomes vital when folksonomies leave the borders of one tagging system, for example, to be exchanged with other systems or stored in a central repository. This is addressed by Gruber who proposes an alternative tuple representation for folksonomies that includes “some notion of source” [108].

Besides time and source, there are further elements that must be considered in a comprehensive conceptualization of tagging. One example are comments entered along with the tags by the users (cp. the text box for comments in the DELICIOUS user interface of Figure 2.1). Another example are different kinds of tag relations, such as the hierarchical relations that can be defined in some tagging systems (e.g., BIBSONOMY [4]). Hotho et al. [129] extend the folksonomy hypergraph to include hierarchical tag relations. They define the set $\prec \subseteq U \times T \times T$ for this purpose and add it as fifth element to the folksonomy quadruple introduced in Section 2.4.2.

Note that \prec contains the set of the users U , i.e., each tag relation is linked to the account of the user who created it. This is important, as it cannot be assumed that the user-defined tag relations are valid for the whole folksonomy. Such a generalization would quickly result in conflicts and disagreements between users having different understandings of how tags are related. Instead, hierarchical tag relations are usually only valid for the personomy of the user who defined them.

There are other cases where it makes sense to represent relations between tags on the folksonomy level (e.g., in tag clustering [46] or for the generation of *folksonomy graphs*, as they will be introduced in Chapter 6). Figure 2.5c sketches such a global tag relation by linking tags t_2 and t_3 . Likewise, there can also be direct links between resources (e.g., hyperlinks) or between users (e.g., group links), as illustrated in Figure 2.5c for r_2 and r_3 as well as u_1 and u_2 . Although these links are not part of the folksonomy itself, they need to be taken into account in a comprehensive conceptualization of the domain of tagging.

Yet other relations are only valid for single tags (e.g., links to well-defined entities as they may be used with semantic tags, see Definition 10). This is also true for other tag properties, such as the order in which tags are entered by the users. Regardless of whether this order is of importance to the users or not, they would expect it to remain the same whenever they access their

taggings. In general, a conceptualization must carefully distinguish between individual tags, as entered by the users, and aggregated tags, as used for the generation of tag clouds and other representations.

A powerful way to formally describe all this information is an *ontology*. In computer science, an ontology is briefly defined as “an explicit specification of a conceptualization” [107]. It describes the concepts and relationships in an area of knowledge in a logic-based language that is machine-processable and provides means for automated reasoning. In the following, we will detail the possibilities of conceptualizing tagging and folksonomies with ontologies.

Ontologies for Tagging and Folksonomies

Ontologies gained much popularity with the rise of the Semantic Web as a way to give information well-defined meaning [49, 120]. They are used by people and software systems that need to share information about a domain. Numerous ontologies for various domains have been developed in the last years. Most of the modern ones are implemented in the *OWL Web Ontology Language* [185], which is based on the *Resource Description Framework (RDF)* [147] and closely related to *RDF Schema (RDFS)* [59]. All three languages are recommendations of the *World Wide Web Consortium (W3C)*. The latest version of OWL is OWL 2, which became a W3C recommendation in late 2009 and is available in its second edition since late 2012 [106].

There is also quite a number of ontologies for the domain of tagging and folksonomies. In this chapter, we will review the available ontologies and discuss their strengths and weaknesses. Older reviews of a part of these ontologies are provided by Kim et al. [145, 144]. Although these reviews provide a good starting point, they are not detailed enough for our purpose. Furthermore, they are mainly focused on own developments (i.e., the later discussed SCOT and MOAT ontologies), neglecting some relevant conceptualizations that have been proposed elsewhere. We reviewed and compared

Parts of this chapter have already been published in [166] and [169].

the available tagging ontologies in detail with the goal to identify the most adequate ways of conceptualizing the domain of tagging.

3.1 Review of Tagging Ontologies

In a survey of the literature and web, we found nine ontologies for the domain of tagging and folksonomies. Table 3.2 lists these ontologies in chronological order by their release dates. Our review is based on the most recent RDF/XML serializations of the ontologies, as they were available at time of writing. The dates of the latest updates are also given in Table 3.2.¹

Popular ontologies are typically abbreviated by their common namespace prefixes. Since there are yet no established prefixes for many of the reviewed ontologies, we partly used own prefixes as given in Table 3.2. The namespace URIs and prefixes of all vocabularies referenced in this thesis are listed in Table A.1 in Appendix A. Abbreviations in uppercase refer to the ontologies as a whole, while lowercase abbreviations denote certain classes, properties, or individuals of the ontologies. Following conventions, classes and individuals start with a capital letter after the namespace prefix and colon, while properties are in small letters.

Relevant work on the ontology-based description of tagging and folksonomies started in 2005 with the creation of a first tagging ontology by Newman et al. [197]. In the same year, Gruber gave two talks in which he outlined his ideas for an “ontology of folksonomy” [108, 109], followed by the formation of the *TagCommons* initiative [25] to further develop these ideas. These early conceptualizations model the key structure and elements of tagging and folksonomies. They also consider some further concepts of tagging discussed in Section 2.4.3, such as date and time information [197], relations between tags [197], or the source of tagging (i.e., the used tagging system) [108]. While Gruber’s conceptualization is a rather informal description of ideas, the conceptualization of Newman et al. is a well-defined ontology implemented in OWL.

3.1.1 Newman’s Tag Ontology

Newman’s ontology entitled *Tag Ontology (TAGS)* defines the fundamental conceptual structure of tagging. A basic decision in the design of TAGS was to

¹In order to ensure that the reviewed versions of the ontologies continue to be available in the near future, we provide copies of them on the web at <http://purl.org/muto>.

Name	Abbr. ¹	Author(s)	Release date (latest update)	Main purpose	Newly introduced concepts	Reused vocabularies ²
Tag Ontology	TAGS	Newman et al.	2005-03-23 (2005-12-21)	First tagging ontology	Fundamental concepts, restricted tagging	DC, FOAF, SKOS
Tagging Ontology	TO	Knerr	2006 (2007-01-15)	Domain description	Tagging source and comment, private and group tagging	DC, DCTERMS, DCTYPE, FOAF, SKOS
Ontology of Folksonomy	OF	Echarte et al.	2007 (-)	Domain description	Aggregated tag, order of tags, tag type	-
Social Semantic Cloud of Tags	SCOT	Kim et al.	2007-03-23 (2008-06-13)	Tag clouds	Tag cloud, frequencies, linguistic variations	FOAF, SIOC, TAGS
Meaning of a Tag	MOAT	Passant & Lautblet	2008-01-15 (-)	Semantic tagging	Semantic and automatic tagging	FOAF, SIOC, TAGS
Upper Tag Ontology	UTO	Ding et al.	2008 (-)	Upper ontology	Voting with tags	(DCTERMS, FOAF, SIOC, SKOS)
Common Tag	CTAG	Tori et al.	2009-06-08 (-)	Minimal ontology	Author vs. reader tags	DCTERMS, (MOAT, SIOC, SIOCT, SKOS, TAGS)
TAGora Tagging Ontology	TT	Szomszor et al.	2009 (2010)	Automatic tag disambiguation	-	-
NiceTag Ontology	NT	Limpens et al.	2009-01-09 (2010-09-09)	Tagging as speech act	Typed tag relations	FOAF, IRW, SIOC, RDFG

¹Abbreviations as used in this work.²The general-purpose vocabularies RDF, RDFS, OWL, and XSD are not listed, as they are used by all ontologies (except CTAG which does not use XSD). Vocabularies in brackets are not linked in the ontology itself but in a separate schema or the aligned version of the ontology.**Table 3.2** — Reviewed tagging ontologies

assign all key elements, including tags, a URI for unique identification on the web [197]. Accordingly, TAGS defines a class `tags:Tag` for the representation of tags instead of using purely literals. Likewise, a class `tags:Tagging` is defined that describes the “n-ary relationship between a tagger, a tag, a resource, and a date” [197]. Such a class is key to any conceptualization of tagging, as discussed in Section 2.4. To emphasize its central role, we prefer the term *tagging ontology* over *tag ontology* for ontologies that describe the domain of tagging.²

Though Newman precisely defines the concept of *tagging*, he does not clearly state what is considered a *tag*. He mentions that representing tags as instances with URIs gives them unique identity on the web and thus allows for “repeated references to the same tag” [197]. He also provides examples for URIs of tags (e.g., `http://www.holygoat.co.uk/owl/redwood/tag/great`) that follow common patterns of tagging systems (e.g., `http://www.delicious.com/tag/great`). However, since these URIs are generated from the tag labels, it remains unclear how tags with same labels but different meanings are represented. Though Newman distinguishes between “the same tag” and “different tags with the same label” [197], he does not elaborate on how these different cases are treated by the TAGS ontology. This can result in confusion and inconsistent use of the ontology in software systems and other ontologies. Most of the systems that use TAGS seem to aggregate tags with same labels. An example is the reviewing and rating website REVYU [119] that uses the aforementioned pattern and assigns the same URI to all tags with identical spelling (e.g., `http://revyu.com/tags/party`). This is also how the later discussed SCOT and MOAT ontologies seem to reuse TAGS (see Section 3.1.4 and Section 3.1.5).

Another fundamental design decision in the development of TAGS was to reuse existing ontologies for concepts that are not tagging-specific. For instance, `foaf:Agent` from the *Friend of a Friend* (FOAF) vocabulary is used to represent users, the tagging date (`tags:taggedOn`) is a subproperty of `dc:date` from the *Dublin Core Metadata Element Set* (DC) [54] and `tags:Tag` is a subclass of `skos:Concept` from the *Simple Knowledge Organization System* (SKOS) [21]. The label of a tag (`tags:name`) is defined as a subproperty of both `dc:title` and `rdfs:label`. Furthermore, TAGS distinguishes two types of tag relations: `tags:equivalentTag` (defined as subproperty of `owl:sameAs`) to relate synonym tags, and `tags:related` (defined as subproperty of `skos:`

² A possible alternative would be the term *ontology of folksonomy*, as used by Gruber [108] and [88] (see Section 3.1.3).

`semanticRelation`) for all other kinds of tag relations (e.g., hierarchical tag relations).

Finally, TAGS introduces the subclass `tags:RestrictedTagging` for taggings that have “precisely one associated resource and one associated tag” [196]. While this is in-line with the three-uniform folksonomy representation presented in Section 2.4.2, it is not the most accurate conceptualization, as we also discussed there.

A problematic part of TAGS is the property `tags:taggedWithTag` that directly links tags and resources. Although this direct link facilitates access to the tag-resource relations from a pragmatic point of view [240], it is imprecise and redundant from a conceptual point of view, as tags and resources are not directly connected but indirectly by tagging instances. Such a direct linking can be problematic for private tagging, as we will discuss in Section 4.4.

3.1.2 Knerr’s Tagging Ontology

Newman’s TAGS ontology was followed by a number of other tagging ontologies in the subsequent years, each focusing on different aspects of tagging. One such ontology, entitled *Tagging Ontology* (TO), was developed by Knerr in 2006 [148]. It takes up the idea of Gruber [108] and references the source of tagging in the ontology, with the class `to:ServiceDomain`. As mentioned in Section 2.4.3, this information can be important if folksonomies are exchanged between systems or merged. The name and the URL of the tagging system can be represented with `to:hasServiceName` and `to:hasServiceHomepage`, which are subproperties of `foaf:name` and `foaf:homepage`.

Apart from that, TO considers some other important concepts we introduced in Chapter 2. Figure 3.1 depicts an illustration of the main concepts and relationships of TO, as provided by Knerr [148]. Group tagging (see Definition 13) is supported by linking not only `foaf:Person` but also `foaf:Group`. Private taggings (see Definition 7) can be expressed with the class `to:VisibilityEnum` defining the visibility of taggings with the individuals `to:Private`, `to:Public`, and `to:Protected`. According to Knerr, public taggings are visible to everyone, while private taggings are only visible to their creators and protected taggings to their creators and a “selected group of people (e.g., friends)” [148]. However, Knerr does not further detail how this group of people is referenced in TO.

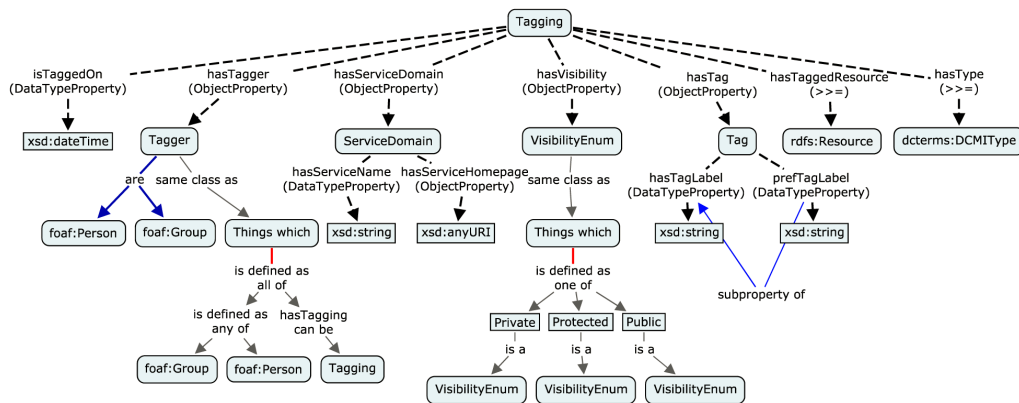


Figure 3.1 — Tagging Ontology by Knerr [148].

TO is also capable to represent information on the type of resources that are tagged. For this purpose, it links to the *DCMI Type Vocabulary* (DCTYPE). DCTYPE is a specification of the *Dublin Core Metadata Initiative* (DCMI) and related to the DC vocabulary that has already been used by TAGS. It defines twelve terms that roughly describe the “genre” of a resource (e.g., `dctype:Image` or `dctype:Text`) [53]. Finally, TO considers comments added to taggings (see Section 2.4.3), which are represented by the property `to:hasNote` (not included in Figure 3.1).

Like TAGS, TO aligns to concepts from FOAF, SKOS, and *Dublin Core* (DC, DCTERMS, DCTYPE), though not in all cases (e.g., `to:prefTagLabel` has no link to `skos:prefLabel`). In contrast to TAGS, TO does not purely use links (i.e., `rdfs:range` and `rdfs:domain`) and subsumptions (i.e., `rdfs:subClassOf`, `rdfs:subPropertyOf`) for these alignments, but also an equivalence relation (`owl:equivalentProperty` between `to:hasType` and `dc:type`).³

Some modeling in TO is unnecessarily complicated (e.g., the introduction of the `to:VisibilityEnum` class) or even semantically incorrect. For instance, making `to:hasTagging` a functional property allows for only one tagging per user, and defining it with two domain axioms (`foaf:Person` and `foaf:Group`) results in a conjunction of the domains, and not in a disjunction as it seems to be intended by Knerr. Similar to TAGS, it remains unclear how tags

³ Note that property equivalence is different from property equality. While equivalent properties “have the same ‘values’ (i.e., the same property extension)”, they “may have different intensional meaning (i.e., denote different concepts)” according to [76].

are represented in **TO**. On the one hand, the ontology states that **to:Tag** represents “a single tag” and that “each tag has a unique URI” [149]. On the other hand, Knerr writes that instances of **to:Tag** “can have multiple labels [...] and a default label” to “allow for variations in spelling or language” [148]. The latter contradicts the idea of tagging where a tag always consists of exactly one text label (see Axiom 4).

The lack of clarity in the definition of what a tag is and how it is represented in the ontology can be regarded as one of the main drawbacks of both **TO** and **TAGS**. In particular, it is unclear how tags are aggregated in these ontologies. While **TAGS** seems to intend an aggregation of identically spelled tags (i.e., all tags with same spelling have the same URI), **TO** rather seems to aim for a semantic aggregation that merges tags which denote the same thing, even if they are differently spelled. This would require some kind of ‘semantification’ of tags, as outlined in Section 2.2.3, which is, however, not further discussed by Knerr. Overall, both **TO** and **TAGS** define classes for aggregated tags, but they do not consider the individual tags as entered by the users. Hence, they can also not represent related properties, such as the order in which tags are entered or other information of the individual tags. However, this kind of information can be important and has to be included in a comprehensive tagging ontology, as we will detail in Section 4.2.6.

3.1.3 Ontology of Folksonomy

In contrast to **TO** and **TAGS**, the *Ontology of Folksonomy* (**OF**) published by Echarte et al. in 2007 [88] makes a clear distinction between individual and aggregated tags. **OF** defines two classes for this purpose: While **of:Tag** is semantically nearly identical to **to:Tag** (except from that it is not a subclass of **skos:Concept**), **of:AnnotationTag** represents the actual tags as entered by the users, i.e., without merging tags that have the same spelling (like in **TAGS**) or meaning (like in **TO**). It means that a new instance of **of:AnnotationTag** is created for each entered tag, without any aggregation. Instead, the instances of **of:AnnotationTag** are linked to instances of **of:Tag**, which are called “tag concepts” by Echarte et al. [88] and contain different variations of a tag that denote the same thing (like for **to:Tag**).

This modeling of taggings (**of:Annotation**) that are first linked to tags (**of:AnnotationTag**) and from there to well-defined entities (**of:Tag**) is actually a conceptualization of semantic tagging (see Definition 11). Every “tag concept” must have exactly one preferred label (**of:hasPrefLabel**)

according to **OF**, and can have additional hidden labels (`of:hasHiddenLabel`, e.g., for frequent misspellings) or alternative labels (`of:hasAltLabel`, e.g., for synonym tags). This is very similar to the modeling of `to:Tag` and the general idea of `skos:Concept` with its distinction between *preferred*, *alternative*, and *hidden* labels.

OF does not define any links to other vocabularies, neither to well-known vocabularies like **SKOS**, **FOAF**, or **DCTERMS** nor to the previously published tagging ontologies **TO** and **TAGS**. Like **TAGS**, it defines a direct relation between tags and resources (`of:hasRelatedResource`) for “efficiency” reasons [88]. As we already pointed out in Section 3.1.1, this is conceptually imprecise and can be problematic, especially for the representation of private taggings (see Section 4.1.4). Like **TO**, **OF** incorporates source information, as introduced by Gruber [108], with the class `of:Source`. It also adopts Gruber’s idea of “negative tagging”, with the class `of:Polarity` [108]. However, as this idea never found its way into tagging systems, it can be neglected in a conceptualization of tagging.

The clear distinction between individual and aggregated tags enables **OF** to define individual tag properties. For instance, the aforementioned tag order is represented by adding an integer value to each instance of `of:AnnotationTag` (with `of:hasPosition`) that represents the position in the user-entered list of tags. **OF** is the only tagging ontology that is capable to represent this information.

Finally, **OF** makes a first attempt to distinguish between different types of tags by introducing two basic tag categories (see Section 2.3.3): `of:TagPersonal` and `of:TagCommon`. While the first captures all tags of “personal type”, i.e., those related to the “planning of personal tasks” or “self-reference tags”, the second is for “the rest of tags” according to Echarte et al. [88]. Unfortunately, these two categories are defined as subclasses of `of:Tag`, not of `of:AnnotationTag`, which is problematic if the same tag has a “personal” function in one context and a “common” function in another.

3.1.4 Social Semantic Cloud of Tags

The *Social Semantic Cloud of Tags* (**SCOT**) [142] has been developed by Kim et al. with a different goal than the previously discussed ontologies, as already indicated by its name. Instead of describing the domain of tagging, it focuses on the sharing of numeric data and selected sets of tags, i.e., information that is often visually presented in the form of *tag clouds* (see Definition 9).

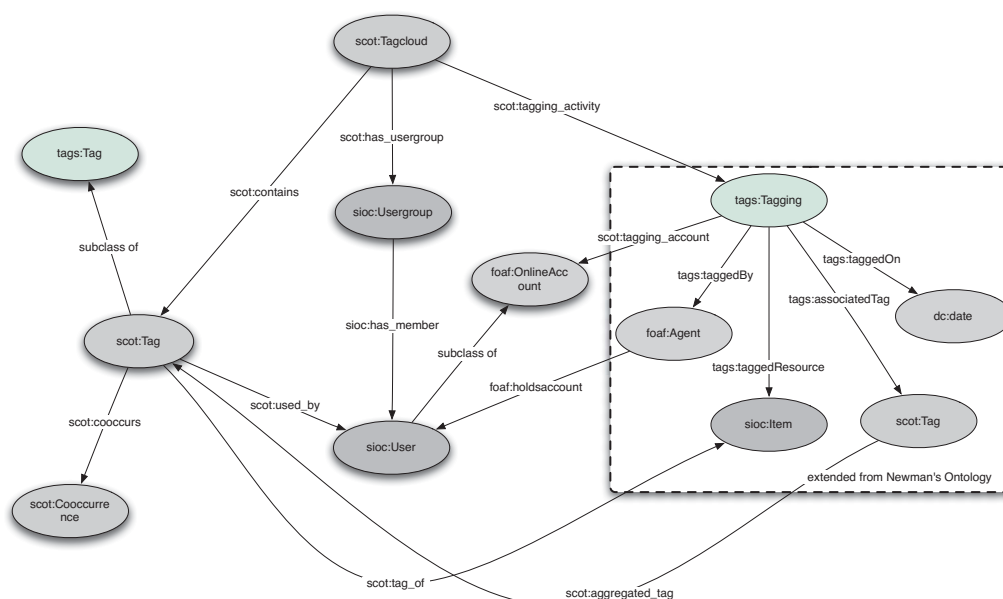


Figure 3.2 — Social Semantic Cloud of Tags by Kim et al. [144].

For this purpose, SCOT reuses Newman’s TAGS ontology and extends it by the class `scot:Tagcloud` (see Figure 3.2). According to Kim et al., this class can be regarded as a “specific type of container for grouping metadata relevant to tagging” [143]. For instance, it defines datatype properties for the total number of tags contained in the tag cloud (`scot:total_tags`) or for the number of associated resources (`scot:total_items`). In addition, it contains information about the creator of the tag cloud (`scot:created_by`), associated user groups (`scot:has_usergroup`), and the source of tagging (`scot:tagospace`). Taggings and tags are linked to `scot:Tagcloud` by the properties `scot:contains` and `scot:tagging_activity` (see Figure 3.2).

Apart from `scot:Tagcloud`, SCOT defines the classes `scot:Tag` and `scot:Cooccurrence`. The first is a subclass of `tags:Tag` and “describes a tag that is aggregated from individual tagging activities” [144]. Unfortunately, this statement is very vague, leaving it unclear what `scot:Tag` actually denotes, as it was already a problem in TO and TAGS. It seems that instances of this class are created by aggregating tags with same labels, i.e., it might be used the same way as `tags:Tag` is commonly used. This interpretation is also indicated by the properties of `scot:Tag` that describe the absolute and relative usage frequencies of a tag, among others. The class `scot:`

Cooccurrence, on the other hand, describes which tags are used together in taggings and how frequently they are used together.

However, all this information can be inferred from other tagging concepts and does not need to be described redundantly. A separate description might be beneficial for technical reasons – e.g., to facilitate querying or to shorten query response times when the ontology is used for data storage – but it is better avoided in a general conceptualization of tagging we aim for in this thesis, since it makes the ontology unnecessary complex, partly redundant and prone to inconsistencies. In our view, a tag cloud is rather one type of visualization for folksonomies than a part of the conceptualization. Although the visualization concept of tag clouds can be formally described as an ontology, we would not integrate it into the core conceptualization. Tag clouds are rather generated by tagging systems based on the conceptualization, as we will further detail in Chapter 5.

More relevant to this work are the many tag relations defined by **SCOT** which describe some of the linguistic variations discussed in Section 2.2.2. For instance, if a tag is an abbreviation of another tag, this can be indicated by the property `scot:acronym`. There are also property relations for plural forms (`scot:plural`) or synonym tags (`scot:synonym`). Finally, **SCOT** introduces properties that represent the type of delimiter used to separate multiple word tags (cp. Section 2.3.3), such as `scot:underscored` or `scot:spaced`. However, as discussed in Section 2.2.3, it can be difficult to automatically detect tag variations that go beyond simple cases, such as different spellings (e.g., British vs. American English) or tag delimiters (e.g., underscore).

Apart from that, **SCOT** is the first tagging ontology that integrates the *Semantically-Interlinked Online Communities* (**SIOC**) vocabulary [55]. **SIOC** is linked with the purpose “to define the range values of tagging properties more specifically” [144]. For instance, **SIOC** classes are used to represent creators of tags (`sioc:User`), sources of tag clouds (`sioc:Site`), or user groups associated with tag clouds (`sioc:Usergroup`). Furthermore, `sioc:Item` is defined as range for tagged resources (as expressed by `scot:tag_of`). Even though explicit ranges for properties usually increase the interoperability of the conceptualization, they can also make the ontology less flexible with regard to different application contexts. In this case, it means that **SCOT** should only be used with resources that are instances of `sioc:Item`, which limits its applicability. Furthermore, **SCOT** can only be used along with **TAGS**, as it is not a complete tagging ontology but an extension of **TAGS** describing information related to tag clouds.

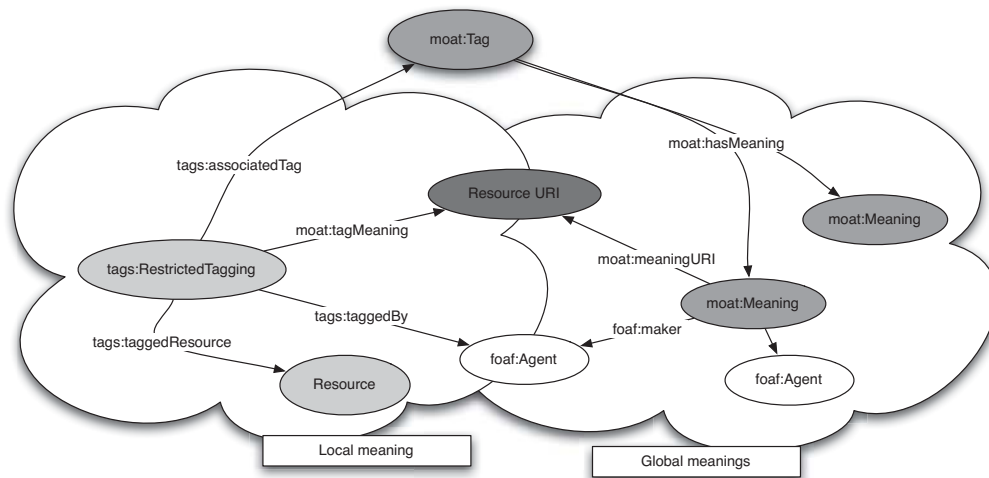


Figure 3.3 — Local and global meanings represented with MOAT [144].

3.1.5 Meaning of a Tag

MOAT [204] is the second tagging ontology that explicitly reuses and extends Newman’s TAGS ontology. Like SCOT, it is thus not a complete tagging ontology but has been developed with the goal to describe the “meanings of tags” [14], i.e., to support *semantic tagging* (see Definition 11). This is realized with concepts that enable a disambiguation of tags by linking them to URIs of well-defined entities from the Semantic Web, such as *DBpedia* resources [51].

MOAT distinguishes between *local* and *global* meanings: While the first denotes a particular meaning of a tag, as assigned by a user in a certain tagging context, the second represents the list of all meanings assigned to a tag by different users in various tagging contexts. The latter is described by the class `moat:Meaning` that contains the URI of the Semantic Web resource (`moat:meaningURI`) and the user who linked that URI (`foaf:maker`). Several `moat:Meaning` instances can be added to the same tag, as illustrated on the right hand side of Figure 3.3.

The information of which users added what meanings is additionally represented by the local meaning concept, as shown on the left hand side of Figure 3.3. Here, the class `tags:RestrictedTagging` from the TAGS ontology is reused and enriched by the property `moat:tagMeaning` that allows to link a tagging to a resource of the Semantic Web. This is conceptually imprecise, as not whole taggings but single tags are to be disambiguated (cp.

Section 2.2.3). However, it works in the case of `tags:restrictedTagging`, because instances of this class consist of only one tag. In addition, the property `moat:localMeaning` can be used to directly link a tag to its ‘local meaning’ (not shown in Figure 3.3). Although this relation is conceptually more accurate, it does not provide information on which users added what meanings to a tag. Similarly, MOAT allows to define a direct link between the resource that is tagged and the resource that gives the meaning for that tag (with the property `moat:taggedWith`). However, this relation also misses user information and does not even contain the label of the tag. It can also be problematic for private tagging (see Section 4.4).

In summary, though the general idea of disambiguating tags is useful, the modeling approach of MOAT is comparatively complicated and redundant. This is mainly due to the limitations of the TAGS ontology that describes only aggregated but not individual tags, as discussed before.

Apart from semantic tagging, MOAT introduces further interesting extensions to the TAGS ontology. First, it specifies that each tag has exactly one label – a key principle of tagging we defined in Axiom 4. This is done by making the class `moat:Tag` a subclass of `tags:Tag` and equipping it with a corresponding cardinality restriction for the property `tags:name`. Another interesting extension is the integration of *automatic tagging* (see Definition 15) in the ontology. Here, MOAT defines the class `moat:TagType` with individuals `moat:AutomatedTagging` and `moat:ValidatedTagging`, indicating whether a tagging has been automatically added (e.g. by the tagging system) or manually by a human being. This concept has been added since “some tools may rely on automatic semantic tagging approaches” and it can be “worthwhile to know if the tag has been made by the user himself or by a dedicated software agent” [203]. While this statement may suggest that automatic tagging is strongly related to semantic tagging, it is important to note that both are rather independent concepts. For instance, imagine a case where tags are automatically extracted from a text-based resource, without any disambiguation.

The class `moat:TagType` is linked with the taggings and not with the tags because of the aforementioned limitation of TAGS that provides only a class for aggregated tags but not for individual ones. Linking the concept for automatic tagging with the representation of individual tags instead would result in a more flexible and accurate conceptualization, as we will detail in Section 4.2.3.

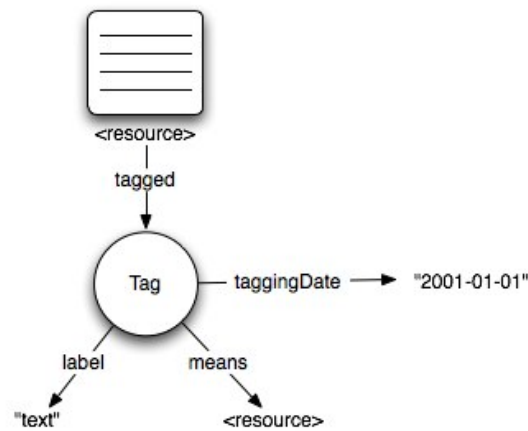


Figure 3.4 — Basic structure of the CTAG ontology [18].

3.1.6 Common Tag

The idea of linking tags with well-defined resources from the Semantic Web is also addressed by the *Common Tag* (CTAG) ontology that was released in 2009 by a consortium of people from research and industry [5]. It has been designed as a “minimal” tagging vocabulary that can be easily embedded in XHTML via RDFa. The basic structure of CTAG is very compact, as illustrated in Figure 3.4: It does not distinguish between taggings and tags but consists of only one class (`ctag:Tag`) that is linked to the tagged resource, a tagging date, a tag label, and the URI of the resource providing the ‘meaning’ for that tag.

In addition, three subclasses are defined for `ctag:Tag`. One of the subclasses (`ctag:AutoTag`) is for automatic tags (see Definition 14). The other two subclasses are for manually assigned tags, distinguishing between 1) tags added by the authors of a resource (`ctag:AuthorTag`), and 2) tags used by people consuming the resource (`ctag:ReaderTag`). Finally, a class `ctag:TaggedContent` is defined providing a “rel-tag-like minimal tagging option” [132] in combination with the property `ctag:isAbout`, i.e., it allows to directly link a resource with URIs that add “meaning” to the resource, similar to the MOAT property `moat:taggedWith` described above.

CTAG defines all concepts in its own namespace, without integrating concepts from other vocabularies. The only exception is the property `ctag:taggingDate` which is defined as a subproperty of `dcterms:created` in the CTAG ontology. However, mappings to related concepts from SIOC, SIOCT,

TAGS, MOAT, and SKOS are given in a separate “mappings” module [133] deployed together with the ontology.

CTAG includes no cardinality constraints, so that any number of properties can be used along with `ctag:Tag`. In the understanding of CTAG, a minimal tagging does not even require a tag label but can purely consist of a link giving meaning to a resource. The opposite case of adding several labels to a tag would, however, not only be against Axiom 4 but would also be problematic against the context of semantic tagging, as it could not be clearly said to which label the linked meaning belongs.

However, missing cardinality constraints are only one drawback of CTAG. Even more problematic is the fact that CTAG does not include a concept for taggings. Thus, it cannot express which tags are part of the same tagging. This information can also not be represented with tag relations, as `ctag:Tag` seems to serve as a class for aggregated tags, like `tags:Tag` and `moat:Tag`. This interpretation is at least suggested by the “mappings” module (see above) that makes `ctag:Tag` a subclass of `tags:Tag` and `moat:Tag`, among others. However, the authors of CTAG seem to interpret `tags:Tag` and `moat:Tag` differently from how it is commonly used by developers. Their interpretation rather follows Newman’s distinction between same tags and tags with same labels but different meanings. Otherwise, it cannot be explained why the property `ctag:means` is linked to `ctag:Tag`.

Another core concept of tagging that is missing in CTAG is the user who added the tags. In one of their examples, the authors use `foaf:maker` to include this information [18], but this is not further specified in the ontology. Instead, the authors state that they expect CTAG to be extended as needed with “additional information from other RDF vocabularies” [18]. Hence, CTAG rather relies on the emergence of conventions than offering a comprehensive specification for the domain of tagging.

For all these reasons, CTAG cannot be regarded as a complete tagging ontology. From the four core concepts of tagging introduced in Section 2.3, only *resources* and *tags* are included, but there are no descriptions for *users* and *taggings*. In contrast to SCOT and MOAT, CTAG does not even reuse these concepts from another tagging ontology, such as TAGS.

3.1.7 Upper Tag Ontology

Similar to CTAG, the *Upper Tag Ontology* (UTO) [78] by Ding et al. was developed with the goal of “making it easy and simple to use” [256]. However,

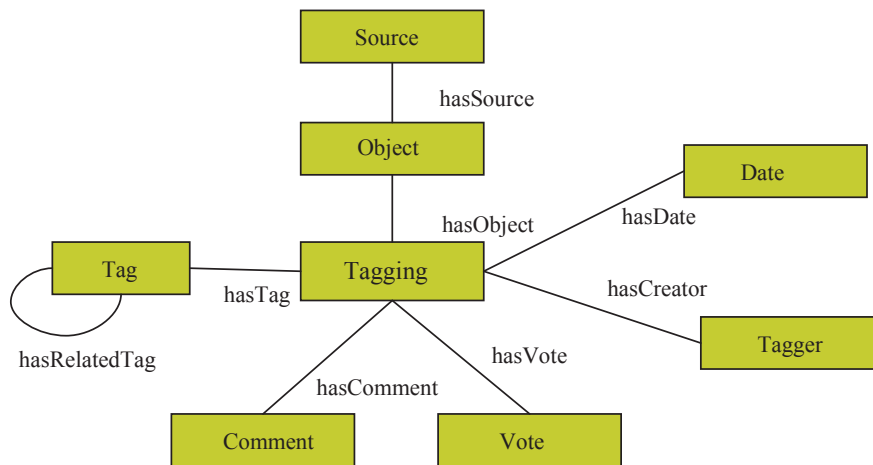


Figure 3.5 — Concepts and relationships of the Upper Tag Ontology [78].

in contrast to **CTAG** it is not intended as a minimal vocabulary but aims to provide a comprehensive yet simple structure for tagging. This structure consists of the central class `uto:tagging` that is linked to a number of other tagging concepts, as illustrated in Figure 3.5.

UTO does not reuse concepts from other ontologies but defines all concepts in its own namespace. The concepts are modeled as a combination of a class and a property that are all labeled according to the same pattern. Besides the fact that this modeling approach leads to more classes than necessary, it can cause irritation when applying **UTO**. For instance, while the date of tagging is a datatype property in all other tagging ontologies (i.e., it links to data values), it is an object property in **UTO**, linking to the class `uto:Date`. Interestingly, the authors of **UTO** use a simple text string instead of an object to represent the tagging date in the example they present [256]. Furthermore, they state in the ontology that `uto:Date` should be a timestamp in the form “MmmYY”. The question remains why the tagging date and similar concepts (e.g., tagging comment, vote) are represented as classes and not as literals in **UTO**. The reason may be that **UTO** is intended to serve as an “upper ontology” for tagging, i.e., an ontology that other tagging ontologies can be aligned to. Unfortunately, this is not further discussed by the authors.

Related to this, **UTO** does not define a separate concept for tag labels. However, as we know from Axiom 4 and the other tagging ontologies, tag labels are an essential element of tagging. Instead, it seems that **UTO** expects this information to be part of the URI of the tag instances (e.g.,

`http://www.delicious.com/tag/great`) – a common practice in tagging systems we already mentioned in the discussion of Newman’s TAGS ontology in Section 3.1.1. In this approach, the character string following the last slash sign in the URI is the tag (i.e., ‘great’ in the above case of DELICIOUS). This means that UTO aggregates tags with same labels, which results in some limitations, as we also already discussed in the context of TAGS (see Section 3.1.1).

While UTO defines all concepts in its own namespace, it aligns these concepts with DCTERMS, FOAF, SIOC, and SKOS.⁴ For instance, the class that represents the tags (`uto:Tag`) is aligned to `skos:Concept`, as in most of the other tagging ontologies. However, instead of using a subclass relation for this alignment, UTO uses an equivalence relation (`owl:equivalentClass`). Other classes and properties are also aligned with equivalence relations (e.g., `uto:Tagger`, `uto:hasCreator`). In some cases, these equivalence relations may lead to semantic inconsistencies, for instance, when `uto:hasRelatedTag` is defined to be equivalent to all three SKOS properties `skos:narrower`, `skos:broader`, and `skos:related`.

Inspired by Gruber’s thoughts on “voting” with tags [108], Ding et al. equipped UTO with the class `uto:Vote`. According to them, instances of `uto:Vote` can be “the number of different Taggers who assign a Tag to a bookmark in Delicious, a photo indicated as a favorite in Flickr, or the number of stars given to a video in YouTube” [78]. This is a very broad understanding of what can be represented with `uto:Vote`. The class seems to denote many different things what limits its semantic interoperability and makes it hardly applicable in practice. From the examples provided by Ding et al. [78, 256], it seems to be mainly used to represent the number of users who tagged a resource. However, this value can easily be inferred by summing up the number of taggings (or the number of users) that exist for that resource. As in the case of `scot:Cooccurrence`, it is unclear why a separate definition of this frequency value is needed in the conceptualization. The main reason seems to be faster access to this value, as indicated by the examples [78, 256].

Furthermore, UTO considers comments added to taggings (`uto:Comment`), like they are mentioned in Section 2.4.3 and are also defined by TO. Finally,

⁴Note that the version of the ontology published under the namespace URI of UTO does not include these alignments. However, they are formally defined in two papers on UTO [78, 256] and in the version of the ontology that is attached to one of these papers [78].

it contains a class called `uto:Source`. However, this class does not describe the source of tagging as proposed by Gruber (see Section 2.4.3) but the “place where the object is hosted” that is being tagged [78]. Although both the source of a tagging and of the tagged resource can be identical (as e.g., in FLICKR or YOUTUBE), they can also be different (as e.g., in DELICIOUS). It seems that UTO does not make this distinction but uses `uto:Source` for both notions of source [78].

3.1.8 Tagora Tagging Ontology

Another tagging ontology has been developed within the larger context of the TAGora project [27], as part of the *TAGora Sense Repository (TSR)* [28]. Similar to MOAT and CTAG, the TSR services aim to support *semantic tagging* (see Definition 11). However, while the aforementioned approaches address mainly the *manual* disambiguation of tags, the TSR services focus on an *automatic* disambiguation [230]. The services are based on three ontologies: The *TAGora Tagging Ontology (TT)*, a second ontology for enriching DBpedia resource descriptions, and a third one that models the tag sense disambiguations. While the latter two ontologies contain concepts needed for the automatic disambiguation, TT is used to represent the taggings of the users.

Figure 3.6 shows an illustration of the concepts and relationships of TT.⁵ Taggings are described by `tt:Post` and are linked to resources (`tt:Resource`), users (`tt:Tagger`), tags (`tt:TagSegment`), and the date and time of tagging (`tt:taggedOn`). This is similar to the basic structure in most other tagging ontologies and in accordance with the hypergraph model presented in Section 2.4.2.

Apart from `tt:TagSegment`, there are other classes representing tags in TT. As in OF, it is distinguished between individual tags as entered by the users (`tt:TagSegment`) and aggregated tags (`tt:Tag`). Among others, this allows to define the order in which tags are entered, as described by the property `tt:hasNextSegment`. The subclass `tt:FinalTagSegment` is used for the tag that is entered last in a tagging.

`tt:TagSegment` is in turn linked to `tt:UserTag`. This class seems to aggregate individual tags with same spelling, similar to `of:Tag` from the OF

⁵The illustration is from the website of the TSR services (<http://tagora.ecs.soton.ac.uk/tsr/>) that is no longer available. A similar illustration of an earlier version of TT can be found in [231].

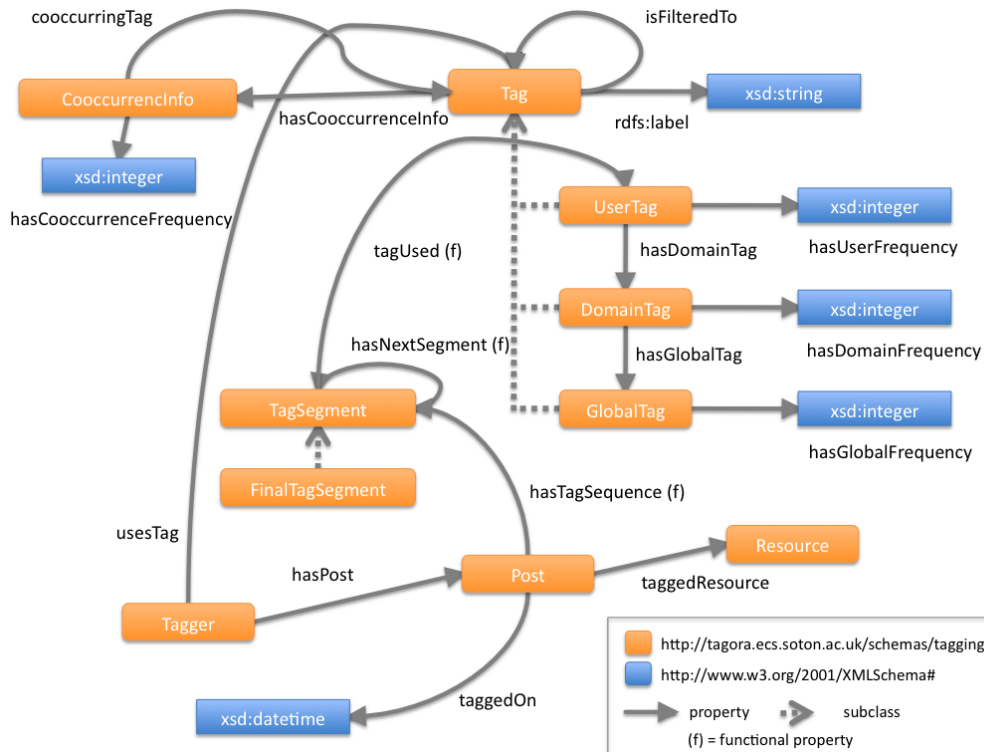


Figure 3.6 — Tagora Tagging Ontology by Szomszor.

ontology (see Section 3.1.3). This is at least indicated by the property `tt:hasUserFrequency` which seems to represent the number of times a tag is assigned by some user. Apart from the aggregation on the user level, TT also describes aggregations on the domain and global level with the classes `tt:DomainTag` and `tt:GlobalTag`, each having its own property to represent corresponding frequency values (`tt:hasDomainFrequency` and `tt:hasGlobalFrequency`). Unfortunately, we could not find any documentation of TT so that we cannot say for sure how these aggregation levels differ from each other and what they are good for.

Finally, TT defines the class `tt:CooccurrenceInfo` that seems to describe which tags are used together and how frequently they are used together. It has thus the same semantics as `scot:Cooccurrence` from the SCOT ontology. As we already discussed in Section 3.1.4, this information does not need to be explicitly represented from a conceptual point of view, as it can be inferred from other tagging concepts. However, since TT has a rather technical nature,

a separate description seems reasonable, as it was also the case in SCOT (see Section 3.1.4).

TT was primarily designed to support the TSR services within the TAGora project. It is not intended to serve as a general conceptualization of the domain of tagging and has only limited semantic interoperability. For instance, apart from three functional properties (linking `tt:Post`, `tt:TagSegment`, and `tt:UserTag`), it does not define any cardinality constraints. In contrast to most other tagging ontologies, it includes no alignments to related ontologies, neither to tagging ontologies nor to more general ontologies like SKOS or SIOC (see Table 3.2). As already mentioned, it is only poorly documented and does not offer textual descriptions or comments that clearly state the purpose and semantics of the concepts.

3.1.9 NiceTag Ontology

The last tagging ontology we examined for this review is the *NiceTag Ontology* (NT) developed by Limpens et al. [163, 192]. It was first published on the web in January 2009, under the name *Semantically Related Tag Ontology* [162]. Since then, it underwent several changes with its latest update in September 2010 [193].

In particular, NT addresses the “diverse nature and uses of tags” [161, p. 62] partly discussed in Section 2.3.3. Limpens et al. consider taggings as “social actions” that are comparable to *speech acts* [192]. Following this analogy, they call the taggings “tag actions” and regard tags as “signs”. With NT, they aim to describe the “manifold relations between a tagged resource and a tag” in more detail [161, p. 62]. For this purpose, NT defines the property `nt:isRelatedTo` along with a number of subproperties. Examples are `nt:makesMeFeel` to represent *affective tags* (see Section 2.3.3), `nt:elicitsAction` if a “resource elicits an action to be performed” [193], or `nt:hasForMedium` for tags that describe the media format of the resource (e.g., “video”, “mp3”, etc.)

Likewise, NT defines several subclasses for the tagging class `nt:TagAction` “to capture other dimensions of the tag including context and other pragmatic dimensions” [192]. For instance, `nt:SetTask` is for tags like “todo” or “toread”, while `nt:Evaluate` represents tag-based ratings like “nice” or “***” (see Section 2.3.3). Using subclasses of `nt:TagAction` to describe tag functions is possible, as its instances consist of exactly one resource and one tag. It is therefore similar to the class `tags:RestrictedTagging` from the TAGS

ontology (see Section 3.1.1) that defines the same restriction. Consequently, `nt:TagAction` has also similar limitations as `tags:RestrictedTagging`, including that taggings consisting of more than one tag are split into multiple ternary relations.

In addition, `nt:TagAction` is defined as a subclass of `sioc:Item` to “account for the shareable nature of tags, which can be seen as some sort of post” [192]. This allows to describe the source of tagging with `sioc:has_container` and the creator of “tag actions” with `sioc:has_creator`. In addition, `nt:TagAction` is defined to be either of type `nt:OwnerTagAction` or of type `nt:VisitorTagAction`. These classes have the same semantics as `ctag:AuthorTag` and `ctag:ReaderTag` from the CTAG vocabulary, i.e., they allow to distinguish between tags that are added by the authors of the resources and tags used by the people consuming the resources (see Section 3.1.6). Another concept that can similarly be found in CTAG is `nt:autoTagAction` for the representation of *automatic tags* (see Definition 14). NT also supports *semantic tagging* (see Definition 11) with the subclasses `nt:DisambiguatedTagAction` and `nt:WebConceptTagAction`. Finally, it includes a class for tag collections (`nt:TagCollection`), which has a similar function as `scot:Tagcloud` from the SCOT ontology.

Apart from that, NT introduces some interesting new concepts, such as the subclass `nt:MachineTagAction` to represent *machine tags* (also known as *triple tags*, see Section 2.3.3). It also supports *deep tagging*, i.e., the tagging of resource parts (see Section 2.3.1), with the class `nt:Point`. If tagging is used to provide access rights to a resource for someone else, this can be defined with the subclass `nt:GiveAccessRights`, while `nt:Share` is used for tags that are added to share resources with others. Unfortunately, it is not explained by Limpens et al. when to use subclasses of `nt:TagAction` and when subproperties of `nt:isRelatedTo`. It seems that both are used in combination, such that `nt:GiveAccessRights` can, for instance, either be used along with `nt:canBeReadBy` or along with `nt:cannotBeReadBy`. However, the benefit of describing one and the same tag function with partly redundant concepts remains unclear.

As for taggings (`nt:TagAction`) and tag relations (`nt:isRelatedTo`), Limpens et al. aim at a more nuanced description of the resources that are being tagged. For this purpose, NT integrates the *Identity of Resources on the Web* (IRW) ontology that distinguishes between resources (`irw:Resource`) and their representation on the web (`irw:WebRepresentation`). This accounts for the fact that resources can, in principle, be anything, even people



Figure 3.7 — Tagging represented with the NiceTag ontology [161].

or events (see Section 4.4), but that only the web representations of the resources can be tagged – at least in web contexts, which is the primary application domain of NT. Instead of directly linking `irw:Resource` and `irw:WebRepresentation`, NT defines its own classes `nt:TaggedResource` and `nt:AnnotatedResource` which are set to be equivalent.

Another difference of NT compared to the other tagging ontologies is that it represents each “tag action” as a *named graph* [61]. This is realized by making `nt:TagAction` a subclass of `rdfg:Graph` from the *Named Graphs* (RDFG) vocabulary. Limpens et al. decided for this modeling, as it enables NT to serve as “a meta-model of tagging, able to bridge the gap between existing ontologies” [192]. This is illustrated in Figure 3.7 which shows a tagging of an image from the website TWITPIC.COM [161]. The named graphs containing the RDF triples of the “tag actions” are indicated by

dotted ellipses. All “tag actions” are of type `nt:ManualTagAction`, i.e., they are not *automatic tags* (see Definition 14) but have been entered by a human being. In addition, they have all been created by the same user, as it is indicated by the `sioc:has_creator` property relations that all point to the same URI. Since each resource can be tagged at most once by each user account (according to Axiom 2), all shown tag actions belong to same tagging. Most importantly, Figure 3.7 illustrates how NT can serve as a container for concepts from other tagging ontologies. In this case, it integrates concepts from CTAG, SCOT, and MOAT.

As NT focuses on the nature and functions of tags, it should not be regarded as a comprehensive domain ontology. Most concepts are rather vaguely defined, without making use of the full capabilities of OWL. Other concepts are not included at all. For instance, NT does not contain a concept for tags but argues that nearly everything can serve as a tag. Instead, it provides descriptions of the manifold relations that can exist between resources and tags. However, the question of how to derive these differentiated relations from common taggings has not yet been sufficiently answered.

3.2 Related Ontologies and Vocabularies

Many of the reviewed tagging ontologies reuse concepts from more general and well-known vocabularies, in particular from DCTERMS, FOAF, SKOS, and SIOC (cp. Table 3.2). They either directly link concepts from these vocabularies (e.g., `foaf:Person` to describe users) or define tagging-specific classes and properties as specializations (e.g., make the tag class a subclass of `skos:Concept`).

However, none of these more general vocabularies (i.e., neither SIOC, nor SKOS, FOAF, or DCTERMS) is capable to sufficiently describe the domain of tagging by itself. Most importantly, they do not contain a concept that links resources, tags, and users, and could thus serve as tagging class. Also other domain-specific concepts, such as private tagging, automatic tagging, or the tag order, cannot be represented with these vocabularies. This is also not changed by available extensions to these vocabularies, such as the *SIOC Types Ontology Module* (SIOCT) [20] that defines only a class for tags (`sioc:Tag`) but not for taggings.

The missing tagging class is also the reason why the *SNaP Tag Ontology* (PNT) [114] cannot be regarded as a complete tagging ontology. It has

been created by Harman et al. in 2011 as one module of the *Simple News and Press Ontologies* [184] and defines only two classes, one for resources (`pnt:Taggable`) and one for tags (`pnt:Tag`)

The same is true for microformat vocabularies like `rel-tag` [90] and `xFolk` [101]. `Rel-tag` specifies how to embed tags into web pages by utilizing the `rel` attribute of (X)HTML for typed hyperlinks. It introduces the value “tag” for the `rel` attribute to indicate that the text of a hyperlink is the tag label and its `href` attribute refers to the URI of that tag. If used in combination with `xFolk`, the `rel-tag` information can be decoupled from the web page and even used to describe tags of other web resources. In addition, it can be placed in a container (with the class `xfolkentry`) and equipped with a description (with the class `description`).

Overall, `rel-tag` and `xFolk` are rather simple data models, as it is typical for microformats [13, 141]. They do not provide a comprehensive domain description but mainly define semantic markup that can be embedded into web pages, similar to RDFa [39, 104].

3.2.1 Annotea Schemas

Since tagging is a form of annotation (see Definition 2), we also looked at more general annotation ontologies and examined their suitability to describe tagging and folksonomies. One well-known project in this area dating from the early 2000s is Annotea [2]. It defines two ontologies entitled *Annotea Annotation Schema* (A) and *Bookmark Schema* (B). Figure 3.8 shows two example instances that illustrate these ontologies, as provided by the authors [136, 152]. Especially the Bookmark Schema might be used to represent basic tagging. It defines the two classes `b:Bookmark` and `b:Topic` that are linked by `b:hasTopic` and assigned to resources by `b:recalls`. Despite the different terminology, these classes are capable to represent taggings and tags, as demonstrated by Koivunen [151]. Koivunen also shows how the Bookmark Schema can be enriched with concepts from DC and FOAF to represent users, comments, and dates along with taggings and tags [151].

Since the Bookmark Schema has not been specifically designed for tagging (it was already released in 2003), it can also not be regarded as a conceptualization of the domain of tagging. Though it may be appropriate to represent basic tagging, it does not provide descriptions for more advanced concepts, such as private, semantic, or group tagging. Similar to CTAG, it also misses a clear description of how users are represented. Finally, the Bookmark

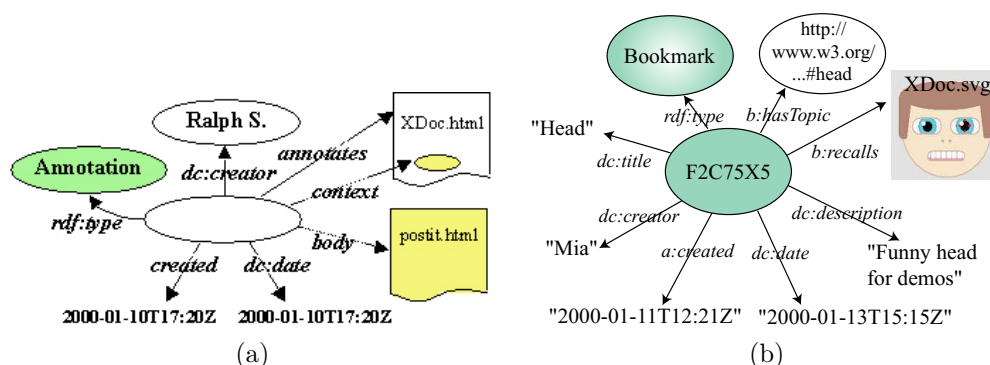


Figure 3.8 — Annotea schemas for (a) annotations [136] and (b) bookmarks [152].

Schema provides only textual descriptions but no formal specifications of the domains, ranges, and cardinalities of properties, limiting the possibilities for machine interpretation and automatic validation of the folksonomy.

3.2.2 Other Annotation Ontologies

Inspired by the Annotea project, Ciccarese et al. started development on the *Annotation Ontology* (AO) in 2010 [64, 67]. AO serves a similar purpose as Annotea but aims to overcome some of its limitations. One difference is that it does not distinguish between annotations and bookmarks but integrates both in one ontology. Furthermore, AO uses the *Provenance Authoring and Versioning* (PAV) module [65] of the SWAN ontology [66] to include a “curation process” for annotations [67]. However, it remains unclear how tagging would benefit from such a curation process. It rather seems to address more complex forms of collaborative annotation.

Similar to NT, AO reuses other ontologies to represent tagging, such as MOAT and TAGS. Concepts from these ontologies are linked by the property `ao:hasTagging`. Figure 3.9 shows an example of semantic tagging provided by the Ciccarese et al. [67] that contains concepts from MOAT and TAGS. However, the resulting representation is comparatively complex and it would be much easier to directly use MOAT and TAGS.

An interesting aspect of AO is the annotation of resource parts, such as portions of images, like it is illustrated in Figure 3.9. This capability to represent *deep tagging* (see Section 2.3.1) is one of the key characteristics of

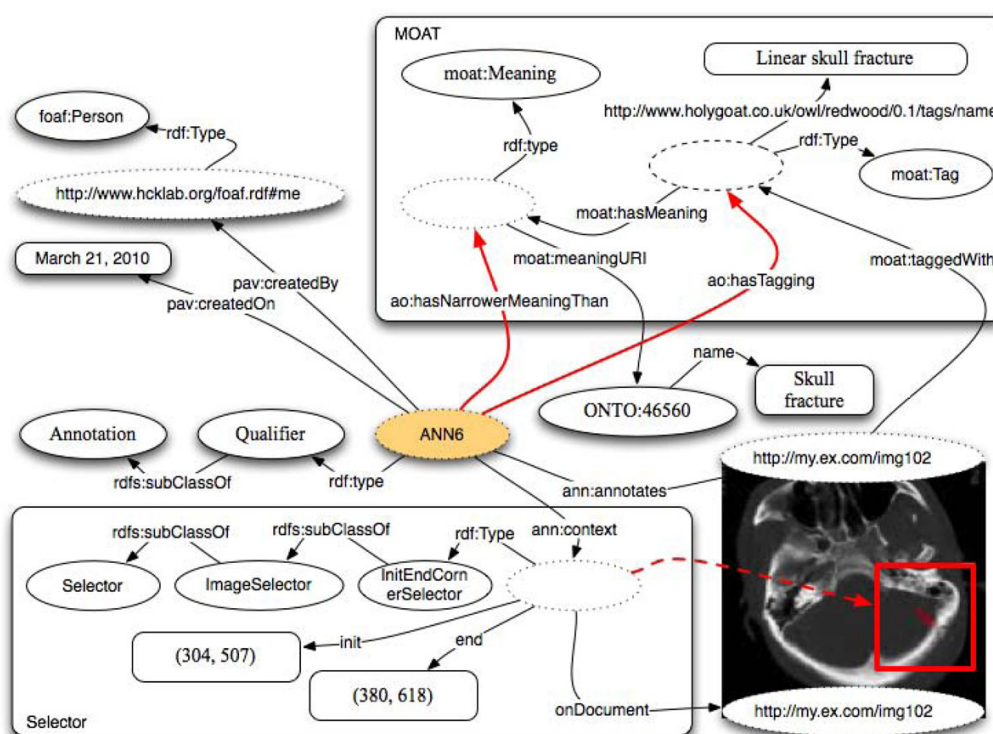


Figure 3.9 — Annotation of an image with the Annotation Ontology [67].

AO. However, it would also be possible with most of the reviewed tagging ontologies, as long as the tagged part of the resource has its own URI.

A project similar to AO had been started by Sanderson and Van de Sompel in 2011. It was also inspired by Annotea and focused on the development of an ontology for *Open Annotation* (OAC) [216]. The core structure is very similar to that of the *Annotea Annotation Schema*, consisting of the three classes `oac:Annotation`, `oac:Body`, and `oac:Target`. Therefore, it has also similar limitations as Annotea when it comes to the representation of tagging and folksonomies.

Recognizing the close relation of their work, the authors of AO and OAC joined forces in 2012 with the goal to integrate their conceptualizations and create a “common, RDF-based, specification for annotating digital resources” [16]. Since then, they published several draft specifications, with the latest version being released as W3C community draft for an *Open Annotation Data Model* (OA) in early 2013 [215]. OA implements a similar structure as OAC but additionally incorporates ideas of AO and related attempts.

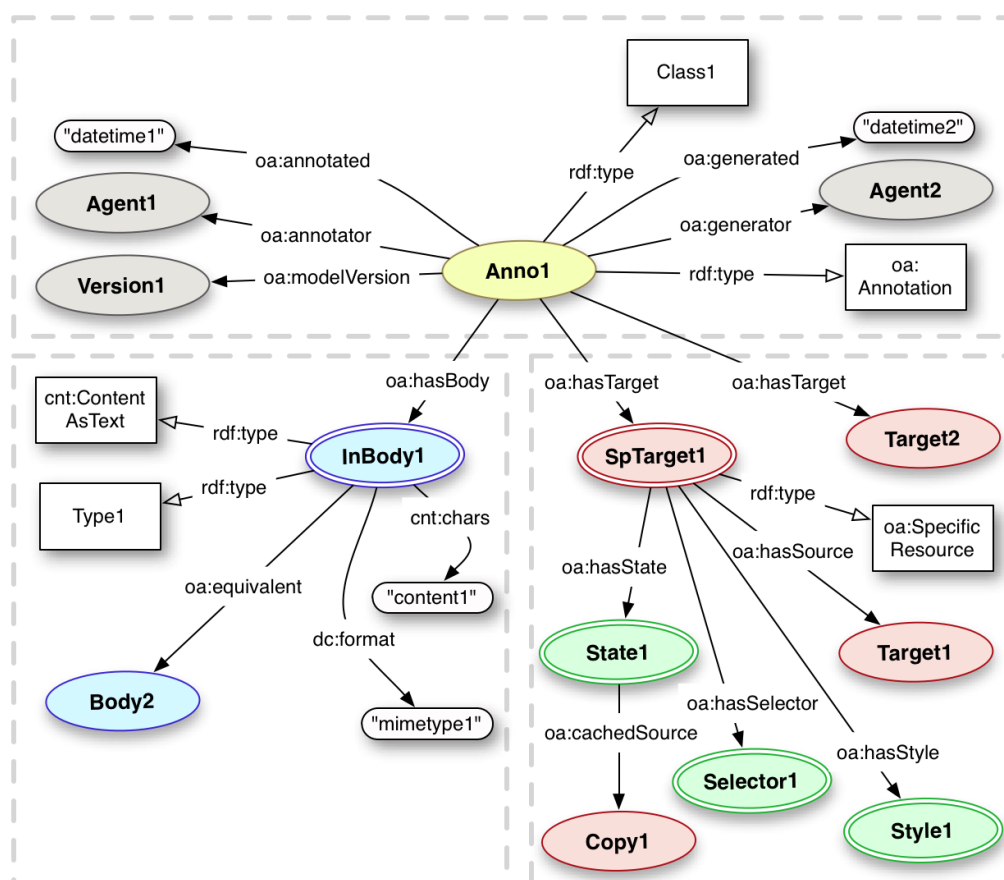


Figure 3.10 — Open Annotation Core Data Model [214].

Figure 3.10 depicts a summary of OA from one of the draft specifications [214]. It illustrates the conceptual separation of annotation, body, and target, as it was already used in OAC. Each element is further detailed by a number of concepts. However, as the core structure of OA is very similar to OAC and the Annotea Annotation Schema, it misses some concepts that are central for the domain of tagging, such as a class for tags and related properties. Like the above ontologies, OA does also not include concepts that could be used to represent more specific forms of tagging, such as private, semantic, or group tagging (see Chapter 2).

Another annotation ontology that explicitly addresses tagging is the *NEPO-MUK Annotation Ontology* (NAO) [217]. It has been developed as part of the *Social Semantic Desktop* project in 2007 [15]. Since much of the indexing is based on tagging in that project, NAO provides a class for tags (`nao:Tag`).

However, like most of the aforementioned vocabularies, it does neither contain a tagging class nor other concepts of tagging, such as users or resources. Instead, it is rather comparable to SKOS by offering preferred and alternative labels as well as descriptions for tags.

There are several other ontologies that address the annotation of resources. Examples include Co-annotea [131], Lemo [115], and M3O [211] that all define metadata vocabularies for multimedia resources. However, as these approaches do not explicitly describe tagging and folksonomies, they are not further discussed here. None of them can be regarded as a comprehensive description of the domain of tagging and folksonomies.

3.3 Towards a Unified Tagging Ontology

The reviewed ontologies revealed a large number of concepts that are important for a comprehensive description of tagging and folksonomies. Based on the review, we classified the concepts into four categories:

1. **Core concepts:** These tagging-specific concepts are essential for an interoperable conceptualization of tagging and folksonomies. They must be included in any tagging ontology.
2. **Generic concepts:** These concepts are essential but not tagging-specific. They are already defined in related ontologies and better reused than redefined.
3. **Inferable concepts:** These concepts are also important, but there is no need to define them in a tagging ontology, as they can be inferred from other concepts.
4. **Rare concepts:** These concepts have been proposed as extensions to tagging but are rarely used in practice. They do not need to be part of the core ontology but can be integrated with extensions when needed.

The result of the classification is shown in Table 3.4. Note that some of the concepts are relatively abstract and may be described by several classes and properties. Others are the result of grouping similar concepts from different tagging ontologies (e.g., `moat:AutomatedTagging` and `ctag:AutoTag` are grouped into ‘automatic tagging’). Note that ‘semantic tagging’ was considered a core concept as well as private and automatic tagging, since these forms of tagging are key to the idea of the Semantic Web.

1. Core	2. Generic
Tagging	User
Private tagging	User group
Automatic tagging	Resource
Semantic tagging	Date & time
Tag	Tag relations
Tag label	Source
Tag order	Comment
3. Inferable	4. Rare
Tag cloud	Tag category
Tag co-occurrence	Resource type
Tag frequency	Restricted tagging
Author vs. user tag	Tag polarity
Votes	Tag variations

Table 3.4 — Classification of tagging concepts.

None of the reviewed ontologies includes all concepts needed for a comprehensive domain description (i.e., all concepts listed under the categories ‘core’ and ‘generic’ in Table 3.4). Taking one tagging ontology and extending it is difficult due to conceptual incompatibilities and limitations. As discussed before, many of the reviewed ontologies define direct relations between tags and resources, which makes it difficult to integrate private tagging. Further problems are caused by partly quite restrictive domain and range axioms or imprecise alignments with concepts from other ontologies. Most importantly, only few of the reviewed ontologies are capable to represent properties of individual tags, such as the order in which tags are entered by the users or hierarchical relations on the individual tag level. The representation of individual tag properties is also required for an accurate and flexible description of semantic tagging, as we will show in Section 4.2.4.

An integration and alignment of different tagging ontologies results in similar problems. The only exceptions are **MOAT** and **SCOT** that have already

been aligned to TAGS during their development [144]. But even in these cases results an unnecessarily complex conceptualization, as can be seen in Figure 3.11. Additional extensions would further complicate matters and likely result in the aforementioned limitations (i.e., restrictive domain and range axioms and alignments, difficulties in representing individual tag properties, etc.).

For these reasons, we decided to develop a new ontology that takes the best parts of the reviewed ontologies, adds missing pieces and combines all in a coherent conceptualization. The next chapter describes the developed ontology in detail.

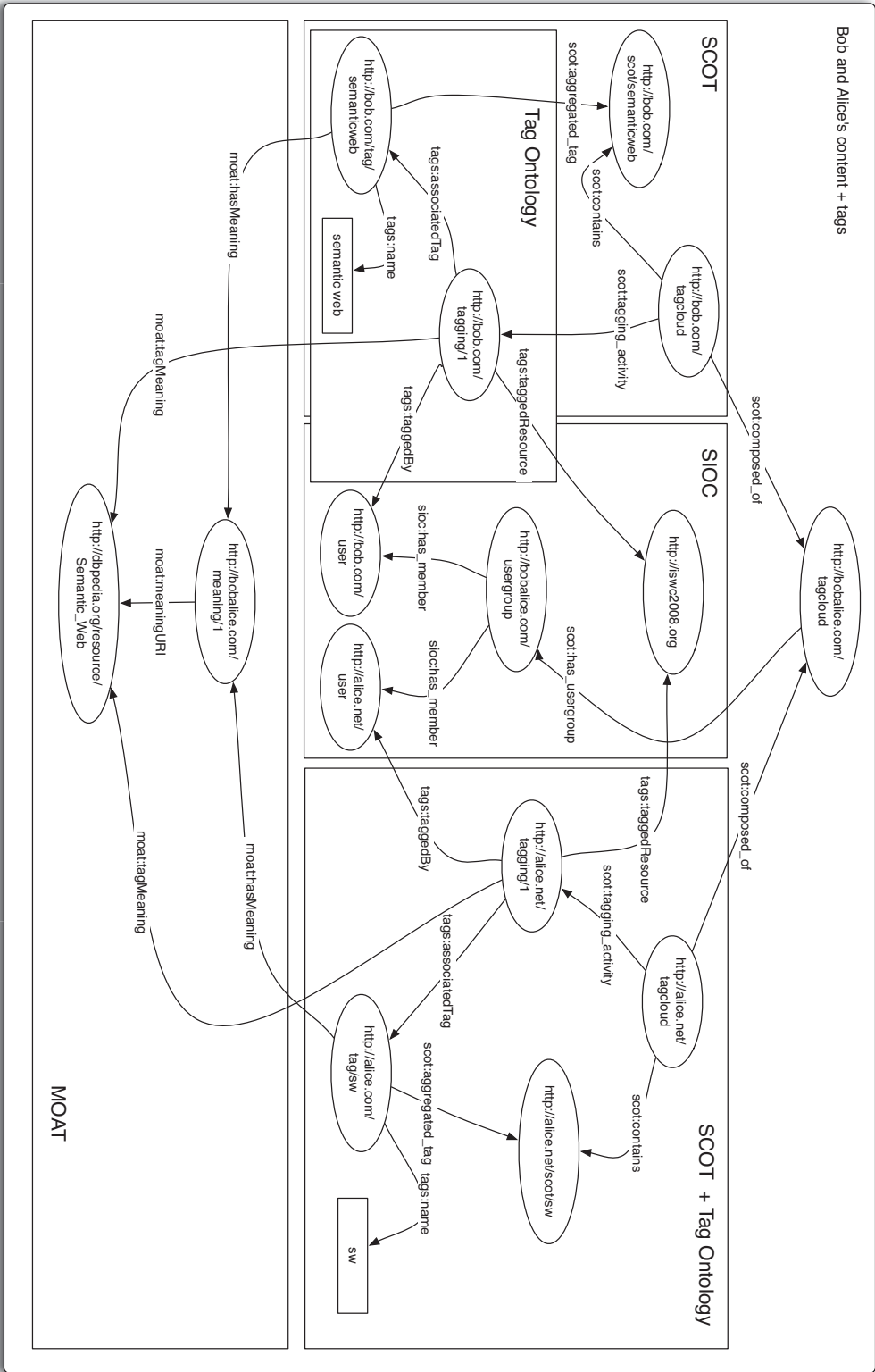


Figure 3.11 — Integration of TAGS, MOAT, and SCOT [144].

The Modular Unified Tagging Ontology (MUTO)

Based on our review, we developed the *Modular Unified Tagging Ontology* (MUTO) as a unifying conceptualization of the domain of tagging. MUTO supports various forms of tagging, including semantic, group, private, and automatic tagging, and is easily extensible.

We chose a modular design that separates the core concepts from the generic and advanced ones. Like most of the reviewed ontologies, we reused concepts from well-known ontologies, such as SIOC and SKOS, for the generic parts. Advanced concepts can be added via modules, as it is known from other vocabularies, such as SIOC (with its `access`, `types`, and `services` modules [55]) and RSS (with its `DublinCore`, `Syndication`, and `Content` modules [45]). Furthermore, we defined a module for mappings between MUTO and other tagging ontologies (currently restricted to TAGS), but separated these mappings from the core ontology, similar to as it was done in CTAG. This modularization reduces the complexity of the ontology and leads to a compact and understandable core conceptualization. It also avoids conceptual inconsistencies and varying levels of expressiveness. Finally, it helps to keep the core ontology stable with regard to future evolutions and advancements in the domain of tagging.

Parts of this chapter have already been published in [166], [169], [175], and [194].

The ontology is implemented in OWL, more specifically in the sublanguage OWL Lite, to allow for the highest possible tool support. In *Description Logic (DL)* terminology, its expressivity is $\mathcal{ALHIF}(\mathcal{D})$, i.e., it contains inverse, functional, and datatype properties, as well as some subproperty relations, but we avoided to use OWL constructs that are not part of OWL Lite, such as `owl:disjointWith` or `owl:unionOf`. With regard to the profiles defined by OWL 2 – which are all more restrictive than OWL DL [106] – MUTO is compliant with OWL 2 RL. This profile fits particularly well with the idea of tagging ontologies, as it is recommended for “relatively lightweight ontologies [that] are used to organize large numbers of individuals” and approaches “where it is useful or necessary to operate directly on data in the form of RDF triples” [106].

In addition, MUTO defines several cardinality constraints to ensure that the fundamental principles of tagging (as defined by the axioms in Section 2.4) are not violated. Since these fundamental principles are important for the interoperability and processability of folksonomies, we finally decided to define them globally, i.e., we used `owl:FunctionalProperty` instead of `owl:Restriction`.¹ This decision is also motivated by the fact that the maximum cardinality is one for all properties that need to be constrained in MUTO. Furthermore, functional properties do not force their use but can also be omitted [76], keeping MUTO flexible despite all constraints.

Figure 4.1 illustrates the main concepts and relationships of version 1.0 of the MUTO core ontology in a compact diagram.² The ontology defines two central classes, one for taggings (`muto:Tagging`) and one for tags (`muto:Tag`), which are the ‘heart’ of the conceptualization. They are both specializations of more general classes from the well-known SIOC [55] and SKOS [189] vocabularies. The other two key concepts, i.e., resources and users, are not unique to tagging. We do not need to define new classes or specializations here, as we can directly reuse concepts from existing vocabularies, namely `sioc:UserAccount` and `rdfs:Resource` in our case.

Furthermore, the MUTO core ontology includes the domain-specific concepts of *private* and *automatic tagging* (see Definition 7 and 15). They are defined as specializations of the central classes `muto:Tagging` and `muto:Tag`. Note that there is no need to define the ‘counterpart’ concepts of public and

¹ Note that this is different from earlier versions of MUTO where we used property restrictions exclusively [169].

² Note that inverse properties (`owl:inverseOf`) and subproperties (`rdfs:subproperty`) are not shown in this compact diagram.

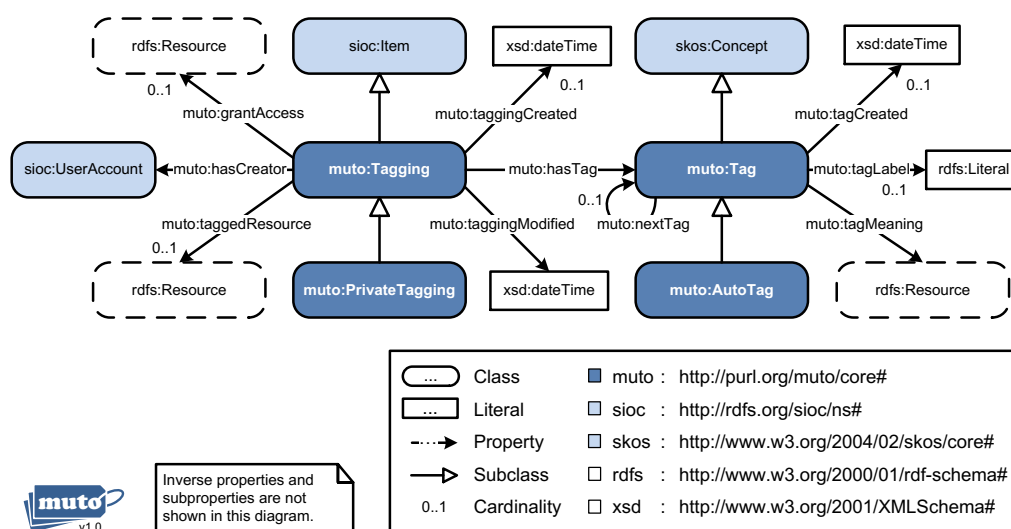


Figure 4.1 — Modular Unified Tagging Ontology (MUTO).

manual tagging, as these are the default tagging modes, i.e., the usual case is public tagging with user-assigned tags. Therefore, taggings that are not instances of `muto:PrivateTagging` are public by default. Likewise, all tags that are not instances of `muto:AutoTag` are assumed to be manually entered.

A more detailed and complete depiction of the MUTO core ontology in the form of a UML class diagram is given in Figure 4.2. We created it according to the *Ontology Definition Metamodel (ODM)* specification of the *Object Management Group (OMG)* [201]. ODM defines a family of metamodels for mappings between languages of the Semantic Web (e.g., RDF and OWL) and of Software Engineering (e.g., UML and MOF). The UML diagram shows version 1.0 of the ontology – like the compact diagram in Figure 4.1.

Note that domain and range information of inverse properties is not shown in Figure 4.2 and that the global cardinality constraints of functional properties are not visually encoded. In addition, we used a special compact notation for property relations, as they are defined by `rdfs:domain`, `rdfs:range`, and `owl:inverseOf`, in combination with the “class notation” of ODM [201] for the representation of the properties themselves. This allows us to show further property details, such as inverse properties, subproperty relations, and property types (e.g., `owl:FunctionalProperty`).

Figure 4.2 includes all concepts of SIOC, DC, and SKOS that are reused in MUTO. To roughly indicate the ontology parts where they are used, we added

boxes for the three vocabularies on top of the UML diagram. The complete MUTO specification is publicly available on the web at the persistent URL <http://purl.org/muto>. Its RDF/XML serialization is also given in Listing C.1 in Appendix C of this thesis.

Based on the four key element sets of taggings, tags, users, and resources defined in Section 2.4, we describe the MUTO ontology in more detail in the following. We also discuss modeling alternatives, as used by the ontologies we reviewed in Chapter 3, and illustrate the application of MUTO in an example scenario. Finally, we evaluate MUTO by comparing it with the reviewed ontologies. We also report on two projects where the ontology has been successfully applied.

4.1 Taggings

The central class `muto:Tagging` describes the taggings, i.e., the set of annotations A we defined in Section 2.4. Its instances describe the n -ary relations that link the resources, tags, and users, as illustrated in Section 2.4.2. They are enriched with further information, such as the date and time of tagging (see Section 2.4.3). Using classes to represent n -ary relations is well-known from many modeling languages (like UML with its *association class* construct) and common practice in OWL [199]. It thus comes as little surprise that nearly all reviewed tagging ontologies use this kind of modeling (see Chapter 3).

Furthermore, we define `muto:Tagging` to be a subclass of `sioc:Item`. We regard this as an adequate alignment, since SIOC has been designed to describe “user-generated content [from] online community site[s]” [19]. Oddly enough, apart from NT, none of the reviewed tagging ontologies made this alignment. Although SCOT and CTAG also link to `sioc:Item`, they use it to describe the resources that are being tagged and not for the taggings themselves. Though this use of `sioc:Item` is semantically possible, it is very restrictive, as we will detail in Section 4.4.

Apart from `sioc:Item`, there is a number of other concepts from the SIOC vocabulary that can fruitfully be reused in the domain of tagging. For instance, we do not need to create a new concept for comments assigned to taggings, as we can take `sioc:note`. Likewise, we do not need a new concept for the tagging source but can reuse the concepts that are already provided by SIOC. In particular, we can first group taggings with `sioc:Container`

and then link them to a joined source with `sioc:has_space`. This way of describing the source of tagging with SIOC concepts has also been used by NT (see Section 3.1.9).

4.1.1 Tagging Comments

Apart from `sioc:Item`, there are a number of other concepts from the SIOC vocabulary that can fruitfully be reused in the domain of tagging. Since we made `muto:Tagging` a subclass of `sioc:Item`, it can be directly enriched with SIOC properties. For instance, we do not need to create a new property for comments assigned to taggings, as we can take `sioc:note`.

UTO and TO are the only two reviewed ontologies that explicitly consider tagging comments. In contrast to MUTO, UTO does not directly use `sioc:note` but defines its own property (`uto:hasComment`), making it equivalent to `sioc:note` (via `owl:equivalentProperty`). The range of this property is given with `xsd:string`. TO also defines its own property `to:hasNote` to represent tagging comments, though without making any alignments or defining an explicit range.

4.1.2 Tagging Source

Likewise, we do not need to define a new concept for the source of tagging (as done in TO or OF), but can reuse the concepts that are already provided by SIOC. In particular, we can first group taggings with `sioc:Container` and then link them to a joined source with `sioc:has_space` (see example in Section 4.5). Providing a concept for the source of tagging is important, for example, if folksonomies from different tagging systems are merged (see Section 2.4.3).

TO defines the class `to:ServiceDomain` and the properties `to:hasServiceDomain`, `to:hasServiceName`, and `to:hasServiceHomepage` to represent the source of tagging. OF uses a similar modeling structure with `of:Source`, `of:hasSource`, and `of:hasSourceName`.

Though SCOT and UTO also define source concepts, these concepts do not represent the source of tagging but of other ontology elements: In SCOT, the source concept is for tag clouds (`scot:Tagcloud`). It links to `sioc:Site`, a subclass of `sioc:Space` that can also be used in MUTO instead of `sioc:Space`. UTO, in turn, defines the source for the resource that is tagged (`uto:Object`) with the class `uto:Source` and the property `uto:hasSource`. It aligns these

concepts to SIOC by making `uto:Source` a subclass of `sioc:Community` and `uto:hasSource` equivalent to `sioc:has_host`. However, this alignment is semantically problematic, as `sioc:has_host` and `sioc:Community` are not linked with each other in SIOC.

In addition, UTO links to DCTERMS by defining an equivalence relation between `uto:hasSource` and `dcterms:source`. The property `dcterms:source` would indeed provide a possible alternative to link to the tagging source, but we decided to stay in the SIOC vocabulary, as `muta:Tag` is already a subclass of `sioc:Item`. Moreover, `sioc:Container` and `sioc:Site` provide efficient means to group taggings and reference the source in our case. A future extension to the MUTO core ontology might further detail the source description by defining tagging-specific subclasses and properties for the SIOC concepts.

4.1.3 Date and Time of Tagging

As already mentioned in Section 2.4.3, date and time of tagging are important information that should be part of the core conceptualization. This is also reflected by the large number of reviewed tagging ontologies that consider this information: All ontologies but CTAG (which has no tagging class) include a corresponding concept. While OF and UTO define each their own concept for this purpose, TAGS reuses the *Dublin Core Metadata Element Set* [54] by making `tags:taggedOn` a subproperty of `dc:date`. Since MOAT and SCOT are based on TAGS, they also include this property (see Section 3.1.4 and 3.1.5). TO uses `dcterms:date` which is semantically the same as `dc:date` but additionally specifies a range of type `rdfs:Literal` [52].

However, specifying `rdfs:Literal` as range is still very general. Although DCTERMS recommends the use of some kind of “encoding schema”, such as the *W3CDTF profile of ISO 8601*, it allows to “express temporal information at any level of granularity” [52]. A part of the reviewed ontologies include more precise specifications of the date and time format. UTO and A (the Annotea Annotation Schema) describe the expected format in `rdfs:comment` statements: While UTO expects only information about the month and year of tagging (“MmmYY”), A recommends to write date and time information as standardized by ISO 8601, i.e., in the form “yyyy-mm-ddThh:mm:ssZ”. Though these specifications inside ontology comments are helpful for human readers, they are not sufficient for machine interpretation and automatic validation of the folksonomy data.

In order to define this information as explicitly and precisely as possible in MUTO, we decided to take up the modeling of `T0`, `OF`, and `TT` and used `xsd:dateTime` from the “XML Schema Datatype” vocabulary as range for `muto:taggingCreated`. The datatype `xsd:dateTime` implements the aforementioned ISO 8601 standard that is also supported by W3C documents [246, 251]. This explicit range increases the interoperability of the ontology, as it forces the use of a standardized format for date and time information that can be unambiguously interpreted. This is important, as a wrong interpretation of the date and time information could result in a wrong use of the folksonomy data and biased statistics.

The clearly specified range was one reason why we did not directly use `dcterms:created` in the MUTO core ontology but defined our own subproperty. Another reason was that we wanted to make `muto:taggingCreated` a functional property with a global cardinality of one. This ensures that there is one unique creation date for each instance of `muto:Tagging`. It is again the same modeling as in `T0`, while it is similar to the modeling in `OF` that uses a local cardinality constraint instead (i.e., with `owl:Restriction` and `owl:onProperty`).

Apart from the creation date and time, MUTO is also capable to represent the date and time when taggings are edited. This can be useful information for tagging systems, for instance, to sort taggings by date of last modification. The only other tagging-related vocabulary that explicitly considers modifications is the Annotea Annotation Schema defining `a:modified` and `a:created` as subproperties of `dc:date`. In contrast, we made `muto:taggingModified` a subproperty of `dcterms:modified`. This is conceptually more accurate but only possible in DCTERMS, since DC does not contain such a concept. As for `muto:taggingCreated`, we set the range to `xsd:dateTime`.

4.1.4 Private Tagging

MUTO supports the special case of private tagging (see Definition 7) with its dedicated subclass `muto:PrivateTagging`. Apparently, MUTO can only provide a description of the concept; the correct implementation of privacy constraints remains the duty of the interactive system. As mentioned above, taggings that are not instances of this subclass are public by default. Also note that private taggings can be made available to specific users or user groups with `to:grantAccess`, as we will detail in Section 4.3.2.

The only other reviewed tagging ontology that contains a concept for private taggings is **T0**. It defines the class `to:VisibilityEnum` with individuals `to:Private`, `to:Public`, and `to:Protected`. We decided against this solution, as it decouples the privacy information from the tagging instances which can be problematic (e.g., if only parts of the ontology are used or accessed).

4.1.5 Linking Taggings and Tags

Taggings and tags are linked by the property `mut0:hasTag`. The cardinality of this property is not restricted, as a tagging can theoretically consist of an unlimited number of tags. **MUTO** even allows for taggings without any tags to support cases where users first purely index a resource and add tags later, as it is possible in some systems (e.g., **DELICIOUS**). However, the cardinality of the inverse property `mut0:tagOf` is restricted, since each instance of `mut0:Tag` belongs to exactly one tagging (therefore, the functional property type). This results from the fact that tags with the same label are not merged in **MUTO** but that each tag is a unique instance (see Section 4.2).

Some of the reviewed tagging ontologies limit the number of tags per tagging. For instance, the already mentioned `tags:restrictedTagging` class of the **TAGS** ontology is specified to have “precisely one associated resource, and one associated tag” [196]. **MOAT** reuses `tags:RestrictedTagging` and thus also applies this restriction. Such restrictions are conceptually inaccurate, as they contradict Axiom 1 stating that a tagging can consist of more than one tag. Unfortunately, the authors of **TAGS** and **MOAT** do not further explain why they define these restrictions. One can only speculate that they aim to model the tripartite character of folksonomies discussed in Section 2.4.2. However, the benefit of this modeling remains unclear, especially as it likely results in a redundant representation of certain tagging information (e.g., the date and time of tagging). **OF**, in contrast, defines a minimum cardinality of one tag for its `of:hasAnnotationTag` property, which is more restrictive than **MUTO**, since it does not allow for taggings without tags.

4.2 Tags

The second core class of the **MUTO** ontology is `mut0:Tag`. Each tag is an instance of that class with its own URI, as in nearly all of the reviewed tagging ontologies. However, in contrast to most other tagging ontologies, tags with same labels are *not* merged in **MUTO**. In our understanding, such an

aggregation of tags is not part of the core conceptualization, as discussed in Chapter 3. Among others, this would take the possibility to define properties for individually entered tags, such as the later described `mut0:tagMeaning` or `mut0:nextTag` (see Section 4.2.1). We rather consider the aggregation of tags a feature of the tagging system (e.g., required for the generation of tag clouds).

However, if aggregated tags should also be part of the conceptualization, the MUTO core ontology may be extended by a specific class or module for this purpose, similar to the modeling in OF and TT. These two ontologies are the only reviewed tagging ontologies that clearly distinguish between aggregated and individual tags. OF defines the class `of:AnnotationTag` and TT the class `tt:TagSegment` to represent the actual tags as entered by the users, which is similar to the modeling in MUTO.

Semantically, tags are related to what is commonly represented by `skos:Concept`. We thus made `mut0:Tag` a subclass of `skos:Concept`, like in many of the reviewed tagging ontologies. The benefits of subclassing `skos:Concept` are similar to those described above for the subclassing of `sioc:Item`. For instance, it allows us to reuse SKOS concepts in MUTO, such as `skos:narrower` and `skos:broader` for the representation of hierarchical tag relations (see Section 4.2.5).

4.2.1 Tag Label

However, not all SKOS concepts can reasonably be used with `mut0:Tag`. Especially the application of `skos:altLabel`, `skos:hiddenLabel`, and `skos:prefLabel` to `mut0:Tag` does not make sense, as a tag has always exactly one label according to Axiom 4. Accordingly, a tag that has more than one label is, strictly speaking, not a tag. This is implemented in MUTO by making `mut0:tagLabel` a functional property, which allows for only one label per tag instance. If additional labels are added to the tag, the tag would transform into a meaningful concept that would then be linked with `mut0:tagMeaning` in MUTO (see Section 4.2.4).

This is different in TO which defines the subproperty `to:prefTagLabel` in addition to `to:hasTagLabel` for tags with multiple labels. Similarly, OF defines the specialized properties `of:hasAltLabel`, `of:hasHiddenLabel`, and `of:hasPrefLabel`, which are semantically identical (though not linked) to the aforementioned SKOS properties. However, these are the only two tagging ontologies that distinguish between different types of tag labels, and

it remains unclear why they do not directly use the semantically identical properties of `skos:Concept`.

The only other tagging ontology that explicitly specifies Axiom 4 is MOAT. It equips the class `moat:Tag` with a cardinality constraint that permits only one label per tag instance. While this is similar to our modeling, we decided for a functional property, since we regard the cardinality constraint an inherent characteristic of the `muto:tagLabel` property.

4.2.2 Date and Time of Tags

The date and time information for tags (`muto:tagCreated`) is conceptually separated from that for taggings described in Section 4.1.3. This is useful if certain tags of a tagging are added at a later time. Omitting the separate date and time information in these cases may result in biased tag statistics and wrong conclusions about the evolution of the folksonomy. As for `muto:taggingCreated`, the range of `muto:tagCreated` is `xsd:dateTime` to force a standardized format and increase the ontology's interoperability (see Section 4.1.3). In addition, `muto:tagCreated` is once again a functional property to ensure that there is no more than one date and time of creation for each instance of `muto:Tag`.

However, since the creation date and time of a tag is usually the same as the creation date and time of the associated tagging, we made `muto:tagCreated` an optional property to prevent a redundant representation of this information. If no separate date and time information is given for a tag, it is assumed that the tag has been created at the same date and time as the associated tagging (i.e., `muto:tagCreated = muto:taggingCreated`).

As discussed before, a tag consists of exactly one label. In other words, editing a tag label means the substitution of one tag by another. This is the reason why MUTO does not define a separate modification date and time for tags. If a tag label is edited by the user, a new tag instance is created having the same property values as the original one except from the creation date and label. If other properties of a tag are changed (e.g., the tag order), this can be considered as a change of the tagging and can be described with `muto:taggingModified`. Prohibiting changes to tag labels helps to prevent misuse and wrong interpretation of folksonomy statistics.

The only two reviewed tagging ontologies that define date and time information for tags are SCOT and CTAG. Instead of the creation date and time, SCOT represents the date when a tag was last used (`ctag:last_used`).

However, this information only makes sense for aggregated tags and is thus not required in MUTO. In case of CTAG, the definition of the property `ctag:taggingDate` is necessary, as CTAG does not define a tagging class. The property is thus rather comparable to `muto:taggingCreated` than to `muto:tagCreated`.

4.2.3 Automatic Tags

The MUTO core ontology is also capable to represent automatic tagging (see Definition 15) by defining the subclass `muto:AutoTag` for `muto:Tag`. Describing manual and automatic tagging in the same ontology makes sense, as it avoids a redundant modeling and facilitates the transformation of automatic tags into manual (i.e., user validated) ones. In addition, it allows to associate both types of tags with a single tagging instance, which is conceptually in line with how automatic tags are often applied, namely as a complement to the tags entered by the user.

This is the main difference to CTAG, NT, and a prior version of MUTO [169] that all define automatic taggings as a specialization of the tagging class. In these cases, it is not possible to have manual and automatic tags in the same tagging instance. The same holds for MOAT which defines an extra `moat:TagType` class with individuals `moat:AutomatedTagging` and `moat:ValidatedTagging`. As this class is linked to `tags:RestrictedTagging`, it can contain only one tag and hence not a combination of manual and automatic tags.

Note that MUTO defines no separate class for manual tags in contrast to CTAG, NT, and MOAT. As mentioned above, every tag that is not an instance of `muto:AutoTag` is assumed to be manually added, as this is the default case of tagging. If there are automatic tags, it is assumed that they have been created by the tagging system. Since the tagging system can be represented along with the tagging class as source information (see Section 4.1.2), there is no need for an additional property representing this information. If the source of automatic tags is not the tagging system itself but some external service, this information can be represented by using `skos:note` and adding a corresponding comment to the instances of `muto:AutoTag`.

4.2.4 Semantic Tags

MUTO follows Axiom 4 and strictly distinguishes between tags (which have exactly one label) and concepts (which can have more than one label). However, it supports the mapping between tags and concepts through the property `muto:tagMeaning`. This is particularly useful in semantic tagging, where the meaning of tags is disambiguated by linking them to well-defined resources (see Definition 11 and Section 3.1.5). Basically, we can distinguish between two forms of disambiguation: Linking a tag to an existing concept, such as a DBPEDIA resource [51] or WORDNET term [191], or transforming it into a new one, for instance, by adding a definition and/or further labels to the tag. Both forms can be expressed via `muto:tagMeaning`. Like `muto:taggedResource`, it has no explicit `rdfs:range` and can thus be linked to any type of resource, i.e., all instances of `rdfs:Resource`, as indicated in Figure 4.1.

This approach of disambiguating tags by linking them to well-defined resources is similar to the ones of MOAT and CTAG. However, MOAT and CTAG disambiguate complete taggings, not single tags, which is conceptually less precise (see Section 2.2.3). It furthermore requires that a new tagging instance is created for every single tag, which is not wanted in MUTO for the aforementioned reasons (see Section 4.1.5).

The property `muto:tagMeaning` can also be used to indicate synonym tags (see Section 2.2.2), simply by linking all tags with identical meaning to the same resource. This includes tags that are different variations of the same term (e.g., if one tag has an underscore as delimiter and another a hyphen).

Note that a tag has always a label on its own (Axiom 4), even if it is linked to a concept having the same label. This is important, as `muto:tagLabel` represents the label that is chosen by the user (e.g., manually entered or selected from a list of suggestions) – at least in manual tagging. It may not be identical to the label defined for the referenced concept by some authority (e.g., via `skos:prefLabel`). Also note that `muto:tagMeaning` is an optional property that is only needed for semantic tagging.

The cardinality of `muto:tagMeaning` is not restricted, as there may be several resources describing the semantics of a tag (e.g., a *Wordnet* term and a *DBpedia* resource). In other cases, it might be difficult to assign a unique meaning to a tag, for instance, in the automatic disambiguation of tags [42, 99]. MUTO defines the subproperty `muto:autoMeaning` to indicate automatic disambiguations. Like in the above cases, all disambiguations

that are not of type `muto:autoMeaning` are assumed to be specified by the user who assigned the tag. Thus, it can always be inferred which user added a meaning to a tag.

4.2.5 Tag Relations

Since `muto:Tag` is a subclass of `skos:Concept`, we can reuse the properties of `skos:Concept` to define basic relations between tags. In particular, we can use the aforementioned `skos:narrower` and `skos:broader` properties for hierarchical relations and `skos:related` for associative ones. These relations are a good example why it is important to have a concept for individual tags in the ontology: Only because one user defines a relation between two tags, this does not mean that other users would also agree with that relation.

As we know from Chapter 3, some of the reviewed tagging ontologies define own tag relations: TAGS specifies the property `tags:relatedTag` and UTO the property `uto:hasRelatedTag` that describe both some kind of relation between tags. However, we see no reason to define such a property in MUTO, as we can directly reuse `skos:related`. This solution is also proposed by TO [148].

A second tag relation defined in TAGS is `tags:equivalentTag` which describes an equivalence relation between tags, expressing that “whenever one [tag] is associated with a resource, the other tag can be logically inferred to also be associated” [196]. TO defines a semantically identical relation with `to:sameTag`. However, we do not need such a relation in MUTO, as we can express semantic equivalence between tags by linking them to the same resource with the property `muto:tagMeaning` (see Section 4.2.4).

4.2.6 Tag Order

The only tag relation we explicitly defined in the MUTO core ontology is `muto:nextTag` (and its inverse counterpart `muto:previousTag`) to describe the sequential order in which tags are entered by the users during the act of tagging. A comprehensive tagging ontology must be able to represent this information, as users would expect the ordering of tags to remain the same whenever they access a tagging (see Section 2.4.3).

The only two reviewed tagging ontologies that include information about the order of the tags in a tagging are OF and TT. OF defines the property `of:hasPosition` to assign each tag an integer value representing its position

in the list of entered tags (`xsd:int`). This solution can cause problems of non-unique positioning, for instance, if counting starts at zero instead of one or if the same integer value is assigned to several tags by accident. Thus, we finally decided to use the modeling with `muto:nextTag` and `muto:previousTag` (after adopting OF's modeling in an earlier version of MUTO [169]). Since there should be no more than one preceding and/or following tag, we made both `muto:nextTag` and `muto:previousTag` functional properties. This is similar to the modeling in TT, though MUTO does not define an extra concept to indicate the last tag of the sequence in contrast to TT.

Using such property relations to describe sequences is common practice in OWL. RDF constructs like `rdf:List` or `rdf:Seq` are no alternatives in our case, as they are not compatible with OWL Lite and limit the possibilities for reasoning. Other alternatives could be the *Ordered List Ontology* [38] or the *List Ontology* [82]. However, such approaches would be conceptually and computationally too expensive for a concept as simple as tag order.

4.3 Users

MUTO reuses `sioc:UserAccount` to represent the accounts of the users who created the taggings (i.e., the set of users U). Linking users by their accounts is more accurate and flexible in this case than linking them directly (e.g., by using `foaf:Agent` as in TAGS), since it allows one user to have several accounts (e.g., one for work-related and one for personal taggings).

An alternative to `sioc:UserAccount` would have been the semantically nearly identical class `foaf:OnlineAccount`. We decided for the SIOC variant because we also used other concepts of this vocabulary along with `muto:Tagging` and can thus stay in one namespace. Moreover, it provides a good possibility to represent group tagging, as we will describe below. However, since `sioc:UserAccount` is a subclass of `foaf:OnlineAccount`, concepts from the FOAF vocabulary can easily be linked and used to describe users and user-related information (e.g., `foaf:gender` or `foaf:age`). Other useful concepts from the SIOC vocabulary are, for instance, `sioc:email` or `sioc:follows`. The latter tells that one user is following the taggings of another and is represented by linking the corresponding instances of `sioc:UserAccount`.

MUTO does not directly use `sioc:has_creator` to link the user accounts, but defines its own subproperty `muto:hasCreator`. It is a functional property, i.e.,

it has a global cardinality of one. It thus implements Axiom 1 by specifying that each tagging is linked to exactly one user account. We used the same combination of a subproperty relation and a functional property for the earlier described `muto:taggingCreated` and `muto:tagCreated` properties.

A functional property is also used by T0 to define this cardinality constraint, while OF uses a local property restriction (with `owl:cardinality`). TAGS and its extensions SCOT and MOAT do not define this cardinality, neither do UTO, CTAG, and NT. Furthermore, none of the reviewed tagging ontologies links to the user account. Instead, they link either directly to the representation of the user (e.g., to `foaf:Agent` like TAGS) or define their own class for this purpose (e.g., `of:User` or `uto:Tagger`). However, we decided for `sioc:UserAccount` for the above reasons.

In addition, CTAG distinguishes between `ctag:AuthorTag` and `ctag:UserTag` (see Section 3.1.6). Similarly, NT differs between `nt:OwnerTagAction` and `nt:VisitorTagAction` (see Section 3.1.9). We did not include similar concepts in the MUTO core ontology for two reasons: 1) this information is often not available, i.e., the author of a resource is not known or not provided in a machine-readable form, and 2) if the author of the tagged resource is known, it can be easily be inferred if he or she is the same as the creator of the tagging. Hence, it does not need to be separately represented.

Note that there is no direct link between users and tags in MUTO, as there is also no direct link between users and resources. It would not only be redundant and conceptually imprecise to add such direct links, but could also be problematic with regard to private tagging (see Section 4.2 and 4.4).

4.3.1 Group Tagging

Since MUTO links to user accounts and not to users, it is flexible with regard to the representation of group tagging (see Definition 13). It enables the representation of different forms of group tagging, including the three variants outlined in Section 2.3.2. Our recommended modeling is to use `sioc:Usergroup` which can be linked to `sioc:UserAccount` by `sioc:member_of`. That way, it remains transparent which user added which taggings (assuming that each user account is used by only a single user).

The representation of group tagging is more restrictive in the reviewed ontologies that include this concept: T0 links to `foaf:Group` in addition to `foaf:Person`, SCOT uses `sioc:Usergroup` to indicate the creators of tag clouds, and `uto:Tagger` is not only a subclass of `foaf:Agent` but also of

`sioc:Usergroup`. However, all these modeling alternatives are less flexible than the one we used in `MUTO`, as they do not link to the user accounts but directly to the representation of the user group. This also results in a lower traceability compared to the recommended `MUTO` modeling, as it cannot be said which user actually created the tagging.

4.3.2 Access Control

As indicated by its name, the `mut0:grantAccess` property grants access to selected taggings for certain users (e.g., a friend) or user groups (e.g., all friends), as already mentioned in Section 4.1.4. It is mainly meant to provide exclusive access rights for private taggings, i.e., to allow others besides the creator of a private tagging to access this tagging. The range of this property is not further specified in `MUTO`.³ The property may be linked to instances of `sioc:UserAccount` or `sioc:Usergroup`, but it could also be linked to instances of `foaf:Agent`, `dcterms:Agent`, or similar. Ultimately, the actual range depends on the application context and the concrete implementation in the tagging system.

The property `mut0:grantAccess` may, however, also be used in public tagging, for instance, to state that a tagging has been sent to another user, as it is possible in some tagging systems (e.g., `BIBSONOMY`). Generally speaking, the property simply states that there is some kind of relation between a tagging and a user who is not the creator of that tagging. This rather vague specification is intended to offer a flexible representation for all features that define such a relation. Though none of the reviewed tagging ontologies includes such a property, its inclusion in the core ontology makes sense, not only but especially to allow for the sharing of private taggings with other users.

4.4 Resources

Resources are linked by the property `mut0:taggedResource`. Like `mut0:tagMeaning` and `mut0:grantAccess`, the property has no explicit range and can thus be linked to all instances of `rdfs:Resource`, as indicated in Figure 4.1. Since `rdfs:Resource` is “the class of everything” and “all other classes are subclasses of this class” [59], it means that taggings can be linked

³Note that this was different in an earlier version of `MUTO` that specified a range (in combination with `owl:unionOf`) [169].

to any kind of resource.⁴ This is in line with the general idea of tagging, where the tagging system determines what is considered a resource (see Section 4.4).

While this modeling is also used by the early TAGS and TO ontologies, many of the later developed tagging ontologies define a specific class that resources must be instances of. SCOT reuses `sioc:Item` for this purpose, which is still relatively generic but nevertheless limited to the thematic context of SIOC, i.e., user-generated content in online community sites [19]. Similarly, CTAG defines the class `ctag:TaggedContent` as subclass of `sioc:Item` in its “mappings” module (see Section 3.1.6). UTO is even more restrictive by aligning `uto:Object` to `foaf:Document`, `foaf:Image`, and `sioc:Post` (a subclass of `sioc:Item`). OF defines its own class `of:Resource`. All these modelings are too restrictive in our view if we take into account that a resource can be “anything that is uniquely addressable within a software system or computer network” (see Definition 12).

`muto:taggedResource` is semantically close to `sioc:about`, a property of `sioc:Item` specifying that the item “is about a particular resource” [55]. We thus made `muto:taggedResource` a subproperty of `sioc:about` (see Figure 4.2). Furthermore, we made `muto:taggedResource` a functional property, since a tagging is always uniquely linked to a single resource according to Axiom 1. Note that the inverse property `muto:taggedWidth` does not define any cardinality constraint, as several taggings from different users are often associated with one and the same resource.

This is again the same modeling as in TO where `to:hasTaggedResource` is also a functional property. OF, by contrast, does not implement this cardinality constraint globally for the property but locally for the tagging class using `owl:cardinality`. `owl:cardinality` is also used in TAGS to define this constraint – unfortunately, without any explanation why the constraint is applied to the subclass `tags:restrictedTagging` and not to `tags:Tagging`. Since this cardinality constraint is an inherent characteristic of the `muto:taggedResource` relation, we decided to use the modeling of TO and made it a functional property in MUTO.

In contrast to many of the reviewed tagging ontologies, MUTO does *not* define a link between the tag and the resource that is being tagged. Tags and resources are only indirectly linked by `muto:Tagging` in MUTO. Though a

⁴ However, note that the linked resources must be of type `owl:Thing` if compliance with sublanguage OWL Lite and OWL 2 RL should continue to be satisfied.

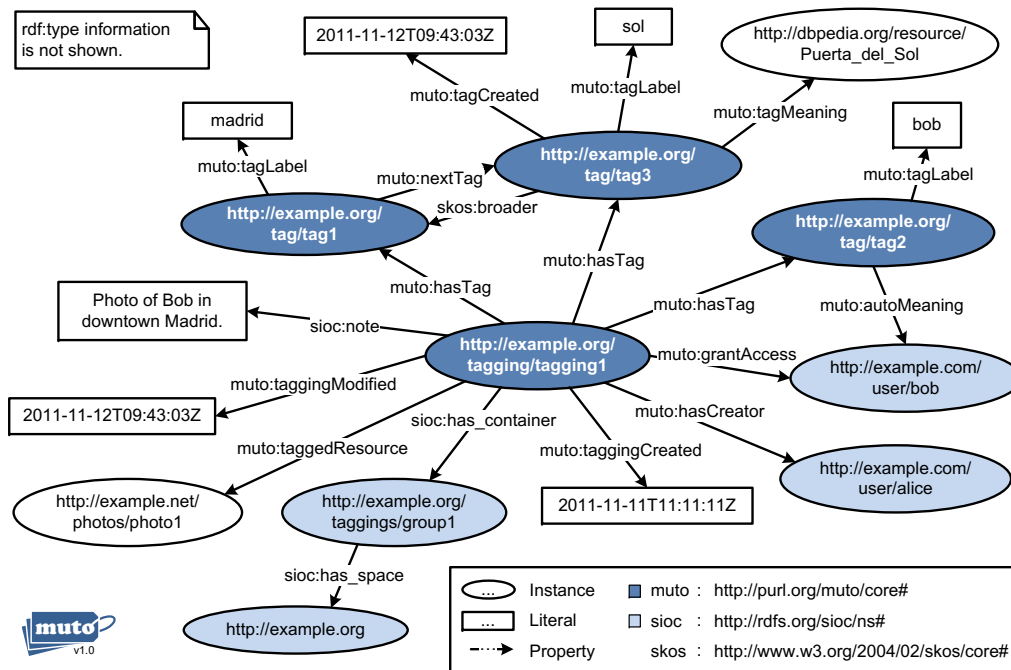


Figure 4.3 — Example RDF graph depicting a private tagging.

direct link between tags and resources may be beneficial in some cases (e.g., when querying the folksonomy [240]), it is redundant and conceptually imprecise, as we already discussed in Section 3.1.1. In addition, it would be problematic in private tagging, as the privacy information is part of the taggings, not of the tags. For instance, if the direct link is used to retrieve tags for a certain resource, there is a danger that the privacy information is not considered and tags from private taggings are also retrieved and shown to users who shall not have access to them.

4.5 Example Scenario

Figure 4.3 depicts a scenario of using the MUTO ontology with the social bookmarking system EXAMPLE.ORG. It shows the RDF graph of user Alice who annotated a photo from the website EXAMPLE.NET.⁵ Assume that Alice interacts with a well-designed user interface. She does not get in touch with the ontology but it is rather used for the design and internal representation

⁵ Visualizing RDF graphs as node-link diagrams is very common and also used in the RDF-related recommendations of the W3C [147, 179].

in the interactive system and/or for sharing the folksonomy with other systems. Listing 4.1 provides the OWL code of the example in RDF/Turtle format.

```

@prefix muto: <http://purl.org/muto/core#> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
<http://example.org/tagging/tagging1> a muto:PrivateTagging;
  muto:taggedResource <http://example.net/photos/photo1>;
  muto:hasCreator <http://example.com/user/alice>;
  muto:hasTag <http://example.org/tag/tag1>,
    <http://example.org/tag/tag2>,
    <http://example.org/tag/tag3>;
  muto:taggingCreated "2011-11-11T11:11:11Z";
  muto:taggingModified "2011-11-12T09:43:03Z";
  muto:grantAccess <http://example.com/user/bob>;
  sioc:has_container <http://example.org/taggings/group1>;
  sioc:note "Photo of Bob in downtown Madrid." .
<http://example.org/tag/tag1> a muto:Tag;
  muto:tagLabel "madrid" ;
  muto:nextTag <http://example.org/tag/tag2>.
<http://example.org/tag/tag2> a muto:Tag;
  muto:tagLabel "sol" ;
  muto:tagCreated "2011-11-12T09:43:03Z";
  muto:tagMeaning <http://dbpedia.org/resource/Puerta_del_Sol>;
  skos:broader <http://example.org/tag/tag1> .
<http://example.org/tag/tag3> a muto:AutoTag;
  muto:tagLabel "bob" ;
  muto:autoMeaning <http://example.com/user/bob> .
<http://example.org/taggings/group1> a sioc:Container;
  sioc:has_space <http://example.org> .

```

Listing 4.1 — OWL code of the example in RDF/Turtle format (‘a’ = shortcut for `rdf:type`).

Imagine the following scenario that led to the creation of the tagging instance:⁶ Alice logs into her account of the social networking service EXAMPLE.COM (`sioc:hasCreator`). From there, she uses the social bookmarking system EXAMPLE.ORG to annotate a photo she uploaded to the media sharing website EXAMPLE.NET (`muto:taggedResource`). As the photo shows her friend Bob in downtown Madrid, she starts tagging with entering the tag ‘madrid’ (`muto:Tag`). Then, she recognizes that the system has automatically identified Bob on the photo and added his name as a tag (`muto:AutoTag`). The system got his name (and further information) from the social networking service EXAMPLE.COM, of which Bob is also a member. In addition, the system links Bob’s name to his account (`muto:autoMeaning`, a subproperty of `muto:tagMeaning`). Though Alice marked the tagging as private

⁶ In brackets, we give the ontology classes and properties used to represent the information.

(`muto:PrivateTagging`), she decides to share it with Bob and grants him access (`muto:grantAccess`). She also adds a comment to the tagging describing the contents of the photo (`sioc:note`).

One day later, Alice looks at the photo again and recognizes that it was taken at Puerta del Sol, a central square in Madrid. She opens the tagging and adds the tag ‘sol’ (`muto:Tag`) to the previously assigned tags ‘madrid’ and ‘bob’. Furthermore, she decides to ‘semantify’ the tag so that she will later remember what ‘sol’ means. First, she makes ‘sol’ a subtag of ‘madrid’ (`skos:broader`) to indicate that it is a specific location in Madrid. Second, she gives the tag explicit meaning by linking it to the corresponding DBpedia resource (`muto:tagMeaning`).

The information that the tag ‘sol’ was added at a later time than the other two tags is given by its property `muto:tagCreated`. This property would not be necessary if the tag would have been entered along with the others. Accordingly, the timestamp of `muto:tagCreated` is the same as of `muto:taggingModified` but different from the one of `muto:taggingCreated`. Source information is linked with `sioc:has_space` after the tagging has been assigned to a container with `sioc:has_container`.

4.6 Discussion

The example illustrates an advanced case of tagging that showcases the expressive power of MUTO by using many of its concepts. The most basic variant of tagging – a list of tags without disambiguations, hierarchical relations, comments, or automatic tags – can be described with much fewer concepts from the ontology and results in a smaller RDF graph. This capability of supporting different forms of tagging, from simple to semantic, from manual to automatic, and from public to private, was one of the main goals in the development of the MUTO ontology. On the other hand, we avoided to make MUTO unnecessary complex but tried to keep it understandable to the people who use it. Finding a good balance between comprehensiveness and compactness was thus another major goal in the development of the MUTO ontology, as we already mentioned in Section 3.3.

The example also indicates the benefits of a precise domain description for the development of tagging systems. There would be many different ways to represent the information from the scenario; having one common conceptualization helps to create a joint understanding. In particular, it can

increase the interoperability between different tagging systems, as illustrated by the scenario: It links taggings of a social bookmarking system with photos of a media sharing website and user profiles of a social networking service. It would even be possible to represent the “tagging of tags” with MUTO, as it was proposed by [232].

Two major challenges in the application of ontologies are performance and scalability. In this work, we focus mainly on a precise conceptualization rather than on a technical optimization for large folksonomies. In such cases, other representations that allow for a fast processing and efficient storage of the folksonomy might be more useful. Furthermore, specific modules may be integrated into the MUTO ontology that speed up processing, such as property relations that directly link the resources, tags, and users in order to avoid the indirection via taggings. However, such pragmatic extensions should be used with care as they may lead to conceptual inconsistencies (e.g., direct relations between tags and resources can conflict with the concept of private tagging).

Apart from these conceptual constraints, we took care to design MUTO in a way that allows for efficient reasoning. By making it compliant to OWL 2 RL, polynomial time reasoning algorithms can be used that operate directly on the RDF triples that make up the folksonomy [106]. This is different to the majority of the reviewed ontologies that are implemented in OWL Full (see Table 4.2) and hence do not support efficient reasoning. The only two reviewed tagging ontologies that are also compliant to OWL 2 RL are TT and UTO (except from its aligned version, see Section 3.1.7.).

On the one hand, we tried to specify the conceptualization as precisely as possible in MUTO. We defined global cardinalities to ensure that the fundamental principles of tagging, as defined in the axioms of Section 2.4, are not violated. In addition, we used *typed* instead of *plain* literals [147] in most cases to clearly define the expected range of datatype properties. On the other hand, we avoided to overly specify the ontology but kept it flexible where appropriate. In addition, we did not want to unnecessarily increase the formal complexity of the ontology.

If we compare MUTO with the reviewed ontologies, it is most closely related to the Tag Ontology [197] and the *Bookmark Schema* of the Annotea project [151]. However, it additionally considers several concepts that are missing in these approaches, such as some advanced tagging concepts described in OF and MOAT. We did not include links to other tagging ontologies in the MUTO core ontology, as this would have inevitably resulted in concep-

Name	Abbr. ¹	OWL sublanguage	OWL 2 profile
Tag Ontology	TAGS	OWL Full	–
Tagging Ontology	TO	OWL Full	–
Ontology of Folksonomy	OF	OWL DL	–
Social Semantic Cloud of Tags	SCOT	OWL Full	–
Meaning of a Tag	MOAT	OWL Full	–
Upper Tag Ontology	UTO	OWL Lite	OWL 2 RL
Common Tag	CTAG	OWL Full	–
Tagora Tagging Ontology	TT	OWL Lite	OWL 2 RL
NiceTag Ontology	NT	OWL Full	–
Modular Unified Tagging Ontology	MUTO	OWL Lite	OWL 2 RL

¹ Abbreviations as used in this work.

Table 4.2 — OWL sublanguages of the tagging ontologies.

tual inconsistencies. Instead, we decided to use a separate mappings module that defines such alignments, as in was also done in **CTAG**. However, note how tagging-specific concepts from the **MUTO** ontology are combined with concepts from more general ontologies in the above example. For instance, Alice’s comment is added with `sioc:note`, the hierarchical relation between the two tags ‘sol’ and ‘madrid’ is described by `skos:broader`, and the used tagging system is linked with `sioc:has_space`.

4.7 Application

The **MUTO** ontology has been applied in different contexts and projects. In the following, we report on two such projects in which the author of this thesis was involved.

4.7.1 SWORE Ontology of the SoftWiki Project

One application context where MUTO has been successfully applied is the *Semantic Web Ontology for Requirements Engineering* (SWORE). SWORE provides the semantic basis for the methods and tools that were developed in the SoftWiki project [171, 172], a research project focusing on semantic collaboration in requirements engineering. The project aimed at enabling large groups of spatially distributed stakeholders to collect, semantically enrich, classify, and aggregate software requirements. Semantic technologies were used to support the collaboration as well as the interlinking and exchange of requirements information. Several methods and tools have been developed in the project that enable the elicitation, analysis, and exchange of software requirements [171].

SWORE describes several concepts of requirements engineering that are relevant in many software development projects. Figure 4.4 depicts the main concepts and relationships of version 2.0 of the SWORE ontology, as published on the web in early 2012 [209]. The core classes of SWORE are `req:Requirement`, `req:Source`, and `req:Stakeholder`. Furthermore, it provides a set of relation types that describe dependencies between requirements, such as `details`, `entails`, or `invalidates`. It also supports the definition of pointers to parts of a system or prototype that requirements may refer to [208, 175].

SWORE integrates several other vocabularies. For instance, stakeholders are represented by the FOAF class `foaf:Agent` and its subclasses `foaf:Person`, `foaf:Group`, and `foaf:Organization`. Further FOAF concepts can be used to express additional information on stakeholders, such as contact details or projects they are involved in. Online discussions between the stakeholders are represented with SIOC. A stakeholder's account is described by the SIOC class `sioc:UserAccount`, comments are a subclass of `sioc:Post` and ratings are represented by the class `sioc:Poll`. SKOS is used to represent the topic hierarchy that assists in the classification of requirements. Each topic is a `skos:Concept` and the hierarchical relations are described with `skos:broader` and `skos:narrower`.

Since its version 2.0, SWORE also integrates the MUTO ontology [209]. In particular, it uses the classes `muto:Tagging` and `muto:Tag` along with some of their properties to represent the tagging of requirements. This possibility of tagging requirements is a rather unique feature of the SoftWiki

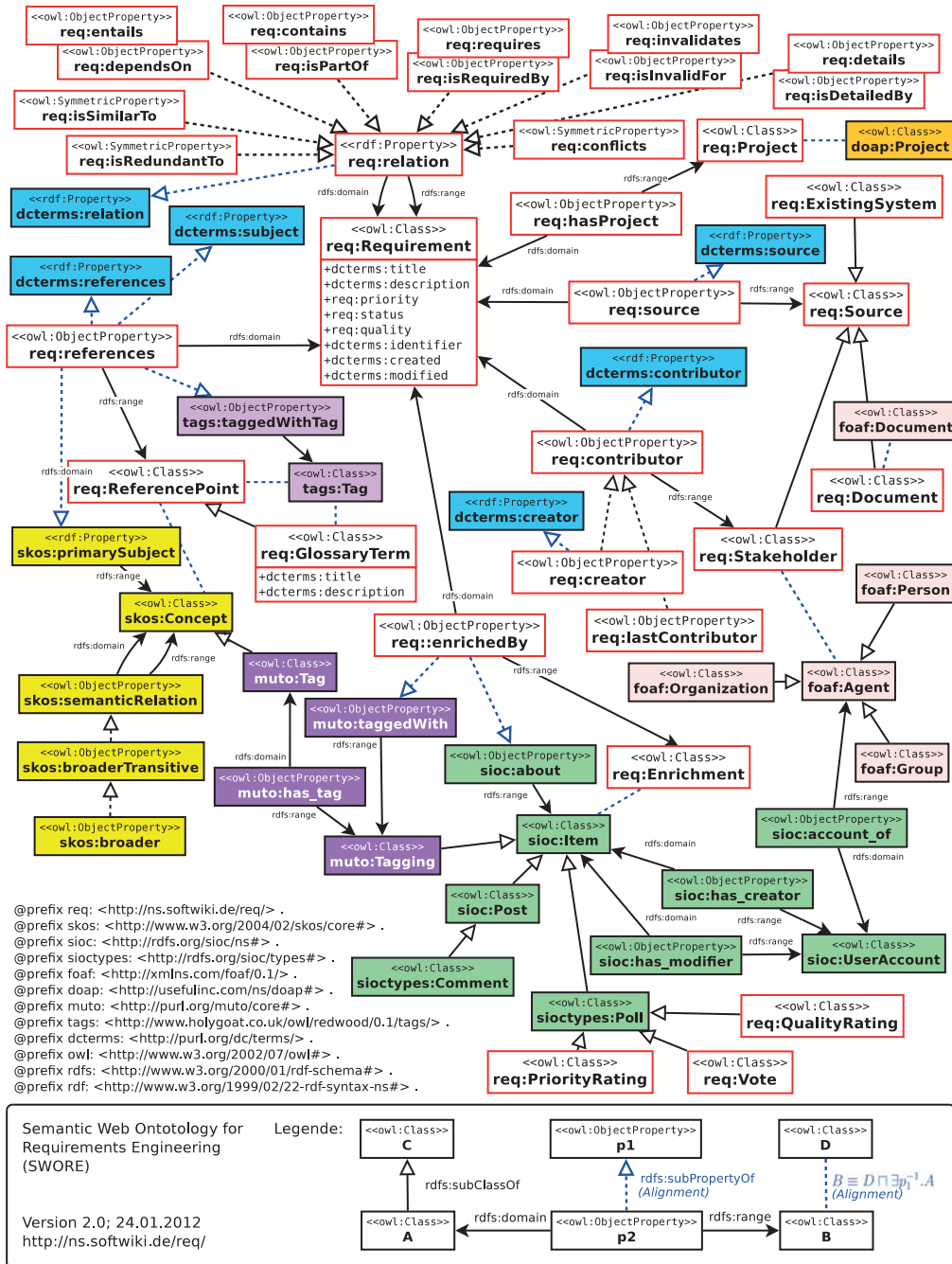


Figure 4.4 — Version 2.0 of the Semantic Web Ontology for Requirements Engineering (SWORE) [209].

approach [172]. It allows for a collective categorization of requirements based on tags in addition to the typical taxonomy classification.

The tagging feature has been implemented in the central web platform developed within the project [170]. The platform enables a large number of geographically distributed stakeholders to collaboratively collect, semantically enrich, and classify software requirements. The requirements can be edited, tagged, and discussed by the users of the web platform. Figure 4.5 shows a screenshot of the web platform implemented within the OntoWiki framework [126].

The web platform offers two ways of classifying requirements: They can either be assigned to a category of a pre-defined taxonomy or associated with freely chosen tags. These tags are aggregated and visualized in the form of a tag cloud in the user interface of the web platform (see Figure 4.5). As `mut0:Tag` is a subclass of `skos:Concept`, tags can easily be transformed into concepts. This ‘semantification’ of tags is explicitly supported in the web platform by offering a possibility to enrich tags with definitions. Such ‘defined tags’ are indicated by a different color in the tag cloud. They may be added to the taxonomy at some point in the requirements engineering process. Note that tags with same spelling are assumed to have the same meaning in this case. This is important in requirements engineering to reduce the ambiguity of the used terms as much as possible. It affects particularly

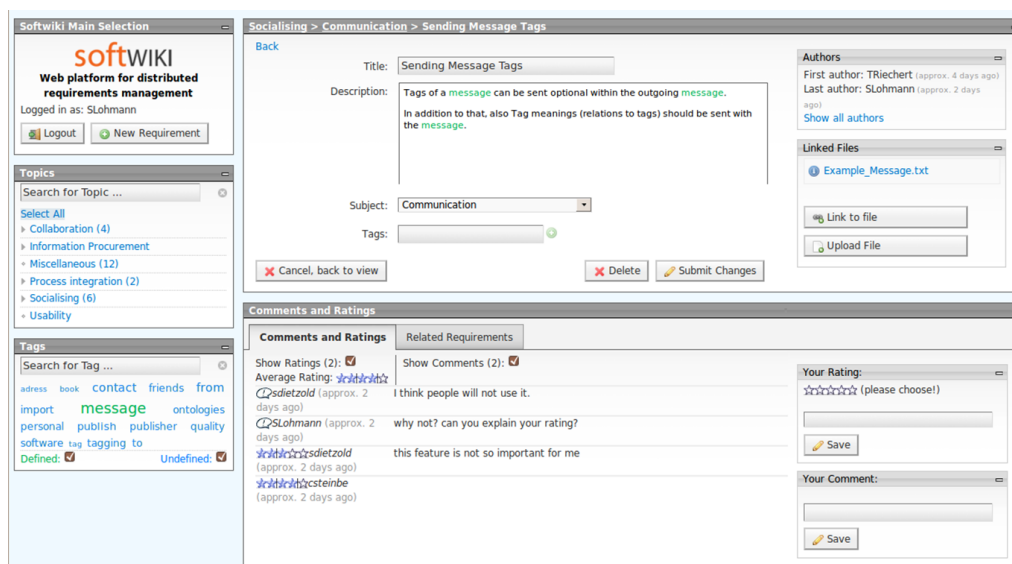


Figure 4.5 — Web platform for collaborative requirements engineering.

the ‘defined tags’, which are represented with `skos:Concept` and not with `muto:Tag`.

The web platform provides access to the requirements and related data with the RDF query language SPARQL [105]. In addition, instance data of SWORE and all integrated ontologies can be exported and imported in RDF format [175]. This flexible access opens up a large range of possibilities. Among others, it supports the utilization of the requirements data in other tools while semantic interoperability is maintained.

4.7.2 Visual Notation for OWL (VOWL)

A second application of the MUTO ontology was in a research project concerned with the visual representation of ontologies. The project was motivated by the observation that there are nearly no well-specified visual representations for OWL ontologies besides the UML profile of the ODA specification [201] introduced in Section 4. Many authors use their own visualization, as can be seen, for instance, by the various illustrations we found in the review of available tagging ontologies (see Chapter 3).

These custom visualizations often focus on specific aspects of the ontologies and are partly hard to read for casual users. In addition, most of these visualizations do not give an explicit description of the visual notation, i.e., a specification that clearly defines the semantics of the graphical elements. In many cases, they rather provide only a basic overview of the ontology that is not sufficient to gain a deeper understanding of the concepts and relationships in a domain of knowledge.

Motivated by the work on MUTO, we aimed for a comprehensive and comprehensible visualization that is also understandable to users less familiar with ontologies. It should be printable but also provide intuitive ways to interactively explore the visualized ontology. Having these goals in mind, we developed the *Visual Notation for OWL Ontologies (VOWL)* [195] that provides a well-specified visual language for the user-oriented representation of ontologies. It defines graphical depictions for many OWL elements that are combined to a graph visualization representing the ontology.

The experiences gained with MUTO were very helpful in the development of VOWL, and MUTO served as one of the key ontologies on which the VOWL visualization was tested. In particular, MUTO was used in a qualitative user study that investigated the strengths and weaknesses of VOWL in comparison to UML. The study examined version 1.0 of VOWL that has

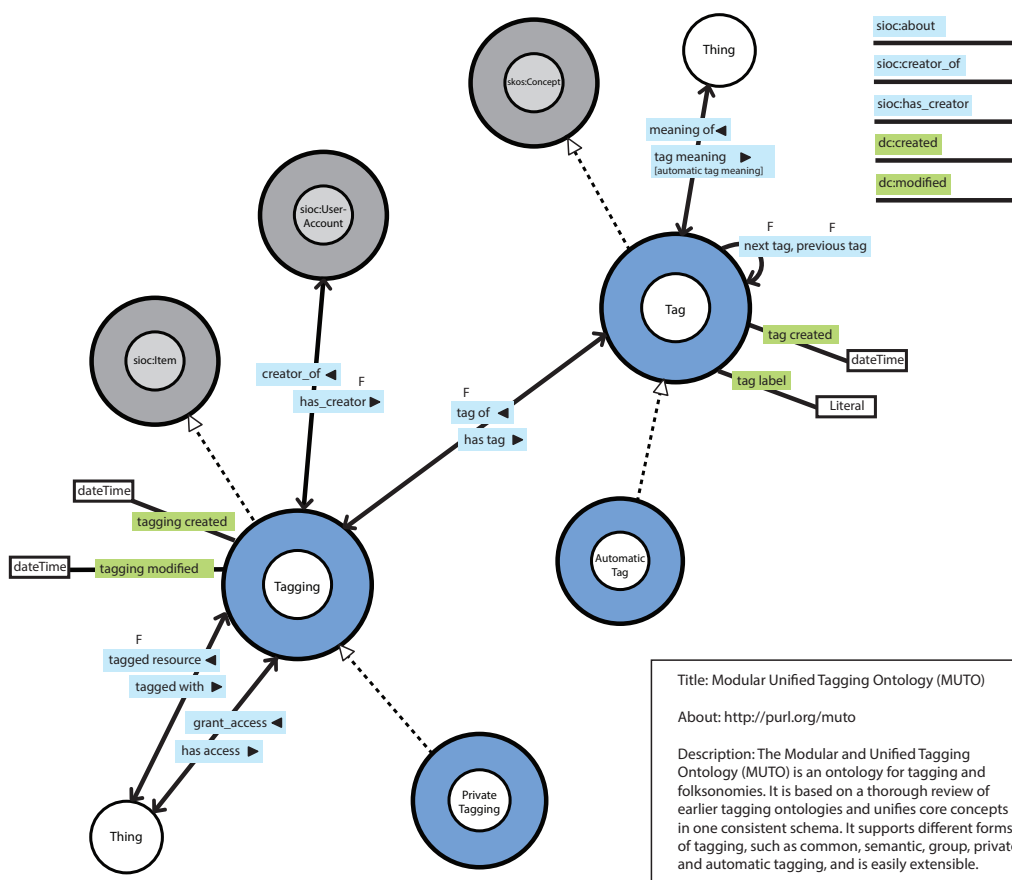


Figure 4.6 — VOWL notation applied to the MUTO ontology [194].

been released in early 2013 by Negru and the author of this thesis.⁷ Figure 4.6 depicts the conceptual layer of the VOWL notation applied to the MUTO ontology [195]. It was one of the visualizations shown to the participants of the user study.

Both visual notations provide a comparatively complete and comprehensive set of graphical elements for the representation of OWL ontologies. Furthermore, they are well-specified and also understandable to users less familiar with ontologies. On the other hand, they follow quite different approaches: While one reuses the well-established UML notation, the other has been specifically designed for OWL. Still, the notations are comparable, as they both employ some kind of node-link diagram.

⁷The VOWL specification is publicly available on the web at the persistent URL <http://purl.org/vowl>.

The study focused on qualitative aspects of the visualizations with the goal to identify strong and weak points of each notation. The overall goal was to get insights that help in the development of a unified visual notation for OWL ontologies. *MUTO* was selected as example ontology for the study because of “its manageable size and because it is relatively easy to understand for users who have not come in contact with ontologies before” [194].

The compactness of the visual representation was considered as one of the main strengths of the *VOWL* notation by the study participants. Most participants preferred that notation over the *UML* one. They found it “less cluttered” and liked that “one immediately sees the concepts of the ontology” [194]. On the other hand, many participants found it troublesome to read the directional indicators of inverse properties (e.g., of `creator_of` and `has_creator`). Others had problems understanding the dashed arrows (the graphical representation of the `subclassOf` property) and the meaning of datatype properties, which, unlike object properties, had no arrow at either end of the line (e.g., `taggingcreated`). More information on the user study and its results can be found in [194].

The comparative study revealed several key challenges regarding the visualization of OWL ontologies. Examples include the treatment of special classes like `owl:Thing` and `owl:Nothing`, the adequate representation of domain and range axioms, and the indication of subclass and subproperty relations. Apart from that, it illustrated the compactness of the *MUTO* core ontology. It also demonstrated that *MUTO* is easily understandable, as the study participants reported no serious problems in understanding the ontology and solving the tasks.

However, in the remainder of this thesis, we will focus on the visualization of folksonomies rather than ontologies. We are interested in different possibilities of visualizing folksonomies and how they can be effectively used in interactive systems. The *MUTO* ontology will serve as the conceptual basis for these considerations.

Tag Cloud Visualizations

Various visualizations of folksonomies have been developed since tagging emerged on the web in late 2003. For instance, there are visualizations of sets of tags using maps (e.g., *TagMaps* [24, 134], *Topigraphy* [96]) or timelines (e.g., *TagLines* [26, 83], *Aging Tag Clouds* [186]). Other approaches depict the links that emerge in tagging, such as the folksonomy graph visualizations we will discuss in Chapter 6. However, certainly the most popular type of visualization for folksonomies is the *tag cloud* we already introduced in Chapter 2 (see Definition 9).

Viégas and Wattenberg call tag clouds “vernacular visualizations” [242], as they are simple yet exciting visualizations that originated outside the research community. They became popular in the context of community-oriented websites that use tagging as indexing method, such as FLICKR [8], DELICIOUS [6], or TECHNORATI [31]. Meanwhile, they have evolved as a core technique of information visualization and are applied in many different contexts [242, 226].

In this chapter, we first take a closer look at tag cloud performance and perception by investigating different tag cloud layouts and their ability to support users in typical information seeking tasks. Based on these results, we examine how tag clouds can be extended to better support the visual analysis of folksonomies.

Parts of this chapter have already been published in [60], [125], [167], and [176].

5.1 Research on Tag Clouds

Most research on tag clouds falls in one of two categories: It either investigates the effectiveness and visual perception of tag clouds or develops improvements and extensions to tags clouds. Both research directions are addressed in this chapter. In the following, we start with an overview of basic tag clouds types and summarize work that studies the performance and perception of tag clouds

5.1.1 Types of Tag Clouds

Several extensions to the basic tag cloud visualization have been presented in the last couple of years. Many of them address layout issues of tag clouds, such as large white spaces or the restriction to specific boundaries. Kaser and Lemire [138], for instance, use slicing trees, nested tables, and rectangle packing to optimize the distribution of space in HTML-based tag clouds. Seifert et al. [219] present a related algorithm for white space optimization that can cope with differently shaped tag clouds. It places tags in a circular fashion with the most frequent ones in the center and those with lower frequency towards the boundaries. Advanced layouts can also be generated with freely available tag cloud generators, such as Wordle [34], Tagxedo [30], or Tagul [29]. They offer several options to customize the tag cloud visualization by adapting typography, color, word orientation, or the general shape of the tag cloud.

Other works use clustering techniques along with different kinds of tag cloud layouts, ranging from line-by-line layouts [116, 138, 218] to force-directed layouts [62] and topographical tag landscapes [96]. An example of a clustered tag cloud created by Clark [69] is shown in Figure 5.1b. It displays frequent tags of the famous speech “I Have a Dream” by M. L. King Jr.

However, the most popular tag cloud layout for folksonomy visualizations on the web is presumably still a rectangular tag arrangement with alphabetical sorting in a sequential line-by-line layout. Figure 5.1a shows an example of this well-known type of tag cloud from the website of AMAZON.COM. We already introduced this layout with the FLICKR tag cloud in Chapter 2 (see Figure 2.2). One reason for the popularity of this layout might be its ease of implementation. However, a sequential arrangement of tags does not provide optimal support for all purposes a user consults a tag cloud for.

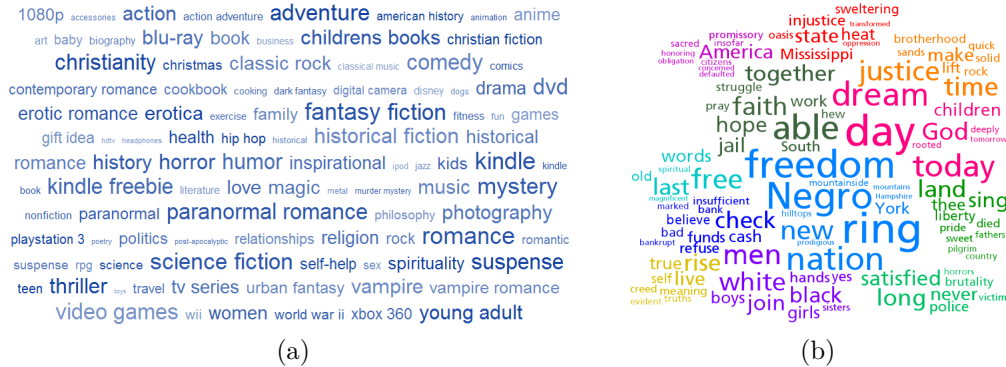


Figure 5.1 — Examples of tag cloud layouts: (a) sequential layout with alphabetical sorting, showing the “most popular tags” on AMAZON.COM [37], (b) clustered tag cloud, displaying frequent tags of the famous speech “I Have a Dream” by M. L. King, Jr. [69].

Although little is known about the benefits of tag clouds and the situations in which they are consulted, some typical user goals have been identified – from getting an overview or general impression of a website’s contents to searching for specific topics or tags [112, 210, 225]. It has also been conjectured that tag clouds serve a social purpose by conveying a sense of activity in a web community [118]. The ability of different tag cloud layouts to support certain user goals and information needs, however, is still a largely open research issue.

5.1.2 Performance and Perception of Tag Clouds

There are several works that study the effectiveness and perception of tag clouds. Most of them investigate tag clouds at a general level by comparing them with other types of user interfaces. For instance, Halvey and Keane compare tag clouds with unweighted, horizontal and vertical lists by asking participants to find and select specific tags in both interface types [112]. The results indicate that unweighted lists perform better than tag clouds and that alphabetical sorting further accelerates search. In addition, the tags’ font sizes had a strong effect on search speed and tags in the upper left corner of the cloud were found most quickly. Based on the times the study participants needed to find tags in specific target regions of the cloud, the authors also concluded that tag clouds are rather scanned than read. Kuo et al. got similar performance results by comparing tag clouds and

lists as alternative presentation forms for search result summarization in the biomedical domain [154]. Though the study participants solved the descriptive tasks more quickly with the lists, they attributed the tag clouds a higher level of satisfaction.

Sinclair and Cardew-Hall [225] conducted an experiment in which the subjects could either use a tag cloud or a search box to complete given tasks. While participants preferred the search box to enter specific tags, they favored the tag cloud for more open-ended tasks. They concluded that tag clouds are not a replacement of but a valuable extension to other types of user interfaces.

Other work takes a closer look at the visual features of tag clouds. Bateman et al. [44] systematically varied eight tag cloud properties and measured their effects on the users' attention by counting how often certain tags are selected. They found that the properties with the largest effect on the users' attention are font size and weight, while the intensity of a tag's color or its number of characters play minor roles. Furthermore, they found that tags in the middle of the cloud were selected more often on average than tags near the borders.

Rivadeneira et al. conducted two experiments [210]: The first examined the effects of font size, location, and proximity-to-the-largest-tag. The user task was to recall tags that were previously presented in tag clouds. In accordance with Halvey and Keane [112] and Bateman et al. [44], a strong effect of font size was observed, while proximity-to-the-largest-tag had no significant impact. Furthermore, tags in the upper left quadrant were recalled more often, but the authors attributed this to the sparseness of the tag clouds used in the study. In the second experiment, Rivadeneira et al. investigated the effects of font size and tag cloud layout on impression formation and recognition. Font size had again a strong effect on the results. The tag cloud layouts had no impact on recognition but affected the accuracy of impression formation.

5.2 Comparison of Tag Cloud Layouts

While the presented studies provide valuable findings on tag cloud performance and are first steps towards a better understanding of tag cloud perception, they mainly focus on sequential tag cloud layouts without considering further ways to arrange tags. The only study that evaluates different

tag cloud layouts [210] is very limited in its conclusions as these layouts were highly diverse, leading to many dependencies in the visual features. Moreover, layout performance was only examined regarding recognition and impression formation in that study but not with respect to more typical situations of tag cloud interaction.

Additionally, the experimental material is comparatively artificial in many of the reported studies. For instance, the tag clouds consist of a very limited number of tags in some cases (e.g., only 10 tags [112] or 13 tags [210] in total) or few font size variations (e.g., only 3 different font sizes [112]). Most importantly, none of the studies addressed the visual exploration of tag clouds directly. All conclusions regarding tag cloud perception have been indirectly inferred from the tags that were either selected or recalled by the study participants.

5.2.1 Experimental Design

In contrast to related work, we pursued the following objectives with our user study:

- Defining user tasks that simulate situations of nearly realistic tag cloud interaction.
- Presenting tag clouds with a common number of tags and font size variations.
- Generating tag cloud layouts that differ only in tag arrangement but no other visual properties.
- Using eye tracking to measure the actual attention areas and perception patterns of tag clouds.

User Tasks

From the range of purposes tag clouds might be used for, we selected three search tasks that are very common according to the literature [44, 112, 154, 210, 225]:

1. Finding a specific tag
2. Finding the most popular tags
3. Finding tags that belong to a certain topic

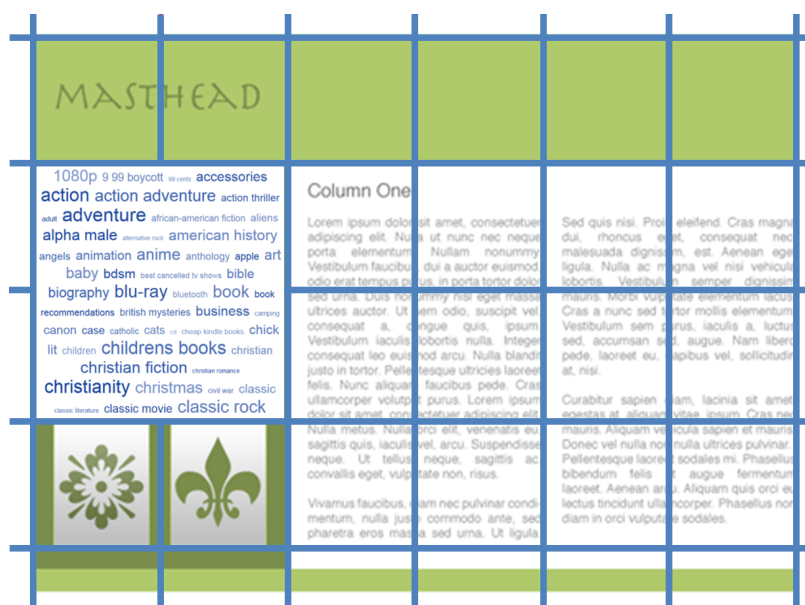


Figure 5.2 — Most user interfaces follow a grid layout that conceptually structures the interface by intersecting horizontal and vertical lines.

By providing a scenario along with the tasks, a somewhat realistic interaction situation was simulated. For example, one text for the first task was: “You always wanted to visit Nizza. At the moment, you do not have enough money for traveling and can only dream about it. . . Please click on the tag ‘Nizza’ in the following tag cloud.” Accordingly, the participants were asked to search for a target tag and click on it in this task. In the second and third task, the participants had to search and select three tags of choice that were either among the most popular ones (task 2) or related to a given topic (task 3). All study participants reported that they had understood the tasks; none needed further explanation or complained in the post-test questionnaire.

Tag Cloud Layouts

In order to reduce the countless variations of possible tag arrangements to a manageable and reasonably representative number, we made the following assumption: Most graphical user interfaces are based on a two-dimensional grid layout that conceptually divides the interface into a number of rectangles by horizontal and vertical lines, as illustrated in Figure 5.2. Consequently, a typical requirement for a balanced integration of tag clouds into user

interfaces is that they (1) are displayed in a rectangular area and (2) fill this area with tags as completely as possible. Based on these requirements, three general ordering principles can be distinguished:

1. Sequential layout, with either a horizontal or vertical arrangement of tags, sorted alphabetically or by some other criteria (e.g., popularity, chronology, etc.)
2. Circular layout, with the most popular tags in the center and tags with decreasing popularities towards the borders (or vice versa)
3. Clustered layout, in which the distance between tags follows a certain clustering criteria (e.g., semantic relatedness) and related tags are positioned in close proximity (cp. [69, 116])

We generated one prototypical tag cloud for each of these three layout strategies. In addition, we created a fourth tag cloud with no variation in the tags' font sizes as reference layout (see Figure 5.3).

Tag Corpora

Since we aimed to compare the different tag cloud layouts independently of interpersonal differences, we decided to present all four layouts in a series to every study participant. We used tag corpora from four different domains to prevent learning effects that would likely have resulted if we had presented the same tags in each layout.

Tag corpora from 'real' contexts were not well suited for our controlled experiment as they usually contain tags with heterogeneous connotations and little semantic consistency. We aimed at minimizing any bias caused by personally affecting or political tags (such as 'terrorist'). Furthermore, we wanted to present tags that are well-known to the study participants to avoid effects that result from different interests and educational backgrounds. Additional requirements for the tag corpora included a reasonable way of categorizing the tags for the clustered tag cloud and the usage of tags that are common in Germany, since the study participants were all German-speaking. For these reasons (and due to a lack of alternatives) we decided to develop our own tag corpora for the study that consisted of neutral tags from common knowledge areas (France, sports, furniture, animals) and could be used to create nearly realistic interaction scenarios. Each corpus consisted of 100 tags with varying popularity values.



Figure 5.3 — Tag cloud layouts for the corpus ‘France’: (a) sequential with alphabetical sorting, (b) circular with decreasing popularity, (c) clustered with thematic clusters, (d) reference with sequential, alphabetical sorting and no weighting of tags.

Generation of the Tag Clouds

Since our goal was to evaluate general layout types and not specific algorithms, we decided to generate the tag cloud layouts by our own instead of using available algorithms. That way, we could apply the same design guidelines for all tag clouds and strictly control tag distribution. Nevertheless, the tag clouds were designed to largely follow typical algorithms and were not perceived as artificial or unusual by the study participants.

We generated the tag clouds for all four layouts as follows: We used an equally sized rectangle with an aspect ratio of 3:2 and filled it with the 100 tags of one of the four corpora. We mapped the tags’ popularity values on six discrete font sizes, resulting in one tag of 30 pt up to 27 tags of 15 pt for each tag cloud. Every quadrant of the tag cloud got the same number of

tags of each font size to avoid biases caused by an unbalanced presentation. In accordance with the ordering principle of the circular layout, the 30 pt tag was placed in the middle of the cloud. In the sequential and clustered layout, the 30 pt tag was placed in another quadrant for each of the four corpora.

We also considered the different user tasks in the tags' distribution: We varied the size and quadrant position of the tags the participants were asked for in the first task. Likewise, we distributed the thematic clusters that were of interest in the third task among all four quadrants of the tag clouds. We kept all other visual features, such as font styles, weights, colors, or intensities, constant in order to avoid interdependencies as reported in [44].

Figure 5.3 shows the resulting four layouts that were generated for the corpus 'France'. The colored lines, circles, and arrows indicate quadrant separations and ordering principles of the layouts for the readers of this thesis; they were, of course, not visible to the study participants.

Procedure

36 participants, mainly students, with an average age of 26 (min 17, max 57) took part in the experiment. All had normal or corrected-to-normal vision. The general familiarity with tag clouds was given with a median of 4 on a scale of 1 to 10. Five participants could not remember having seen any tag clouds before. In order to create a nearly identical understanding of tagging and tag clouds, the experiment started with a short introduction into these topics, consisting of an oral explanation accompanied by a paper demonstration (see Figure 5.4a). In addition, we presented three tag clouds from popular websites (FLICKR, DELICIOUS, and LAST.FM [11]) that all followed a sequential layout with alphabetical sorting. We expected that many participants had already come across this well-known layout – effects caused by a higher familiarity cannot be avoided but somewhat compensated by presenting this type of tag cloud to all participants in advance.

After the introduction, the participants were randomly assigned to one of the three tasks, resulting in 12 participants per task. We applied a Graeco-Latin square design to counterbalance the layout-corpora-combinations in the series of trials for each experimental group. Table 5.1 shows the resulting four series for the first task. The second and third experimental groups (task 2 and 3) were assigned to corresponding layout-corpora-combinations also following a Graeco-Latin square design.

SG	Tr1	Tr2	Tr3	Tr4
1	L1 / C1	L2 / C3	L3 / C4	L4 / C2
2	L2 / C2	L1 / C4	L4 / C3	L3 / C1
3	L3 / C3	L4 / C1	L1 / C2	L2 / C4
4	L4 / C4	L3 / C2	L2 / C1	L1 / C3

Table 5.1 — Graeco-Latin square design for the first task (presented to one group of participants): Combinations of tag cloud layouts (L) and tag corpora (C) for the four subgroups (SG) and four trials (Tr).



Figure 5.4 — a) Paper demonstrator to explain tagging and tag clouds, b) Presentation of the tag clouds on a 17" TFT monitor with embedded eye tracking system.

The tag clouds were presented on a 17" TFT monitor with a screen resolution of 1280 x 1024 px. They were placed in the middle of a blank screen in an area of 20 x 13.3 cm (see Figure 5.4b). Before the presentation of each layout, a short text was displayed explaining the scenario and task. The text of the task was identical within each group; the scenario was adapted to the corresponding tag corpus. The participants were asked to carefully read the task before moving on. When one tag cloud layout was completed by clicking on the tag(s), the group's task was again displayed along with the next scenario and layout. After all four layouts had been presented, the participants filled out a questionnaire in which they had to state their layout preference among others. All four layouts were reprinted in the questionnaire

to aid recognition. The click times were recorded by the presentation software. Gaze data was captured by an eye tracking system that was embedded in the TFT monitor.

5.2.2 Task-Layout-Performance

We measured the task-layout-performance as the time the participants needed to accomplish the tasks. For the first task, we measured the time until the target tag was selected. For the second and third tasks, we calculated the mean value of the time needed for the three selections as indicator for task-layout-performance.

We ran a Kruskal-Wallis test on the data to get an objective ranking on how the layouts perform for each of the three tasks. In addition, we calculated a subjective ranking by counting the study participants' votes for the layouts in the post-test questionnaire. Table 5.3 shows these values for all three tasks along with the mean values and standard deviations (in seconds) of the times needed for task completion. In the following, we discuss the results for each task in more detail.

Results for Task 1

For the first task (finding a specific tag), the reference layout (the layout without any weighting of tags) performed best with a value of 17.8 when comparing the mean ranks, followed by the sequential tag cloud layout (mean rank: 22.5). Since both layouts were alphabetically sorted and hence highly supported scanning for a particular tag, we expected these layouts to score well in this task, though the differences were not significant ($p > .05$). Furthermore, we found some evidence that tags with large font sizes are selected more quickly than small tags in this task. However, we cannot provide reliable values as this effect seemed to be influenced by other visual features, such as a tag's number of characters or its position and neighboring tags in the cloud.

Interestingly, only 3 of this group's 12 participants voted for the reference layout in the questionnaire (two of them have never come in contact with tag clouds before). Although the font sizes did not provide additional value in this task, most participants (8 out of 12) preferred the sequential tag cloud layout. It seems that their choice was not only driven by rational factors but largely influenced by aesthetic aspects, since many subjects mentioned

Task	Layout	N	KW Mean Rank	User Votes	Mean	SD
1	sequential	12	22.5	8	8.6	5.5
	circular		27.8	1	13.6	11.1
	clustered		30.0	0	14.3	9.6
	reference		17.8	3	6.9	4.7
2	sequential	12	20.8	1	2.6	2.3
	circular		18.7	8	2.2	1.9
	clustered		24.4	3	3.1	2.9
	reference		34.1	0	4.5	3.5
3	sequential	12	26.5	4	7.3	6.9
	circular		22.0	3	5.4	4.4
	clustered		19.4	5	4.9	4.1
	reference		30.2	0	6.9	4.0

Table 5.3 — Performance values of the tag cloud layouts for the 12 participants (N) of each experimental group: Kruskal-Wallis (KW) mean rank, user votes, mean value, and standard deviation (in sec).

the more appealing appearance of the tag clouds compared to the reference layout in the questionnaire.

Results for Task 2

In the second task (finding the most popular tags), the circular layout showed the significantly best performance results (mean rank of 18.7, $p < .05$). It was also selected as the preferred layout by most participants (8 out of 12). Unsurprisingly, the reference layout failed both in click times and user ratings as it does not contain any information about tag popularity.

Interestingly, the two participants that needed the longest time to accomplish the task preferred the clustered layout – it seems that the circular layout supports the identification of objectively popular tags (i.e., tags with large font sizes) whereas the clustered layout encourages subjective decision making

on most important tags (i.e., considering not only a tag's font size but also its position in the clustered layout). However, this interpretation needs further validation.

Results for Task 3

The results of the third task (finding tags that belong to a given topic) also met our expectations as the clustered tag cloud layout performed best (though not significantly, $p > .05$). However, not all participants seemed to recognize its ordering principle; some even selected tags from topic areas we did not ask for in the task. Interestingly, most participants that chose tags in accordance with the task voted for the clustered tag cloud in the questionnaire.

Font size had again a strong effect on tag selection, as the selected tags had a mean size of 23.6 pt (median: 25 pt) compared to a mean size of 19.6 pt (median: 18 pt) for all tags, even though the study participants could freely choose any tags that belong to the given topic area, independently of their font size.

5.2.3 Analysis of Eye Tracking Data

We analyzed the eye tracking data in an exploratory manner by looking for typical patterns in the visual search behavior. We focused on the spatial distribution of fixations over the three tag cloud layouts (the reference layout was not of interest here). All fixations of the subjects' eyes with a minimum duration of 100 ms were considered, using a fixation radius (i.e., the smallest distance that separates fixations) of 50 px. We analyzed the fixation data of the first six seconds of each trial, which represent well the main phase of the search process.

In a first analysis, we divided the whole tag cloud areas into 5 x 5 rectangular, equally sized subareas and counted fixations in these areas. Figure 5.5 shows the distribution of fixations over the 25 subareas. In the circular layout, fixations are quite strongly focused on the central part of the tag cloud. They are more dispersed and oriented towards the upper left quadrant in the sequential layout. In the clustered layout, they are more evenly distributed. The lower right attracts less attention in all three layouts.

The uneven distribution of fixations among the quadrants becomes even more apparent when we aggregate the 25 subareas into partly overlapping

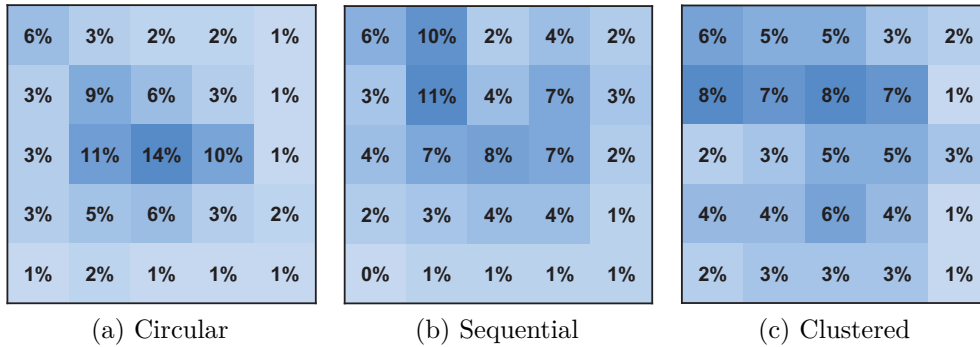


Figure 5.5 — Distribution of fixations in percent over the 5 x 5 subareas for the three tag cloud layouts. The coloring illustrates the pattern of the distribution.

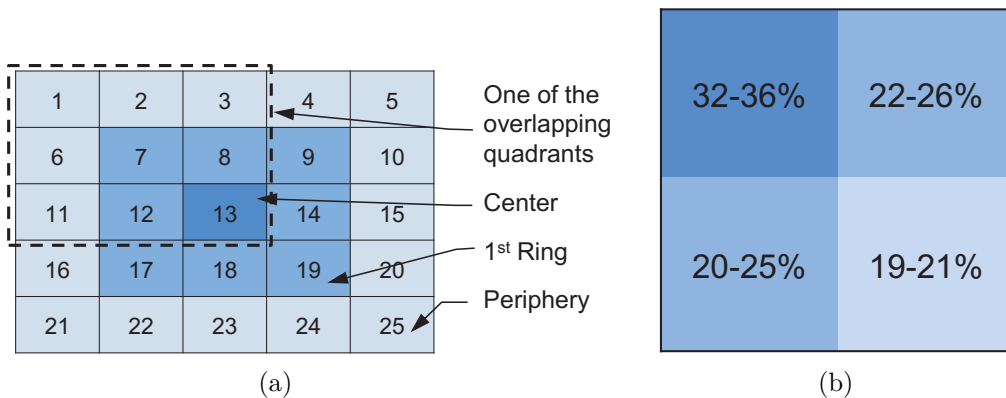


Figure 5.6 — a) Definition of areas of interest, b) Distribution of fixations over the quadrants.

quadrants (see Figure 5.6a). Compared to a strict quadrant separation, fixations near the quadrant borders are counted for more than one quadrant in this variant what further strengthens the results in our case. Interestingly, the distribution of fixations over the quadrants varies only marginally among the layouts (see Figure 5.6b). In all layouts, the upper left quadrant got the most fixations (32-36%) and the bottom-right the fewest (19-21%). This might be explained by Western reading habits.

To further analyze the tendency towards the center we aggregated the 25 subareas to three larger, concentric zones as illustrated in Figure 5.6a: 1) a center zone consisting of the single rectangle in the middle, 2) a first ring

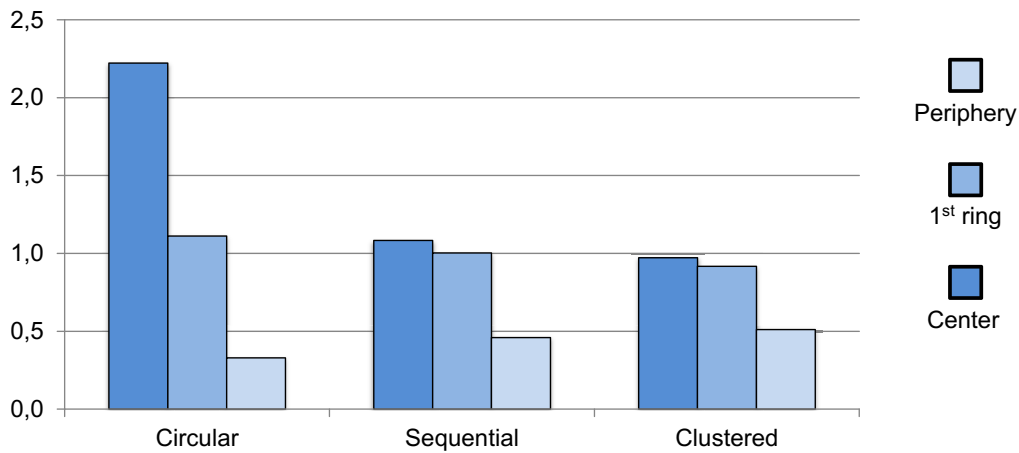


Figure 5.7 — Fixation density for the central-to-peripheral zones.

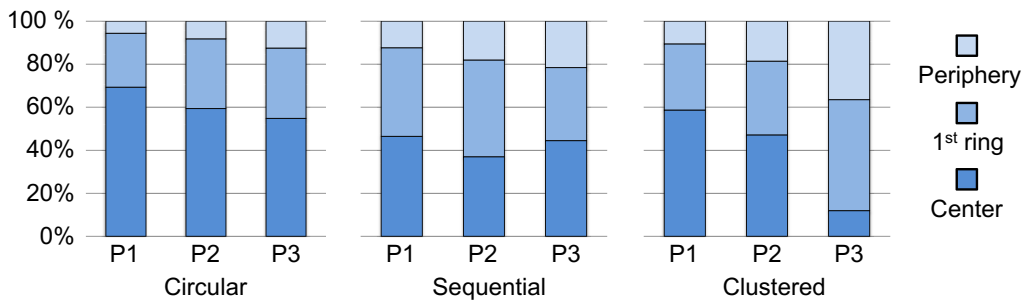


Figure 5.8 — Distribution of fixation density in the center, first ring, and periphery zones for the three time periods.

of rectangles around the center, and 3) a second ring of rectangles (we call ‘periphery’). In this case, we calculated the number of fixations per unit of area (fixation density). The data for the center, first ring, and peripheral zones show that the highest proportion of fixations lies in the central parts of the cloud (center + first ring), regardless of the layout (see Figure 5.7). For the circular layout, the small center rectangle has even more than double the fixation density than the surrounding first ring, indicating a very strong central focus.

In order to gain insight into the temporal variation of the gaze focus, we divided the overall time interval analyzed (seconds 0-6) into three subintervals: seconds 0-1 (period 1), seconds 1-3 (period 2), and seconds 3-6 (period 3). We chose three periods of increasing length since we were interested in contrasting a short initial phase indicative of a subject’s orientation at the

beginning of the trial with the gaze focus in later intervals where the visual search becomes more expansive and covers larger areas. Figure 5.8 shows the gaze distribution over the center-to-periphery zones for the three time periods. As already observed in the overall gaze distribution, the circular layout generates the highest central orientation, which is slightly decreasing over time for the benefit of the first ring and peripheral zones. The change in distribution is less pronounced for the sequential layout. In the clustered layout, the decrease in central orientation is strongest. A possible explanation is that the gaze tends to stay in the target cluster once the thematic relation with it has been recognized.

5.2.4 Discussion

The results of our comparative study on tag cloud layouts show clearly that there exists no single best way to arrange weighted tags in a cloud – as common in interaction design, the optimal solution depends strongly on the specific user goals and intentions of the designer. However, regarding the layout classes and user goals investigated in our experiment, the following task-layout-combinations are supported by our results:

1. Finding a specific tag: Sequential layout with alphabetical sorting.
2. Finding the most popular tags: Circular layout with decreasing popularity.
3. Finding tags that belong to a certain topic: Thematically clustered layout.

Generalizing from the layouts and tasks, we got some results that are largely in line with the findings and assumptions made in related work. Table 5.5 summarizes the findings by contrasting the general results of our study with the results of related work (cp. Section 5.2). As already indicated in previous studies [44, 112, 210], we also found that tags with large font sizes seem to ‘pop out’ to the viewers – they were on average identified more quickly in the first task and more often selected in the third. Although tag clouds perform worse than unweighted lists on average [112, 154], they seem to accelerate the identification of very frequent tags.

When aggregating the eye tracking data of all layouts and tasks, we observed a tendency towards the center: Tags in the middle of the cloud attracted more user attention than tags near the borders. Also, an effect of position could be observed in the eye tracking data: The upper-left quadrant received

Aspect	Description & literature source	Evaluation based on our results
Tag size	Large tags attract more user attention than small tags [44, 210] and are found more quickly [112].	Basically, our results support these findings, though they also indicate that further properties, such as the number of characters, the position of a tag in the cloud, or neighboring tags, influence this effect (cp. [44]).
Scanning	Users scan rather than read tag clouds [44, 112].	The eye tracking data clearly support this finding. However, in contrast to the assumption of [112], no general scanning direction could be identified.
Centering	Tags in the middle of the cloud attract more user attention than tags near the borders [44].	This seems to be true for most kinds of tag clouds though the layout can increase (e.g., circular) or decrease (e.g., clustered) this effect.
Position	Tags in the upper left quadrant are better recalled [210] and are found more quickly [44].	The eye tracking data show that the upper left quadrant receives the most fixations in all layouts.
Layout	The layout of a tag cloud influences its perception [210].	This finding is strongly supported by our eye tracking data.
Exploration	Tag clouds provide suboptimal support when searching for specific tags [112, 154, 225].	In general, our results support this finding. However, very popular tags can be found comparatively quickly.

Table 5.5 — Findings on tag cloud perception and performance, contrasting related work and our own results (including both quantitative and qualitative observations).

more fixations than the others. Thus, a tag’s general visibility is largely influenced by its position in the tag cloud, which might be used by designers to direct users’ attention to certain tags.

The feedback and ratings of the study participants suggest that fun and aesthetic aspects largely affect the user’s perception of tag clouds. Participants partly preferred layouts that did not yield the best performance results. Thus, such aspects must be included in any design decision and usability evaluation of tag clouds. Furthermore, our findings clearly show that interface designers must consider the possible user tasks before deciding for a certain type of tag cloud or for using tag clouds at all.

5.3 Prefix Tag Clouds

As we already mentioned in Chapter 2, a major limitation of folksonomies is that they 2.2.2 treat different forms of the same word as individual tags. Typical examples are inflections (e.g., singular and plural), nominalizations (e.g., the gerund *-ing*), or spelling differences (e.g., British vs. American English). Such variations of words either also appear in the tag cloud or are not shown at all if not frequent enough. In the first case, they take up screen space that could better be used to display other information or additional tags. In the second case, the viewer has no indication whether there are other forms of a word in the text or not.

Especially the latter is problematic: Assume, for example, an information retrieval context where a user is interested in all resources related to the topic of “visualization”. Searching for this tag would likely not return all relevant documents, as the tag would not be associated with all resources. However, if other word forms like “visualisation” or “visually” are less frequently used, they might not be shown in the tag cloud. Hence, there would be no hint that would tell the user what else to try as search term.

To overcome this limitation, we have developed an extension of the tag cloud visualization that we call *prefix tag cloud*. It creates a prefix tree that groups different word forms and visualizes the subtrees as tag cloud. Color is used to emphasize the grouping, while the relative frequencies of the word forms are indicated by font size, as common in tag clouds. The grouped word forms are arranged in a circular tag cloud layout that supports the quick identification of the most frequent words and word forms.

The process of creating prefix tag clouds consists of three main components: First, a prefix tree is generated from a set of tags. This requires us to order the tags lexicographically and to compute a prefix hierarchy from the resulting list. In a second step, the prefix subtrees are rendered as node-link diagrams. The font sizes of the prefixes in the diagrams are scaled according to the tag frequencies. Finally, the tag cloud is composed from the subtrees and placed in a given drawing area. We use a circular tag cloud layout that makes efficient use of the available screen space. In the following, we will detail these three components of the visualization process.

5.3.1 Prefix Tree Generation

Initial input is a non-empty set of tags $T = \{t_1, \dots, t_n\}$ with individual tags t_i , $1 \leq i \leq n$, as we already defined it in Section 2.4. The tags are composed of a finite sequence of characters from the alphabet $\Sigma := \{\sigma_1, \dots, \sigma_m\}$. Since $t_i \in \Sigma^+$, each tag t_i contains at least one character and is not an empty string.

In addition, each tag is associated with a quantitative value expressing its occurrence frequency. This mapping is given by the function $f_{freq} : T \rightarrow N$. Likewise, we define T_{freq} to denote that the tag set T is ordered by frequency, while we use T_{lex} if it is ordered lexicographically.

The prefix tree P is generated from the set of tags T with the function $f_{tree} : T \rightarrow P$. Algorithm 5.1 provides the pseudo code for that function which consists of the following steps:

- **Adding the empty string λ to T :** It serves as prefix¹ for all tags $t_i \in T$ that do not have another tag $t_j \in T$ as prefix.
- **Ordering T lexicographically:** Fast sorting algorithms such as merge sort can do this in $\mathcal{O}(n \log n)$ time. As a result, we obtain the lexicographically ordered list T_{lex} with the empty string λ as first element.
- **Generating the prefix tree from T_{lex} :** This step processes the entire list T_{lex} once and checks if tag $t_j \in T_{lex}$ is a prefix of tag $t_i \in T_{lex}$, where $j < i$. If true, it subtracts t_j from t_i and saves the resulting string (i.e., the prefix) as vertex $v_i \in V$. In addition, it creates

¹Note that the term *prefix* is not used in its linguistic sense but in its meaning related to data structures in this work.

```

PrefixTree( $T$ ):
1: //  $T$ : set of tags
2:  $T := T.add(\lambda)$ ; // add empty string  $\lambda$  to  $T$ 
3:  $T_{lex} := lex\_order(T)$ ; // order  $T$  lexicographically
4:  $n := size(T_{lex})$ ;
5:  $V := T_{lex}$ ; // vertices of prefix tree
6:  $E := \emptyset$ ; // edges of prefix tree
7: for  $i := n$  to 1 do
8:    $j := i - 1$ ;
9:   while  $(!(t_j \prec_{pref} t_i) \wedge (j \geq 1))$  do
10:     $j --$ ;
11:   end while
12:    $v_i := t_i - t_j$ ; // subtract tag  $t_j$  from tag  $t_i$ 
13:    $E := E \cup (v_i, v_j)$ ; // add edge to prefix tree
14: end for
15: return  $P = (V, E)$ ; // prefix tree  $P$ 

```

Algorithm 5.1 — Prefix tree generation.

an edge between v_i and $v_j \in V$. Since it uses two nested loops, one being conditional, it has $\mathcal{O}(n^2)$ time complexity in worst case and $\mathcal{O}(n)$ in best.

The result of Algorithm 5.1 is the prefix tree $P := (V, E)$ for T , i.e., a graph where the vertices V represent the prefixes and the edges $E \subset V \times V$ describe the hierarchical structure of the prefix tree. The root node $v_{root} \in V$ of the tree is the empty string λ .

Figure 5.9 illustrates the generation of the prefix tree P_x for a sample list of tags T_{lex_x} . The list contains a small number of lexicographically ordered tags starting with the letter “v”. The occurrence frequencies f_{freq} of the tags are also depicted, as they will later be used to scale the font sizes in the prefix subtrees. Figure 5.9b depicts the hierarchical structure of the prefix tree, while Figure 5.9c shows the actual prefix tree as it results from Algorithm 5.1. The tree is displayed in a left-to-right orientation that is later also used to display the subtrees in the prefix tag cloud.

The mapping between prefixes and tags is described by the bijective function $f_{map} : V \rightarrow T$, i.e., each prefix $v_i \in V$ is associated with exactly one tag $t_i \in T$ and vice versa. In the example of Figure 5.9, it means that the prefix “visual” is associated with the tag “visual”, while the prefix “ization”

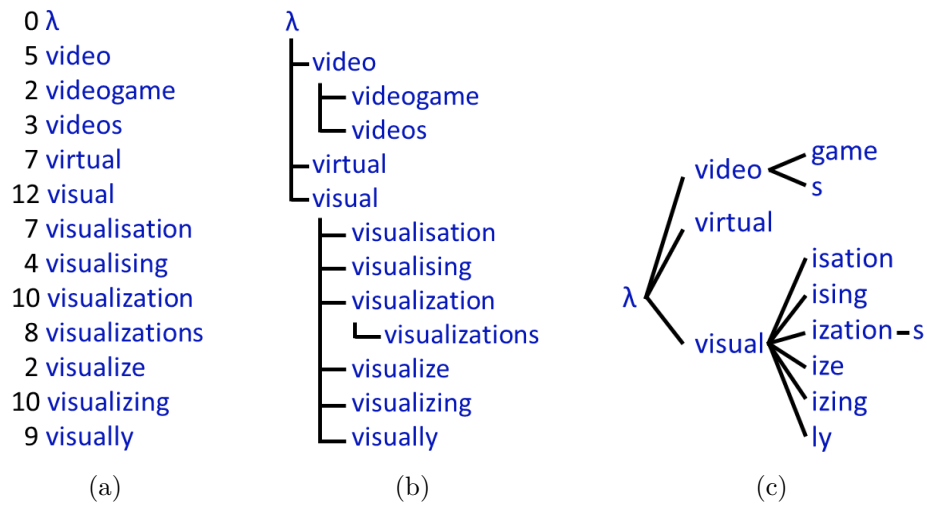


Figure 5.9 — Transforming a list of tags into a prefix tree: (a) Lexicographically ordered list of tags and their frequencies. (b) Node-link diagram illustrating the prefix tree structure, with tags shown in full length. (c) Node-link diagram of the vertically aligned prefix tree, with tags split into prefixes.

is mapped to the tag “visualization”. Note that the ending “s” following the prefix “ization” is also called *prefix* in our case. It represents the tag “visualizations” and is treated like the other prefixes.

5.3.2 Subtree Rendering

Before we generate the prefix tag cloud, we split the prefix tree P at its root node v_{root} into a set of subtrees $P' := (V', E')$ with $V' \subset V$ and $E' \subset E$. Each subtree P'_i represents either a group of word forms (if $|V'_i| > 1$) or a single word (if $|V'_i| = 1$). The latter is the case for tags that have v_{root} as prefix and no child node, such as the word “virtual” in Figure 5.9.

The subtrees of P' are visualized as node-link diagrams in a left-to-right orientation. Parent nodes are placed in the vertical center and to the left of their child nodes, as it was sketched for the whole prefix tree in Figure 5.9c. This left-to-right orientation is consistent with the dominant reading direction in English and other languages. The diagrams have different background colors so that they can be more easily distinguished. However, color is not necessary to read and understand the visualization.



Figure 5.10 — Alternative renderings of the same prefix subtree: (a) showing the complete tags, (b) showing only the prefixes.

The font sizes of the prefixes in the subtrees are scaled according to the frequency values (given by f_{freq}) of the associated tags (given by f_{map}). Since a linear change of the font size has a roughly quadratic effect on the text area, we use the square root of the frequency values for scaling. Using a linear scale instead would overstate the larger tags, while a logarithmic scale would understate them. There may be cases where such scales are more suitable, even though they increase the “lie factor” [236] of the visualization. For instance, a logarithmic scale is often used if the frequency values of the tags tend to follow a power law distribution, as in many folksonomies [225].

Figure 5.10 shows two alternative renderings of subtree $P'_{visual} \subset P_x$ that was already depicted in Figure 5.9c: One visualizes the tags $t_i \in T_x$ and the other the prefixes $v_i \in V_x$. As the latter diagram is less redundant and more space-efficient, we use it for the prefix tag clouds.

5.3.3 Tag Cloud Generation

The subtrees are visualized in a circular tag cloud layout with the most frequent tags in the center and tags with decreasing frequencies towards the boundary. As we found in the user study (see Section 5.2), this layout supports particularly well the identification of popular tags, which we consider a key task related to prefix tag clouds.

We create the circular layout with a simple yet effective algorithm that is similar to the one presented in [93]. It places the subtrees of P' along a spiral path, as sketched in Figure 5.11.²

²Note that the actual distance between turns of the spiral must be much smaller than sketched to get results like in this work.

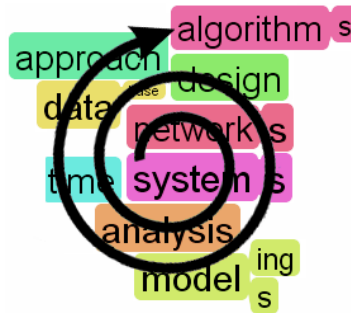


Figure 5.11 — Spiral placement of the prefix subtrees to create the circular tag cloud layout.

First, the drawing area is defined, which can, in principle, be of any shape. As graphical user interfaces are usually based on 2D grid layouts, a rectangle may be the ‘shape of choice’ in most cases. Drawing starts with the subtree that contains the tag with the highest frequency value, i.e., tag t_1 from the descending ordered list of tags T_{freq} . This first subtree is placed in the center of the drawing area. In the example of Figure 5.11, it consists of the tags “system” and “systems”, i.e., the diagram shows the prefixes “system” and “s” accordingly.

Subsequently, subtrees with decreasing frequencies are placed along the spiral path. This is done by traversing T_{freq} and rendering the associated subtrees as described above. As long as a subtree would intersect with any previously placed subtrees in the drawing, it is moved further along the spiral path. It is also moved if it would be placed outside the drawing area. If there is not enough free space left to place a subtree in the drawing area, it is continued with the next subtrees until some predefined threshold or until the last element of T_{freq} is reached. Rendered subtrees are marked as rendered to avoid multiple placement of the same subtree.

If we would strictly adhere to the circular layout as described above (Section 5.2.1), we would not be allowed to place tags with small font sizes close to the center of the cloud. However, we would waste a lot of space in such a strict layout, as we do not only place tags of rectangular shape but also subtrees with more complex shapes. Hence, we decided to fill free space between larger subtrees with smaller ones. We do this by starting the spiral placement for each subtree in the center of the drawing area so that smaller subtrees are rendered whenever there is enough space, also between already drawn subtrees.

Subtree drawing continues until one of three conditions is met: 1) all subtrees of P' have been rendered, 2) a user-defined number of subtrees of P' has been rendered, or 3) the drawing area is completely filled with subtrees. The first two conditions result in a circle-shaped tag cloud due to the spiral placement of the subtrees. The circle may be cut by the border of the drawing area if it does not completely fit. The third condition results in a tag cloud with the shape of the drawing area.

Finally, we implemented a method that combines the above conditions. It automatically scales the font sizes of the prefixes so that a given number of subtrees fits exactly in the pre-defined drawing area. Different scaling factors are tested until a solution is reached, while the relative differences between the font sizes are retained as much as possible. In the next section, we provide examples of a circle-shaped tag cloud and a rectangle-shaped one.

5.3.4 Application Examples

Apart from the visualization of folksonomies, a popular application area of tag clouds is text summarization. In this context, tag clouds are typically used to depict the words that occur most often in texts. The font sizes of the tags indicate the word frequencies, i.e., the larger a tag the more often it occurs in the text [93]. Such tag clouds can provide an overview on the topics of a text and serve as a starting point for a deeper analysis [243, 225]. For instance, they help to judge whether a given text is relevant to a specific information need or not. When a tag cloud visualization is used this way, the ‘tags’ are words from a text. For this reason, the term *word cloud* is often preferred over the term *tag cloud* in these contexts. However, we will keep using the term *tag cloud* for this kind of visualization in this work.

We also applied prefix tag clouds in the context of text summarization to evaluate their usefulness. The investigated text corpus is a large dataset containing publication information of the computer science bibliography DBLP [159]. We used the XML file of the DBLP dataset, dating from February 25, 2013. It contains bibliographic information for more than 2.1 million publications in the field of computer science. We transformed the publication titles into individual tags with techniques common in tag cloud generation [81, 93]: We first used regular expressions to remove special characters and separate words by spaces. We then converted the tags to lowercase and removed stop words, i.e., common words like “the”, “is”, or



Figure 5.12 — Differently shaped prefix tag clouds, each showing 150 words from publication titles indexed in the computer science bibliography DBLP: (a) most frequent words of all publication titles, b) most frequent words of publication titles containing the string “visual”.

“at” that do not carry relevant meaning. Finally, we counted the frequencies with which the words appear in the titles. As a result, we got the data required to generate prefix tag clouds, i.e., a set of tags T and the mappings $T \rightarrow N$ defined by f_{freq} .

Figure 5.12a depicts a rectangle-shaped prefix tag cloud showing the 150 words that appear most frequently in all publication titles. Very frequent words, such as “system”, “network”, and “analysis”, become immediately apparent. These words occur around 100,000 times in the titles. Words with small font sizes, such as “integration” or “random”, occur around 14,000 times. Words below this frequency value are not displayed in this highly filtered view.

The grouping in the prefix tag cloud reveals that the singular and plural forms of several words occur with similar frequencies in the paper titles, as the plural “s” has often nearly the same font size (e.g., “system” or “network”). Other words occur more often in either singular (e.g., “image” or “method”) or plural (e.g., “graph”). The word “model” is not only used in singular and plural but additionally in its gerund form. The gerund is also popular for the words “process” and “test”. Finally, the words “data” and

“database” are grouped, while “data” is used more frequently, as indicated by its larger font size.

Figure 5.12b depicts another prefix tag cloud generated from the DBLP dataset. It has a circle shape and shows all words that occur most frequently together with the word “visual” or any other word having “visual” as prefix. We generated it from the subset of publication titles that resulted after filtering for the string “visual”. As this string appears in all publication titles of the subset (which is 31,802 times), it has the largest font size in the prefix tag cloud.

The prefix tag cloud contains several subtrees that are similar to the ones visualized in Figure 5.12a. There are also new subtrees, such as one consisting of the prefixes “real” and “ity” or another grouping “audio” and “visual”. Furthermore, we can observe that the word “program” is often used in its gerund form while “technique” is mostly used in plural. The subtree of “visual” is similar to the one that already served as an example in Figure 5.10. The prefix is most often used in the word “visualization”, while it also appears frequently in the words “visualizing”, “visualisation”, “visually”, and the plural form of “visualization”. Interestingly, even the German word “visualisierung” is part of the subtree, even though publications in computer science are usually written in English.

5.3.5 Discussion

While there are several works on layout issues of tag clouds, only few address the grouping of different word forms. One approach is presented by Cui et al. who use the Porter stemming algorithm to collapse words with the same stem and only include the most frequent word form in the tag cloud [72]. A similar approach is taken by Dörk et al. [81]. However, the viewer gets no idea which word forms have been collapsed, as this information is not shown in the tag clouds.

Stemming algorithms, on the other hand, are indeed a valid alternative to the prefix trees used in our approach. However, they require knowledge about the language of the tags, while prefix tag clouds can in principle be used with any language that is based on Latin script and features inflected word forms. They rely purely on lexicographical ordering, which makes them versatile and easy to implement.

Another related approach is the work on Word Trees by Wattenberg and Viégas [248]. They use a suffix tree to present search terms in their textual

context. Instead of single words, they analyze complete sentences and visualize the sentence structure as tree. It is the only work we found that uses a data structure similar to prefix trees for visualization purposes. It is conceptually related to our approach but serves a different purpose by displaying keywords in context instead of grouping word forms in tag clouds.

5.4 Considering Time and Tag Relations

As we have illustrated in Chapter 2 (especially in Section 2.4.1), many relationships emerge if tags are used together. These relationships are not considered in the tag clouds presented so far. While prefix tag clouds show the relationships between different word forms, they do not reveal relationships between different tags in the cloud. It is, for instance, not visible from Figure 5.12b which tags are used together with ‘visualization’.

Another limitation of tag clouds is that they depict tag frequencies summed up over time. They do not provide information on time-varying tag use. If we refer to the example in Figure 5.12b again, we cannot see any trends, i.e., we cannot say which tags have been used exceptionally often in paper titles in the last couple of years.

To overcome these limitations, we developed an interaction technique we call *time-varying co-occurrence highlighting*. It uses visual highlighting of co-occurrences combined with colored histograms to indicate time-dependent tag relations in tag clouds.

5.4.1 Time-Varying Co-occurrence Highlighting

Essentially, the technique is based on two visual features:

- Each tag in the cloud is enhanced by a histogram that visually depicts the variation in tag use over time.
- Whenever a tag is selected in the cloud, related tags are visually highlighted and co-occurrence information is shown.

Both visual features are integrated as follows:

- Along with the co-occurrence highlighting, those bars in the histogram are colored that represent the point in time of co-occurrence.



Figure 5.13 — Small tag cloud enhanced by (a) plain and (b) time-varying co-occurrence highlighting.

Figure 5.13a shows a small example tag cloud that has been enhanced by a feature we call *co-occurrence highlighting*. Hovering over a tag leads to a highlighting of all co-occurring tags in the cloud. In the example, the tag ‘center’ has been selected, as indicated by its red color and border. Tags that are used together with ‘center’ are highlighted in yellow, with the number of co-occurrences given in parentheses.

Tags that are used very often together with ‘center’ are additionally colored in red, such as the tag ‘tennis’ that co-occurs 391 times with ‘center’ in this example. By contrast, the tag ‘game’ has a black color in Figure 5.13a, as it co-occurs only five times with ‘center’. In general, the more often a tag has been used together with the selected one, the stronger its red color, i.e., the font color of the highlighted tags correlates with the co-occurrence frequency. We decided to use this indirect method of visual highlighting, as drawing direct links between the tags would result in visual clutter. We will discuss this issue and alternative solutions Chapter 6.

Clicking on a tag reduces the set of tags displayed in the cloud to those that co-occur with the selected one (including the selected tag itself). This allows to explore a certain subset of possibly related tags separated from the others. If there is no distraction by other tags, relevant relationships and trends may be found more easily in the set of tags.

In addition, histograms are plotted in the background of the tags. They show the variation in tag use over the analyzed time span. Those bars in the tags’ histograms that represent the point of co-occurrence are marked in red, while the other bars remain blue. For instance, in case of the tag ‘2011’,

only the left bars of the histogram are red, indicating that all co-occurrences of ‘center’ and ‘2011’ appear in the first half of the analyzed time span.

We distinguish between two forms of time-varying co-occurrence highlighting: In the *absolute* variant, each co-occurrence leads to a complete coloring of the corresponding bar in the histogram, as in Figure 5.13b. In the *relative* variant, the bars are only partly colored, indicating the number of co-occurrences in relation to the tag frequency at the corresponding point in time.

5.4.2 Application Example

As for prefix tag clouds (see Section 5.3.4), we applied the interaction technique of time-varying co-occurrence highlighting in the context of text summarization. In this case, we used short messages from the microblogging service TWITTER [32] as text corpus. We decided for the application context of microblogging, as it is strongly time-dependent. However, we could have also used folksonomies which are also very time-dependent [111, 226, p. 101].

We retrieved the microblogging messages from a large database of TWITTER [32] posts created and maintained by Thom et al. [233]. The database contains all publicly available posts that come with geo-coordinates. These posts are typically sent from mobile devices, such as smartphones. For our example dataset, we used the geo-coordinates of the posts as initial filter and selected a subset restricted to an area of New York. Furthermore, we filtered the posts by date, using a time span from September 3 to 19, 2011. We then applied the *Spatiotemporal Anomaly Detector* offered by Thom et al. to find anomalies in the Twitter data that correspond to certain events. It analyses the spatiotemporal density, word usage, and number of contributing users to identify posts that are likely from local witnesses of an event (see [233] for a detailed description of the approach and the related algorithms).

We found events related to tennis, the New York Yankees, and Electric Zoo for days 3 to 5, and events related to Maker Faire as well as Occupy Movement for days 16 to 18. Because the tag ‘party’ occurred most frequently, we used it to pre-filter the microblogging messages. We then removed it from the text corpus, as it would otherwise be associated with most messages across all times, thus not adding any value. We finally got a sample of 4,934 posts that we used to examine the applicability and usefulness of time-varying co-occurrence highlighting.

We processed the microblogging messages with NLP techniques typically used in tag cloud generation, such as tokenization, lowercase conversion,



Figure 5.14 — Visual interface showing a tag cloud where the time-varying co-occurrences of the tag ‘tennis’ are highlighted.

and stop word removal [34, 81]. That is, we first split the text streams at whitespace characters to get the individual words, then converted these words to lowercase, and removed stop words, i.e., common words (such as ‘the’, ‘is’ and ‘at’) that usually do not carry meaning. Finally, we aggregated syntactically identical words and printed them on the screen, with the font size scaled to their frequency of use. Figure 5.14 shows the tag cloud generated for the text corpus we retrieved from the database of Twitter messages, embedded into the visual interface that enables to configure the visualization.

The visual interface offers several parameters to adapt the visual appearance of the tag cloud to the user’s needs (see Figure 5.14). The users can, for instance, switch between linear and logarithmic scaling of the tags’ font size, depending on the distribution in tag frequencies. Furthermore, they can adapt the overall size of the tag cloud by specifying the maximum number of tags that are shown, or the minimum number of times a tag must be used to be displayed in the cloud. Finally, they can change the font and background colors, the horizontal and vertical spacing, and the minimum and maximum font sizes of the tags.

The histograms can also be configured. Among others, the user can specify the aggregation interval of the histogram, i.e., how much time is represented

by each bar in the histogram. In the example of Figure 5.14, an aggregation value of 10 was chosen, meaning that each bar in the histogram represents 10 hours of microblogging (except from the last bar which is the remainder). Since the aggregation method was set to ‘average’, the presented values are the average occurrences within the 10 hour time frames. As for the tag clouds, one can specify further parameters like the color of the histograms or if the histograms are visible at all.

5.4.3 Discussion

There are related approaches that encode time in tag clouds, such as *SparkClouds* [155] or *ParallelTagClouds* [70]. Parallel Tag Clouds combine the ideas of tag clouds and parallel coordinates to show tag frequencies at multiple points in time simultaneously. They allow to compare changes in tag use, but they do not directly visualize the evolution of the tags. This is different in SparkClouds where a sparkline (i.e., a simplified line graph) is added to each tag to explicitly show changes in tag use over time. While this is similar to our histogram visualization, using histograms with discrete bars instead of continuous sparklines to encode time has some advantages: On the one hand, it allows to color-code certain bars, what is needed to indicate time-dependent co-occurrences. On the other hand, it lets the user specify the aggregation interval, i.e., how much time is represented by each bar in the histogram.

Neither SparkClouds nor Parallel Tag Clouds show tag relations and are thus not able to visualize time-varying co-occurrences. This is also true for other visualizations that use line graphs or stack graphs to illustrate tag evolution, such as *Topic Streams* [81] or *Twitter StreamGraphs* [68]. The TIARA [164] system combines tag clouds with river flows similar to the ThemeRiver visualization [117] to show trends in tag use. Finally, there are approaches that enable users to interactively explore the tag use over time with GUI elements such as sliders [186, 233]. However, they do also not include co-occurrence information.

Co-occurrence information is, in turn, considered by approaches that visualize tag relations in tag clouds, such as the *clustered tag clouds* mentioned above. They implicitly indicate tag relations by spatial proximity, for instance, by placing frequently co-occurring tags close to each other. Co-occurrence information will also be used in the folksonomy graphs we present in the next chapter. They make the tag relations explicit by connecting co-occurring

tags with edges. Related to this are visualization techniques that combine tag clouds and trees. For instance, Tree Clouds [98] arrange tags on a tree to visually depict their relatedness. However, in contrast to the above works, all these approaches do not contain information about time-varying changes in tag use.

Time-varying co-occurrence highlighting combines both directions of extending tag clouds for visual analysis. It integrates information about the temporal distribution and co-occurrence of tags in one interactive visualization. It therefore takes into account that both can be important information in the analysis of folksonomies and text contents and is best explored in an integrated manner.

A possible drawback of the technique is the increased complexity of the tag cloud visualization. Though we still regard tag clouds enhanced with time-varying co-occurrence highlighting as relatively easy to understand and use, they are not as simple and puristic as common tag cloud visualizations. Especially the histograms create additional visual information that may negatively affect the readability of tag clouds.

The co-occurrence highlighting, by contrast, should not have a relevant effect on the perception of tag clouds, as it does not change the general appearance of the tag cloud but purely equips it with interactive highlighting. The user first sees a common tag cloud and can then use the interactive highlighting to further explore the contents. That way, it adds contextual information that can be helpful to identify multiword expressions or ambiguities.

5.5 Interactive Analysis with Tag Clouds

This approach of using a common tag cloud visualization and enhancing it with interactive features was also the guiding principle in the tag cloud based analysis system we developed. We call it the *Tag Cloud Explorer*, as it uses tag clouds as its main visualization method. The tag clouds are equipped with advanced natural language processing, intuitive interaction techniques, and a high level of control for users in order to support different kinds of analysis tasks. The Tag Cloud Explorer combines several ideas of the previously presented work in one consistent analysis framework. It makes heavy use of natural language processing to enrich the tag clouds with linguistic information. It is therefore particularly useful for the analysis of text contents, though several features can also be used to analyze folksonomies.

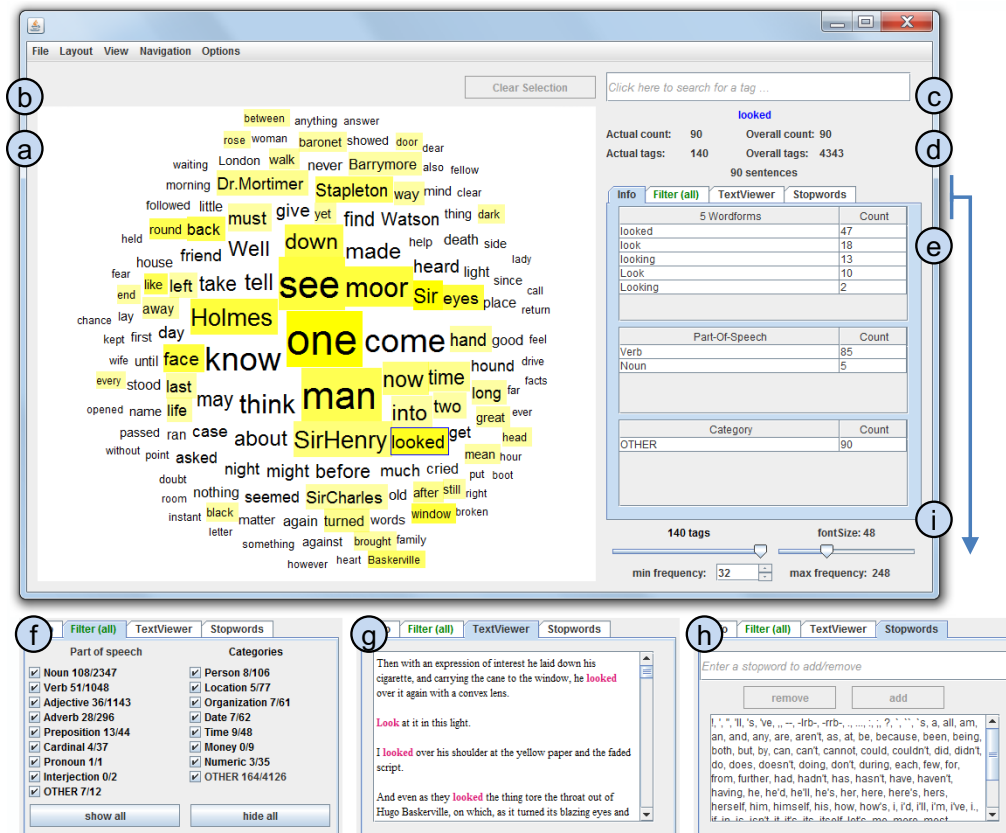


Figure 5.15 — Tag Cloud Explorer consisting of the following components: (a) tag cloud visualization, (b) tag filter, (c) search box, (d) tag statistics panel, (e) info panel, (f) part-of-speech and named entity filters, (g) text viewer, (h) stopword editor, and (i) cloud control panel.

5.5.1 Tag Cloud Explorer

In the following, we provide an overview of the Tag Cloud Explorer, which is implemented in Java and depicted in Figure 5.15. As an example text for this screenshot, we used the popular Sherlock Holmes novel “The Hound of the Baskervilles” by Arthur Conan Doyle. The Tag Cloud Explorer consists of the central tag cloud view and a number of other components providing additional information and functionality for the analysis. The individual components are marked with letters in Figure 5.15. In the following, we will describe their functionality and explain how they support text analysis.

Text Processing

One of the drawbacks of all tag cloud approaches presented so far is that they provide a purely statistical summary of the tags without taking deeper linguistic knowledge about the tags and their relations into account. To overcome this limitation, we equipped the Tag Cloud Explorer with a component for linguistic analysis. We use the Stanford CoreNLP tools [23] for this purpose and perform several steps of natural language processing, consisting of tokenization, sentence splitting, part-of-speech tagging, lemmatization, and named-entity recognition.

Based on the results of the part-of-speech tagger, we additionally implemented a detector for nominal multiword expressions. It joins all continuous sequences of proper nouns that occur in the same sentence. With this simple heuristic, we can detect most compound nominals and proper names in the text (see [213] for a comprehensive summary of different multiword phenomena).

The separate display of multiword expressions is important for many analysis tasks, especially those involving the identification of person or place names which are often multiwords (e.g., “Michael Jordan” or “New York”). Another benefit of considering multiwords is that the frequency counts of the individual tags are not artificially increased (e.g., “new” as part of “New York”).

Tag Cloud Layouts

The tag cloud forms the central visualization in the Tag Cloud Explorer. We implemented three different tag cloud layouts that the users can choose from: two sequential line-by-line layouts, one sorted alphabetically, the other by frequency, and a circular layout showing the tags with the highest frequency in the center of the cloud and the lower frequency tags closer to the perimeter – similar to the one that was investigated in Section 5.2 and used in the prefix tag clouds (see Section 5.3.3).

While the alphabetical layout supports users in quickly spotting specific tags they are looking for, the frequency ordered layout lets them arrange tags according to how often they occur in the text. The circular layout complements the sequential line-by-line layouts as a space-efficient and visually appealing alternative. Font size is scaled linearly with the occurrence frequency of the tags for all layouts. As the Tag Cloud Explorer allows for a free placement of tags, additional tag cloud layouts can easily be added to

the implementation. Also, the mapping of the frequencies to the font sizes of the tags can easily be adapted.

The Tag Cloud Explorer uses the information about different word forms provided by the lemmatization component to subsume them under one representative tag in the cloud. This means that, for example, not all inflections of a verb are shown as separate tags in the cloud, but that their counts are added up and only their most frequent representative is displayed. Detected multiwords are displayed in camel case to make them easily recognizable as one entity. However, both features can be disabled in the menu if users do not want a special treatment of multiwords or the merging of different word forms.

In addition, we implemented the above described technique of *co-occurrence highlighting* for the tag clouds (see Section 5.4.1), i.e., users can hover over tags to highlight related ones. In the current implementation, two tags are related if they co-occur within the same sentence. However, the co-occurrence context can easily be changed, i.e., to link all tags that are used within the same tagging (cp. 2.4.2). In contrast to the implementation of co-occurrence highlighting presented in Section 5.4.1, the degree of co-occurrence is not expressed by font color in the Tag Cloud Explorer. Instead, the saturation of the yellow box around the related tags indicates the co-occurrence frequency. The box of tags that appear often with the selected one have a higher saturation than boxes of tags that co-occur only few times. This allows to use font color for different purposes, such as the part-of-speech coloring shown in Figure 5.17.

The second effect of hovering over a tag is that further information about it is displayed in the tag statistics panel (d) and info panel (e) we will describe in the following.

Tag Statistics Panel

The tag statistics panel (d) displays information about a selected tag. This is illustrated for the tag “looked” in Figure 5.15. The tag statistics panel lists the number of occurrences of the tag within the filtered set of sentences (*Actual count*), the number of tags currently present in the tag cloud (*Actual tags*), the number of occurrences of the selected tag in the whole text corpus (*Overall count*), and the overall number of tags in the corpus (*Overall tags*). Finally, it gives the total number of sentences in which the focused tag

occurs (which is identical to the above values in this case, as no filters are selected).

The information from the tag statistics panel can, for instance, be used for sanity checks. One apparent disadvantage of tag clouds is that the difference in frequency between tags as judged according to their font size can give a false impression about the true frequency count ratio of the tags. Showing the absolute frequency values to users lets them easily correct false impressions.

Info Panel

The info panel (e) is part of the tabbed pane below the tag statistics panel. It displays linguistic information about the focused tag, consisting of the detected word forms, part-of-speech tags, and named entity types for that tag, along with the respective frequency counts.

In the example of Figure 5.15, five different word forms have been detected for the focused tag. The word form “looked” is chosen as the representative because it appears most often in the text (as given by its count value). Furthermore, it has been detected that the tag is mainly used as a verb in the text and that it is not a named entity (indicated by the category name “OTHER” in the list of named entities, see below).

This supports the analysis in two respects: First, it can be used to learn more about a tag in the cloud. For instance, it might be relevant for an analysis task if a tag occurs mainly in present or past tense, or if it is used as verb or noun. Second, it can be used to cross-check the results of the linguistic analysis. Although the accuracy of the used text processing techniques is generally high, they are sometimes prone to errors depending on the type and quality of the text.

Filtering and Searching

Tags can be selected by clicking on them. They are then added to the tag filter (b). The tag cloud displays only those tags that co-occur with *all* of the selected tags, i.e., it changes from a tag cloud for the whole text to a ‘co-occurrence cloud’ as soon as tags are selected. Figure 5.16 shows a co-occurrence cloud for a text corpus with research abstracts from the field of visualization (cf. Section 6.3.4). The selected tags are colored red (in this case, “text” and “visualization”). They can be added and removed from the tag filter in any order and at any time, which triggers an update of the tag cloud accordingly.

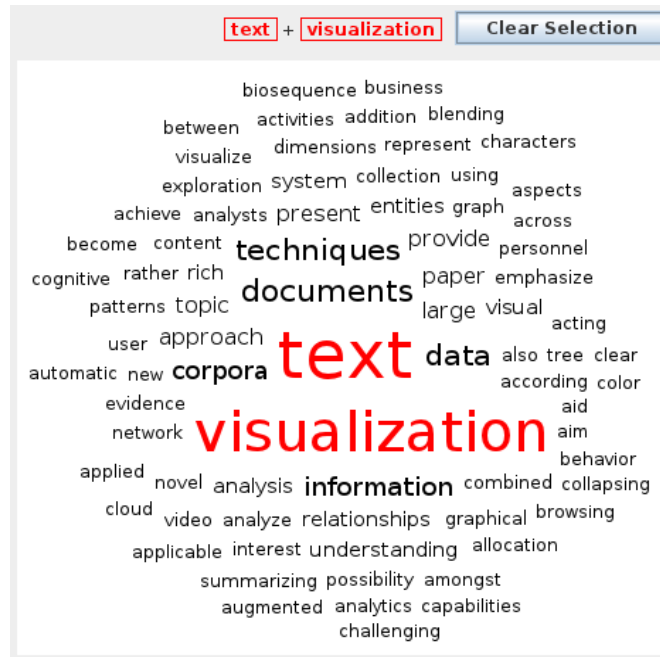


Figure 5.16 — The tags *text* and *visualization* have been added to the tag filter and are highlighted in the resulting co-occurrence cloud.

The tag filter functionality can be used to focus entirely on the co-occurrences by removing all tags from the cloud that do not share any sentence with the selected ones. Thus, an effective drill down to relevant information is supported that facilitates iterative analytic processes.

Another way of adding tags to the tag filter is by using the search box (c). Here, users can enter tags independently of whether they are part of the tag cloud. If a word is entered that occurs in the text, the tag statistics panel (d) and the info panel (e) on the right display information about it. If it is part of the tag cloud, it is highlighted, along with all co-occurring tags. Users can add words from the search box to the tag filter list. The search box thus allows the construction of co-occurrence clouds for tags whose frequency is too low to be displayed in the tag cloud.

Part-of-Speech and Named-Entity Filters

The filter tab (f) enables users to explore the tag cloud according to different parts-of-speech (POS) and different named entities (NE). All POS types of the Penn Treebank tagset [181] are supported by the CoreNLP POS

Tagger [234]. They are condensed to nine major POS types in the user interface of the Tag Cloud Explorer. Likewise, the NE types of the CoreNLP NE Recognizer [94] are listed in the user interface (labeled with ‘categories’ for better understanding).

Users can hover over the POS and NE types to highlight all respective tags in the cloud and see which tags belong to which category. As for the co-occurrence highlighting, we used a yellow box whose saturation indicates what fraction of occurrences of the tag has been categorized with the respective type. The two numbers after each type name denote the number of tags of the respective type being part of the cloud and of the overall text corpus, respectively. For convenience, the two buttons on the bottom of the filter tab allow users to select or deselect all types at once. Disabling one POS or NE type has the effect that counts of this type are subtracted from the overall frequency of the respective tags.

Being able to filter by part-of-speech and named entities is a powerful feature of the Tag Cloud Explorer that enables users to explore specific aspects of the tags and corresponding contents. The tag cloud can, for instance, be set up to show only locations that occur together with a certain person, or users can look at adjectives that co-occur with a specific organization.

Text Viewer and Stopword Editor

If the Tag Cloud Explorer is used to analyze text content, the text viewer (h) can be used to list all sentences in the text that contain the selected tags (as shown for the tag “looked” in Figure 5.15). This enables the users to review tags in their original context and verify hypotheses, for instance, about the relationship between two persons.

The text viewer has a very limited functionality in the current implementation. It might be extended to show more text context and allow for sophisticated searching, filtering, and further interaction on its own. Another option would be to reuse an existing text viewer and integrate it into the Tag Cloud Explorer.

With the stopwords editor (g), users can modify the stopwords list and adapt it to their analysis context. Any modification of the list triggers an instant update of all views. The list can also be externally changed by editing the corresponding text file.



Figure 5.17 — Tag cloud for the text “The Hound of the Baskervilles”, with the tags colored according to their most frequent part-of-speech.

Configuration of the Tag Cloud

The cloud control panel (i) allows to dynamically change the size of the tag cloud and of the displayed tags. Users can set the maximum number of tags, the maximum font size of tags, or the minimum frequency that tags require to appear in the cloud. When one of these values is changed, the other values are adapted accordingly.

In the menu of the Tag Cloud Explorer, users can disable stopword filtering, define a cutoff frequency for the co-occurrence calculations, disable the concatenation of multiword expressions, and turn the lemmatizer off. The latter has the effect that different word forms are no longer merged but that each tag represents a single word form, as in common tag clouds.

Another feature offered in the menu is a coloring of the tags according to their most frequent part-of-speech. A screenshot of a tag cloud with this functionality activated is depicted in Figure 5.17.

5.5.2 Use Case

As for the previously presented approaches, we will demonstrate the analytical power of the Tag Cloud Explorer on the example of analyzing text contents. For this purpose, we downloaded Reuters Corpus Volume 1 (RCV1), which is a large collection of manually categorized news articles made available by Reuters Ltd. for research purposes [158].

We use the first week of articles from the corpus, ranging from August 20 to August 26 of the year 1996. Three major global sports events were dominating the news during that week. First, the summer Olympic games

0-0 1-0 100 6-1 6-2 6-3 6-4 7-5 7-6 about added **after** again **against** aggregate ahead allowed **also** American Atlanta Attendance
Australia Australian Austria away **back** ball **beat** become **before** began behind Belgium best between break Britain came captain career
champion **championship** chance close club coach Cup CzechRepublic **day** defeat division double down **draw** drove early eight end
England English European event face felt fifth **final** finished **first** five following forced former **four** fourth France French Friday
game gave German **Germany** **get** goals good group half Halftime held **hit home** homer hope hours including injury **innings**
international **into** Italy Japan **just** keep know **last** later **lead** league left leg like looking loss **lost** **made** manager **match** may
meeting Men **metres** million **minutes** missed Monday month move needed New NewYork NewZealand next nine now number Olympic **one**
Open Oval Page Pakistan past penalty pitch place **played** **players** **points** put qualifier qualifying **race** rally ranked reached really
record **Results** return right **round** rugby **runs** Russia **Saturday** saw **scored** Scorers **season** **second** seconds **seed** series **set**
seven shot side since six sixth soccer SouthAfrica **Spain** sport squad Standings **start** still straight struck **Sunday** Sweden tabulate **team**
tennis **test** think **third** **three** Thursday tie **time** title told **took** top tour tournament trying Tuesday **two** **U.S.** U.S.Open until
victory vs. walked want way Wednesday week weekend well wickets Wimbledon winner Women **WON** world WorldCup **year**

Figure 5.18 — Initial tag cloud depicting one week of Reuters sports news in a sequential line-by-line layout with alphabetical sorting.

that took place in Atlanta, GA that year. Second, the Wimbledon Tennis Championships taking place in London a little earlier. Third, the U.S. Open that started in New York in that week. We therefore restrict the Reuters corpus to sports news by selecting all articles that have been categorized accordingly. As analysis example, we use a task that was also part of the qualitative user study reported in the next section: “Who are the most frequently mentioned male and female Olympic champions during that week?”

In the Tag Cloud Explorer, we first choose an alphabetically ordered, sequential tag cloud layout as shown in Figure 5.18. The most frequent tags in this cloud are common words from the domain of sports, such as *won*, *played*, *match*, and *game*. We can also spot some names of countries, cities, and sports events, such as the aforementioned *Wimbledon*, *Olympic* games, and *U.S. Open*. Since we are interested in Olympic champions, we hover over the tag *Olympic*. In the tag statistics panel, we see that it occurs in 87 sentences of the text corpus. The info panel further informs us that it occurs 90 times in those 87 sentences, seven of which are occurrences within multiwords like *Olympic Committee*.

We switch to the co-occurrence cloud of *Olympic* (Figure 5.19a) and see that the tag *champion* is used most often with it. Among the other tags in the cloud, there are many athletes with their first and last names contracted by the multiword feature. We further see numbers denoting scores, years, distances, etc.

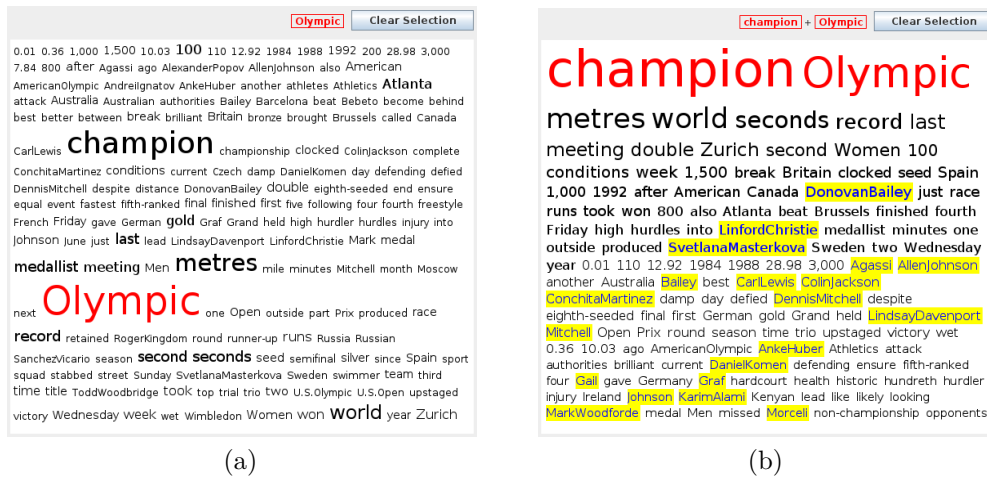


Figure 5.19 — (a) Alphabetically ordered co-occurrence cloud for the tag *Olympic*, (b) Frequency ordered co-occurrence cloud for the tags *Olympic* and *champion* with highlighted person names.

Next, we add *champion* to the tag filter and choose the frequency based ordering (Figure 5.19b). We then use the named entity filter to show only persons in the cloud (Figure 5.20a). We can see right away that *Donovan Bailey* is the most frequently mentioned Olympic champion in the analyzed part of the corpus. To read more about him, we open the text viewer that lists all sentences containing his name. We learn that he is a Canadian sprinter who set a speed record at the 1996 Olympic games.

Looking at the other names in the frequency ordered co-occurrence cloud of Figure 5.20a, we can quickly spot the female Olympic champion most often mentioned. Her name is the third tag in the cloud: *Svetlana Masterkova*. We select this tag and deselect the *Person* filter to get the co-occurrence cloud shown in Figure 5.20b. It reveals further information about Svetlana Masterkova, for instance, that she seems to be Russian and that there appears to be some connection to the distances of 800, 1,000, and 1,500 meters.

To verify these assumptions and get further information, we switch to the text viewer and learn that Svetlana Masterkova is indeed a Russian sprinter and double Olympic champion in the 800 and 1,500 meter distances. We also read that she set two world records shortly after her double win at the Olympic games, one in the 1,000 meter track at a competition in Brussels and another in the mile distance of the track and field event Weltklasse

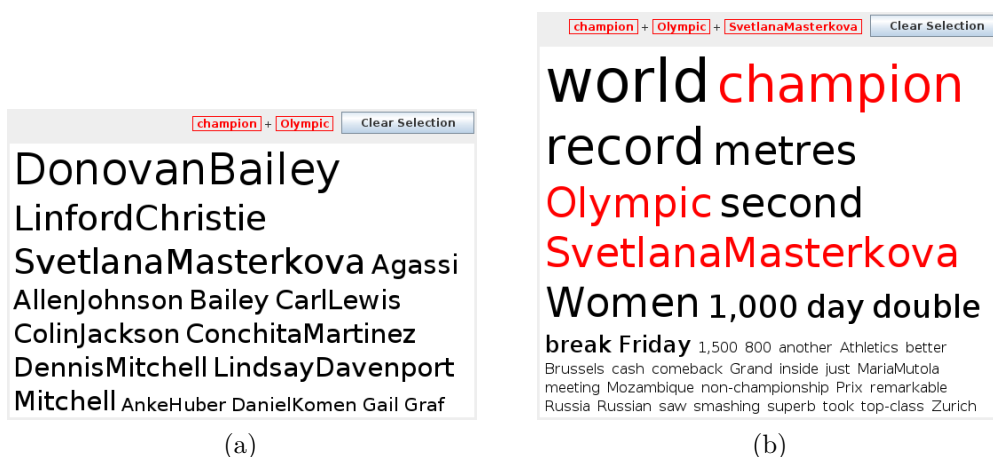


Figure 5.20 — (a) Frequency ordered co-occurrence cloud for the tags *Olympic* and *champion* filtered by person names, (b) Frequency ordered co-occurrence cloud for the tags *Olympic*, *champion*, and *Svetlana Masterkova*.

Zürich. Those four wins within a short time span caused a lot of attention in the news during the week in focus.

The application example showcases some of the strengths of the Tag Cloud Explorer. At the beginning of the analysis, users quickly get a rough idea of the tag space by skimming through the cloud. The most frequent tags and topics are thus conveyed immediately, offering a good starting point for the analysis. We started the analysis with the tag *Olympic* in the application example.

The different types of tag clouds facilitate the structured exploration by letting users arrange the tag space according to their information needs. In the example, the frequency-based layout helped to identify the persons most often mentioned together with the tags *Olympic* and *champion*. Using the co-occurrence clouds, users can filter the tag space and analyze parts of it in more detail, as we did with the name *Svetlana Masterkova*. Co-occurrence highlighting, on the other hand, was useful to interactively explore and discover tag relations. The named entity feature allowed us to focus only on person names. Finally, we could use the text viewer to refer back to the relevant parts of the original contents in order to verify assumptions and get further useful information.

5.5.3 Qualitative User Study

We conducted a qualitative user study to gain further insight into the effectiveness of the Tag Cloud Explorer and general approach.

Material and Tasks

We used three corpora in the study which we have already introduced in the previous sections: The first was the novel “The Hound of the Baskervilles” (see Section 5.5.1) that served as a training corpus in the study. The second was Reuters Corpus Volume 1 (RCV1), restricted to one week of sports news, as in Section 5.5.2. The third contained the abstracts of all publications of the IEEE VIS conference series from the years 1998–2011 [124]. We selected the latter corpus to include some research-oriented tasks in the study. A part of it was already shown in the co-occurrence cloud of Figure 5.16.

For the Reuters and VIS corpus, we designed questionnaires, each of which contained twelve tasks of varying difficulty. Basically, the tasks can be categorized into three major groups: (1) frequency based tasks, (2) exploration tasks, and (3) tasks asking for specific tags, with many of the tasks belonging to more than one of these groups. Example tasks include “Who was mentioned most often in connection with ‘Ferrari?’”, and “What is the Nyquist theory about?”.

Procedure

The participants were five members of the Institute for Visualization and Interactive Systems (VIS) of the University of Stuttgart. They were between 25 and 31 years of age and mostly male (one was female) with very good to excellent English skills according to their own judgment. All participants had experience with expert systems and analysis tasks, and most were familiar with the topic of text analytics. We considered this beneficial as we aimed for informed feedback from experts in this area.

The procedure for each of the participants consisted of the following five steps:

1. **Color vision deficiency test:** Each participant was tested for color vision deficiencies with the Ishihara color plates.
2. **User training:** We explained the features of the Tag Cloud Explorer using the “The Hound of the Baskervilles” corpus. The participants

could ask questions and try out the system until they felt confident to use it.

3. **Task completion:** We asked the participants to solve the analysis tasks, beginning with the Reuters sports news corpus followed by the corpus with the VIS abstracts. In case participants were stuck during a task, we kept hints on a separate sheet of paper that they could consult. To get additional insight, we encouraged the participants to articulate their thoughts during task completion according to the think-aloud method.
4. **Questionnaire:** We asked the participants to complete a questionnaire with demographic questions and questions about their experience with and thoughts on the approach.
5. **Discussion:** Finally, we discussed the general approach and specific analysis tasks with the participants.

Results

The study participants were surprised by the possibilities offered by such a simple and straightforward visualization as tag clouds if enriched with context information and sophisticated interaction techniques. Overall, they performed very well in solving the analysis tasks with the Tag Cloud Explorer. The paper with the hints was hardly ever used and the tasks were generally solved quickly and correctly without any help. We were impressed by this result, as we expected the participants to have more difficulties working with the Tag Cloud Explorer.

With respect to the tag cloud layouts, an interesting finding was that all participants preferred the sequential layouts over the circular one, although they rated the circular one to be aesthetically most appealing. When asked about this apparent contradiction, most participants answered that they found it easier to visually compare relative tag sizes using the line-by-line layout. This is because the lines could be used as visual anchors which facilitate to compare font height. Furthermore, we could observe the participants switching between the frequency and alphabetically ordered layouts according to whether they were interested in high frequency tags or searching for a specific tags. This indicates that it is important to provide different tag cloud layouts that users can choose from depending on the analysis task.

However, most participants preferred the search box over the tag cloud when searching for a specific tag. This result is not surprising, as using the search box is fast and also allows to find tags that are not part of the displayed tag cloud. Furthermore, it is in line with the finding of Sinclair and Cardew-Hall [225] that a search box is preferred for specific tasks while a tag cloud for more general ones.

The participants were in disagreement about the usefulness of the part-of-speech coloring. Some considered it a useful feature, while others found that it has little analytical value. It was argued that the part-of-speech of most words does not need to be visually communicated, as users normally know the part-of-speech of a word once they read it. Using the part-of-speech categories as a filter for the tag cloud was therefore considered more useful.

The named entity feature was unanimously found helpful and the participants used it frequently to solve the analysis tasks. The aggregation of multiwords and different word forms were also positively evaluated by all participants. Overall, the participants assigned the linguistically and interactively improved tag clouds many positive attributes (such as “tidy”, “clear”, “efficient”, “useful”). The user study revealed that the main advantages of tag cloud based analysis seems to be flexibility and intuitiveness, which indicates that tag clouds might be particularly beneficial in environments where training times should be minimal.

5.5.4 Discussion

There are other analysis systems that make use of tag clouds. Examples can be found in domains such as patent analysis [150], opinion mining [254], or investigative analysis [228]. However, most systems use tag clouds rather statically as a visual summary of text documents.

An interactive tag cloud variant has been implemented in the VisGets system [80]. It is used to filter tags in web-based information retrieval. Multiple tags can be concurrently selected as filters, restricting the resources in the result list to only those that are indexed with these tags. Related tags in the tag cloud are highlighted, as well as related elements in other views, such as temporal bar charts or geographic maps. However, tag clouds are used as one visualization component among many in the VisGet system. Although it is connected with other views by brushing and linking, the analytical possibilities of the tag cloud itself are limited.

A noteworthy exception is the POSvis system [245]. It is a literary analysis system helping scholars to review the vocabulary of novels, filter it by parts-of-speech, and explore networks of characters from the novel. It uses tag clouds as one of its main views and is therefore closely related to our work. It provides two main tag cloud views: One that displays extracted character names and another showing words from selected part-of-speech categories. In addition, it offers a third view consisting of multiple tag clouds, each dedicated to a specific part-of-speech category. Users can customize the order, size, and color of the tags in the cloud.

However, the analytical and interactive features of the tag clouds in the POSvis system are limited compared to our approach. The Tag Cloud Explorer offers generic focus-and-context techniques and direct interaction with the tag cloud. It provides a broad range of interactive features and is highly configurable. This makes it suitable for a wide variety of text analytics problems, while POSvis has mainly been designed for the analysis of novel characters and their relationships.

In future work, we plan to address the handling and comparison of multiple tag sources. An interesting question in this respect is how to extend the tag cloud to allow for the comparison of several tag spaces at a time. This could, for instance, be done using different colors, similar to the tag cloud component of the ManyEyes website [243]. Other alternatives could be Parallel Tag Clouds [70] or the use of a tag cloud matrix [245, 254]. In general, we aim to integrate the presented approach with related work to allow for even more comprehensive analysis.

Folksonomy Graph Visualizations

Though tag clouds give a good impression of the folksonomy vocabulary, a folksonomy is more than the sum of its tags. As we know from Chapter 2, the ‘heart’ of folksonomies are not the tags but the taggings that link the sets of resources, tags, and users. This core structure can be represented as a hypergraph, as it was detailed in Section 2.4.

However, hypergraphs are rather cumbersome to visualize and difficult to understand for users [177, 188]. We thus need to reduce the hypergraphs to simpler graph structures that represent relevant parts of the folksonomy. Depending on what we are interested in, different subgraphs can be derived from the folksonomy hypergraph. We call this process the *decomposition of the folksonomy hypergraph* and will describe it in the following. We will then use the resulting subgraphs to classify existing work on visualizing folksonomies as graphs. With the ChainGraph, we will finally show our own development for the visualization of folksonomy graphs and similar data structures.

6.1 Folksonomy Hypergraph Decomposition

Figure 6.1 shows a complete decomposition of the non-uniform and three-uniform hypergraphs discussed in Section 2.4.2. Compared to the examples

Parts of this chapter have already been published in [121], [168], and [174].

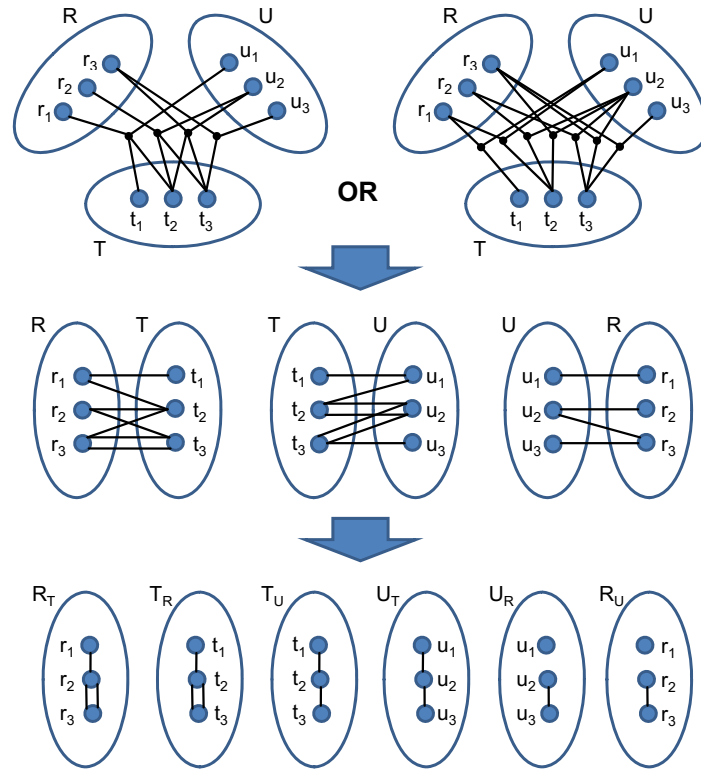


Figure 6.1 — Complete decomposition of the folksonomy hypergraph (which is identical for the non-uniform and three-uniform variant due to Axiom 2).

from that section, the hypergraphs have some more edges this time to better illustrate the decomposition. However, both hypergraphs contain the same information due to Axiom 2, stating that one and the same resource can only be tagged once by each user (see Section 2.4). They are first split into three bipartite graphs $G(RT)$, $G(TU)$, and $G(UR)$ that can then each be further reduced into a pair of unipartite graphs, i.e., finally result the six graphs $G(R_T)$, $G(T_R)$, $G(T_U)$, $G(U_T)$, $G(U_R)$, and $G(R_U)$ (cp. Figure 6.1).

Note that each of the resulting graph pairs has the same structure, i.e., the edges in graphs $G(R_T)$ and $G(T_R)$, $G(T_U)$ and $G(U_T)$, as well as $G(U_R)$ and $G(R_U)$ are connected to the same vertices. More interestingly, each element set is represented by two graphs that are based on different link types: While $G(R_T)$ links the resources based on the associated tags, $G(R_U)$ links the resources by the users who annotated them. Likewise, $G(T_R)$ and $G(T_U)$ each provide a different perspective on the tag set, while $G(U_T)$ and

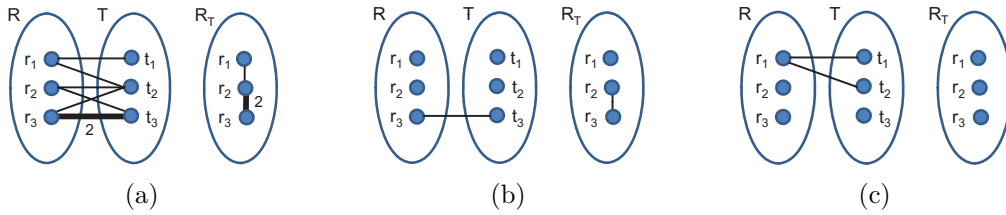


Figure 6.2 — Folksonomy subgraphs with (a) weighted edges instead of multi-edges, (b) edges filtered by weight, (c) edges restricted to a single user.

$G(U_R)$ describe different types of links in the set of users. These subgraphs nicely reveal the implicit links of tagging we introduced in the context of the three-layer model in Section 2.4.

The folksonomy subgraphs can be further reduced for the visualization. A common reduction would be the transformation into weighted graphs by merging the multi-edges, as illustrated in Figure 6.2a for the example graphs $G(RT)_{ex}$ and $G(R_T)_{ex}$ from Figure 6.1. The edge weights represent the number of links that exist between two vertices. Usually, such a transformation significantly reduces the number of edges, leading to a more compact and readable graph visualization.

The weighted graphs can again be further reduced. For instance, we can specify a threshold and hide all edges in the visualization that have a weight below this threshold. Figure 6.2b illustrates this again for the $G(RT)_{ex}$ and $G(R_T)_{ex}$ subgraphs by showing only edges with a weight > 1 . Alternatively, we can rank the edges by weight and visualize exclusively the top k edges. This is similar to idea of tag clouds where only a certain number of most frequently used tags is shown.

Finally, further reductions can be performed by restricting the subgraphs to specific elements, as described for the folksonomy quadruple in Section 2.4.2. Restricting the $G(RT)_{ex}$ subgraph to the taggings of user u_1 results in the graph $G(RT)_{ex_{u_1}}$ and the subgraph $G(R_T)_{ex_{u_1}}$ illustrated in Figure 6.2c.

6.2 Review of Existing Graph Visualizations

We can use the subgraphs resulting from the decomposition to categorize related work in this area. Further useful categories are the folksonomy source

and the graph layout used for the visualization. Another more specific category is the way in which tag frequencies are visualized. Apparently, this category can only be applied to those subgraphs that contain tags (i.e., graphs $G(RT)$, $G(TU)$, $G(T_R)$, and $G(T_U)$).

In the following, we use these four categories to characterize existing graph visualizations of folksonomies we found in the literature and on the web. Table 6.2 lists the reviewed approaches, ordered by the type of folksonomy subgraph they visualize. As shown by the table, most of the work visualizes the $G(T_R)$ graph or its restricted subgraph $G(T_R)_u$. These visualizations are related to tag clouds as they show the set of tags, but they additionally display relations between the tags based on their co-occurrence (or co-use in case of the $G(T_U)$ graph). Some of the visualizations even adapt the tag sizes to their frequencies of use, as in tag clouds. There is also a visualization for the $G(RT)$ graph, but we could not find any visualization for the other folksonomy subgraphs we derived in Section 6.1.

Visualizations of Subgraph $G(T_R)$

Visualizations of the $G(T_R)$ graph are provided by Eaton [86] and Halpin et al. [111]. Both display folksonomy data from DELICIOUS [6] and draw the graphs in a force-directed layout. Eaton’s DELTAGS tool lets the user enter a tag and parses the “related tags” part of the HTML page which is generated for that tag by DELICIOUS. It then draws a force-directed graph from these related tags using the TOUCHGRAPH library for Java. A screenshot of the resulting graph visualization is shown in Figure 6.3a.

DELTAGS provides several parameters that enable the user to customize the visualization. For instance, one can specify a search radius (i.e., the number of edges traversed), limit the search results (i.e., the number of edges shown), or restrict the displayed nodes by the number of incoming edges. Unfortunately, DELTAGS is not working with the current version of DELICIOUS and is not documented, so that we cannot further examine the approach.

The $G(T_R)$ graph visualized by Halpin et al. [111] consists of 15 tags from DELICIOUS that are all related to the tag “complexity” (including the tag “complexity” itself). The tags are visualized as circle nodes, with the node size indicating the tags’ usage frequencies, similar to the font size in tag clouds. The edges are weighted by a cosine distance measure which calculates

Name	Author(s)	Folks. subgraph	Data source	Graph layout	Tag frequencies	Software (library)
Deltags	Eaton	$G(T_R)$	Delicious	Random force	Not shown	Java (TouchGraph library)
Inter-tag Correlation Graph	Halpin et al.	$G(T_R)$	Delicious	Random & radial force	Node size	Pajek
Delicious Soup	Zitvogel	$G(T_R)_u$	Delicious	Grid & radial force	Node size	Flash
Elastic Tag Maps	Stefaner	$G(T_R)_u$	Delicious	Elastic map	Font size	Flash
Co-occurrence Network	Michlmayr & Cayzer	$G(T_R)_u$	Unknown	Random force	Not shown	Java (JUNG framework)
Map of Tags	Shaw	$G(T_U)$	Delicious	Semidef. embedding	Node & font size	Unknown
TagGraph	Oliphant	$G(RT), G(RT)_u$	Flickr	Random force	Not shown	Flash (SpringGraph component)

Table 6.2 — Reviewed folksonomy graph visualizations.

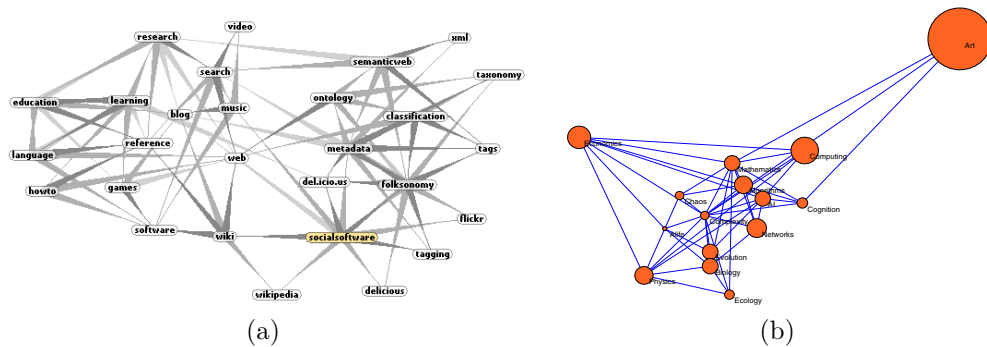


Figure 6.3 — Visualizations of subgraph $G(T_R)$ by (a) Eaton [86] and (b) Halpin et al. [111].

the similarity of tags by their degree of co-occurrence. This similarity is expressed by the distance of connected tag nodes in the graph visualization.

Halpin et al. used the network analysis and visualization tool PAJEK [17] to create the graph visualizations. They created two different types of graphs, one showing only direct connections between the tag “complexity” and the 14 related tags in a radial layout, and the other considering 30 additional connections that reached the highest co-occurrence values in the investigated set of tags. Figure 6.3b shows the latter visualization created with the Kamada-Kawai algorithm [137] for force-directed placement. Note that the visualizations of Halpin et al. are generated for illustrative purposes and do not provide any interaction possibilities.

Visualizations of Subgraph $G(T_R)_u$

This is different in the interactive visualizations of Zitvogel [257] and Stefaner [229, p. 56ff.]. Initially, their visualizations show only the tag nodes, with the node or font size logarithmically scaled to the tag frequency, like in tag clouds. The relations of a tag with other tags are shown on demand if the user selects the tag. On the one hand, this reduces the number of visible edges and hence visual clutter, but on the other hand, it does not allow for analytical tasks which require to view the relations between all displayed tag nodes at a time, such as the visual identification of highly connected nodes.

Both Stefaner and Zitvogel use data from DELICIOUS again, though only from single user accounts, i.e., they visualize the restricted subgraph $G(T_R)_u$.

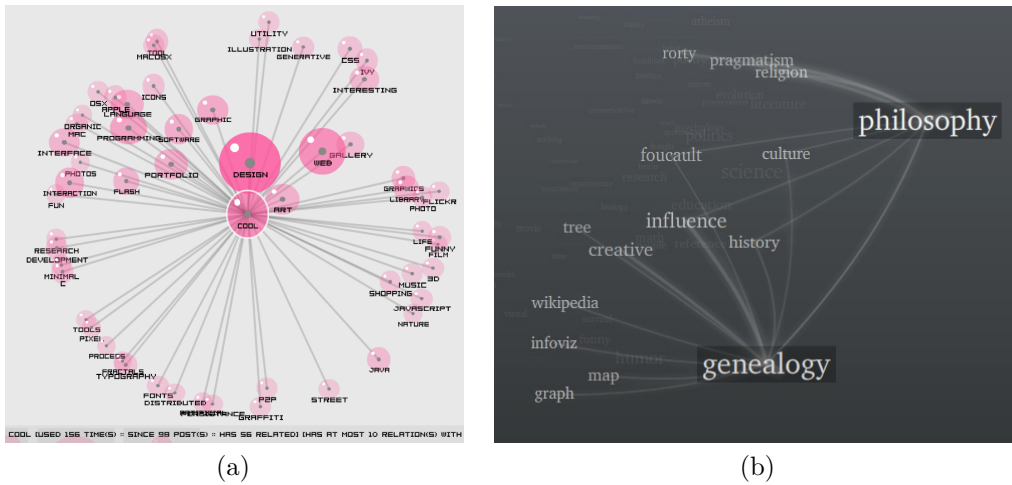


Figure 6.4 — Visualizations of subgraph $G(T_R)_u$ by (a) Zitvogel [257] and (b) Stefaner [229].

Zitvogel’s tool, called DELICIOUS SOUP, visualizes the tag nodes as animated bubbles of different sizes that correlate to the tags’ usage frequencies (see Figure 6.4a). Hovering over a node displays the frequency value at the bottom of the visualization, along with information on the age of the tag (i.e., since how many taggings it is used) and the number of related tags (i.e., tags that have been used together with the tag). In addition, single nodes or groups of related nodes can be marked with different colors.

Initially, Zitvogel uses a grid layout with random order that is similar to a *sequential tag cloud* described in Section 5.2.1. If the user selects a tag, the grid changes to a radial layout that is comparable to the second visualization of Halpin et al. mentioned above. The distances of the radially arranged nodes from the selected node in the center represent the relatedness of the tags. However, it remains unclear how this relatedness is calculated and how the radial layout is generated, as the work also misses a documentation.

Stefaner uses a dimensionality reduction method called *elastic maps* to layout the $G(T_R)_u$ graph, in combination with a *principal components analysis (PCA)* [229, p. 56ff.]. In addition, he performs a *curvilinear component analysis (CCA)*, resulting in a visualization that places frequently co-occurring tags close to each other and looks roughly like the inverse of a *circular tag cloud* (see Section 5.2.1). Like in tag clouds, the tag nodes are simply text labels with a font size that scales logarithmically to the tag’s usage

frequency. The frequency values are given at the bottom of the visualization in a bar chart that nicely depicts the power law distribution that can often be found in folksonomies [111].

In contrast to the approach of Zitvogel, one can select multiple tags at a time in the work of Stefaner to display their relations (see Figure 6.4b). The relations are ranked according to different criteria, while only those with the highest rank are displayed. The demo implementation is restricted to 14 pre-processed user accounts, each consisting of 150 tags.

A third visualization of subgraph $G(T_R)_u$ is presented by Michlmayr and Cayzer [187]. Like in the visualizations of Halpin et al., they use a force-directed layout where the node distance is roughly in inverse proportion to the number of tag co-occurrences, although they do not adapt the font size of the tags to their frequencies counts. Furthermore, they rank the tag relations and display only the top-ranked ones with their adjacent nodes to reduce the size and density of the graph. This is similar to the approach of Stefaner with the difference that Michlmayr and Cayzer base their rankings purely on the co-occurrence frequency. An example for the resulting “co-occurrence network” is shown in Figure 6.5a. It has been generated with the *Java Universal Network/Graph Framework (JUNG)* using curved edges and the force-directed algorithm of Fruchterman and Reingold [95].

A special focus of the work was on visualizing the evolution of a user’s tag set. Michlmayr and Cayzer generated an animation where the tags are iteratively added to the graph in the same order as they have been created by the users of the tagging system. The graph visualization moves upwards with every iteration, leading to a ‘river’ metaphor similar to the one presented in [26, 83]. In addition, they applied an evaporation technique to their visualization that let the edge weights decrease with the age of the tags.

Visualization of Subgraph $G(T_U)$

The $G(T_U)$ graph represents the links that result if different users assign the same tag, and not if the same tag is used for different resources, as for subgraph $G(T_R)$. We could only find one single approach that visualizes this type of graph. It has been created by Shaw [221] from DELICIOUS data consisting of 200 tags from nearly 1500 users. A visualization of 100 tags from this dataset is shown in Figure 6.5b. Shaw uses weighted nodes and *semidefinite embedding* for dimensionality reduction. This results in a

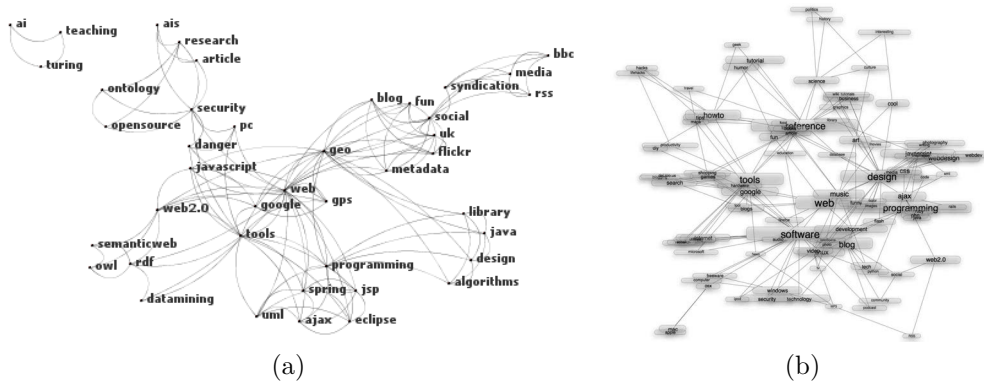


Figure 6.5 — Visualizations of subgraphs $G(T_R)_u$ and $G(T_U)$ by (a) Michlmayr & Cayzer [187] and (b) Shaw [221].

statistically grouped visualization that bears a resemblance to *clustered tag clouds* (see Section 5.2.1), if enriched with tag relations. The tag distances have been calculated using the *Kullback-Leibler divergence*. The main focus of the work was on finding an optimal connectivity matrix for the graph visualization. Unfortunately, the tag nodes overlap in the visualization, so that many of them are not readable. This is also not significantly increased by making the nodes slightly transparent (cp. Figure 6.5b).

Visualizations of Subgraphs $G(RT)$ & $G(RT)_u$

The $G(RT)$ graph does not only consider tags and their relationships but also the resources the tags are connected with (see Section 6.1). Also for this graph type, we found only one work that visualizes it. It is a tool created by Oliphant [200] that uses a force-directed layout to visualize tag relations between images from FLICKR. Figure 6.6a shows an example of a part of a graph generated with the tool. The resource nodes are thumbnails of images that are interconnected by the tags they share. The graph is iteratively created by the user, starting with a certain tag of interest. This tag becomes the central graph node and a number of images indexed with it are arranged radially around it. Users can add more images and/or tags to the graph. If they select one of the tags, it becomes the focus and is turned into the central graph node. This way, it is possible to iteratively create an $G(TR)$ graph visualized in a force-directed layout.

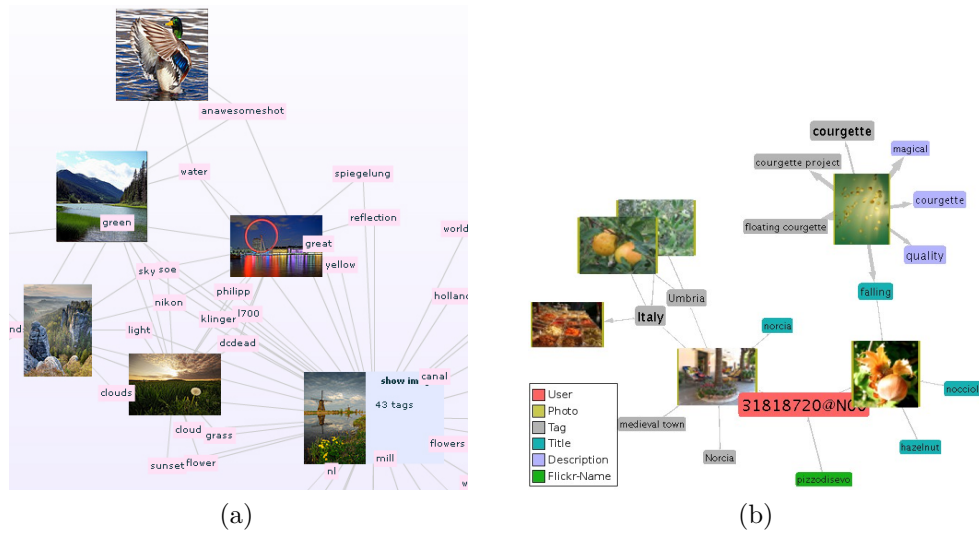


Figure 6.6 — Visualization of (a) subgraph $G(RT)$ by Oliphant [200] and (b) extended folksonomy graph by Kern et al. [140].

Other Visualizations of Folksonomy Graphs

There are related approaches that use simple visualizations of the $G(T_R)$ graph to depict *tag hierarchies* [128] or *tag clusters* [46, 224] derived from folksonomies. Others work extends folksonomy graphs by additional elements. For instance, FOLKSOVIZ visualizes tag hierarchies by utilizing structural information from Wikipedia [156]. As another example, Kern et al. [140] extend folksonomy graphs with additional metadata associated with images, such as the title or a description of the image. They use FLICKR as data source and apply spreading activation techniques to query the graph. That way, the images are not only connected by tags but also by user IDs and image titles, among others (see Figure 6.6b). Finally, there are also graph visualizations of the social network underlying tagging systems, such as for FLICKR [249] or DELICIOUS [87].

Discussion

The reviewed graph visualizations reach their limits in cases where resources are highly interconnected with tags, as in many folksonomies. In these cases, the following two drawbacks of graph visualizations can often be observed [247] (cp. Figure 6.3 to 6.6):

- Tag relations are partly hard to visually spot and track, making it difficult to figure out how the tags and/or resources are interrelated.
- High graph densities and many crossing edges additionally impede the visual tracking of tag relations.

These drawbacks can only in part be solved by optimizing the graph drawings. A more effective solution rather needs to take the preceding step, the generation of the graph representation, into account. We thus developed the *ChainGraph*, a graph visualization that uses an alternative representation to overcome the mentioned limitations. As we will show in the following, it particularly supports the visual exploration of tag relations in folksonomy graphs. We will focus on the bipartite folksonomy subgraph $G(RT)$. However, ChainGraphs can also be used to represent other folksonomy subgraphs, such as $G(UR)$ or $G(TU)$.

6.3 The ChainGraph Visualization

The ChainGraph approach is based on the idea of multiplying certain nodes in the graph representation. In this case, all tags that are associated with more than two resources are represented by several nodes in the graph so that all tag relations become binary. Accordingly, the ChainGraph forms an aggregation of several *simple paths* that we call ‘chains’ and that require a special interpretation due to the node multiplication.

The basic idea of the ChainGraph is best illustrated by comparing it with the straightforward way of visualizing a folksonomy graph. Figure 6.7a depicts such a common graph that is similar to the one by Oliphant introduced above (see Section 6.2). Like the visualization of Oliphant, it depicts tag relations between images as they could be described by a folksonomy. Figure 6.7b shows a ChainGraph that visualizes exactly the same folksonomy subgraph. However, in contrast to the common graph visualization, all tag nodes along one and the same chain represent the same tag, not different ones, this time. Vice versa, all resources that are interconnected by the same tag chain are related with each other via this tag. The ordering of the resource nodes in the chains is irrelevant for this interpretation, since all relations within a chain are transitive. Reordering the nodes in the chains has thus no effect on the semantics of the visualization; however, it affects its readability, as we will detail in Section 6.3.2.

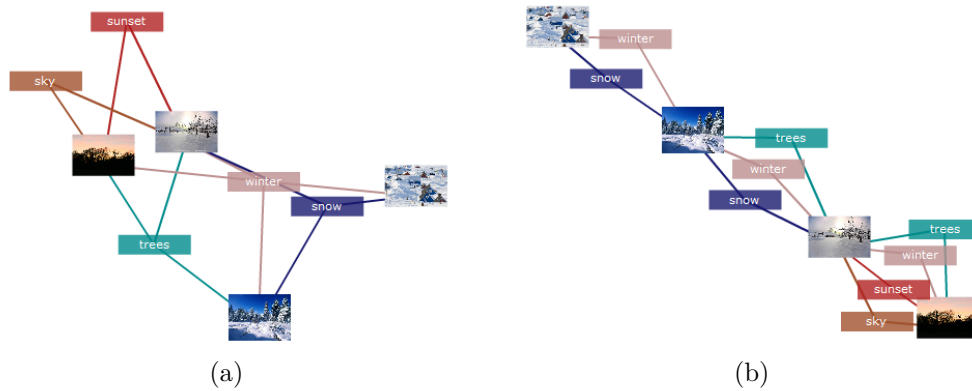


Figure 6.7 — (a) Common graph and (b) ChainGraph visualization of the same folksonomy subgraph.

A comparison of the two types of graph visualizations in Figure 6.7 reveals that the ChainGraph overcomes the above-mentioned limitations of common graph visualizations of folksonomies by having the following advantages:

- Tags are arranged in chains running in parallel between the resources they connect, making the number and kind of tag relations immediately visible.
- The visual tracking of tag relations is facilitated by a generally lower graph density and fewer edge crossings than in the common graph visualization.

These advantages hold true for nearly every ChainGraph. However, besides a coloring and force-directed placement, an optimal ordering of the resource nodes in the chains is crucial to get good results. This is one issue we will discuss in the following. We will also discuss limitations of the ChainGraph that result from the node multiplication and the special interpretation of tag relations. First, we will describe the general process of creating a ChainGraph.

6.3.1 Creating a ChainGraph

To create a ChainGraph for the folksonomy subgraph $G(RT)$, we take the set of resources $R = \{r_1, \dots, r_l\}$ and the set of tags $T = \{t_1, \dots, t_m\}$, as we defined them in Section 2.4. In addition, we have a function $f : R \rightarrow \mathcal{P}(T)$ that represents the links between the resources and tags in the folksonomy

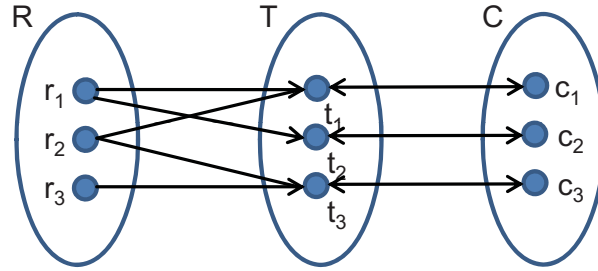


Figure 6.8 — Resources are linked to tags which are bijectively mapped to chains.

subgraph. From this input, a set of chains $C = \{c_1, \dots, c_n\}$ can be built. Each chain $c \in C$ represents one tag by the mapping $g : C \longleftrightarrow T$, as illustrated in Figure 6.8, and contains an ordered set of resources that are linked with this tag, such that:

$$\forall c \in C, r \in R : g(c) \in f(r) \Leftrightarrow r \in c \quad (6.1)$$

The order of the resources in each chain is defined by a relation o that is irreflexive, asymmetric, and transitive, thus describing a strict total order. For a certain chain c_x , the set of all possible orders $O_x = \{o_{x_1}, \dots, o_{x_m}\}$ contains $m = |c_x|!$ different orders, whereas each chain can only be in one order $o_x \in O_x$ at a time (i.e., sorted one way). The initial orders of the resources in the chains C are randomly chosen and form the initial configuration $\delta_{init} = (o_1 \in O_1, \dots, o_n \in O_n)$.

Algorithm 6.1 describes the generation of the ChainGraph. All resources in R are represented by resource nodes that are connected via edges with tag nodes according to their orders in the chains.

6.3.2 Optimizing the Order in the Chains

The order of the resources in the chains is highly relevant for the readability of the resulting ChainGraph visualization. Figure 6.9 shows two alternative ChainGraphs generated for the same folksonomy subgraph $G(RT)_x$. Whereas the resources are ordered in a clear and consistent manner in the left ChainGraph, the right one is comparatively hard to read. The difference is only caused by the ordering of the resources in the chains, the configuration, which we call $\delta_{optimal}$ for the left ChainGraph and δ_{random} for the right.

```

1:  $Nodes = R$ 
2:  $Edges = \emptyset$ 
3: for all  $c \in C$  do
4:    $r_{prior} = null$ 
5:   for all  $r \in c$  do
6:     if  $r_{prior} \neq null$  then
7:        $newTagNode = newNode(g(c))$ 
8:        $Nodes = Nodes \cup newTagNode$ 
9:        $newEdge1 = (r_{prior}, newTagNode)$ 
10:       $newEdge2 = (newTagNode, r)$ 
11:       $Edges = Edges \cup newEdge1 \cup newEdge2$ 
12:     end if
13:     $r_{prior} = r$ 
14:   end for
15: end for
16:  $ChainGraph = (Nodes, Edges)$ 

```

Algorithm 6.1 — Generation of the ChainGraph from a set of chains.

The general rule is that the more similar the orders in the chains of a configuration, the better the resulting ChainGraph visualization. For example, the two chains c_3 and c_4 in the left ChainGraph connect the resources r_1 , r_3 , and r_4 both in the order (r_4, r_3, r_1) , whereas in the right ChainGraph, the orders of the resources in c_3 and c_4 differ, with $c_3 = (r_1, r_2, r_3, r_4)$ and $c_4 = (r_3, r_1, r_4)$. Due to the different orders of the chains, nodes in the right visualization are rather crosswise than parallel connected what limits the benefits of the ChainGraph.

To avoid these problems, an algorithm is needed that determines the optimal orders in the chains, the configuration $\delta_{optimal}$, and thus generates a ChainGraph representation that leads to an optimal visualization. However, we first have to define a similarity measure that allows us to compare the orders of two arbitrary chains.

Similarity Measure

Basically, the similarity of two chains depends on a) the resources they connect, and b) the order in which they connect these resources. To compute a unique value for the similarity of two chains, we first transform each chain into an isogram over the alphabet R with each resource being a letter and the orders of the letters in the isograms being given by the orders of

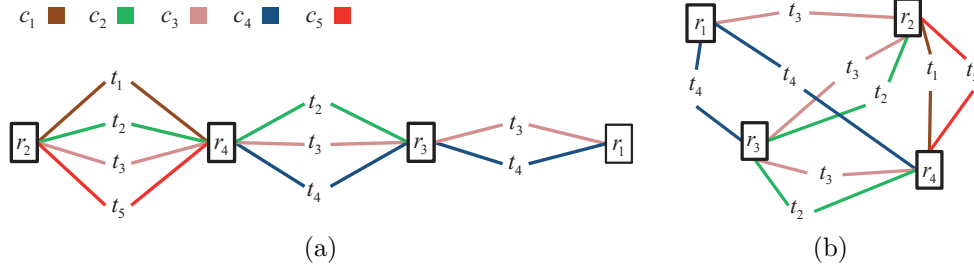


Figure 6.9 — An (a) optimal ChainGraph differs from a (b) random one in the ordering of the resources in the chains.

the resources in the chains. To give an example, the isograms for the two chains $c_3 = (r_1, r_2, r_3, r_4)$ and $c_4 = (r_3, r_1, r_4)$ from the random configuration δ_{random} of Figure 6.9 are $iso(c_3) = r_1r_2r_3r_4$ and $iso(c_4) = r_3r_1r_4$.

Having the chains transformed into isograms allows us in a second step to apply the *Levenshtein distance* to compute the similarity between these chains. The Levenshtein distance defines the minimum number of edit operations needed to transform one isogram into another [157]. Possible edit operations are *insertion*, *deletion*, and *substitution* of single letters. For instance, the Levenshtein distance between $iso(c_3) = r_1r_2r_3r_4$ and $iso(c_4) = r_3r_1r_4$ is three, since the following three edit operations change $iso(c_3)$ into $iso(c_4)$, and there is no way to do it with fewer than three edit operations:

1. $iso(c_3) = r_1r_2r_3r_4 \rightarrow r_1r_3r_4$ (deletion of r_2)
2. $r_1r_3r_4 \rightarrow r_1r_4$ (deletion of r_3)
3. $r_1r_4 \rightarrow r_3r_1r_4 = iso(c_4)$ (insertion of r_3)

The isograms of the analogous chains c_3 and c_4 from the optimal configuration $\delta_{optimal}$ of Figure 6.9 require only one edit operation:

1. $iso(c_3) = r_2r_4r_3r_1 \rightarrow r_4r_3r_1 = iso(c_4)$ (deletion of r_2)

An overall similarity measure *sim* for the orders of all chains C , with $|C| = n$, in a certain configuration δ_x can therefore be computed on the basis of the sum of all Levenshtein distances *ld* between all pairs of isograms, as described in the following equation:

$$sim(\delta_x) = \frac{1}{1 + \sum_{k=1}^{n-1} \sum_{l=k+1}^n ld(iso(c_k), iso(c_l))} \quad (6.2)$$

Accordingly, two or more tag chains that connect exactly the same resources in exactly the same order have a similarity of 1, since the sum of their Levenshtein distances is 0.

Exact Solution

An exact solution for the optimal configuration $\delta_{optimal}$ of a ChainGraph C_x requires the computation of the similarity measures between all chains of all possible configurations. The complexity of this computation can be approximated as follows:

Given a set of chains C , with $|C| = n$, and a set of resources R , with $|R| = l$, the maximum number of different configurations is $n * l!$, since each resource can at most occur once in each chain and hence l defines the maximum number of resources in a chain.

For the similarity measure sim of a certain configuration δ_x , n^2 Levenshtein distances have to be computed. Computing the Levenshtein distance between two isograms of maximal length l has $\mathcal{O}(l^2)$ complexity [223]. The complexity of computing an exact solution for an optimal configuration $\delta_{optimal}$ can thus be approximated by: $\mathcal{O}(n * l! * n^2 * l^2) = \mathcal{O}(n^3 * l! * l^2) \approx \mathcal{O}(n^3 * l!)$.

Hence, the computational complexity grows polynomially with the number of chains n (i.e., with the number of tag relations) and factorially with the maximum length of the chains l (i.e., the maximum number of resources that are linked with the same tag) in worst case. A heuristic solution that speeds up the computation by approximating an optimal ChainGraph visualization in reasonable time is therefore highly desirable. In the following, we present such a heuristic solution that we developed specifically for the ChainGraph.

Heuristic Solution

The heuristic solution uses an alternative approach to create the ChainGraph. It adds the resources R stepwise to the chains C , as described in Algorithm 6.2. Two basic decisions are required in every step: Which resource to add next and where to add it. In an optimal visualization, we want resources that share many tags to be positioned as close as possible to each other.

Accordingly, we start the graph building with the two resources that share most tags (see Algorithm 6.2, lines 1 and 2).

```

1:  $FirstPair = \{a, b \in R \mid \forall c, d \in R : (|f(a) \cap f(b)| \geq |f(c) \cap f(d)|)\}$ 
2:  $addToChains(FirstPair, C)$ 
3:  $Unbound = R - FirstPair$ 
4: while  $Unbound \neq \emptyset$  do
5:    $next = null$ 
6:    $maxConstraintLevel = -1$ 
7:   for all  $r_x \in Unbound$  do
8:     if  $constraintLevel(C, r_x) > maxConstraintLevel$  then
9:        $next = r_x$ 
10:       $maxConstraintLevel = constraintLevel(C, r_x)$ 
11:     end if
12:   end for
13:    $addToChains(next, C)$ 
14:    $Unbound = Unbound - next$ 
15: end while

```

Algorithm 6.2 — Stepwise creation of the chains.

Next to the number of shared tags, some additional aspects need to be considered when determining the order in which the resources are added to the chains. We thus define a variable *constraintLevel* that takes these issues into account and weights the selection priority of the resources accordingly. Given a set of chains C , the *constraintLevel* of a certain resource r_x is computed by the following formula:

$$constraintLevel(C, r_x) = \frac{numSharedTags(C, r_x)}{(numConnRes(C, r_x) + |Alternatives(C, r_x)|)^2} \quad (6.3)$$

The formula is based on three heuristic strategies:

1. *numSharedTags*: If a resource r_x shares many tags with the resources that are already added to the chains C it would produce many new edges (i.e., more shared tags = higher constraint level).
2. *numConnRes*: If a resource r_x would be connected to only a few of the already added resources it produces a more parallel arrangement of the chains (i.e., fewer connected resources = higher constraint level).

3. *Alternatives*: If there exist only few alternatives to connect a resource r_x to the already added resources it should be added as early as possible (i.e., fewer connection alternatives = higher constraint level).

In each generation step, the *constraintLevel* is calculated for all resources that have not yet been added to the chains C (see Algorithm 6.2, lines 7-12). The resource with the highest *constraintLevel* is added next (line 13). The resource is added in a way that it is connected to as few of the already drawn resources as possible.

The computational complexity of this solution can be approximated as follows: The complexity of the two loops in Algorithm 6.2 (the while-loop in line 4 and the for-loop in line 7) is in $\mathcal{O}(l^2)$ (with $l = |R|$). Getting the *constraintLevel* for a certain resource r_x requires the computation of *numSharedTags*(C, r_x), *numConnRes*(C, r_x), and *Alternatives*(C, r_x) that together have a complexity of $\mathcal{O}(n * l)$ (with $n = |C|$). The overall complexity of computing the heuristic solution can thus be approximated by $\mathcal{O}(l^2 * n * l) = \mathcal{O}(n * l^3)$.

Compared to the exact solution, the complexity does not grow polynomially but linear with the number of chains n and not factorially but polynomially with the number of resources l in the heuristic solution. This results in a much lower complexity to compute an optimal ChainGraph. However, it has to be noted that the resulting configuration is possibly only a local optimum.

6.3.3 Use Case

We implemented an interactive ChainGraph visualization for an example graph $G(RT)_{ex}$. It simulates a folksonomy subgraph as it could ideally result from tagging on a photo sharing website like FLICKR [8]. For instance, it could be the result of a query that asks for images indexed with the tags ‘paris’ and ‘france’. As these two tags would be associated with all retrieved images, they do not provide any value for the visualization and are hence not part of the ChainGraph. Instead, the implemented ChainGraph visualizes all other tags that are associated with at least two of the retrieved images. Tags that are associated with only one image are not included in the visualization, as they do not represent tag relations.

We implemented the application example in APACHE FLEX [3] (formerly ADOBE FLEX) using the SPRINGGRAPH component [222] to draw the force-directed layout. The resource nodes are thumbnails of the images. They

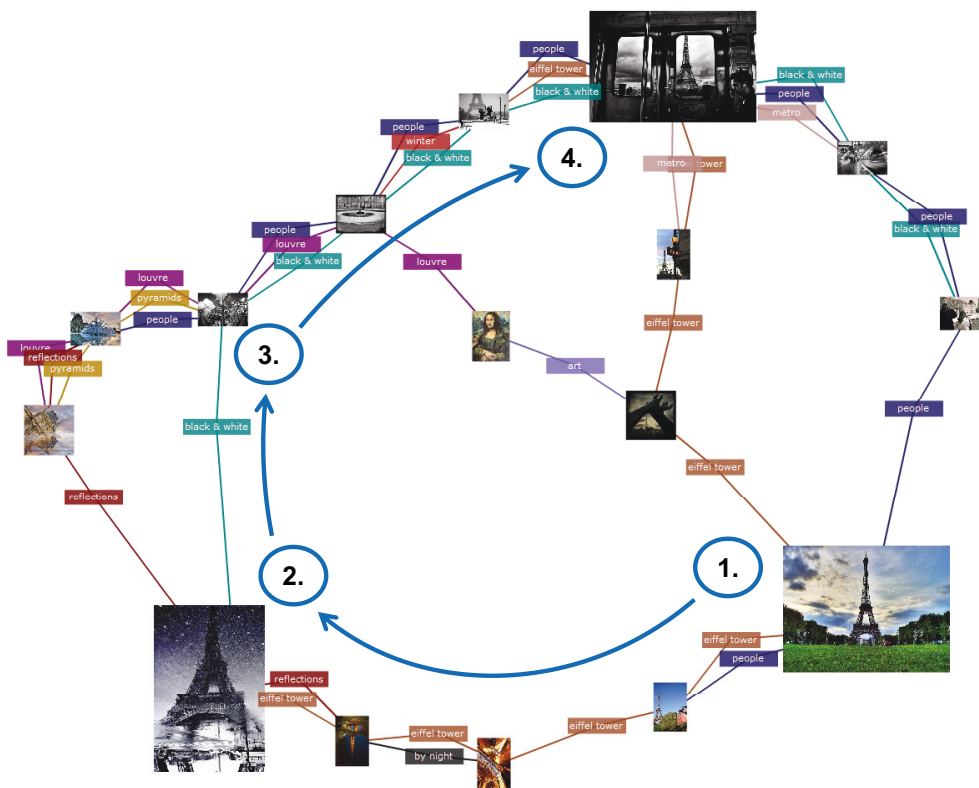


Figure 6.10 — Using the ChainGraph to visually explore tag relations between images of Paris.

can be moved by drag-and-drop and clicked on to enlarge. The tag nodes connect the resource nodes in chains according to the ChainGraph idea. Figure 6.10 shows a screenshot of the implementation.¹

Assume a user is searching for a decorative image of Paris that she would like to use as an illustration for a text about the French capital. She first glances at the image that shows the Eiffel Tower in summer (see Figure 6.10, step 1). After enlarging this image, she is not satisfied as it looks a bit boring in her opinion. So she continues her search by following the chain labeled with the tag “eiffel tower”. Another image of the Eiffel Tower catches her attention and she enlarges it (step 2). She likes the monochromatic style of the image and decides to look for further images of this kind. Thus, she now follows the “black & white” chain. She recognizes that another chain,

¹ A demo of the ChainGraph implementation is publicly available at <http://chaingraph.visualdataweb.org>.

labeled with “people”, meets the “black & white” chain and runs in parallel with it (step 3) – she is on the ‘right path’, since people on an image help to make it lively and interesting what is in line with her search goals. Finally, she reaches a black-and-white image that shows the subway of Paris with passengers inside (step 4). Since the image belongs also to the “eiffel tower” chain, this symbol of Paris is shown on the image as well, visible in the background through the window of the subway. After enlarging the image, she decides that it perfectly meets her needs and takes it as a decoration of her text about Paris.

As illustrated by this scenario, the ChainGraph can provide helpful assistance in *exploratory search* processes [250]. Relatively vague user needs can be iteratively refined by discovering and following certain chains of interest. If the same set of tags has been associated with several resources, their chains tend to run in parallel, making it easy for users to explore related resources and to select the ones that fit best with their needs.

The use case illustrates how the chains can serve as ‘paths’ for visual exploration. According to this metaphor, resource nodes are ‘crossroads’ where one path can be left in favor of another. Similar to the idea of *information scent* [63], users keep switching paths (i.e., chains) until they finally reach the resource they are looking for.

The use case gives also some idea of the manifold possibilities to tailor the ChainGraph visualization. Besides the variety of coloring a ChainGraph, interactive functionality, such as filtering or highlighting, can additionally support the exploratory search process. The most appropriate implementation depends, of course, on the needs of the use case that the ChainGraph is applied in.

6.3.4 Evaluation of the ChainGraph

We performed a user study in which we compared the ChainGraph with the common graph visualization. Our goal was to get some insights into the user acceptance, understandability, and effectiveness of the ChainGraph as a visualization of folksonomies.

Experimental Design and Material

We generated three pairs of graph visualizations for the experiment, using the heuristic optimization approach for the creation of the ChainGraphs (see Section 6.3.2). Each pair consisted of one ChainGraph and one common

	1	2	3	4	5	6
A	x	x		x		
B		x	x	x	x	
C	x					x
D				x	x	x
E	x	x	x		x	
F			x		x	

(a)

	1	2	3	4	5	6
			x	x	x	
B	x	x		x		x
C	x	x	x		x	x
D	x		x	x		x
				x	x	
F	x	x	x	x	x	x

(b)

	1	2	3	4	5	6	
x	x	x	x	x	x	x	A
	x	x	x	x	x	x	B
x	x	x	x	x	x	x	C
x			x	x	x	x	D
x	x		x		x		E
x	x	x		x	x		F

(c)

Table 6.3 — Connectivity matrices for the three graphs $G(RT)_i$ of the experiment with (a) 18, (b) 24, (c) 30 tag relations. The numbers in the table heads indicate the tags, the letters in the leftmost and rightmost columns indicate the resources.

graph visualizing the same folksonomy subgraph $G(RT)_i$. The various tag chains were differently colored for better distinction. We kept the sets of resources and tags for all three pairs the same ($|R| = |T| = 6$) but gradually increased the density of the graphs $G(RT)_i$ (i.e., the number of the tag relations). Note that this means also an increase in density for the common graphs, while it leads to an increase in the total number of nodes for the ChainGraphs.

The minimum number of tag relations that is required to connect all six resources of the graph is twelve (if each of the six tags connects exactly two resources). However, including this extreme case in the evaluation would be useless, since both graph visualizations (ChainGraph and common graph) would look identical (i.e., both would form a *cycle graph*). The inclusion of a maximally connected folksonomy subgraph, where all resources are connected to all other resources via all tags, would also unnecessarily restrict the experimental design. We thus decided to create more realistic cases, consisting of three folksonomy subgraphs $G(RT)_i$ containing 18, 24, and 30 tag relations respectively. The tag relations were randomly distributed in a way that every resource was connected to at least one tag and every tag to at least two resources (as we were only interested in tag relations, not in tags that are associated with only a single resource). Table 6.3 shows the connectivity matrices for the three graphs $G(RT)_i$ we created.

We defined three user tasks in accordance with the goals of the ChainGraph approach:

1. Give the pair of resources that shares most tags. Also give the number of tags that is shared by that pair of resources.
2. Give all tags that are shared by resources x and y .
3. Give all tags that are shared by resources x , y , and z .

The variables x , y , and z were replaced by letters indicating resources from the folksonomy subgraphs $G(RT)_i$ (cp. Table 6.3). These resources were randomly selected for every single task and pair of graph visualizations.

In sum, we thus applied a 2x3x3 within-subjects design with variables *graph type* (common graph vs. ChainGraph), *task type* (task 1, 2, 3), and *graph density* (18, 24, 30 tag relations).

Procedure

Twelve participants with a mean age of 29 years (standard deviation (SD) = 6.2) took part in the experiment, mainly male PhD students with a computing-related background. The general familiarity with graphs was given with a mean value of 7.7 (SD = 2.6) on a ten-point scale (with 1 = “no familiarity” and 10 = “high familiarity”). All subjects reported normal or corrected-to-normal vision capabilities and no color blindness.

The graph visualizations were presented on a 17" TFT monitor with a screen resolution of 1280 x 1024 px. Both graph types were separately introduced and explained by an example. The graphs and tasks were displayed one at a time, and all graphs of one type were presented in series before the other type was shown. A change in the interpretation of the graph visualization was thus demanded only once during the experiment and was accompanied by an explanation.

We counterbalanced the order in which the graph types were presented to cancel out carryover effects: One group of participants started with the series of common graph visualizations, whereas the other group was first confronted with the ChainGraph visualizations. The presentation order of the tasks and graph visualizations within each series was the same and as given above, i.e., with tasks 1, 2, and 3 in this order and increasing graph densities (18, 24, 30) for each task. We did not change the order as we expected the difficulty of task completion to raise with the graph density, and wanted the same increase in difficulty for all subjects. While this likely resulted in learning effects *within* the experimental conditions (especially for the similar tasks 2 and 3), we did not consider these a problem, as we were

only interested in differences *between* the conditions. We rather considered it important to keep the order of the stimuli constant within both conditions to control learning effects and ensure comparability of the two conditions.

All participants had to fill out an evaluation sheet after each condition (i.e., after all graphs of one type have been presented). The graph types had to be rated according to 23 pre-defined items on a five-point Likert scale (with 1 = “strongly disagree” and 5 = “strongly agree”). The items were then mapped to aggregated values on the dimensions *effectiveness*, *understandability*, *attractiveness*, and *controllability* (accumulating 5-7 items each). The items and mappings are given in Table B.1 in the appendix of this thesis.

After completion of the experiment, the subjects had to compare both graph types in a questionnaire (the graphs were neutrally labeled as ‘Graph A’ and ‘Graph B’ throughout the whole experiment). Furthermore, we measured the time needed to complete the tasks and the accuracy of the solutions.

Results

The overall results of the user study are promising. Nine of the twelve participants preferred the ChainGraph visualization to solve the tasks of the experiment. The participants quickly understood how to interpret the ChainGraph and reported no serious difficulties in using it. They considered the larger sizes of the ChainGraph visualizations as a possible disadvantage but complained more about the many crossing edges in the common graph visualizations. They liked the parallel chain arrangement and the general ‘tidiness’ of the ChainGraph and considered the coloring helpful to solve the tasks.

The positive reception of the ChainGraph is also reflected by the ratings of the participants on the evaluation sheet. Table 6.4 list the average ratings on the four evaluation dimensions (higher value = better rating), along with the results from a Wilcoxon signed-rank test. We converted the ratings into ranks and carried out this non-parametric test due to the relatively small sample size of $N = 12$. As can be seen, the ChainGraph achieved slightly better results on all four evaluation dimensions, with significant differences in its perceived effectiveness and understandability ($p < 0.5$ and $p < 0.1$).

The positive ratings seem to stem especially from the first task of the experiment, as indicated by the time measures and the feedback of the study participants. Table 6.6 lists the average time and accuracy values, along with

	Co. Graph			ChainGraph		Wilcoxon	
	N	Mean	SD	Mean	SD	Z	p
Effectiveness	12	3.1	0.3	3.6	0.6	-2.1	<.05
Understandability	12	2.9	0.4	3.7	0.8	-2.8	<.01
Controllability	12	3.3	0.7	3.7	0.9	-0.9	>.05
Attractiveness	12	2.8	0.3	3.2	0.8	-1.4	>.05

Table 6.4 — Aggregated user ratings on the four evaluation dimensions.

results from corresponding Wilcoxon tests. The differences in the completion times for the first task, i.e., the identification of the pair of resources that shares most tags, are notable and significant for all three graph densities. This finding is supported by comments from the questionnaire: Two subjects reported difficulties in solving the first task with the common graph and another three subjects stated that the first task was much easier to solve with the ChainGraph. The accuracy values show a similar tendency for the first task, though no significant differences for the small sample of subjects.

The differences in the time and accuracy for the second and third task are not so clear. The ChainGraph performed better in some cases, the common graph in others – the common graph achieved even significantly better values in two cases. However, the differences are not systematic but both graph types reached relatively balanced results over all densities. Thus, the ChainGraph was neither better nor worse for these types of tasks regarding time and accuracy.

Overall, the ChainGraph was well received and positively evaluated by the study participants. It reached significantly better performance results than the common graph for the first task and similar results for the second and third. However, the tasks of the user study were rather narrow focused and served particularly the benefits of the ChainGraph approach. Also, the sample size was relatively small, and so were the presented graph visualizations. Future evaluations should include more complex user tasks and investigate the ChainGraph also in real use cases (in addition to controlled experiments), such as the one presented in Section 6.3.3.

		Time (in seconds)						Accuracy (in percent)						
Task	N	Connec.	Mean	SD	Mean	SD	Z	p	Mean	SD	Mean	SD	Z	p
1	12	18	89	49	27	21	-3.1	<.01	83	27	100	0	-1.9	>.05
		24	71	21	27	33	-2.4	<.05	75	32	89	26	-0.9	>.05
		30	73	32	36	22	-2..7	<.01	72	34	89	16	-1.3	>.05
2	12	18	20	9	21	7	-0.3*	>.05*	100	0	96	14	-1.0	>.05*
		24	17	6	24	8	-2.4*	<.05*	96	14	100	0	-1.0	>.05
		30	30	11	26	7	-1.8	>.05	83	14	95	9	-1.9	>.05
3	12	18	25	8	22	9	-1.2	>.05	100	0	100	0	0.0	>.05
		24	25	7	32	20	-1.1*	>.05*	96	14	100	0	-1.0	>.05
		30	28	7	37	17	-2.2*	<.05*	96	10	90	23	-0.8	>.05*

* Values based on negative ranks (i.e., the common graph performed better in these cases)

Table 6.6 — Time and accuracy values for the three user tasks.

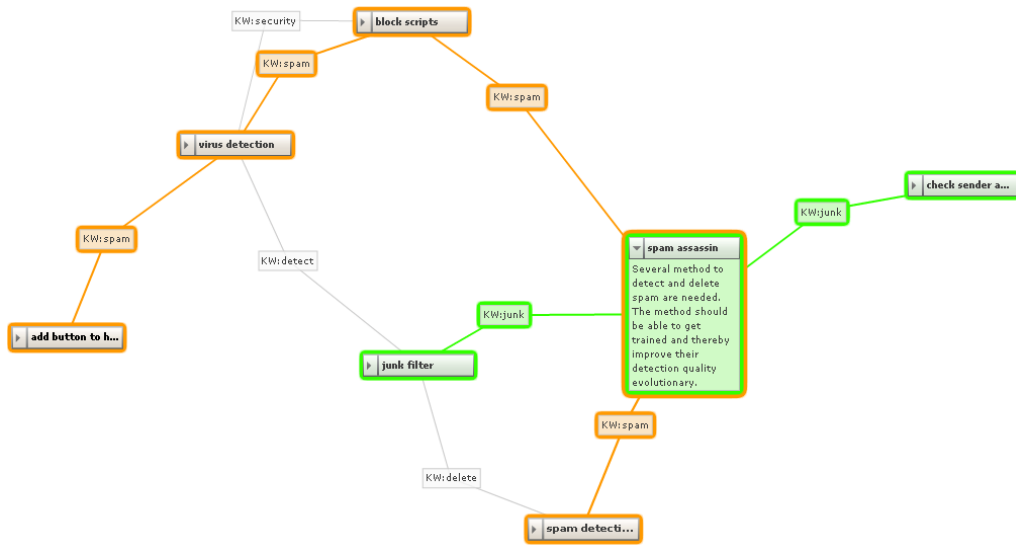


Figure 6.11 — Using the ChainGraph to visualize shared metadata relations between software requirements.

6.3.5 Discussion

With the ChainGraph, we introduced a graph visualization tailored to the visual exploration of folksonomies. While we described its creation and application for the folksonomy subgraph $G(RT)$, it can also be used with the $G(TU)$ and $G(UR)$ subgraphs we introduced in Section 6.1. The general ChainGraph approach is not even restricted to folksonomies but can also be applied to similar data structures and bipartite graphs.

For instance, we used the ChainGraph in the *SoftWiki* project mentioned in Section 4.7.1 to analyze tag relations between software requirements [122]. An example of this application is depicted in Figure 6.11. It shows the chains of a selection of tags, with the ones for the tags “junk” and “spam” highlighted.

Although there are many application possibilities, the ChainGraph should not be regarded as a general alternative to common graph visualizations. It rather offers a special kind of visualization serving a specific purpose. Due to its large size, the ChainGraph is best used on large screens. Figure 6.12 shows examples of applying our ChainGraph implementation on a powerwall and on a tabletop display.

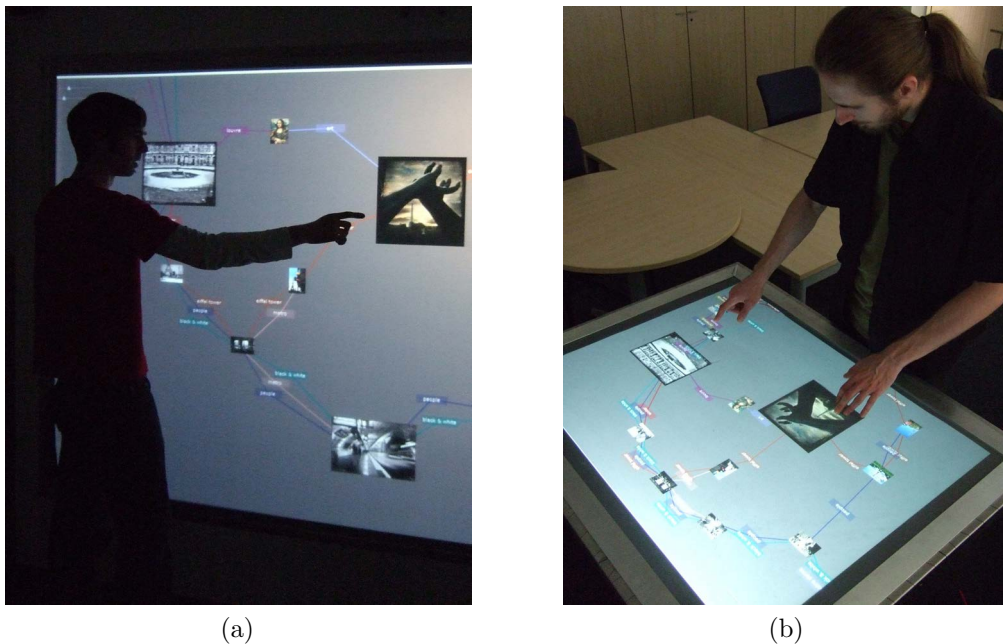


Figure 6.12 — Exploring a folksonomy with the ChainGraph visualization (a) on a powerwall and (b) on a multi-touch table.

Related Work

Much research has been conducted with the aim of drawing more understandable and readable graphs [77]. Popular optimization goals are a reduction of crossing edges, an avoidance of overlapping nodes, or edges of more or less uniform length. While an exact solution for these optimization problems is usually NP complete, many approaches propose heuristics such as simulated annealing [75], multidimensional scaling [153], or a force-directed placement [95] to compute nearly optimal solutions in much shorter time.

We used the force-directed placement to draw the ChainGraph representation on the screen. It helps to minimize node overlaps, what is especially valuable in the case of parallel running chains. An alternative solution for the visualization of parallel running chains could be drawings where edges are visualized as polygonal lines. Though these drawings provide greater flexibility, it can, however, be difficult to follow edges with several bends by the eye [198].

In contrast to this line of research, the ChainGraph proposes not a new drawing but a new type of graph representation. In this representation, tag

nodes are multiplied and composed to chains. The general idea of node multiplication is not new. It has, for instance, been proposed for graph-based structures in the areas of network or compiler optimization [135]. However, only few attempts have been made to use node multiplication as a technique in information visualization. The work that is maybe most closely related to ours is the *TensionSplit* algorithm [85] that multiplies nodes if they are energized by conflicting connections in a force-directed layout. The algorithm is based on the assumption that these nodes would most likely produce crossing edges what can be avoided by their multiplication. The problem of this layout-driven approach is that the reason for node multiplication is no longer visible after application. The structure-driven node multiplication of the ChainGraph, by contrast, remains clear and comprehensible also in the resulting graph visualization.

Limitations

The ChainGraph requires a special interpretation due to the multiplication and chain arrangement of nodes. This can be regarded as the main limitation of the approach, since it is less intuitive than the interpretation of a common graph and requires extra explanation. Another limitation of the ChainGraph is its larger size that results from the node multiplication. This can, for instance, be a problem if display space is limited and if common interaction techniques, such as scrolling, panning, or zooming, have to be avoided.

Apart from these limitations, there are virtually no other constraints that restrict the application of the ChainGraph. It is important to note that the ChainGraph considers only *indirect* relations but no *direct* ones. Direct (and other kinds of) relations can, of course, be easily added to the graph visualization; however, this might lead to irritations, since the relations must then be differently interpreted. More suitable might be the inclusion of tags that are only linked with one resource (i.e., non-shared tags), even though this can result in a large number of additional nodes in many cases.

Usually, exploration with the ChainGraph is only one in a series of activities in a retrieval process. For instance, the ChainGraph in the scenario of Section 6.3.3 could be the result of a search for the tags ‘paris’ and ‘france’. The approach is less suitable for the visualization of complete folksonomy subgraphs. Such cases require more compact graph representations.

Summary and Outlook

This thesis addressed the conceptualization and visualization of tagging and folksonomies. After defining the key concepts, we gave an overview and comparison of basic tagging models, followed by a comprehensive review of existing tagging ontologies and related vocabularies. With the *Modular Unified Tagging Ontology (MUTO)*, we presented a well-defined and coherent conceptualization of tagging incorporating the best parts of the reviewed ontologies.

MUTO supports different forms of tagging, including common, semantic, group, private, and automatic tagging. It provides a compact and understandable core conceptualization that can easily be extended by modules. The modular approach reduces the complexity of the ontology and increases its stability with regard to future developments in tagging. Furthermore, it reflects the tripartite character of folksonomies by connecting the independent domains of resources, tags, and users.

We designed MUTO for broad applicability and scalable reasoning. It is implemented in the sublanguage OWL Lite to allow for the highest possible tool support. By making it also compliant to OWL 2 RL, polynomial time reasoning algorithms can be executed directly on the RDF triples that constitute the folksonomy. However, the focus of this work was rather on a precise conceptualization than on performance. If performance is an issue, specific modules may be integrated into the MUTO ontology that speed

up processing, such as a module defining direct links between tags and resources.

Apart from such pragmatic extensions, further modules may be developed in the future that detail certain parts of the domain. One possible extension could be a module defining specific types of tags (geotags, tag-based star ratings, etc.) and different tag relations (synonymy, part-of, etc.). However, the question of how to derive typed tag relations from common tagging has not yet been sufficiently answered.

The goal of the conceptualization was not only to contribute to a better understanding of tagging but also to improve the semantic interoperability of folksonomies and related visualizations. If folksonomies are represented with the MUTO ontology, they can be accessed and shared in a uniform manner and validated against the conceptualization. This allows for the development of reusable visualizations that are independent of individual tagging systems and that provide different perspectives on the folksonomies.

In this thesis, we focused on tag clouds and folksonomy graphs as two popular types of folksonomy visualizations. We first looked into tag clouds and examined their visual perception and performance. We reported on a comparative user study that analyzed different tag cloud layouts with regard to how well they support typical information seeking tasks. The study results suggest that there exists no single best way to arrange tags in a cloud but that the optimal layout depends on the user goals and intentions of the designer – as it is often the case in interaction design. Still, we found some task-layout-combinations that worked particularly well in the study. Furthermore, we derived some interesting findings on visual attention patterns in tag clouds from an analysis of the eye tracking data recorded in the study.

Drawing upon these findings, we examined how tag clouds can be extended to provide better support for the visual analysis of folksonomies and other data. In particular, we were interested in the possibilities offered by tag clouds to visually analyze text content. As a first approach, we developed *prefix tag clouds*, a variation of tag clouds that uses a prefix tree to group different word forms and visualizes the subtrees in a circular tag cloud layout. The approach relies solely on lexicographical ordering, which makes it language-independent and easy to implement.

As a second approach, we presented *time-varying co-occurrence highlighting*. It supports the visual analysis by enhancing tag clouds with information

about tag relations and time-dependent tag use. We showed that the consideration of these two dimensions can be of value for the interactive exploration of folksonomies and text content. However, the increased complexity of the tag cloud visualization can be considered as a drawback of the technique.

Therefore, we decided for a less intrusive way to extend tag clouds when developing the *Tag Cloud Explorer*. The Tag Cloud Explorer is based on a common tag cloud visualization enhanced by interaction techniques and linguistic information in an intuitive way. It combines several ideas of our previous work and makes heavy use of natural language processing. It is particularly powerful in the analysis of text content but can also be used to explore folksonomies and similar data.

Finally, we examined possibilities of visualizing the graph structure of folksonomies. Such graph visualizations can reveal interesting patterns and relationships that are not visible in simple tag clouds. Since a direct visualization of the folksonomy hypergraph is not useful in most cases, we first discussed the decomposition of the hypergraph into simpler subgraphs representing specific parts of the folksonomy. Based on these subgraphs, we classified existing visualizations of folksonomy graphs and identified common limitations.

To overcome these limitations, we developed the *ChainGraph* visualization to support the exploration of folksonomies. While we described this visualization approach on the example of a folksonomy subgraph representing tag-resource relations, it can also be used for other folksonomy subgraphs and bipartite graphs in general. For instance, we applied the ChainGraph in the context of the SoftWiki project to visually analyze relations between software requirements [122].

A key challenge related to folksonomy graphs in general and the ChainGraph in particular is the large size of the visualizations. We discussed different strategies to reduce this size, from displaying only selected nodes and edges (e.g., the top weighted ones) to adaptive approaches where the user decides on the elements shown in the graph. A related research question is the optimal mapping of folksonomy subgraphs to a two-dimensional plane. While most of the work discussed in this thesis uses force-directed algorithms for this purpose, alternative graph layouts may provide even better results.

Several outcomes of this thesis have been successfully applied in different research projects. The MUTO ontology has been integrated in the latest ver-

sion of the *Semantic Web Ontology for Requirements Engineering (SWORE)* to represent taggings in the context of the *SoftWiki* project [209].

The developed tag cloud extensions proved useful in visual text analysis. *Prefix tag clouds* helped to group and analyze paper titles, while *time-varying co-occurrence highlighting* supported the time-based analysis of microblogging content. The *Tag Cloud Explorer* was used to explore different kinds of text corpora, ranging from sport news to abstracts of research papers.

The *ChainGraph* visualization was applied in image retrieval and requirements engineering, as discussed before. Furthermore, it was a source of inspiration for related works the author of this thesis was involved in. One example is the *RelFinder*, which uses a visualization similar to the Chain-Graph to show relationships between resources in RDF datasets [123, 173].

All these examples indicate the broad applicability of the works developed in this thesis. Many of the outcomes are not limited to tagging and folksonomies but can also be used in other contexts. It is our hope that the contributions of this thesis will be of value for related projects and future research in the domain of tagging and beyond.



Referenced RDF Vocabularies

Table A.1 lists the names and URIs of all RDF vocabularies referenced in this thesis. In addition, all namespace prefixes as used in the thesis and in the MUTO ontology are given.

Vocabulary name	Prefix	URI reference
Annotea Annotation Schema	A	http://www.w3.org/2000/10/annotation-ns#
Annotation Ontology	AO	http://purl.org/ao/
(Annotea) Bookmark Schema	B	http://www.w3.org/2002/01/bookmark#
Common Tag	CTAG	http://commontag.org/ns#
DCMI Metadata Terms	DCTERMS	http://purl.org/dc/terms/
DCMI Type Vocabulary	DCTYPE	http://purl.org/dc/dcmitype/
Friend of a Friend	FOAF	http://xmlns.com/foaf/0.1/

Continued on next page

Table A.1 — Continued from previous page

Vocabulary name	Prefix	URI reference
Identity of Resources on the Web	IRW	http://www.ontologydesignpatterns.org/ont/web/irw.owl
Meaning of a Tag	MOAT	http://moat-project.org/ns#
Modular Unified Tagging Ontology	MUTO	http://purl.org/muto/core#
NEPOMUK Annotation Ontology	NAO	http://www.semanticdesktop.org/ontologies/2007/08/15/nao#
NiceTag Ontology	NT	http://ns.inria.fr/nicetag/2010/09/09/voc
Open Annotation Data Model	OA	http://www.w3.org/ns/oa#
Open Annotation	OAC	http://www.openannotation.org/ns/
Ontology of Folksonomy	OF	http://www.eslomas.com/tagontology-1.owl
Ordered Lists Ontology	OLO	http://purl.org/ontology/olo/core#
OWL 2 Schema vocabulary	OWL	http://www.w3.org/2002/07/owl#
Provenance Authoring and Versioning	PAV	http://purl.org/pav/
SNaP Tag Ontology	PNT	http://data.press.net/ontology/tag/
RDF Vocabulary	RDF	http://www.w3.org/1999/02/22-rdf-syntax-ns#
(RDF) Named Graphs	RDFG	http://www.w3.org/2004/03/trix/rdfg-1
RDF Schema	RDFS	http://www.w3.org/2000/01/rdf-schema#
RDF Site Summary	RSS	http://purl.org/rss/1.0/

Continued on next page

Table A.1 — *Continued from previous page*

Vocabulary name	Prefix	URI reference
Social Semantic Cloud of Tags	SCOT	http://scot-project.org/scot/ns#
Semantically-Interlinked Online Communities	SIOC	http://rdfs.org/sioc/ns#
SIOC Types Ontology Module	SIOCT	http://rdfs.org/sioc/types#
Simple Knowledge Organization System	SKOS	http://www.w3.org/2004/02/skos/core
Tag Ontology	TAGS	http://www.holygoat.co.uk/owl/redwood/0.1/tags/
Tagging Ontology	TO	http://bubb.ghb.fh-furtwangen.de/TagOnt/tagont.owl
TAGora Tagging Ontology	TT	http://tagora.ecs.soton.ac.uk/schemas/tagging
Upper Tag Ontology	UTO	http://info.slis.indiana.edu/~dingying/uto.owl
XML Schema	XSD	http://www.w3.org/2001/XMLSchema#

Table A.1 — Names, namespace prefixes, and URIs of all referenced vocabularies.

Items of the ChainGraph Evaluation

	N	Co. Graph		ChainGraph	
		Mean	SD	Mean	SD
<i>Effectiveness:</i>					
I could quickly solve the tasks.	12	3.1	0.8	3.9	0.8
Searching for the resources that share most tags was easy.	12	2.0	1.2	3.1	1.3
It was easy to find all tags that were shared by two resources.	12	3.3	1.2	3.8	1.1
It was easy to find all tags that were shared by three resources.	12	2.8	1.1	3.0	1.0
Solving the tasks took too long.	12	3.5	0.9	3.6	0.8
The graph visualizations made it difficult to solve the tasks.	12	3.2	0.9	3.4	1.2

Continued on next page

Table B.1 — Continued from previous page

	N	Co. Graph		ChainGraph	
		Mean	SD	Mean	SD
The colors helped me in solving the tasks.	12	4.1	1.1	4.5	0.9
<i>Understandability:</i>					
I immediately understood the structure of the graphs.	12	4.1	1.1	3.6	1.2
I found my way well in the graphs.	12	2.9	0.9	3.6	1.0
The graphs presented the information in an understandable and clear way.	12	3.2	0.7	3.3	1.0
I do not understand the way the information is arranged.	12	1.4	0.7	4.3	1.0
The graph visualizations are too complex.	12	2.8	0.8	3.7	0.7
<i>Controllability:</i>					
I felt unsure when solving the tasks.	12	3.3	1.0	3.4	1.1
I am not sure if I solved all tasks correctly.	12	3.8	1.0	2.7	1.4
Some information was concealed in the graphs so that I could not read all.	12	4.0	0.9	4.7	0.7
I partly felt lost in the graphs.	12	3.1	0.9	3.6	1.1
I felt overwhelmed by all the information.	12	2.7	0.9	3.9	0.9

Continued on next page

Table B.1 — Continued from previous page

	N	Co. Graph		ChainGraph	
		Mean	SD	Mean	SD
The overlapping lines in the graphs confused me.	12	2.9	1.3	3.8	1.5
<i>Attractiveness:</i>					
I liked the optical appearance of the graphs.	12	2.9	0.9	3.4	1.2
The visualizations are innovative.	12	2.0	1.0	2.9	1.1
The visualizations are conventional.	12	3.7	0.9	3.4	1.2
The way of presenting the graphs appears unprofessional.	12	2.8	1.1	3.7	1.0
I think the graphs are stylish.	12	2.3	1.1	2.4	1.1

Table B.1 — Items of the ChainGraph evaluation grouped by category (negative items were positively coded), along with the ratings of the participants. The items were presented in random order.



RDF/XML Serialization of MUTO

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:sioc="http://rdfs.org/sioc/ns#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <!-- /// OWL DL compliance statements /// -->
  <!-- Annotation properties -->
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/
    elements/1.1/creator"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/
    elements/1.1/description"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/
    licence"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/
    elements/1.1/title"/>
  <owl:AnnotationProperty rdf:about="http://purl.org/dc/terms/
    issued"/>
```

```

<owl:AnnotationProperty rdf:about="http://xmlns.com/foaf/0.1/
  homepage"/>
<owl:AnnotationProperty rdf:about="http://xmlns.com/foaf/0.1/
  depiction"/>
<!-- Classes -->
<owl:Class rdf:about="http://www.w3.org/2004/02/skos/core#
  Concept"/>
<owl:Class rdf:about="http://rdfs.org/sioc/ns#Item"/>
<owl:Class rdf:about="http://rdfs.org/sioc/ns#UserAccount"/>
<!-- Object properties -->
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#about
  "/>
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#
  has_creator"/>
<owl:ObjectProperty rdf:about="http://rdfs.org/sioc/ns#
  creator_of"/>
<!-- Datatype properties -->
<owl:DatatypeProperty rdf:about="http://purl.org/dc/terms/
  created"/>
<owl:DatatypeProperty rdf:about="http://purl.org/dc/terms/
  modified"/>
<!-- /// MUTO Core ontology /// -->
<owl:Ontology rdf:about="http://purl.org/muto/core">
  <dc:title>Modular Unified Tagging Ontology (MUTO)</dc:title
  >
  <rdfs:label xml:lang="en">MUTO Core Ontology</rdfs:label>
  <dc:description>The Modular and Unified Tagging Ontology (
    MUTO) is an ontology for tagging and folksonomies. It is
    based on a thorough review of earlier tagging
    ontologies and unifies core concepts in one consistent
    schema. It supports different forms of tagging, such as
    common, semantic, group, private, and automatic tagging,
    and is easily extensible.</dc:description>
  <dc:creator>Steffen Lohmann</dc:creator>
  <dcterms:issued>2011-11-16</dcterms:issued>
  <dcterms:licence>http://creativecommons.org/licenses/by
    /3.0/</dcterms:licence>
  <rdfs:seeAlso>http://purl.org/muto/core#</rdfs:seeAlso>
  <foaf:homepage>http://purl.org/muto</foaf:homepage>
  <foaf:depiction>http://purl.org/muto/core/muto-compact.png
    </foaf:depiction>
  <foaf:depiction>http://purl.org/muto/core/muto-UML.png</
    foaf:depiction>
  <owl:versionInfo>Version 1.0 – Global changes (compared to
    earlier versions): Some properties have been renamed;
    cardinality constraints in class descriptions (owl:

```

```

Restriction) have been replaced by global cardinality
constraints (owl:FunctionalProperty).</owl:versionInfo>
</owl:Ontology>
<!-- Classes -->
<owl:Class rdf:about="http://purl.org/muto/core#Tagging">
  <rdfs:label xml:lang="en">Tagging</rdfs:label>
  <rdfs:comment xml:lang="en">A tagging links a resource to a
    user account and one or more tags.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://rdfs.org/sioc/ns#Item
"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
  core#"/>
</owl:Class>
<owl:Class rdf:about="http://purl.org/muto/core#
  PrivateTagging">
  <rdfs:label xml:lang="en">Private Tagging</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://purl.org/muto/core#
  Tagging"/>
  <rdfs:comment xml:lang="en">A private tagging is a tagging
    that is only visible to its creator (unless the creator
    has not granted access to others via muto:grantAccess).
    Every tagging that is not an instance of muto:
    PrivateTagging is public by default.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
  core#"/>
</owl:Class>
<owl:Class rdf:about="http://purl.org/muto/core#Tag">
  <rdfs:label xml:lang="en">Tag</rdfs:label>
  <rdfs:comment xml:lang="en">A Tag consists of an arbitrary
    text label. Note that tags with the same label are NOT
    merged in the ontology.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2004/02/
  skos/core#Concept"/>
  <owl:versionInfo>Version 1.0: The owl:disjointWith
    statement was removed to make MUTO conform to OWL Lite (
    the statement is not essential in this case).</owl:
    versionInfo>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
  core#"/>
</owl:Class>
<owl:Class rdf:about="http://purl.org/muto/core#AutoTag">
  <rdfs:label xml:lang="en">Automatic Tag</rdfs:label>
  <rdfs:subClassOf rdf:resource="http://purl.org/muto/core#
  Tag"/>
  <rdfs:comment xml:lang="en">An automatic tag is a tag that
    is automatically associated with a resource (e.g. by a

```

```

    tagging system), i.e. it is not entered by a human being
    .</rdfs:comment>
  <owl:versionInfo>Version 1.0: muto:AutoTag was a subclass
    of muto:Tagging (called muto:autoTagging) in earlier
    versions. Defining it as a subclass of muto:Tag is more
    appropriate and allows for taggings that contain a
    combination of manually and automatically created tags
    .</owl:versionInfo>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
      core#"/>
  </owl:Class>
  <!-- Object properties of class muto:Tagging -->
  <owl:ObjectProperty rdf:about="http://purl.org/muto/core#
    hasTag">
    <rdfs:label xml:lang="en">has tag</rdfs:label>
    <rdfs:comment xml:lang="en">A tagging consists of a (
      theoretically unlimited) number of tags. A tagging may
      also consist of no tags, e.g. if the system allows its
      users to mark a resource first and add tags later.</rdfs:
      comment>
    <rdfs:domain rdf:resource="http://purl.org/muto/core#
      Tagging"/>
    <rdfs:range rdf:resource="http://purl.org/muto/core#Tag"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
      core#"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://purl.org/muto/core#
    tagOf">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      FunctionalProperty"/>
    <rdfs:label xml:lang="en">tag of</rdfs:label>
    <rdfs:comment xml:lang="en">Every tag is linked to exactly
      one tagging. This results from the fact that tags with
      same labels are NOT merged in the ontology.</rdfs:
      comment>
    <rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
    <rdfs:range rdf:resource="http://purl.org/muto/core#Tagging
      "/>
    <owl:inverseOf rdf:resource="http://purl.org/muto/core#
      hasTag"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
      core#"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="http://purl.org/muto/core#
    taggedResource">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
      FunctionalProperty"/>

```



```

<rdfs:label xml:lang="en">tagged resource</rdfs:label>
<rdfs:comment xml:lang="en">Every tagging is linked to
    exactly one resource. This can be any kind of resource (
        i.e. all subclasses of rdfs:Resource), including tags
        and taggings.</rdfs:comment>
<rdfs:domain rdf:resource="http://purl.org/muto/core#
    Tagging"/>
<rdfs:subPropertyOf rdf:resource="http://rdfs.org/sioc/ns#
    about"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
        core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
    taggedWith">
    <rdfs:label xml:lang="en">tagged with</rdfs:label>
    <rdfs:comment xml:lang="en">A resource can have several
        taggings from different users. Tags are never directly
        linked to resources but can be inferred from the
        taggings.</rdfs:comment>
    <rdfs:range rdf:resource="http://purl.org/muto/core#Tagging
        "/>
    <owl:inverseOf rdf:resource="http://purl.org/muto/core#
        taggedResource"/>
        <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
            core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
    hasCreator">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
        FunctionalProperty"/>
    <rdfs:label xml:lang="en">has creator</rdfs:label>
    <rdfs:comment xml:lang="en">Every tagging is linked to at
        most one user account. This property can be omitted for
        automatic taggings. In contrast to its superproperty
        sioc:has_creator, it is functional and with an explicit
        domain. Use sioc:member_of to define groups for group
        tagging or link to foaf:Agent, foaf:Person, or foaf:
        Group via sioc:account_of.</rdfs:comment>
    <rdfs:domain rdf:resource="http://purl.org/muto/core#
        Tagging"/>
    <rdfs:range rdf:resource="http://rdfs.org/sioc/ns#
        UserAccount"/>
    <rdfs:subPropertyOf rdf:resource="http://rdfs.org/sioc/ns#
        has_creator"/>
        <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
            core#"/>
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
  creatorOf">
  <rdfs:label xml:lang="en">creator of</rdfs:label>
  <rdfs:comment xml:lang="en">A user account can have a (
    theoretically unlimited) number of taggings. Use sioc:
    member_of to define groups for group tagging or link to
    foaf:Agent, foaf:Person, or foaf:Group via sioc:
    account_of.</rdfs:comment>
  <rdfs:domain rdf:resource="http://rdfs.org/sioc/ns#
    UserAccount"/>
  <rdfs:range rdf:resource="http://purl.org/muto/core#Tagging
    "/>
  <rdfs:subPropertyOf rdf:resource="http://rdfs.org/sioc/ns#
    creator_of"/>
  <owl:inverseOf rdf:resource="http://purl.org/muto/core#
    hasCreator"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
    core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
  grantAccess">
  <rdfs:label xml:lang="en">grant access</rdfs:label>
  <rdfs:comment xml:lang="en">A (usually private) tagging can
    be linked to one or more user accounts or user groups
    that should have access to it (apart from the creator).
    This property can also be used in public tagging to link
    a user account or user group to a tagging (e.g. if the
    creator of a tagging wants to suggest the tagging to
    another user).</rdfs:comment>
  <rdfs:domain rdf:resource="http://purl.org/muto/core#
    Tagging"/>
  <owl:versionInfo>Version 1.0: The range information has
    been removed for greater flexibility and OWL Lite
    conformance (no longer owl:unionOf). Classes from
    different vocabularies can now be used here – such as
    sioc:UserAccount, sioc:Usergroup, foaf:OnlineAccount,
    foaf:Group, or dcterms:Agent –, though we recommend the
    use of sioc:UserAccount or sioc:Usergroup to remain in
    the SIOC namespace.</owl:versionInfo>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
    core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
  hasAccess">
  <rdfs:label xml:lang="en">has access</rdfs:label>
  <rdfs:comment xml:lang="en">A user account or user group
    can have access to a private tagging from another user

```

```

        if the access is explicitly permitted by the creator of
        the tagging. This property can also be used in public
        tagging to link a user account or user group to a
        tagging (e.g. if the creator of a tagging has suggested
        the tagging to another user).</rdfs:comment>
<rdfs:range rdf:resource="http://purl.org/muto/core#Tagging
"/>
<owl:inverseOf rdf:resource="http://purl.org/muto/core#
grantAccess"/>
<owl:versionInfo>Version 1.0: see muto:grantAccess</owl:
versionInfo>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
core#"/>
</owl:ObjectProperty>
<!-- Datatype properties of class muto:Tagging -->
<owl:DatatypeProperty rdf:about="http://purl.org/muto/core#
taggingCreated">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
FunctionalProperty"/>
    <rdfs:label xml:lang="en">tagging created</rdfs:label>
    <rdfs:comment xml:lang="en">Every tagging has exactly one
    creation date and time. The datatype of this property is
    xsd:dateTime (in contrast to its superproperty dterms:
    created which has range rdfs:Literal).</rdfs:comment>
    <rdfs:domain rdf:resource="http://purl.org/muto/core#
Tagging"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
dateTime"/>
    <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/terms/
created"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
core#"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="http://purl.org/muto/core#
taggingModified">
    <rdfs:label xml:lang="en">tagging modified</rdfs:label>
    <rdfs:comment xml:lang="en">A tagging can have multiple
    modification dates, as the number of times a tagging can
    be edited (e.g. to add or remove tags) is theoretically
    unlimited. The datatype of this property is xsd:
    dateTime (in contrast to it superproperty dterms:
    created which has range rdfs:Literal).</rdfs:comment>
    <rdfs:domain rdf:resource="http://purl.org/muto/core#
Tagging"/>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
dateTime"/>

```

```

<rdfs:subPropertyOf rdf:resource="http://purl.org/dc/terms/
  modified"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
    core#"/>
</owl:DatatypeProperty>
<!-- Datatype properties of class muto:Tag -->
<owl:DatatypeProperty rdf:about="http://purl.org/muto/core#
  tagCreated">
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#
    FunctionalProperty"/>
  <rdfs:label xml:lang="en">tag created</rdfs:label>
  <rdfs:comment xml:lang="en">The creation date and time of a
    tag. This property can be omitted if muto:
    taggingCreated = muto:tagCreated (i.e. in the common
    case that a tag has been created along with a tagging,
    not in a later edit of the tagging). The datatype of
    this property is xsd:dateTime (in contrast to its
    superproperty dcterms:created which has range rdfs:
    Literal).</rdfs:comment>
  <rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#
    dateTime"/>
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/terms/
    created"/>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
      core#"/>
</owl:DatatypeProperty>
<!-- Object properties of class muto:Tag -->
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
  tagMeaning">
  <rdfs:label xml:lang="en">tag meaning</rdfs:label>
  <rdfs:comment xml:lang="en">The meaning of a tag can be
    expressed by a link to a well-defined resource. This can
    be any resource that clarifies the meaning of the tag (
    e.g. some DBpedia resource).</rdfs:comment>
  <rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
    core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
  meaningOf">
  <rdfs:label xml:lang="en">meaning of</rdfs:label>
  <rdfs:comment xml:lang="en">The number of tags that can be
    linked to one and the same meaning is theoretically
    unlimited.</rdfs:comment>
  <rdfs:range rdf:resource="http://purl.org/muto/core#Tag"/>

```

```

<owl:inverseOf rdf:resource="http://purl.org/muto/core#
tagMeaning"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
autoMeaning">
  <rdfs:label xml:lang="en">automatic tag meaning</rdfs:label
  >
  <rdfs:comment xml:lang="en">This subproperty indicates that
the meaning of a tag has been automatically defined (e.
g. by a tagging system), i.e. it has not been defined by
a human being. The default case is disambiguation by
users via muto:tagMeaning.</rdfs:comment>
  <rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
  <rdfs:subPropertyOf rdf:resource="http://purl.org/muto/core
#tagMeaning"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
core#"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="http://purl.org/muto/core#
tagLabel">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
FunctionalProperty"/>
  <rdfs:label xml:lang="en">tag label</rdfs:label>
  <rdfs:comment xml:lang="en">Every tag has exactly one label
(usually the one given by the user) – otherwise it is
not a tag. Additional labels can be defined in the
resource that is linked via muto:tagMeaning.</rdfs:
comment>
  <owl:versionInfo>Version 1.0: The subproperty relation to
rdfs:label has been removed for OWL DL conformance (rdfs
:label is an annotation property and one cannot define
subproperties for annotation properties in OWL DL).</owl
:versionInfo>
  <rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Literal"/>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
core#"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
nextTag">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
FunctionalProperty"/>
  <rdfs:label xml:lang="en">next tag</rdfs:label>

```

```

<rdfs:comment xml:lang="en">This property indicates the tag
  that follows next in the list of tags. It can be used
  to describe the order in which the tags have been
  entered by the user.</rdfs:comment>
<rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
<rdfs:range rdf:resource="http://purl.org/muto/core#Tag"/>
<owl:versionInfo>Version 1.0: Earlier versions of MUTO
  defined a datatype property muto:tagPosition with
  integer values which has some drawbacks compared to this
  solution.</owl:versionInfo>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
    core#"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://purl.org/muto/core#
  previousTag">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
    FunctionalProperty"/>
  <rdfs:label xml:lang="en">previous tag</rdfs:label>
  <rdfs:comment xml:lang="en">This property indicates the tag
    that is preceding in the list of tags. It can be used
    to describe the order in which the tags have been
    entered by the user.</rdfs:comment>
  <rdfs:domain rdf:resource="http://purl.org/muto/core#Tag"/>
  <rdfs:range rdf:resource="http://purl.org/muto/core#Tag"/>
  <owl:inverseOf rdf:resource="http://purl.org/muto/core#
    nextTag"/>
  <owl:versionInfo>Version 1.0: see muto:nextTag</owl:
    versionInfo>
    <rdfs:isDefinedBy rdf:resource="http://purl.org/muto/
      core#"/>
  </owl:ObjectProperty>
</rdf:RDF>

```

Listing C.1 — RDF/XML serialization of version 1.0 of the Modular Unified Tagging Ontology (MUTO).

Bibliography

- [1] Amazon.com. <http://www.amazon.com>. 2
- [2] Annotea project. <http://www.w3.org/2001/Annotea>. 57
- [3] Apache flex. <http://flex.apache.org>. 158
- [4] Bibsonomy. <http://www.bibsonomy.org>. 22, 33
- [5] Common tag. <http://www.commonitag.org>. 47
- [6] Delicious. <http://www.delicious.com>. 1, 2, 95, 144
- [7] Faviki. <http://www.faviki.com>. 20
- [8] Flickr. <http://www.flickr.com>. 1, 2, 95, 158
- [9] Flickr: Help: Tags. <http://www.flickr.com/help/tags/>. 23, 26
- [10] Gmail. <http://mail.google.com>. 2
- [11] Last.fm. <http://www.lastfm.de>. 24, 103
- [12] LibraryThing. <http://www.librarything.com>. 2
- [13] Microformats. <http://microformats.org>. 26, 57
- [14] MOAT: Meaning of a tag. <http://moat-project.org>. 45
- [15] NEPOMUK – the social semantic desktop. <http://nepomuk.semanticdesktop.org>. 60
- [16] Open annotation community group. <http://www.w3.org/community/openannotation/>. 59
- [17] Pajek. <http://pajek.imfm.si>. 146
- [18] Quick start guide – common tag. <http://commonitag.org/QuickStartGuide>. xiv, 47, 48
- [19] SIOC project – semantically-interlinked online communities. <http://sioc-project.org>. 69, 82

194 Bibliography

- [20] SIOC types ontology module namespace. <http://rdfs.org/sioc/types#>. 56
- [21] SKOS simple knowledge organization system. <http://www.w3.org/2004/02/skos/>. 38
- [22] SlideShare. <http://www.slideshare.net>. 21
- [23] Stanford CoreNLP – a suite of core NLP tools. <http://nlp.stanford.edu/software/corenlp.shtml>. 128
- [24] Tag maps. <http://tagmaps.research.yahoo.com>. 95
- [25] TagCommons. <http://tagcommons.org>. 36
- [26] Taglines. <http://research.yahoo.com/taglines/>. 32, 95, 148
- [27] TAGora. <http://www.tagora-project.eu>. 51
- [28] Tagora. 51
- [29] Tagul. <http://tagul.com>. 96
- [30] Tagxedo. <http://www.tagxedo.com>. 96
- [31] Technorati. <http://technorati.com>. 95
- [32] Twitter. <https://twitter.com>. 123
- [33] Viddler. <http://www.viddler.com>. 21
- [34] Wordle. <http://www.wordle.net>. 96, 124
- [35] Youtube. <http://www.youtube.com>. 2
- [36] Popular tags on flickr. <http://www.flickr.com/photos/tags>, 2010. xiv, 16, 17
- [37] Amazon.com: Most popular tags. <http://www.amazon.com/gp/tagging/cloud>, 2011. xv, 97
- [38] Samer A. Abdallah and Bob Ferris. The ordered list ontology 0.72. <http://purl.org/ontology/olo/core#>, 2010. 79
- [39] Ben Adida, Mark Birbeck, Shane McCarron, and Ivan Herman. RDFa core 1.1 – second edition. <http://www.w3.org/TR/rdfa-core/>, 2013. 57

- [40] Christine Alvarado, Jaime Teevan, Mark S. Ackerman, and David Karger. Surviving the information explosion: How people find their electronic information. Technical Report AIM-2003-006, MIT AI Lab, 2003. 1
- [41] Morgan Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*, pages 971–980. ACM, 2007. 27
- [42] Sofia Angeletou, Marta Sabou, and Enrico Motta. Semantically enriching folksonomies with FLOR. In *Proceedings of the 1st International Workshop on Collective Semantics: Collective Intelligence & the Semantic Web (CISWeb '08)*, volume 351 of *CEUR-WS.org*, 2008. 77
- [43] Michael Arrington. All the cool kids are deep tagging. <http://techrunch.com/2006/10/01/all-the-cool-kids-are-deep-tagging/>, 2006. [last accessed on October 4, 2013]. 21
- [44] Scott Bateman, Carl Gutwin, and Miguel Nacenta. Seeing things in the clouds: The effect of visual features on tag cloud selections. In *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (HT '08)*, pages 193–202. ACM, 2008. 98, 99, 103, 110, 111
- [45] Gabe Begeed-Dov, Dan Brickley, Rael Dornfest, Ian Davis, Leigh Dodds, Jonathan Eisenzopf, David Galbraith, Ramanathan V. Guha, Ken MacLeod, Eric Miller, Aaron Swartz, and Eric van der Vlist. Rdf. 65
- [46] Grigory Begelman, Philipp Keller, and Frank Smadja. Automated tag clustering: Improving search and exploration in the tag space. In *Proceedings of the WWW2006 Workshop on Collaborative Web Tagging*, 2006. 19, 33, 150
- [47] Tim Berners-Lee. A short history of “resource” in web architecture. <http://www.w3.org/DesignIssues/TermResource.html>, 2009. 21
- [48] Tim Berners-Lee, Roy T. Fielding, and Larry Masinter. Uniform resource identifier (URI): Generic syntax. <http://tools.ietf.org/html/rfc3986>, 2005. 21
- [49] Tim Berners-Lee, James A. Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001. 35

- [50] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, pages 193–202. ACM, 2008. xviii, 24, 25
- [51] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia – a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009. 19, 45, 77
- [52] DCMI Usage Board. DCMI metadata terms. <http://dublincore.org/documents/dcmi-terms/>, 2012. 71
- [53] DCMI Usage Board. DCMI type vocabulary. <http://dublincore.org/documents/dcmi-type-vocabulary>, 2012. 40
- [54] DCMI Usage Board. Dublin core metadata element set, version 1.1. <http://dublincore.org/documents/dces/>, 2012. 26, 38, 71
- [55] Uldis Bojars and John G. Breslin. SIOC core ontology specification. <http://rdfs.org/sioc/spec/>, 2010. 44, 65, 66, 82
- [56] Nitin Borwankar. Slicing and dicing data 2.0. <http://tagschema.com/blogs/tagschema/2005/06/slicing-and-dicing-data-20-part-1.html>, 2005. [last accessed on October 4, 2013]. 21
- [57] Simone Braun, Christine Kunzmann, and Andreas Schmidt. People tagging and ontology maturing: Toward collaborative competence management. In Dan Diaper, Colston Sanger, David Randall, and Pascal Salembier, editors, *From CSCW to Web 2.0: European Developments in Collaborative Design*, pages 133–154. Springer, 2010. 21
- [58] Tim Bray, Dave Hollander, Andrew Layman, Richard Tobin, and Henry S. Thompson. Namespaces in XML 1.0. <http://www.w3.org/TR/xml-names/>, 2009. 26
- [59] Dan Brickley and Ramanathan V. Guha. RDF vocabulary description language 1.0: RDF schema. <http://www.w3.org/TR/rdf-schema/>, 2004. 35, 81
- [60] Michael Burch, Steffen Lohmann, and Daniel Weiskopf. Prefix tag clouds. In *Proceedings of the 17th International Conference on Information Visualisation (IV 2013)*, pages 45–50. IEEE, 2013. 95

- [61] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th International Conference on World Wide Web (WWW '05)*, pages 613–622. ACM, 2005. 55
- [62] Ya-Xi Chen, Rodrigo Santamaría, Andreas Butz, and Roberto Therón. TagClusters: Semantic aggregation of collaborative tags beyond tag-clouds. In *Proceedings of the 10th International Symposium on Smart Graphics (SG '09)*, volume 5531 of *LNCS*, pages 56–67. Springer, 2009. 96
- [63] Ed H. Chi, Peter Pirolli, Kim Chen, and James Pitkow. Using information scent to model user information needs and actions on the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*, pages 490–497. ACM, 2001. 160
- [64] Paolo Ciccarese. Annotation ontology. <http://code.google.com/p/annotation-ontology/>. 58
- [65] Paolo Ciccarese. PAV 2.0 – provenance authoring and versioning ontology. <http://code.google.com/p/pav-ontology/>. 58
- [66] Paolo Ciccarese. Semantic web applications in neuromedicine (SWAN) ontology. <http://www.w3.org/TR/hcls-swan/>, 2009. 58
- [67] Paolo Ciccarese, Marco Ocana, Leyla J. Garcia Castro, Sudeshna Das, and Tim Clark. An open annotation ontology for science on web 3.0. *Journal of Biomedical Semantics*, 2(Suppl 2):S4, 2011. xiv, 58, 59
- [68] Jeff Clark. Twitter StreamGraphs. <http://www.neoformix.com/Projects/TwitterStreamGraphs/view.php>. 125
- [69] Jeff Clark. Clustered word clouds. <http://neoformix.com/2008/ClusteredWordClouds.html>, 2008. [last accessed on October 4, 2013]. xv, 96, 97, 101
- [70] Christopher Collins, Fernanda B. Viégas, and Martin Wattenberg. Parallel tag clouds to explore and analyze faceted text corpora. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST '09)*, pages 91–98. IEEE, 2009. 125, 140

- [71] Aaron S. Cope. Machine tags. <http://www.flickr.com/groups/api/discuss/72157594497877875/>, 2007. [last accessed on October 4, 2013]. 26
- [72] Weiwei Cui, Yingcai Wu, Shixia Liu, Furu Wei, M. X. Zhou, and Huamin Qu. Context-preserving, dynamic word cloud visualization. *IEEE Computer Graphics and Applications*, 30(6):42–53, 2010. 120
- [73] Daniel Dahl, Jens Lechtenbörger, Jens Sieberg, and Gottfried Vossen. Learnr: web 2.0-driven learning. In *Proceedings of the 7th IASTED International Conference on Web-based Education (WBE '08)*, pages 259–264. ACTA Press, 2008. 21
- [74] Céline Van Damme, Martin Hepp, and Katharina Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *Proceedings of the ESWC 2007 Workshop on Bridging the Gap between Semantic Web and Web 2.0 (SemNet '07)*, pages 57–70, 2007. 19, 21, 22, 28
- [75] Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996. 168
- [76] Mike Dean, Guus Schreiber, Sean Bechhofer, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn A. Stein. OWL web ontology language reference. <http://www.w3.org/TR/owl-ref/>, 2004. 40, 66
- [77] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications*, 4(5):235–282, 1994. 168
- [78] Ying Ding, Elin K. Jacob, Michael Fried, Ioan Toma, Erjia Yan, Schubert Foo, and Staša Milojević. Upper tag ontology for integrating social tagging data. *Journal of the American Society for Information Science and Technology*, 61(3):505–521, 2010. xiv, 48, 49, 50, 51
- [79] Alan Dix, Stefano Levialdi, and Alessio Malizia. Semantic halo for collaboration tagging systems. In *Proceedings of the Workshop on the Social Navigation and Community-Based Adaptation Technologies*, 2006. 18, 19

- [80] Marian Dörk, Sheelagh Carpendale, Christopher Collins, and Carey Williamson. VisGets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1205–1212, 2008. 139
- [81] Marian Dörk, Daniel M. Gruen, Carey Williamson, and M. Sheelagh T. Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010. 118, 120, 124, 125
- [82] Nick Drummond. List ontology. <http://www.co-ode.org/ontologies/lists/2008/09/11/>, 2008. 79
- [83] Micah Dubinko, Ravi Kumar, Joseph Magnani, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Visualizing tags over time. In *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pages 193–202. ACM, 2006. 32, 95, 148
- [84] Martin Dürst and Michel Suignard. Internationalized resource identifiers (IRIs). <http://tools.ietf.org/html/rfc3987>, 2005. 21
- [85] Peter Eades and Candido F. X. de Mendonça N. Vertex splitting and tension-free layout. In *Proceedings of the Symposium on Graph Drawing (GD '95)*, volume 1027 of *LNCS*, pages 202–211. Springer, 1995. 169
- [86] Alf Eaton. Graph del.icio.us related tags. <http://hublog.hubmed.org/archives/001049.html>, 2005. [last accessed on October 4, 2013]. xvii, 144, 146
- [87] Alf Eaton. Graph del.icio.us subscriptions network. <http://hublog.hubmed.org/archives/001050.html>, 2005. [last accessed on October 4, 2013]. 150
- [88] Francisco Echarte, José Javier Astrain, Alberto Córdoba, and Jesús E. Villadangos. Ontology of folksonomy: A new modelling method. In *Proceedings of the Semantic Authoring, Annotation and Knowledge Markup Workshop (SAAKM '07)*, volume 289 of *CEUR-WS.org*, 2007. 38, 41, 42
- [89] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation.

- In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, pages 385–392. Curran Associates, 2008. 27
- [90] Tantek Çelik and Kevin Marks. rel="tag". <http://microformats.org/wiki/rel-tag>, 2005. [last accessed on October 4, 2013]. 57
- [91] Stephen Farrell and Tessa Lau. Fringe contacts: People-tagging for the enterprise. In *Proceedings of the WWW2006 Workshop on Collaborative Web Tagging*, 2006. 21, 29
- [92] Stephen Farrell, Tessa Lau, Stefan Nusser, Eric Wilcox, and Michael Muller. Socially augmenting employee profiles with people-tagging. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST 2007)*, pages 91–100. ACM, 2007. 21
- [93] Jonathan Feinberg. Wordle. In Julie Steele and Noah Iliinsky, editors, *Beautiful Visualization*, pages 37–58. O’Reilly, 2010. 116, 118
- [94] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL ’05)*, pages 363–370. ACL, 2005. 132
- [95] Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. 148, 168
- [96] Ko Fujimura, Shigeru Fujimura, Tatsushi Matsubayashi, Takeshi Yamada, and Hidenori Okuda. Topigraphy: visualization for large-scale tag clouds. In *Proceedings of the 17th International Conference on World Wide Web (WWW ’08)*, pages 1087–1088. ACM, 2008. 95, 96
- [97] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987. 16
- [98] Philippe Gambette and Jean Véronis. Visualising a text with a tree cloud. In Hermann Locarek-Junge and Claus Weihs, editors, *Classification as a Tool for Research*, pages 561–569. Springer, 2010. 126

- [99] Andres Garcia, Martin Szomszor, Harith Alani, and Oscar Corcho. Preliminary results in tag disambiguation using DBpedia. In *Proceedings of the 1st International Workshop on Collective Knowledge Capturing and Representation (CKCaR '09)*, 2009. 77
- [100] Jesse J. Garrett. An interview with flickr's eric costello. <http://www.adaptivepath.com/ideas/e000519>, 2005. [last accessed on October 4, 2013]. 1
- [101] Bud Gibson. xfolk rc1. <http://microformats.org/wiki/xfolk>, 2005. [last accessed on October 4, 2013]. 57
- [102] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006. 18, 19, 21, 24
- [103] James Governor. On the emergence of professional tag gardeners. <http://redmonk.com/jgovernor/2006/01/10/on-the-emergence-of-professional-tag-gardeners/>, 2006. [last accessed on October 4, 2013]. 19
- [104] Alexander Graf. RDFa vs. microformats. Technical report, DERI, 2007. 57
- [105] The W3C SPARQL Working Group. SPARQL 1.1 overview. <http://www.w3.org/TR/sparql11-overview/>, 2013. 91
- [106] W3C OWL Working Group. OWL 2 web ontology language document overview (second edition). <http://www.w3.org/TR/owl2-overview/>, 2012. 35, 66, 86
- [107] Thomas Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993. 34
- [108] Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges. In *Proceedings of the 1st Conference on Metadata and Semantics Research (MTSR '05)*, 2005. Published in International Journal on Semantic Web and Information Systems, 3(2):1–11, 2007. 21, 22, 33, 36, 38, 39, 42, 50
- [109] Thomas Gruber. TagOntology – a way to agree on the semantics of tagging data. <http://tomgruber.org/writing/tagontology.htm>, 2005. Presentation to Tag Camp [last accessed on October 4, 2013]. 36

- [110] Marieke Guy and Emma Tonkin. Folksonomies: Tidying up tags? *D-Lib Magazine*, 12(1), 2006. 17
- [111] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pages 211–220. ACM, 2007. xvii, 20, 28, 31, 123, 144, 146, 148
- [112] Martin J. Halvey and Mark T. Keane. An assessment of tag presentation techniques. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pages 1313–1314. ACM, 2007. 97, 98, 99, 110, 111
- [113] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), 2005. 1, 13
- [114] Paul Harman, Jarred McGinnis, and Paul Wilton. SNaP tag ontology. <http://data.press.net/ontology/tag/>, 2011. 56
- [115] Bernhard Haslhofer, Wolfgang Jochum, Ross King, Christian Sadilek, and Karin Schellner. The lemo annotation framework: weaving multimedia annotations with the web. *International Journal on Digital Libraries*, 10(1):15–32, 2009. 61
- [116] Yusef Hassan-Monteroa and Víctor Herrero-Solanaa. Improving tag-clouds as visual information retrieval interfaces. In *Proceedings of the International Conference on Multidisciplinary Information Sciences and Technologies (InSciT '06)*. INSTAC, 2006. 96, 101
- [117] Susan Havre, Elizabeth Hetzler, Paul Whitney, and Lucy Nowell. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002. 125
- [118] Marti A. Hearst and Daniela Rosner. Tag clouds: Data analysis tool or social signaller? In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS '08)*, page 160:10. IEEE, 2008. 97
- [119] Tom Heath and Enrico Motta. Revyu: Linking reviews and ratings into the web of data. *Web Semantics*, 6(4):266–273, 2008. 38

- [120] Jeff Heflin. OWL web ontology language use cases and requirements. <http://www.w3.org/TR/2004/REC-webont-req-20040210/>, 2004. 35
- [121] Philipp Heim and Steffen Lohmann. ChainGraph: A new approach to visualize shared properties in resource collections. In *Proceedings of the 9th International Conference on Knowledge Management and Knowledge Technologies (I-KNOW '08)*, pages 106–114. TU Graz, 2009. 141
- [122] Philipp Heim, Steffen Lohmann, Kim Lauenroth, and Jürgen Ziegler. Graph-based visualization of requirements relationships. In *Proceedings of the 3rd International Workshop on Requirements Engineering Visualization (REV '08)*, pages 51–55. IEEE, 2008. 167, 173
- [123] Philipp Heim, Steffen Lohmann, and Timo Stegemann. Interactive relationship discovery via the semantic web. In *Proceedings of the 7th International Conference on the Semantic Web: Research and Applications (ESWC '10) – Volume Part I*, volume 6088 of *LNCS*, pages 303–317. Springer, 2010. 174
- [124] Florian Heimerl, Steffen Koch, Harald Bosch, and Thomas Ertl. Visual classifier training for text document retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2839–2848, 2012. 137
- [125] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. Word cloud explorer: Text analytics based on word clouds. In *Proceedings of the 47th Hawaii International Conference on System Science (HICSS '14)*, pages 1833–1842. IEEE, 2014. 95
- [126] Norman Heino, Sebastian Dietzold, Michael Martin, and Sören Auer. Developing semantic web applications with the ontowiki framework. In Tassilo Pellegrini, Sören Auer, Klaus Tochtermann, and Sebastian Schaffert, editors, *Networked Knowledge – Networked Media*, volume 221 of *Studies in Computational Intelligence*, pages 61–77. Springer, 2009. 90
- [127] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004. xiv, 5, 6
- [128] Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical

- Report 775, Stanford InfoLab, 2006. <http://ilpubs.stanford.edu:8090/775/>. 19, 20, 21, 28, 150
- [129] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *Proceedings of the 3rd European Semantic Web Conference (ESWC '06)*, volume 4011 of *LNCS*, pages 411–426. Springer, 2006. 14, 31, 33
- [130] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In *Proceedings of the 1st International Conference on Semantics and Digital Media Technologies (SAMT '06)*, volume 4306 of *LNCS*, pages 56–70. Springer, 2006. 32
- [131] Jane Hunter and Ronald Schroeter. Co-annotate: A system for tagging relationships between multiple mixed-media objects. *IEEE MultiMedia*, 15(3):42–53, 2008. 61
- [132] Alex Iskold, Peter Mika, Vuk Milicic, Scott Montgomerie, Alexandre Passant, Jamie Taylor, and Andraz Tori. Common tag vocabulary. <http://commontag.org/ns>, 2009. 47
- [133] Alex Iskold, Alexandre Passant, Andraz Tori, Peter Mika, Scott Montgomerie, and Vuk Milicic. Commontag mappings. <http://commontag.org/mappings>, 2009. 48
- [134] Alexandar Jaffe, Mor Naaman, Tamir Tassa, and Marc Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval (MIR '06)*, pages 89–98. ACM, 2006. 95
- [135] Johan Janssen and Henk Corporaal. Making graphs reducible with controlled node splitting. *ACM Transactions on Programming Languages and Systems*, 19(6):1031–1052, 1997. 169
- [136] José Kahan, Marja-Riitta Koivunen, Eric Prud'Hommeaux, and Ralph R. Swick. Annotea: An open rdf infrastructure for shared web annotations. In *Proceedings of the 10th International World Wide Web Conference (WWW '01)*, pages 623–632. ACM, 2001. xiv, 57, 58
- [137] Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989. 146

- [138] Owen Kaser and Daniel Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *WWW' 07 Workshop on Tagging and Metadata for Social Information Organization*, 2007. 96
- [139] Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *Proceedings of the 15th International Conference on Multimedia (MULTIMEDIA '07)*, pages 631–640. ACM, 2007. 1
- [140] Roman Kern, Michael Granitzer, and Viktoria Pammer. Extending folksonomies for image tagging. In *Proceedings of the 9th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '08)*, pages 126–129. IEEE, 2008. xvii, 150
- [141] Rohit Khare and Tantek Çelik. Microformats: a pragmatic path to the semantic web. In *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pages 865–866. ACM, 2006. 26, 57
- [142] Hak Lae Kim, John G. Breslin, Sung-Kwon Yang, and Hong-Gee Kim. Social semantic cloud of tag: Semantic model for social tagging. In *Proceedings of the 2nd KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications (KES AMSTA '08)*, volume 4953 of *LNCS*, pages 83–92. Springer, 2008. 42
- [143] Hak-Lae Kim, Stefan Decker, and John G. Breslin. Representing and sharing folksonomies with semantics. *Journal of Information Science*, 36(1):57–72, 2010. 43
- [144] Hak Lae Kim, Alexandre Passant, John G. Breslin, Simon Scerri, and Stefan Decker. Review and alignment of tag ontologies for semantically-linked data in collaborative tagging spaces. In *Proceedings of the 2nd IEEE International Conference on Semantic Computing (ICSC '08)*, pages 315–322. IEEE, 2008. xiv, 35, 43, 44, 45, 63, 64
- [145] Hak Lae Kim, Simon Scerri, John G. Breslin, Stefan Decker, and Hong Gee Kim. The state of the art in tag ontologies: a semantic model for tagging and folksonomies. In *Proceedings of the International Conference on Dublin Core and Metadata Applications (DC '08)*, pages 128–137. DCMI, 2008. 35

- [146] Margaret E.I. Kipp and D. Grant Campbell. Patterns and inconsistencies in collaborative tagging systems: An examination of tagging practices. *Proceedings of the American Society for Information Science and Technology*, 43(1):1–18, 2006. 24
- [147] Graham Klyne and Jeremy J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>, 2004. 26, 35, 83, 86
- [148] Torben Knerr. Tagging ontology – towards a common ontology for folksonomies. <http://tagont.googlecode.com/files/TagOntPaper.pdf>, 2006. [last accessed on October 4, 2013]. xiv, 39, 40, 41, 78
- [149] Torben Knerr. An ontology describing the concept of a tagging. <http://tagont.googlecode.com/files/tagont.owl>, 2007. [last accessed on October 4, 2013]. 41
- [150] Steffen Koch, Harald Bosch, Mark Giereth, and Thomas Ertl. Iterative integration of visual insights during scalable patent search and analysis. *IEEE Transactions on Visualization and Computer Graphics*, 17(5):557–569, 2011. 139
- [151] Marja-Riitta Koivunen. Semantic authoring by tagging with annotea social bookmarks and topics. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW '06)*, volume 209 of *CEUR-WS.org*, 2006. 57, 86
- [152] Marja-Riitta Koivunen, Ralph Swick, and Eric Prud'hommeaux. Annotea shared bookmarks. In *Proceedings of the K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation (Semannot '03)*, volume 101 of *CEUR-WS.org*, 2003. xiv, 57, 58
- [153] Joseph B. Kruskal and Judith B. Seery. Designing network diagrams. In *Proceedings of the 1st General Conference on Social Graphics*, pages 22–50. U. S. Department of the Census, 1980. 168
- [154] Byron Y-L Kuo, Thomas Hentrich, Benjamin M . Good, and Mark D. Wilkinson. Tag clouds for summarizing web search results. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pages 1203–1204. ACM, 2007. 98, 99, 110, 111

- [155] Bongshin Lee, Nathalie H. Riche, Amy K. Karlson, and Sheelash Carpendale. SparkClouds: Visualizing trends in tag clouds. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1182–1189, 2010. 125
- [156] Kangpyo Lee, Hyunwoo Kim, Chungsu Jang, and Hyoung-Joo Kim. Folksoviz: a subsumption-based folksonomy visualization using wikipedia texts. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pages 1093–1094. ACM, 2008. 150
- [157] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. 155
- [158] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004. 133
- [159] Michael Ley. DBLP: Some lessons learned. *Proceedings of the VLDB Endowment*, 2(2):1493–1500, 2009. 118
- [160] Rui Li, Shenghua Bao, Yong Yu, Ben Fei, and Zhong Su. Towards effective browsing of large scale social annotations. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*, pages 943–952. ACM, 2007. 32
- [161] Freddy Limpens. *Multi-points of view semantic enrichment of folksonomies*. PhD thesis, Université de Nice – Sophia Antipolis, 2010. xiv, 53, 55
- [162] Freddy Limpens, Fabien Gandon, and Michel Buffa. SRTag ontology. <http://ns.inria.fr/srtag/2009/01/09/srtag.html>, 2009. 53
- [163] Freddy Limpens, Alexandre Monnin, David Laniado, and Fabien Gandon. NiceTag ontology: Tags as named graphs. In *Proceedings of the 1st International Workshop on Social Networks Interoperability*, 2009. 53
- [164] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. TIARA: Interactive, topic-based visual text summarization and analysis. *ACM Transactions on Intelligent Systems and Technology*, 3(2):25:1–25:28, 2012. 125

- [165] Steffen Lohmann. Learners as metadata source? A three-layer model of collaborative content annotation. In Max Mühlhäuser, Werner Sesink, Andreas Kaminski, and Jürgen Steimle, editors, *Interdisciplinary Approaches to Technology-enhanced Learning*, pages 279–286. Waxmann, 2011. 11
- [166] Steffen Lohmann. Semantic models for interactive systems: The case of tagging and folksonomies. In Tim Hussein, Heiko Paulheim, Stephan Lukosch, Jürgen Ziegler, and Gaëlle Calvary, editors, *Semantic Models for Adaptive Interactive Systems*, Human-Computer Interaction Series, pages 169–185. Springer, 2013. 11, 35, 65
- [167] Steffen Lohmann, Michael Burch, Hansjörg Schmauder, and Daniel Weiskopf. Visual analysis of microblog content using time-varying co-occurrence highlighting in tag clouds. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*, pages 753–756. ACM, 2012. 95
- [168] Steffen Lohmann and Paloma Díaz. Representing and visualizing folksonomies as graphs: a reference model. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*, pages 729–732. ACM, 2012. 11, 141
- [169] Steffen Lohmann, Paloma Díaz, and Ignacio Aedo. MUTO: the modular unified tagging ontology. In *Proceedings of the 7th International Conference on Semantic Systems (I-SEMANTICS '11)*, pages 95–104. ACM, 2011. 35, 65, 66, 76, 79, 81
- [170] Steffen Lohmann, Sebastian Dietzold, Philipp Heim, and Norman Heino. A web platform for social requirements engineering. In *Software Engineering 2009 – Workshopband*, volume 150 of *LNI*, pages 309–315. GI, 2009. 90
- [171] Steffen Lohmann, Philipp Heim, Sören Auer, Sebastian Dietzold, and Thomas Riechert. Semantifying requirements engineering – the softwiki approach. In *Proceedings of the 4th International Conference on Semantic Systems (I-SEMANTICS '08)*, pages 182–185. TU Graz, 2008. 88
- [172] Steffen Lohmann, Philipp Heim, and Kim Lauenroth. Web-based stakeholder participation in distributed requirements elicitation. In

- Proceedings of the 16th IEEE International Requirements Engineering Conference (RE '08)*, pages 323–324. IEEE, 2008. 88, 90
- [173] Steffen Lohmann, Philipp Heim, Timo Stegemann, and Jürgen Ziegler. The relfinder user interface: interactive exploration of relationships between objects of interest. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10)*, pages 421–422. ACM, 2010. 174
- [174] Steffen Lohmann, Philipp Heim, Lena Tetzlaff, Thomas Ertl, and Jürgen Ziegler. Exploring relationships between annotated images with the chaingraph visualization. In *Proceedings of the 4th International Conference on Semantic and Digital Media Technologies (SAMT '09)*, volume 5887 of *LNCS*, pages 16–27. Springer, 2009. 141
- [175] Steffen Lohmann and Thomas Riechert. Adding semantics to social software engineering: (re-)using ontologies in a community-oriented requirements engineering environment. In *Software Engineering 2010 – Workshopband*, volume 160 of *LNI*, pages 485–494. GI, 2010. 65, 88, 91
- [176] Steffen Lohmann, Jürgen Ziegler, and Lena Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT '09), Part I*, volume 5726 of *LNCS*, pages 392–404. Springer, 2009. 17, 95
- [177] Erkki Mäkinen. How to draw a hypergraph. *International Journal of Computer Mathematics*, 34(3):177–185, 1990. 141
- [178] Ching man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. Discovering and modelling multiple interests of users in collaborative tagging systems. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology – Workshops*, pages 115–118. IEEE, 2008. 31
- [179] Frank Manola and Eric Miller. RDF primer. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, 2004. 83
- [180] Andrea Marchetti, Maurizio Tesconi, Francesco Ronzano, Marco Rosella, and Salvatore Minutoli. Semkey: A semantic collaborative

- tagging system. In *Proceedings of WWW 2007 Workshop on Tagging and Metadata for Social Information Organization*, 2007. 18, 19, 20
- [181] Mitchell P. Marcus, Mary A. Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993. 131
- [182] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. HT06, tagging paper, taxonomy, flickr, academic article, to read. In *Proceedings of the 17th ACM Conference on Hypertext and Hypermedia (HYPERTEXT '06)*, pages 31–40. ACM, 2006. 20, 28
- [183] Adam Mathes. Folksonomies – cooperative classification and communication through shared metadata. <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>, 2004. [last accessed on October 4, 2013]. 15, 16, 18, 24
- [184] Jarred McGinnis, Paul Wilton, Paul Harman, and John O’Donovan. Snap. 57
- [185] Deborah L. McGuinness and Frank van Harmelen. OWL web ontology language overview. <http://www.w3.org/TR/owl-features/>, 2004. 35
- [186] Chirag Mehta. US presidential speeches tag cloud. <http://chir.ag/projects/preztags>, 2007. [last accessed on October 4, 2013]. 32, 95, 125
- [187] Elke Michlmayr and Steve Cayzer. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *Proceedings of WWW 2007 Workshop on Tagging and Metadata for Social Information Organization*, 2007. xvii, 148, 149
- [188] Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the 4th International Semantic Web Conference 2005 (ISWC '05)*, volume 3729 of *LNCS*, pages 522–536. Springer, 2005. 20, 21, 22, 31, 141
- [189] Alistair Miles and Sean Bechhofer. SKOS simple knowledge organization system reference. <http://www.w3.org/TR/skos-reference/>, 2009. 66
- [190] David R. Millen and Jonathan Feinberg. Using social tagging to improve social navigation. In *Proceedings of AH 2006 Workshop on*

the Social Navigation and Community-Based Adaptation Technologies, 2006. 21, 28

- [191] George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 19, 77
- [192] Alexandre Monnin, Freddy Limpens, Fabien Gandon, and David Laniado. Speech acts meet tagging: NiceTag ontology. In *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS '10)*, pages 31:1–31:10. ACM, 2010. 53, 54, 55
- [193] Alexandre Monnin, Freddy Limpens, David Laniado, and Fabien Gandon. Nice tag ontology. <http://ns.inria.fr/nicetag/2010/09/09/voc>, 2010. 53
- [194] Stefan Negru, Florian Haag, and Steffen Lohmann. Towards a unified visual notation for OWL ontologies: Insights from a comparative user study. In *Proceedings of the 9th International Conference on Semantic Systems (I-SEMANTICS '13)*, pages 73–80. ACM, 2013. xv, 65, 92, 93
- [195] Stefan Negru and Steffen Lohmann. A visual notation for the integrated representation of OWL ontologies. In *Proceedings of the 9th International Conference on Web Information Systems and Technologies (WEBIST '13)*, pages 308–315. SciTePress, 2013. 91, 92
- [196] Richard Newman. Tag ontology. <http://www.holygoat.co.uk/owl/redwood/0.1/tags/>, 2005. 39, 73, 78
- [197] Richard Newman. Tag ontology design. <http://www.holygoat.co.uk/projects/tags>, 2005. [last accessed on October 4, 2013]. 36, 38, 86
- [198] Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*. World Scientific Publishing, 2004. 168
- [199] Natasha Noy, Alan Rector, Pat Hayes, and Chris Welty. Defining n-ary relations on the semantic web. <http://www.w3.org/TR/swbp-n-aryRelations/>, 2006. [last accessed on October 4, 2013]. 69
- [200] Forrest Oliphant. Taggraph. <http://taggraph.com>, 2005. xvii, 149, 150

- [201] OMG. Ontology definition metamodel. <http://www.omg.org/spec/ODM/1.0>, 2009. 67, 91
- [202] Alexandre Passant. Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs. In *Proceedings of the 1st International Conference on Weblogs and Social Media (ICWSM 07)*, 2007. 19, 20
- [203] Alexandre Passant. MOAT ontology. <http://moat-project.org/ns#>, 2008. 46
- [204] Alexandre Passant and Philippe Laublet. Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data. In *Proceedings of the WWW2008 Workshop on Linked Data on the Web (LDOW '08)*, volume 369 of *CEUR-WS.org*, 2008. 45
- [205] Isabella Peters. *Folksonomies: Indexing and Retrieval in Web 2.0*. De Gruyter, 2009. 11, 13, 15, 16, 20
- [206] Isabella Peters and Katrin Weller. Tag gardening for folksonomy enrichment and maintenance. *Webology*, 5(3), 2008. 19
- [207] Emanuele Quintarelli. Folksonomies: power to the people. <http://www.iskoi.org/doc/folksonomies.htm>, 2005. [last accessed on October 4, 2013]. 14
- [208] Asarnusch Rashid and Steffen Lohmann. Fostering remote user participation and integration of user feedback into software development. In *Proceedings of the 1st Workshop on the Interplay between Usability Evaluation and Software Development (I-USED '08)*, volume 407 of *CEUR-WS.org*. 88
- [209] Thomas Riechert, Steffen Lohmann, Jens Lehmann, Kim Lauenroth, Philipp Heim, and Sebastian Tramp. Semantic web ontology for requirements engineering (SWORE). <http://ns.softwiki.de/req/>, 2012. xv, 88, 89, 174
- [210] A. Walkyria Rivadeneira, Daniel M. Gruen, Michael J. Muller, and David R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*, pages 995–998. ACM, 2007. 97, 98, 99, 110, 111

- [211] Carsten Saathoff and Ansgar Scherp. M3O: The multimedia meta-data ontology. In *10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies (SeMuDaTe '09)*, volume 539 of *CEUR-WS.org*, 2009. 61
- [212] Harald Sack and Jörg Waitelonis. Integrating social tagging and document annotation for content-based search in multimedia data. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW '06)*, volume 209 of *CEUR-WS.org*, 2006. 21
- [213] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing (CICLing '02)*, volume 2276 of *LNCS*, pages 1–15. Springer, 2002. 128
- [214] Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. Open annotation core data model. <http://www.openannotation.org/spec/core/20120509/index.html>, 2012. xiv, 60
- [215] Robert Sanderson, Paolo Ciccarese, and Herbert Van de Sompel. Open annotation data model. <http://www.openannotation.org/spec/core/>, 2013. 59
- [216] Robert Sanderson and Herbert Van de Sompel. Open annotation: Beta data model guide. <http://www.openannotation.org/spec/beta/>, 2011. 59
- [217] Simon Scerri, Michael Sintek, Ludger van Elst, and Siegfried Handschuh. NEPOMUK annotation ontology specification. <http://www.semanticdesktop.org/ontologies/2007/08/15/nao>, 2007. 60
- [218] Johann Schrammel, Michael Leitner, and Manfred Tscheligi. Semantically structured tag clouds: an empirical evaluation of clustered presentation approaches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*, pages 2037–2040. ACM, 2009. 96
- [219] Christin Seifert, Barbara Kump, Wolfgang Kienreich, Gisela Granitzer, and Michael Granitzer. On the beauty and usability of tag clouds. In *Proceedings of the 12th International Conference Information Visualization (IV '08)*, pages 17–25. IEEE, 2008. 96

- [220] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work (CSCW '06)*, pages 181–190. ACM, 2006. 24
- [221] Blake Shaw. Semidefinite embedding applied to visualizing folksonomies. <http://www.metablake.com/advml/adv-ml-project.pdf>, 2005. [last accessed on October 4, 2013]. xvii, 148, 149
- [222] Mark Shepherd. Springgraph flex component. <http://mark-shepherd.com/blog/springgraph-flex-component/>, 2006. [last accessed on October 4, 2013]. 158
- [223] Jeong S. Sim, Costas S. Iliopoulos, Kunsoo Park, and William F. Smyth. Approximate periods of strings. *Theoretical Computer Science*, 262(1):557–568, 2001. 156
- [224] Edwin Simpson. Clustering tags in enterprise and web folksonomies. In *Proceedings of the 2nd International Conference on Weblogs and Social Media (ICWSM '08)*. AAAI, 2008. 150
- [225] James Sinclair and Michael Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29, 2008. 97, 98, 99, 111, 116, 118, 139
- [226] Gene Smith. *Tagging: People-Powered Metadata for the Social Web*. New Riders, 2008. 1, 15, 16, 20, 21, 22, 24, 26, 29, 32, 95, 123
- [227] Peter Spyns, Aldo de Moor, Jan Vandenbussche, and Robert Meersman. From folksologies to ontologies: How the twain meet. In *Proceedings of the 2006 OTM Confederated International Conferences (OTM '06), Part I*, volume 4275 of *LNCS*, pages 738–755. Springer, 2006. 13
- [228] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7(2):118–132, 2008. 139
- [229] Moritz Stefaner. Visual tools for the socio-semantic web. University of Applied Sciences Potsdam, 2007. Master thesis. xvii, 28, 146, 147

- [230] Martin Szomszor, Harith Alani, Ivan Cantador, Kieron O’Hara, and Nigel Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. In *Proceedings of the 7th International Semantic Web Conference (ISWC ’08)*, volume 5318 of *LNCS*, pages 632–648. Springer, 2008. 51
- [231] Martin Szomszor, Ciro Cattuto, Wouter Van den Broeck, Alain Barrat, and Harith Alani. Semantics, sensors, and the social web: The live social semantics experiments. In *Proceedings of the 7th Extended Semantic Web Conference (ESWC ’10)*, volume 6089 of *LNCS*, pages 196–210. Springer, 2010. 51
- [232] Vlad Tanasescu and Olga Streibel. Extreme tagging: Emergent semantics through the tagging of tags. In *Proceedings of the 1st International Workshop on Emergent Semantics and Ontology Evolution (ESOE ’07)*, volume 292 of *CEUR-WS.org*, pages 84–94, 2007. 86
- [233] Dennis Thom, Harald Bosch, Steffen Koch, Michael Wörner, and Thomas Ertl. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. In *Proceedings of the 2012 IEEE Pacific Visualization Symposium (PacificVis ’12)*, pages 41–48. IEEE, 2012. 123, 125
- [234] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL ’03) – Volume 1*, pages 173–180. ACL, 2003. 132
- [235] Karen H. L. Tso-Sutter, Leandro B. Marinho, and Lars Schmidt-Thieme. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC ’08)*, pages 1995–1999. ACM, 2008. 29
- [236] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition, 2001. 116
- [237] Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. <http://www.vanderwal.net/random/entrysel.php?blog=1635>, 2005. [last accessed on October 4, 2013]. 28

- [238] Thomas Vander Wal. Folksonomy definition and wikipedia. <http://vanderwal.net/random/entrysel.php?blog=1750>, 2005. [last accessed on October 4, 2013]. 14
- [239] Thomas Vander Wal. Folksonomy explanations. <http://vanderwal.net/random/entrysel.php?blog=1622>, 2005. [last accessed on October 4, 2013]. 14
- [240] Thomas Vander Wal. Folksonomy coinage and definition. <http://vanderwal.net/folksonomy.html>, 2007. [last accessed on October 4, 2013]. 1, 13, 14, 30, 39, 83
- [241] Thomas Vander Wal. Wikipedia folksonomy is a mess with collaborative misunderstanding. <http://vanderwal.net/random/entrysel.php?blog=1949>, 2007. [last accessed on October 4, 2013]. 11
- [242] Fernanda B. Viégas and Martin Wattenberg. Tag clouds and the case for vernacular visualization. *interactions*, 15(4):49–52, 2008. 95
- [243] Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. ManyEyes: a site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, 2007. 118, 140
- [244] Jakob Voss. Tagging, folksonomy & co-renaissance of manual indexing? *CoRR*, abs/cs/0701072, 2007. 27
- [245] Romain Vuillemot, Tanya Clement, Catherine Plaisant, and Amit Kumar. What’s being said near “martha”? exploring name entities in literary text collections. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology (VAST ’09)*, pages 107–114. IEEE, 2009. 140
- [246] W3C. Use international date format (ISO). <http://www.w3.org/QA/Tips/iso-date>. 72
- [247] Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002. 150
- [248] Martin Wattenberg and Fernanda B. Viégas. The word tree, an interactive visual concordance. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1221–1228, 2008. 120

- [249] Marcos Weskamp. Flickr graph. <http://marumushi.com/projects/flickrgraph>. [last accessed on October 4, 2013]. 150
- [250] Ryen W. White and Resa A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm*. Morgan & Claypool, 2009. 160
- [251] Misha Wolf and Charles Wicksteed. Date and time formats. <http://www.w3.org/TR/NOTE-datetime>, 1997. 72
- [252] Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *Proceedings of the 17th Conference on Hypertext and Hypermedia (HYPERTEXT '06)*, pages 111–114. ACM, 2006. 21
- [253] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pages 417–426. ACM, 2006. 32
- [254] Yingcai Wu, Furu Wei, Shixia Liu, Norman Au, Weiwei Cui, Hong Zhou, and Huamin Qu. OpinionSeer: Interactive visualization of hotel customer feedback. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1109–1118, 2010. 139, 140
- [255] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the WWW2006 Workshop on Collaborative Web Tagging*, 2006. 24
- [256] Michael Fried Sin-Jae Kang Zhixian Yan Ying Ding, Ioan Toma. Integrating social tagging data: Upper tag ontology (UTO). In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2008 (SMC '08)*, pages 460–466. IEEE, 2008. 48, 49, 50
- [257] Olivier Zitvogel. Delicious soup. <http://www.zitvogel.com/deliciousoup>, 2005. [last accessed on October 4, 2013]. xvii, 146, 147