Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Document
D-08-01

Readings on Knowledge Management

Vol. 1, 2008

Andreas Dengel

Marcus Liwicki

Thomas Roth-Berghofer (Eds.)

August 2008

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 2080
D-67608 Kaiserslautern
Tel: +49 (631) 205-75 0
Fax: +49 (631) 205-75 503

Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Tel: +49 (681) 302-5151
Fax: +49 (681) 302-5341

Robert-Hooke-Str. 6
D-28359 Bremen
Tel.: +49 (421) 218-64 100
Fax.: +49 (421) 218-64 150

E-Mail: Hinfo@dfki.deH WWW: http://www.dfki.de

Deutsches Forschungszentrum für Künstliche Intelligenz

DFKI GmbH

German Research Center for Artificial Intelligence

Founded in 1988, DFKI today is one of the largest nonprofit contract research institutes in the field of innovative software technology based on Artificial Intelligence (AI) methods. DFKI is focusing on the complete cycle of innovation - from world-class basic research and technology development through leading-edge demonstrators and prototypes to product functions and commercialization.

Based in Kaiserslautern, Saarbrücken and Bremen, the German Research Center for Artificial Intelligence ranks among the important „Centers of Excellence“ worldwide.

An important element of DFKI's mission is to move innovations as quickly as possible from the lab into the marketplace. Only by maintaining research projects at the forefront of science DFKI has the strength to meet its technology transfer goals.

The key directors of DFKI are Prof. Wolfgang Wahlster (CEO) and Dr. Walter Olthoff (CFO).

DFKI's research departments are directed by internationally recognized research scientists:

- Image Understanding and Pattern Recognition (Prof. Dr. T. Breuel)
- Knowledge Management (Prof. Dr. A. Dengel)
- Agents and Simulated Reality (Prof. Dr. P. Slusallek)
- Language Technology (Prof. Dr. H. Uszkoreit)
- Intelligent User Interfaces (Prof. Dr. W. Wahlster)
- Institute for Information Systems at DFKI (Prof. Dr. P. Loos)
- Robotics (Prof. Dr. F. Kirchner.)
- Safe and Secure Cognitive Systems (Prof. Dr. B. Krieg-Brückner)
- Augmented Reality (Prof. Dr. D. Stricker)

and the associated Center for Human Machine Interaction (Prof. Dr.-Ing. Detlef Zühlke)

In this series, DFKI publishes research reports, technical memos, documents (eg. workshop proceedings), and final project reports. The aim is to make new results, ideas, and software available as quickly as possible.

Prof. Dr. Wolfgang Wahlster
Director

Readings on Knowledge Management

Vol. 1, 2008

Andreas Dengel, Marcus Liwicki, and Thomas Roth-Berghofer (Eds.)

DFKI-D-08-01

ISSN 0946-0098

Preface

Readings on Knowledge management is a collection of papers that are written for the Seminar Knowledge Management (KM) being part of the master program in Computer Science at the University of Kaiserslautern. The Seminar is held at DFKI in Kaiserslautern in summer 2008.

Our intention was to provide a forum in which students may be introduced into scientific work as it is a matter of fact when publishing research papers at international conferences or workshops. Consequently, students had to investigate defined topics and write papers following pre-given guidelines. We installed a program committee consisting of the supervisor team and the students participating in the seminar. The individual contributions (submissions) have been peer-reviewed using the criteria that are common in international research communities. This reviewing process not only increases the quality of the contributions by giving rich feedback to the authors but makes the seminar similar to a workshop broadening the experience of the students.

This year we have accepted seven articles that are presented in a workshop-like session. They are all well written and describe state-of-the-art approaches the area of knowledge management.

We hope that other researches may profit from this collection and like to thank all authors for their collaboration and their excellent contributions.

Prof. Dr. Andreas Dengel
Dr. Marcus Liwicki
Dr. Thomas Roth-Berghofer

Reviewers

Benjamin Adrian
Marko Brunzel
Manuel Möller
Darko Obradovic
Leo Sauermann
Rafael Schirru
Sven Schwarz
Jens Göddel
Alexander Grothkast
Jörn Hees
Volker Hudlet
Markus Rahm
Daniel Schall
Markus Weber

Table of Contents

A survey of Semantic Annotations for Knowledge Management

Markus Weber

Department of Computer Science
Kaiserslautern University of Technology, Germany m_weber2@cs.uni-kl.de

Abstract. This paper discusses the need in knowledge management to enrich documents with meta data and to combine them with existing domain knowledge. To link existing knowledge to a document, semantic annotations are introduced. Therefore the foundations of semantic annotations are clarified and several approaches, such as annotation frameworks, semantic wikis and a paper-based approach, are discussed.

1 Introduction

In knowledge management (KM), 85% of the information is unstructured, 30% of people's time is spent for searching for relevant information and 45% of a manager's time is spent for working with documents. Finally, according to IDC[1] the volume of data in company networks grew from 3.200 petabyte in the year 2000 to 54.000 petabyte in 2004 [1]. Thus there is a larger amount of unstructured data which is hard to maintain and for people impossible to browse without an appropriate tool support. Handling this unstructured data is a big issue in KM systems.

In order to capture knowledge, using markup techniques and supporting semantic annotations are major techniques for creating meta-data. Semantic annotations represent a specific sort of meta-data which provides references to entities in the form of URIs or other types of unique identifiers[2]. An example for an annotated document is illustrated in figure 1, which shows a document where entities in the text are linked with a unique identity in a semantic repository.

It is difficult to completely process the content of all kinds of documents. Even with technologies based on natural language processing, image processing, machine vision and speech recognition. Thus Semantic Annotations are one of the promising methodologies to define semantic structures on the content.

The use of meta data is the traditional way to add some context information to a document, such as the author and keywords, which summarize the content. An example for this kind of meta-data is the Dublin Core Metadata Initiative®[3], which is a standard for cross-domain information resource description.

[1] International Data Corporation (IDC) - IDC is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications, and consumer technology markets.
[2] http://www.ontotext.com/kim/semanticannotation.html
[3] http://www.dublincore.org/

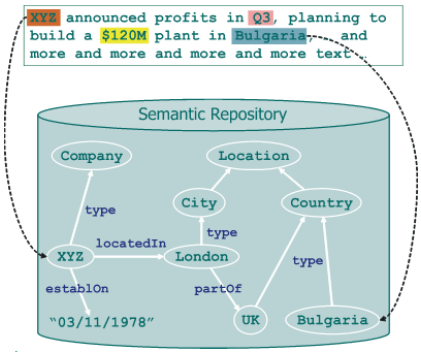

Fig. 1. Semantic annotations.

But this kind of meta data is not sufficient if the document should be machine-readable. The Web is faced with the same problems, as the number of Web pages and hereby the information is growing every day as well. Referring to this situation the Semantic Web is a vision to structure data and make knowledge reusable. The Semantic Web Activity Statement[2] points out the main idea of the Semantic Web:

> "The Web can reach its full potential only if it becomes a place where data can be shared and processed by automated tools as well as by people. For the Web to scale, tomorrow's programs must be able to share and process data even when these programs have been designed totally independently. The Semantic Web is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications."

Thus Semantic Web technologies can be a good approach to structure the data in KM as well.

The purpose of this paper is to give a survey of different state of the art approaches for semantic annotations. In Section 2 the relevance for knowledge management will be discussed. Section 3 describes the foundations of semantic annotations and introduces annotation frameworks and tools. The last sections take a look at other approaches where semantic annotations are used to enrich the content of documents, semantic wikis and a paper-based approach.

2 Knowledge management

Nonaka and Takeuchi stated [3] what a successful KM program needs. On the one hand, to convert internalized tacit knowledge into explicit codified knowledge in order to share it. On the other hand, it also must permit individuals and groups to internalize and build up implicit knowledge they have retrieved from the KM system.

Figure 2 illustrates a scenario in KM, a company with employees who are the users of the internal KM system. Those users possess implicit knowledge. For Nonaka and Takeuchi, the key to build up new knowledge is explication of the implicit knowledge. In other words, users produce documents, explicit knowledge, which are added to the document collection in the KM system and shared in the system.

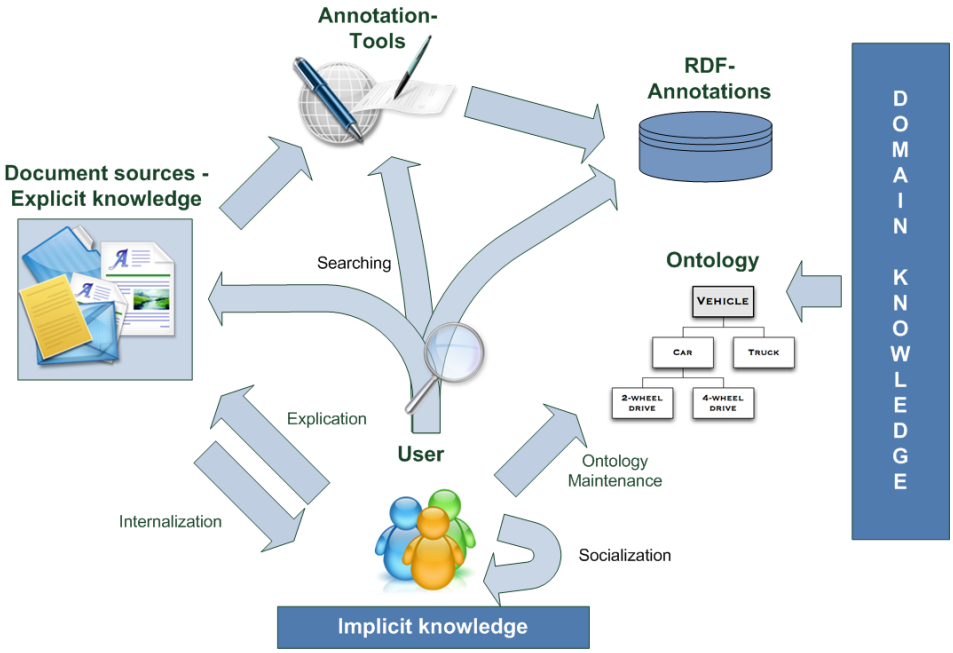

Fig. 2. Knowledge management systems

As there is communication between users, they share their implicit knowledge with each other (socialization). Furthermore by internalization, which means consuming documents, users can also increase their own knowledge. As the number of documents increases over time the need is raised to implement an effective search mechanism, a semantic search [4], in order to receive better search results.

The first step to achieve this goal is to encode the domain knowledge in a standardized way which is also machine-readable. Ontologies can be used for this purpose, so they have to be created and maintained by experts with domain expertise. A framework for a KM system using Semantic Web technologies is introduced in [5].

But one problem still exists. Somehow the instances of the ontologies (e.g. encoded as RDF) have to be linked with the according words or text phrase in the documents. Those annotations will by stored either in a separate RDF storage or in the document itself. So there is a need for supporting annotation

tools which can handle the various types of documents. For instance web-native documents, word processor documents and so on.

The following section investigates semantic annotations. Requirements which result from the scenario will be introduced. Because of the different document formats, there is need to define appropriate persistent types for annotations. Furthermore three representative examples for annotation tools will be presented.

3 Semantic annotations

This section will elaborate the foundations of semantic annotations, such as the requirements they have to meet and the different persistent types of annotations. Furthermore several annotation tools are introduced.

3.1 Requirements

Uren and Cimiano formulated [6] seven requirements for Semantic Annotation systems. Those requirements result from the document centric scenario which is illustrated in figure 2. Four viewpoints have to be considered on this task: the ontologies, the documents, the annotations which link the documents with the ontologies and the users of the system.

The seven requirements are listed in the following:

1. **Standard formats.** This requirement refers to the RDF-Annotations and ontology shown in the scenario. For this issue re-usability is an important factor, as for instance ontologies package existing domain knowledge of the company. Using not-standardized formats complicates to introduce new software tools or guaranty future proof. Furthermore the possibility to share the format in an heterogeneous environment and to access the resources simultaneously improves the annotation process. Another advantage is that there is no constraint to a specific proprietary format respectively a knowledge management system. A standard is W3C's RDF annotation schema[7] and for ontologies the Web Ontology Language OWL[8].
2. **User centered/collaborative design.** In order to improve the process of annotating, the user interface of the annotation tool has to be user friendly. As it could become a bottleneck if the annotation process is to complex and time-consuming. Therefore a good approach is to integrate the annotation support into the user interface, where they create, read, share and edit the resources. Also the system design should consider collaboration between the users, to assure the reuse of the intelligent documents. But it always has to be considered that this exchange has to happen in a trusted environment, if there is confidential knowledge involved, such as a medical environment with patient data.
3. **Ontology support.** Annotation tools should support appropriate formats and multiple ontologies, for instance if there has to be distinguished between two different types of ontologies. Like general meta-data for patients and

technical ontologies for diagnosis in a medical context. It has to be merged or declared to which context they belong to. Further on changes over time, such as adding or modifying existing classes, should not lead to an inconsistent state between the ontologies and the annotations.

4. **Support of heterogeneous document formats.** The document sources, explicit knowledge, in our scenario can exist in different formats, such as web-native formats, word processor file formats, spreadsheets, graphics files, etc.. For this reason annotation tools have to integrate in the existing work practice. For instance a Web editor or a word processor should provide an annotation tool/plug-in to enrich the document with annotations while the user is editing the document. Thus the editing and annotating process becomes one process.
5. **Document evolution.** Besides changing ontologies, the documents sources are in an evolutionary process. So maintenance of the consistency between annotations and documents is a problem. Hence the annotation environment has to inform and help the knowledge worker to maintain the appropriate annotations while the document changes. Again the advantage of including the annotation tools in the editing environment seems to be a good approach.
6. **Annotation storage.** Depending on the environment, the storage of the annotations is handled differently. The model of the Semantic Web decouples the document content and the semantics, whereas word processor models store those kind of information in the document itself. Thus for KM both models have to be considered because both of them are used. The following section will discuss different persistent types for annotations.
7. **Automation.** In enterprise environment larger document collections can be found, so an automated mark-up of documents is vital. In order to achieve this, knowledge extraction technologies have to be part of the annotation environment.

Those requirement have to be considered when annotation tools are designed. As mentioned in the annotation storage requirements, there are different persistent types of annotations [9] respectively ways to store annotations. Those persistent types will be discussed in the following section.

3.2 Persistent types of semantic annotations

The document sources occur as different types of documents, such as web sources, word processor documents and so on. Thus we need different techniques to store annotations for documents. As for instance web sources could have the restriction that there is no write access to the document.

- **Embedded annotations.** This technique embeds the annotations in the document itself. The advantage of this technique is that maintaining and exchanging of the semantic annotations is easier as the tools does not have to access an external resource. But it has to be considered that a semantic search has to extract the annotation from the document before performing a query on it.

Embedded annotations can be used in an environment where write access is granted to the source document. An example is a word processor file or a wiki page where annotations can be added to the original source document.
- **Intrinsic annotations.** In contrast to embedded annotations, intrinsic annotations are stored in an external resource and linked inside the document. So as an advantage annotations do not have to be extracted from the document to perform for instance a semantic search. Thus the assumption is made that the software which accesses the document also can access the RDF data in the external file. Hence offline work is a problem. Furthermore write access is needed to store the links inside the document.
- **Extrinsic annotations.** This techniques takes into account that a user does not always have write permission on a document, for instance a document which is located on a web-server. Hence the meta data has to be stored in an external file and it has to be linked to the document. The link, for instance RDF-Annotations, points to a certain fragment of the document.

Those persistent type are used in annotations tools with respect to the source documents and their restrictions. The following section introduces three typical annotation tools which are used in KM.

3.3 Annotation tools

Having defined the requirements for the annotation tools and different persistent types, three frameworks will be introduced. These frameworks and their tools support the user with the task of linking the documents with the ontologies. They help to improve the process of explication, mentioned in Section 2, as they enrich the documents with semantic information. The chosen frameworks are typical examples of annotation tools used in KM.

Annotea. Annotea is a W3C project under Semantic Web Advanced Development (SWAD)[6] [9] which is an infrastructure for the Web. Because it is not possible to gain write access for all document sources, extrinsic annotations have to be used to annotate documents. Annotea uses the XPointer method to locate annotations within a document. The format used for the annotations is RDF. Those RDF-annotations are stored either on an annotation server or on the local machine.

The Annotea framework is only semiformal as the annotations are intended for users. As its annotations are free text statements, they are less machine-readable. The Annotea framework is used by a number of tools, such as Amaya, Annozilla and Vannotea, which are discussed in detail in [6].

The majority of the requirements defined in Section 3.1 are fulfilled within Annotea. For instance Amaya uses standard formats like RDF(S) XLink[4] and XPointer[10] for linkage of text positions. These techniques are standardized. Furthermore an implementation as Web browser & editor is a good approach as

[4] http://www.edition-w3c.de/TR/2001/REC-xlink-20010627/

it integrates the annotation support within working environment. Also ontology support is realized by using an annotation server. Further on Amaya supports several Web source document standards, such as HTML, XHTML and XML. In order to support document evolution XPointer are used to define where the metadata should be attached to the document. As already mentioned annotations can be stored on the local machine or on an annotation server. Thus there is a decoupling of the document context and the semantics. Automation is not offered, thus all annotations have to be done manually.

CREAM. The CREAM framework is an annotation framework developed by the University of Karlsruhe [6]. It specifies an annotation interface, with automatic extraction of annotations, a document management system and an annotation inference server. XPointers provide the linkage between annotations and the position in the text.

In contrast to Annotea the deep web is also considered. The databases which are used to generate Deep Web pages can be annotated, so that annotations are generated automatically with the pages. The storage model allows the user to decide whether annotations are stored on a separate server as an extrinsic annotation or as embedded annotation within a web page.

OntoMat is a reference implementation of the CREAM framework. OntoMat uses standard formats, such as OWL or XPointer. In order to improve usability OntoMat comes up with features like drag& drop to create instances. Ontology support is realized on a OntoBroker[5] annotation server. Supported document formats are HTML and Deep Web. By using XPointers the consistency of annotations to the document can be maintained. In contrast to Annotea, OntoMat uses an approach called Pattern-based Annotation through Knowledge on the Web (PANKOW)[11] to provide automation.

KIM-Platform. The Knowledge & Information Management[6] (KIM) Platform offers more than just semantic annotations, it also provides services for automatic semantic annotation, indexing and retrieval of documents [12]. The semantic annotations are stored in a knowledge base in the form of named entities (people, places, etc.) using KIM ontologies. There are various plug-ins for front-ends, such as Microsoft's Internet Explorer, available. Thus just HTML documents are supported.

Some of the annotation tools act as plug-ins to provide a possibility to annotate Web documents while editing or reading. The following section introduces a different approach, Semantic Wikis. They act as a whole environment to create, edit and annotate documents.

[5] Ontobroker is a deductive, object oriented database system that has originally been developed as a research prototype at the AIFB Karlsruhe as part of the Semantic Web initiative.

[6] http://www.ontotext.com/kim/

4 Semantic Wiki software

The next approach for the usage of semantic annotations is a Semantic Wiki. Traditional wikis offer collaborative authoring functionality and because of the simplicity of the wiki syntax their usage is popular. The high acceptance of wiki software makes it a good candidate for semantic extensions.

There is a lack of structure in the data, as almost all information in the content is written in natural language. Thus the data is not machine-readable. An example for knowledge reuse would be if a wiki page contains the information, that Dan Brown is the author of "The Da Vinci Code" and Doubleday is his publisher. The book "The Da Vinci Code" should automatically appear on the wiki page of Dan Brown as the author and on the publishers page as published books. Furthermore it should be possible for the user to query "Who is the author of The Da Vinci Code?". To solve those requirements, meta-data has to be assigned [13] again.

Using semantic technologies, Semantic Wikis try to structure the content in wikis and make it maintainable [14]. Therefore Semantic Wikis enable users to additionally describe resources in a formal language like RDF or OWL. One approach is to define semantic annotations in the same way as layout and structural descriptives, so for users the authoring effort is the same as for handling the layout. Hence even less trained people should be able to deal with Semantic Wikis. Furthermore it is possible to exchange data with external applications, for instance an external search. The benefit is obvious, by enriching the resources with meta data, such as "Who is the author of "Da Vinci Code"?", users can search in the system for all authors.

The usage of wiki software already has been established in KM, for instance they are used for project organization. The knowledge about software projects in many companies is shared in wikis (documentation, project plans, bug tickets and so on) [15]. Furthermore there are studies about the usage of Semantic Wikis in KM [13], such as KIWI (Knowledge in a wiki)[7] and NEPOMUK (Social Semantic Desktop)[8].

After introducing the basic ideas of Semantic Wikis, the next section will introduce several examples of Semantic wiki systems.

4.1 Overview Semantic Wiki projects

There are already several Semantic Wiki systems available [15] and most of them are still in a prototypical state. To give a short overview, a few important systems will be mentioned.

Semantic Media Wiki, which is an extension of Media Wiki, has the focus on encyclopedia purpose. It does not demand a certain annotation schema, so users can add annotations even if there is no schema available. Those annotations are embedded in the wiki text.

[7] http://www.kiwi-project.eu/
[8] http://nepomuk.semanticdesktop.org/

Ikewiki is a Semantic Wiki which was developed for KM as a tool for collaborative development of ontologies and the focus is on a wide semantic support for the user. IkeWiki supports the usage and the editing of Web Ontology Language (OWL). The inference engines OWL-RDFS and OWL-DL are currently supported.

SemperWiki [13] is a semantic desktop wiki for personal KM. **OntoWiki** has its focus on offering an easy to handle collaborative interface to maintain ontologies. Furthermore it provides a semantic search and navigation.

Kaukolu[14] differs from the existing Semantic Wiki system. The system is based on JSPWiki[9] and Sesame[10]. In comparison to other wikis, Kaukolu allows to formulate arbitrary RDF by extending the wiki syntax. the subject of the RDF is not required to represent the URI of the page. Furthermore Kaukolu allows to import RDF and RDF Schema, as it is also represented in RDF. So ontologies can be edited using Kaukolu. Beyond it supports using arbitrary strings instead of URIs or labels of the respective property or instance. In order to support the user with entering RDF triples, Kaukolu provides auto-completion. An overview of state of the art Semantic Wikis can be found on [16].

Concerning the requirements, Semantic Wiki systems are also a promising approach for semantic annotation systems. The introduced systems all use standard formats to store annotations as well as ontologies. As suggested in the second requirement the annotation interface is integrated into the user interface. Furthermore features like auto-completion, for instance offered by Kaukolu, support users with the annotation process. As most of the wiki systems are implemented to run on a server as a Web application, collaborative work is possible. Another important aspect is ontology support. For instance IkeWiki has its focus on collaborative engineering of ontologies [15], so it offers functionalities to edit ontologies.

As wiki systems have their own way to store context information, there is no support of heterogeneous document formats, such as word processor or spreadsheet file formats. Some of the wiki systems use embedded annotations, for instance Semantic Media Wiki, thus annotations can easily be maintained while editing the document. The last requirement, offering automation, is part of current research [15].

In contrast to the previous approaches, the following section discusses alternative input devices to capture annotations.

5 Paper based approach

The last approach that is discussed in this paper is a paper-based approach. As the myth of a paperless office still has not become true, paper is still a quite common "input device". Still a lot of people prefer to read longer texts printed on paper. While they are reading, most of the people annotate the text for instance

[9] http://www.jspwiki.org/
[10] http://www.openrdf.org/

by marking the author or writing comments to an important passage of the text, as shown in figure 3. But not all of these annotation are Semantic annotations.

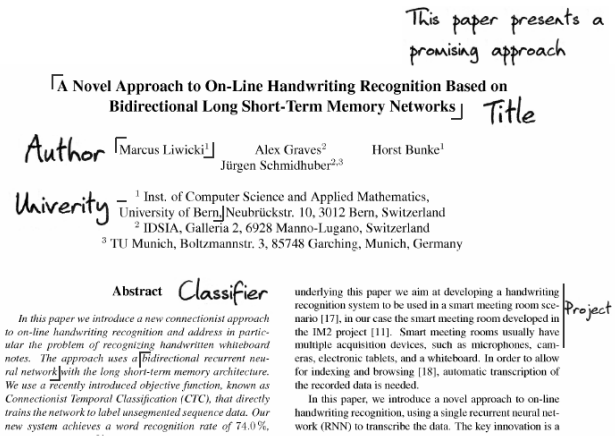

Fig. 3. Annotated document [17].

Nevertheless these annotations are "invisible" for a machine as it is no digital information. So the reader of the document additionally has to transfer his annotations to the digital resource which doubles the effort. As annotating already causes additional work, this might not always be done, so the semantic information is lost. The following section introduces an approach to avoid this extra work of synchronizing the printed with the digital version.

5.1 Semantic E-Ink

Semantic eInk [17] is a project of DFKI Kaiserslautern which has the focus on working with paper or e-paper as an input device for Semantic Annotations. The aim of the system is to act as an interface to the Semantic Desktop[11]. Therefore the handwritten annotations have to be recognized, interpreted and transfered to the Semantic Desktop.

The Anoto Group[12] provides a technology which adds a unique pattern to a printout. By using a digital pen, the position on the paper can be determined and annotations, as shown in figure 3, can be interpreted. Thus the annotations can be added to the digital medium in order to enrich the document with semantic information. Related work is done by ETH Zürich, the PaperProof project[13] also uses the Anoto paper for mapping printed and digital document.

[11] http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/
[12] http://www.anoto.com
[13] http://www.globis.ethz.ch/research/paper/applications/paperproof

Another input device is iLiad[14], a portable e-paper device offered by iRex technologies. As there is a Linux system running on iLiad, the intention of the project is to extend software applications with a semantic component which informs a central service about the user interactions - position of the stylus on the e-paper - with a document, such as a PDF-file. This service identifies annotations again via handwriting recognition and sends the Semantic Annotations to the Semantic Desktop.

So this approach offers an alternative approach for annotation tools mentioned in figure 2.

6 Conclusion

The so called Web 2.0 tagging already is some kind of annotating. As it is a collaborative indexing of resources. Popular indexed objects are blog entries, images or bookmarks. Generally this technique is used in social platforms, like del.icio.us[15], Flickr[16] or Last.fm[17]. But those two terms, tagging and Semantic Annotations, are not equivalent as tagging describes the resource as a whole, for instance the content of a picture and document. Furthermore it is intended to be for people. Whereas semantic annotations is amenable for machine processing as it does not require natural language. Nevertheless the popularity of those platforms indicates that users annotate information and although benefit from this additional work.

The survey of the semantic annotating points out that there are several approaches to annotate documents. Depending on the type of the document the different approaches have their advantages and disadvantages. For instance semantic wikis are quite popular in KM, but still there is a lot of research going on in the area of usability. For example annotation tools likes Annotea are the best approach for annotating the Web pages as there is no write access to the Web servers. Thus each approach has its stakeholders, a combination of all the approaches with a shared storage, to improve collaboration, is essential for KM.

As the process of annotation still is a time consuming process, annotation tools have to focus on simplification of this task. Thus offering semi-automatic or automatic extraction of annotations has to be integrated into annotation systems. Not all of the current prototypes include this feature.

In addition several Semantic Annotations done by users are simply not captured, such as annotations on a printout. The Semantic eInk project considered this problem and is developing a prototype to capture Semantic annotations written on paper. Then these annotations are sent to the NEPOMUK server. As most people prefer to work with printed documents, capturing their interaction with printout is reasonable.

[14] http://www.irextechnologies.com/products/iliad
[15] http://delicious.com/
[16] http://www.flickr.com/
[17] http://www.lastfm.de/

Thus it could be a good approach to investigate alternative approaches to annotate documents. For instance by offering input devices like Anoto paper or monitoring user interaction with the system. Eye-trackers can identify relevant text passages and perform automatic Semantic Annotation extraction. User interaction, such as copying a document to a specific folder on the system, could be used to suggest a Semantic Annotation which links the document with the project. Consequently future work could also focus on approaches to capture user interaction to improve usability. As there is the need to simplify the annotation process without loosing machine-readability.

References

1. Dengel, A.: Knowledge Management (course material) (2008) http://www3.dfki.uni-kl.de/agd/content/e102/e3415/e3527/index_ger.html.
2. Herman, I.: W3C Semantic Web Activity (2001) http://www.w3.org/2001/sw/.
3. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company : How Japanese Companies Create the Dynamics of Innovation. Oxford University Press (May 1995)
4. Guha, R., McCool, R., Miller, E.: Semantic Search (2003)
5. Stojanovic, N., Handschuh, S.: A framework for knowledge management on the semantic web (2002)
6. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Journal of Web Semantics (2005)
7. Brickley, D., Guha, R.: RDF Vocabulary Description Language 1.0: RDF Schema (2004) http://www.w3.org/TR/rdf-schema/.
8. McGuinness, D.L., van Harmelen, F.: OWL Web Ontology Language (2004) http://www.w3.org/TR/owl-features/.
9. Reif, G. In: Semantische Annotation. Springer Berlin Heidelberg (2006) 405–418
10. DeRose, S., Maler, E., Jr., R.D.: XML Pointer Language (XPointer) Version 1.0 (2001) http://www.w3.org/TR/WD-xptr.
11. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web (2004)
12. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: KIM - Semantic Annotation Platform. In: The SemanticWeb - ISWC 2003. Springer-Verlag Berlin Heidelberg (2003) 834–849
13. Oren, E.: Semperwiki: a semantic personal wiki. In: Proc. of 1st Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure, Galway, Ireland. (NOV 2005)
14. Kiesel, M.: Kaukolu: Hub of the Semantic Corporate Intranet. SemWiki2006 (2006)
15. Schaffert, S., Bry, F., Baumeister, J., Kiesel, M.: Semantic Wiki. Informatik Spektrum (2007)
16. NN: Semantic wiki state of the art (2007) http://semanticweb.org/wiki/Semantic_Wiki_State_Of_The_Art.
17. Liwicki, M., Schumacher, K., Dengel, A., Weibel, N., Signer, B., Norrie, M.C.: Pen and paper-based interaction with the Semantic Desktop. In: DAS. (2008)

Relation Types in Medical Ontologies

Daniel Schall

Technische Universität Kaiserslautern
Dept. of Computer Science
67653 Kaiserslautern, GERMANY
d_schall@informatik.uni-kl.de

Betreuer:
Manuel Möller
Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI GmbH)

Zusammenfassung Obwohl Ontologien seit mehr als zehn Jahren erfolgreich verwendet werden, existiert häufig keine mathematisch eindeutige Definition der verwendeten Relationen. Darüber hinaus werden die verwendeten Relationen teils mehrdeutig und inkonsistent in den verschiedenen Ontologien eingesetzt. Das Ziel dieser Arbeit ist es, formale Grundlagen von Relationen zu erläutern und anhand dieser Grundlagen die Probleme und Inkonsistenzen zwischen einigen wichtigen Ontologien der Bioinformatik vor dem Hintergrund des domäneübergreifenden Reasoning aufzuzeigen.

1 Einleitung

„Our vision is that all biomedical knowledge and data are disseminated on the Internet using principled ontologies, such that they are semantically interoperable and useful for improving biomedical science and clinical care.“[1]

Die zunehmende Menge an Fakten, die in einer Wissensdomäne anfällt, macht eine Strukturierung notwendig, da einerseits die Komplexität der Informationen für den Menschen nicht mehr zu erfassen ist, andererseits eine Formalisierung des Wissens eine automatische Verarbeitung ermöglichen würde.

Insbesondere die fortschreitende Spezialisierung innerhalb einer Domäne erschwert das Finden einer gemeinsamen Konzeptualisierung. Ein einheitliches Vokabular und gemeinsam verwendete Definitionen würden jedoch zu Übersichtlichkeit und Verständlichkeit beitragen. Auch das automatische Verarbeiten der Wissensbestände in einem Computersystem würde enorm von einheitlichen und formalen Regelungen profitieren, wie oben zitiert. Daher rührt der Bedarf an Ontologien, die genau diese formale Spezifikation vornehmen.

Während in der Biologie die Taxonomien bisher zur Klassifikation von Konzepten ausreichend waren, zeigen sich in der Bioinformatik deren beschränkte

[1] http://bioontology.org/

Möglichkeiten. Da in Taxonomien alle Konzepte nur über *eine* Verbindung (=Relation) miteinander verknüpft werden, nämlich die Beziehung *ist_ein* (*is_a*), ist es nicht möglich, ausgefeiltere Beziehungen zu modellieren. Eine Klassenhierarchie der Lebewesen wie in der Biologie, die eine rein hierarchische Einordnung von Konzepte in einem Baum bieten, reicht nicht aus, um Fakten wie räumliche Beziehungen oder zeitliche Evolution zu erfassen.

Die erweiterten Möglichkeiten eines Thesaurus, auch *Synomyme* und *verwandte Konzepte* darstellen zu können, ist zwar eine Erweiterung der Taxonomien, allerdings immer noch zu beschränkt in seinem Funktionsumfang, als dass er für die umfassende Darstellung des bioinformatischen Wissens genügen könnte. Zeitliche und räumliche Zusammenhänge können hier bestenfalls rudimentär und über Hilfskonstrukte modelliert werden. Ein solcher Theasurus wurde mit dem *Unified Medical Language System (UMLS)*[2] [1] geschaffen, das mehrere biomedizinische Vokabulare zur Verfügung stellt. UMLS ist zwar kein reiner Thesaurus, da zusätzliche Beziehungen eingearbeitet sind. Da sich der Großteil aber auf hierarchische Relationen wie *is_a* beschränkt, überwiegt der Thesaurus-Charakter.

„An ontology is a specification of a conceptualization."[2] Der Bedarf an einer *formal spezifizierten Begriffsbildung*, in der *Konzepte* und *Beziehungen* zwischen Konzepten definiert werden können, hat zum Einsatz von *Ontologien* geführt, die genau diese Möglichkeiten bieten. Durch diese wird es möglich, ein Wissensgebiet wie etwa die Biologie zu erfassen. Einerseits können die verwendeten Konzepte in einem Vokabular abgebildet werden, zum anderen können beliebige, aber definierte, Relationen zwischen den Konzepten erfasst werden. Dadurch erst wird *automatisiertes Ableiten neuen Wissens* von bereits bestehenden Fakten ermöglicht. Dieser Prozess des Folgerns aus implizitem Wissen und das Umwandeln dessen zu explizitem Wissen wird als *Reasoning* bezeichnet.

Ontologien in der Bioinformatik. Der biomedizinische Sektor hat sich in der Vergangenheit verstärkt mit der Organisation und der Erschließung von Ontologien als Werkzeug zum Wissensmanagement beschäftigt und spezielle Ontologien für viele Fachgebiete geschaffen. Es existieren beispielsweise Ontologien für die menschliche Anatomie wie das *Foundation Model of Anatomy Ontology* (FMA)[3] [3], *SNOMED Clinical Terms* (SNOMED CT)[4][4] oder die *GALEN Core Model Schemata for Anatomy* (GALEN) [5] [5], sowie Ontologien, die Zusammenhänge auf molekularer Ebene beschreiben, wie die *Gene Ontology* (GO)[6] [6] [7]. Außerdem existieren mit den *Open Biomedical Ontologies* (OBO)[7] [8] sogenannte *Upper-Ontologies*, die versuchen, fachspezifische Ontologien wie die vorgestellten

[2] http://www.nlm.nih.gov/research/umls/
[3] http://sig.biostr.washington.edu/projects/fm/index.html
[4] http://www.ihtsdo.org/our-standards/snomed-ct/
[5] http://www.opengalen.org/
[6] http://www.geneontology.org/
[7] http://www.obofoundry.org/

miteinander zu vereinen. Jede einzelne dieser Ontologien ist auf ihr Fachgebiet zugeschnitten und ihre Fachtermini sind entsprechend gewählt worden. Ebenso spiegelt sich die Beschränkung auf ein Spezialgebiet in der Wahl der Beziehungen innerhalb der Ontologie wieder. So hat beispielsweise die Relation *part_of* in der FMA eine abgewandelte Bedeutung als in der GO. [8]

Diese unterschiedliche Verwendung der Relationen (von einer unterschiedlichen Definition, sofern vorhanden, abgesehen) ist ein großes Hindernis beim Mapping zwischen unterschiedliche Ontologien. Daher ist es momentan sehr schwierig, eine Top-Level-Ontologie zu definieren, die die vorgestellten Ontologien einschließt und den bestehenden Wissenskatalog aus jeder Domäne direkt erschließen kann. Wünschenswert wäre eine solche Vereinigung allerdings, um eine verbreiterte Basis für ein automatisches Reasoning über mehrere Fachgebiete verteilt erreichen zu können. Beispielsweise könnten durch eine Vereinigung der *FMA* und des *GO Cellular Branch* Zusammenhänge zwischen makro- und mikroskopischer Anatomie erschlossen werden. Da für ein Fachgebiet teils mehrere Ontologien existieren, wie beispielsweise *SNOMED CT* und *GALEN* für den anatomischen Zweig, jede mit ihren Besonderheiten und exklusiven Wissen, würde eine Vereinigung letzlich ein Reasoning über beide Wissensbestände zugleich ermöglichen und dadurch eine breitere Wissensbasis zur Verfügung stellen.[9]

Diese Vereinigung wird allerdings durch Inkompatibilitäten zwischen den verwendeten Relationen erschwert. Bevor einige Probleme aufgezeigt werden, die beim Mapping von Ontologien auftreten, muss eine formale Grundlage geschaffen werden, auf deren Basis die Relationen analysiert und bewertet werden können.

2 Formale Grundlagen

Prinzipiell müssen Relationen, die zwischen Teilnehmern aus der Ontologie bestehen, unterschieden werden von solchen, die Objekte oder Klassen der Ontologie mit Meta-Daten wie Anmerkungen etc. verknüpfen. Die folgenden Klassifikationen beschränken sich, wie diese Arbeit, auf erstgenannte. Letzere, sogenannte *Annotationen*, werden hier nicht betrachtet, da sie von geringer Bedeutung für automatisches Reasoning sind. [9]

Beziehungsebene. Relationen in und zwischen Ontologien bilden eine Beziehung zwischen zwei Teilnehmern (Klassen oder Instanzen) ab. Dadurch ergeben sich drei Möglichkeiten für binäre Relationen:
<**Klasse, Klasse**>**:** Macht eine allgemeingültige Aussage, die für alle Instanzen der jeweiligen Klassen gültig ist. Zum Beispiel erlaubt die Aussage *Hund is_a Hundeartige*, dass alle Instanzen von *Hund* gleichzeitig auch Instanzen von *Hundeartige* sind.
<**Instanz, Klasse**>**:** Diese Aussage wird zwischen einer bestimmten Instanz und einer Klasse definiert. Beispiel: *Lassie* **instance_of** *Hund*.
<**Instanz, Instanz**>**:** Hier wird eine Relation nur für zwei Instanzen definiert, ohne eine allgemeine Gültigkeit zu implizieren. Beispielsweise ist *Lassies Hals-*

band **part_of** *Lassie*. Damit ist nicht gesagt, dass jeder Hund ein Halsband besitzen muss.

Wie zu sehen, soll für Relationen zwischen Klassen eine kursive Schreibweise verwendet werden, für alle anderen Fettdruck. So soll die Schreibweise diesem De-Facto-Standard, an dem sich unter anderem Ceusters et al., Schulz und Hahn, sowie B. Smith orientieren, angepasst und die Unterscheidbarkeit der Beziehungsebene erleichtert werden.[8]

Mathematische Ordnung. Neben der Beziehungsebene sind die formal-logischen Eigenschaften ein wichtiges Merkmal einer Relation. Hier sind die Merkmale *Reflexivität*, *(Anti-)Symmetrie* und *Transitivität* zu unterscheiden, die für jede Relation eindeutig festgelegt werden sollten. So ist eine sichere automatisierte Verwendung möglich, ohne dass es zu Unklarheiten kommen kann.

Reflexivität bezeichnet die Eigenschaft, ein Objekt (oder eine Klasse) x in Relation R zu sich selbst setzen zu können: $x\ R\ x$. Die Beziehung *is_a* ist beispielsweise reflexiv, wie am Beispiel der Klasse Hundebandwurm verdeutlicht werden soll: *Hundebandwurm is_a Hundebandwurm*, da jeder Hundebandwurm offensichtlich ein Hundebandwurm ist. Eine Relation ist dann reflexiv, wenn diese Bedingung für alle Klassen der Ontologie gilt. Gilt diese Eigenschaft der Relation für keine Klasse, dann ist die Relation irreflexiv.

Symmetrie einer Relation ist die Eigenschaft, dass aus der Beziehung die Umkehrung folgt. Beispielsweise sei die Relation *adjacent_to* (*angrenzend an*) symmetrisch, sodass aus *Linker Lungenflügel adjacent_to Rechter Lungenflügel* folgt, dass *Rechter Lungenflügel adjacent_to Linker Lungenflügel*. **Antisymmetrie** bezeichnet entsprechend das Gegenteil, wenn aus *A in_relation_zu B* nicht automatisch folgt dass *B in_relation_zu A*. So folgt aus der Beziehung *Herzkammer part_of Herz* nicht, dass umgekehrt auch gilt: *Herz part_of Herzkammer*.

Transitivität ist die Übertragbarkeit von Relationen. Beispielsweise ist die Relation *part_of* meist transitiv definiert, so dass gilt: Aus *Kralle part_of Pfote* und *Pfote part_of Hund* folgt nach der Transitivität: *Kralle part_of Hund*.

Diese Merkmale sollten für alle verwendeten Relationen eindeutig definiert werden und dann auch nur gemäß ihrer Definition Verwendung finden, um eine formale Grundlagen zu haben, die eine automatische Ableitung von Wissen ermöglicht.

Weltanschauung. Zusätzlich zu den mathematischen Grundlagen ist es für die Definition einer Relation wichtig, welches *Weltbild der Ontologie* zugrunde liegt. Dabei muss beachtet werden, dass mit der Modellierung der Ontologie der Horizont festgelegt wird, den diese zu erfassen vermag.

Ein gerne zitiertes Beispiel ist die Einbeziehung von *zeitlicher Veränderung* in die Ontologie. In den anfänglichen Versionen der OBO wurde die Veränderlichkeit der Konzepte und die Dynamik von Beziehungen zwischen diesen nicht beachtet. Statische Strukturen wie die Anatomie des gesunden, menschlichen Körpers oder die Modellierung von Zellentwicklungslinien konnten ohne

dynamische Beziehungen erfasst werden. Doch konnte beispielsweise die Zellteilung nicht problemlos modelliert werden, da es keine Möglichkeit gab, das Entstehen von zwei Zellen aus einer darzustellen. Ebenso war es nicht möglich, das Konzept *Lungenkrebs* sinnvoll in Relation zu *Lungengewebe* zu setzen. Die Verknüpfung durch *part_of* ist offensichtlich fehlerhaft, da das veränderte Gewebe nur ein pathologischer Teil des gesunden Gewebes ist. Allerdings ist es nicht möglich, die Entwicklung des entarteten Gewebes aus gesundem Gewebe darzustellen, da es keine zutreffenden Relationen gibt, die diesen Sachverhalt ausdrücken können. Wie zu erkennen ist, muss das grundlegende Modell der Realität bedacht gewählt werden, da die Definition der Relationen entsprechend dem Modell eingeschränkt wird. Es bietet sich an, auf das Modell von *Grenon et al.*, vorgestellt in „SNAP and SPAN: Towards Dynamic Spatial Ontology“ [10] zurückzugreifen, welches ein dynamisches Entstehen von Instanzen und definierte Relationen zur Abbildung von zeitlichen Verläufen zulässt. Hier können neben den statischen Konzepten, die *Continuants* genannt werden, auch Prozesse, Ereignisse und Veränderungen (*Occurents* genannt), dargestellt werden. *SNAP/SPAN* bietet einen Katalog an definierten Relationen, die in den meisten Domänen ohne Veränderung eingesetzt werden können.[11]

3 Fehler in Relationen

Ausgehend von den vorangehend angeführten Merkmalen, finden sich in ontologischen Relationen zahlreiche Fehler. Eine Relation wird nur selten im Stadium der manuellen Eingabe von „Basiswissen“ als fehlerhaft erkannt, da in diesem Schritt die Definition der Relation vermeintlich klar gefasst und für den Menschen unzweifelhaft ist. Beim automatischen Reasoning über den Wissensbestand treten die Fehler dann in Form von falschen Folgerungen zu Tage.[12] Unzulänglichkeiten in den Relationen können auch bei der Evolution der Ontologie auftreten, etwa wenn sich herausstellt, dass eine Relation für die soeben neu erschlossenen Domänen ungeeignet ist und eine feinere Unterscheidung des Beziehungstyps hätte erfolgen müssen. Der klassische Ansatz, um Fehler aufzuspüren ist allerdings die systematisch Fehlersuche, manuell oder automatisch durchgeführt. Beispielsweise haben *Ceusters et. al.* in „Mistakes in Medical Ontologies: Where Do They Come From and How Can They Be Detected?“ für die Ontologie SNOMED-CT eine automatisierte Analyse mit den Tools LinKFactory®[8] und LinKBase®[9] durchgeführt. Hierbei wurden drei unterschiedliche Algorithmen von LinKFactory® verwendet, um den Wissensbestand zu analysieren. Zum einen wurde ein *lexikalischer Vergleich* durchgeführt und es wurden redundante Konzepte und Relationen ausfindig gemacht. Außerdem wurden *synonyme Terme* gesucht und deren Relationen verglichen. So konnten fehlende Relationen, die nur bei Synonymen vorhanden waren, entdeckt werden. Ein weiterer Algorithmus hat den Datenbestand nach *unsinnigen Unter- und Oberklassen* durchsucht und Konzepte, die nur ein Unterkonzept subsummieren, zur Überarbeitung markiert.

[8] http://www.landcglobal.com/pages/linkfactory.php
[9] http://www.landcglobal.com/pages/linkbase.php

Die aus dieser automatisierten Analyse gewonnenen Daten wurden anschließend manuell überprüft und entsprechend korrigiert.

Im Folgenden sind einige der häufigsten Fehler erläutert, die bei dieser und weiterer Analysen gefunden wurden.

3.1 Intra-ontologische Fehler.

Zunächst soll auf Fehler in Relationen innerhalb einer einzigen Ontologie hingewiesen werden, ohne dabei auf die Implikationen für domänenübergreifendes Reasoning einzugehen. Denn selbst innerhalb einer Ontologie führt eine falsche Relation oftmals zu vielen daraus abgeleiteten Fehlern. Beispielsweise kann aus einer einzigen, falschen Relation, wie etwa *Zeh part_of Hand* eine große Menge an falschen Schlussfolgerungen getroffen werden. So kann aus diesem falschen Beispiel weiter abgeleitet werden, dass *Zeh part_of* Arm oder unter Zuhilfenahme des Fakts *Eingewachsener Zehnagel located_at Zeh*, dass diese Erkrankung plötzlich zu einer Erkrankung der oberen Extremitäten wird. Die prominentesten Fehler, die bei der Analyse von biomedizinischen Ontologien gefunden wurden, sind im folgenden mit Beispielen und Entstehungsgeschichte aufgeführt.

Menschliche Fehler. Da jede Ontologie zu Anfang eine Wissensbasis benötigt, die manuell im System hinterlegt werden muss, können in diesem Schritt bereits fehlerhafte Beziehungen erzeugt werden. In SNOMED-CT wurde beispielsweise die Relation *is_a* fälschlicherweise zwischen einer Therapie und der dazugehörigen Krankheiten definiert. So wurde daraus gefolgert, dass die Erkrankung eine Oberklasse der zugehörigen Therapie ist.

Auch andere Relationen wurden nicht bestimmungsgemäß verwendet. Durch automatische Ableitung entstanden so eine Vielzahl weiterer fehlerhafter Beziehungen. Diese menschlichen Fehler folgen teils aus Unachtsamkeit, teils aber auch aus fehlendem Verständnis für die verwendeten Konzepte und Relationen.

Eine weitere Unterteilung ist möglich, um den menschlichen Fehler näher zu bestimmen:

Fehlerhafte Negation Konzepte in SNOMED-CT umfassen auch Krankheiten. Viele dieser Krankheiten wurden in Subklassen sehr genau unterschieden und ihre symptomatische Lokation wurde in das Konzept aufgenommen. Dadurch entstanden lange Ausdrücke, wie etwa *Morbus Dupuytren der Handfläche, Knötchen ohne Kontraktur.*

In SNOMED-CT wurde dieses Konzept von *Kontraktur des Handflächenmuskels* subsummiert, obwohl im Text offensichtlich *fehlende Kontraktur* erwähnt wird. Diese Beziehung ist wohl durch Unachtsamkeit entstanden und führt im Verlauf der automatisierten Verarbeitung zu weiteren Fehlern. Da hier ein Konzept von seiner Negation subsummiert wird, können sehr seltsame Folgerungen entstehen, die sich zwar logisch widersprechen, vom automatischen Reasoning allerdings nicht erkannt wurden.

Ungeeignete Unterscheidung zwischen „partiell“ und „komplett“ *Ceusters et. al.* haben viele dieser Fehler dieser Art in SNOMED-CT gefunden.

Oftmals wurde eine Krankheit, die nur partielle Beteiligung eines Organs beschreibt, von der Krankheit, die die totale Beteiligung des Organs voraussetzt, subsummiert. Dadurch wird automatisiertes Reasoning erschwert, da die implizite Eigenschaft „betrifft das ganze Organ“ in der Subklasse plötzlich nicht mehr gültig ist. Beispielsweise wurde folgende Beziehung hergestellt: *Erkrankung der Nebenniere part_of Erkrankung der Niere*. Durch diese zunächst logisch erscheinende Verknüpfung würde jedoch fehlerhaft abgeleitet werden, dass eine Entzündung der Nebenniere automatisch eine Entzündung der gesamten Niere bedeutet.

Technologiebedingte Fehler. Nicht nur menschliche Falschangaben führen zu Fehlern, auch aus richtigen Basisdaten können durch *fehlerhafte Ableitungen* ungültige Ergebnisse entstehen. Da das anatomischen Fachvokabular Homonyme aufweist, konnte durch lexikalisches Matching der englischen Termini folgende fehlerhafte Ableitung entstehen: *structure of labial vein (Lippenvene) is_a vulval vein (Vene der Vulva)* und gleichzeitig *structure of labial vein is_a structure of vein of head (Vene des Kopfes)*. Dieser Fehler rührt offensichtlich daher, dass das Konzept *labia* sowohl die *Schamlippen*, als auch die *Lippen des Mundes* bezeichnet und in der Konzeptualisierung nicht unterschieden wurde. Beim automatischen Ableiten ist dann die fehlerhafte Folgerung entstanden. Eine Möglichkeit der Vermeidung wäre die explizite sprachliche Unterscheidung der Konzepte gewesen, oder die Einbeziehung des Kontext beim Reasoning. So hätte erkannt werden können, was mit der Bezeichnung *labial vein* gemeint ist. Ein Einbeziehen von Seriennummern für die Konzepte hätte die Unterscheidbarkeit auf maschineller Ebene ermöglicht, allerdings ist der Mensch als Fehlerquelle auch dann nicht auszuschließen.

Bedeutungsverschiebung. Weitere Fehler können bei Änderungen an den grundlegenden Konzepten einer Ontologie entstehen. So können vorhandene Annahmen mit neu hinzukommenden Ideen in Konflikt geraten, ohne dass dies unmittelbar auffallen würde. In SNOMED-CT wurde beispielsweise die Idee der SEP-Tripel aus der Ontolgie SNOMED-RT übernommen. SEP steht für Structure-Entire-Part und schreibt die Unterteilung jedes Organs in drei Konzepten vor: Struktur, Ganzes und Teil. Dadurch wurde die Grundannahme von SNOMED-CT, dass Konzeptklassen wie *Herz* immer für *Herz als ganzes oder einen beliebigen Teil davon* stehen, obsolet und viele Beziehungen wurden durch diese Änderung fehlerhaft. Eine Korrektur der nicht mehr zutreffenden Relationen musste nach Bekanntwerden manuell erfolgen.

Redundante Konzepte. Für ein Konzept aus der Realität sollte nur genau ein Konzept in der Ontologie stehen. Dadurch ist eine Zuordnung und Abbildung von Sachverhalten direkt und zweifelsfrei möglich. Stehen mehrere Konzepte für die gleiche Bedeutung in der Realität, so entsteht Redundanz und es können konfligierende Definition existieren.

Eine Analyse der Ontologie SNOMED-CT hat bei restriktiven Suchparametern bereits 8746 redundante Konzepte ohne Bedeutungsunterschiede gefunden. Viele dieser redundanten Konzepte sind durch die Einverleibung von anderen Ontologien entstanden und existieren parallel zu den bestehenden Klassen. Durch Klassen mit gleicher Bedeutung, aber unterschiedlichen Beziehungen wird das automatisierte Reasoning erschwert,da kein modellierter Zusammenhang zwischen redundanten Klasse existiert.

Fehlende Ausdruckskraft der Ontologie. Auch wenn durch wenige Relationen ein Großteil der realen Zusammenhänge erfasst werden kann, treten Probleme bei wenigen, „exotischen“ Konstellationen auf. Durch die Relation *part_of*, die als totale Subsummierung der einschließenden Objekte definiert ist, können die meisten Sachverhalte skizziert werden, wie etwa *Gehirn part_of Zentralnervensystem* oder *Seite part_of Buch*. Betrachtet man allerdings den *Nervus Tibialis* (Schienbeinnerv), der sowohl Teil des Oberschenkels als auch Teil des Unterschenkels ist, so ist die Relation *part_of*, die totale Subsummierung impliziert, hier nicht mehr angebracht. Trotz dieses systematischen Fehlers wurde die Relation *part_of* beispielsweise bei SNOMED-CT in Ermangelung einer treffenderen Relation zur Beschreibung verwendet. Ähnliche Beispiele finden sich in der GO wieder. Hier wurden in Ermangelung von Relationen, die räumliche Beziehungen ausdrücken, die bestehenden Relationen is_a und part_of zweckentfremdet. Die beispielsweise modellierter Beziehung *Extrazelluläre Region is_a Zelluläre Komponente* ist aber falsch, da die *Extrazelluläre Region*, die die Zelle umgibt, nicht eindeutig genug bestimmbar ist. Die Frage, wo genau die extrazelluläre Region endet, ist nicht beantwortbar. Außerdem fehlt oft das Verständnis für zeitliche Veränderung beim Entwurf der Ontologie und die Ausdruckskraft der Ontolgie wird eingeschränkt. Beim Modellieren von Relationen kann die temporale Komponente dann ebenfalls nicht einbezogen werden. Das Fehlen von Relationen, um den zeitlichen Verlauf auszudrücken, kann auch durch die *zweckentfremdete Verwendung* der bestehenden Relationen nicht mehr kompensiert werden. Es entstehen beim Versuch lediglich falsche Fakten und durch automatisiertes Reasoning wird die Situation dann weiter verschlimmert.

Mehrdeutigkeiten. Viele Relationen werden verwendet, ohne dass eine genaue Definition über die Bedeutung der Relation getroffen wurde. So kann die Relation *part_of* in der GO auf drei verschiedene Arten interpretiert werden. Zum einen kann A *part_of* B eine *wechselseitige Existenzabhänigkeit* bedeuten, also dass jede Instanz von A immer Teil einer anderen Instanz B ist und dass jede Instanz von B eine Instanz von A als Teil enthält (etwa: *Herzbeutel part_of Herz*). Die gleiche Relation wurde auch verwendet, um eine *einseitige Teil-Ganzes-Beziehung* auszudrücken, also dass jede Instanz von A Teil einer Instanz B sein muss (aber nicht, dass jedes B zwingend ein A enthalten muss). Wie etwa im Beispiel *Zellkern part_of Zelle* zu sehen, da es auch Zellen ohne Zellkern gibt. Als dritte Interpretationsmöglichkeit kann *part_of* in der GO als *optionale Teil-Ganzes-Beziehung* bedeuten, dass eine Instanz von A Teil einer

Instanz von B sein kann, allerdings nicht zwingend sein muss. Beispiel: *Arterie part_of Arm* drückt aus, dass Arterien Teil des Arms sind, sagt aber nicht, dass Arterien exklusiv Teil des Arms sind, da diese auch in anderen Körperteilen vorkommen können.

Ununterscheidbarkeit. In OBOs Cell Ontology existieren die Relationen *derives_from* und *develops_from*, die beide für unterschiedliche Beziehungen eingesetzt wurden, allerdings ohne genaue Abgrenzung, inwieweit Unterschieden zwischen den Relationen bestehen. Offenbar sind beide Relationen Synonym zueinander, ohne dass diese Tatsache dokumentiert wurde. Der Fehler wurde mittlerweile behoben, doch die Gefahr, Relationen zu schaffen, die aufgrund ihrer Ähnlichkeit später nicht mehr unterscheidbar sind, bleibt bestehen. Da für automatisiertes Reasoning nicht ersichtlich ist, dass diese Beziehungen identisch verwendet werden, können sie nicht als synonym erkannt werden und schränken so den automatischen Prozess ein. Dadurch werden viel Ableitungen nicht so getroffen, wie erwartet.

3.2 Inter-ontologische Fehler

Die vorgestellten Fehler, deren Auswirkungen bisher nur innerhalb einer Ontologie betrachtet wurden, können beim Reasoning über mehrere Domänen ebenso auftreten und zu fehlerhaften Ableitungen führen. Zusätzlich können beim Zusammenschluss von Ontologien neue Konflikte entstehen, die im folgenden skizziert werden sollen. Ohne eine vorab festgelegte Definition der verwendeten Relationen und Konzepte ist es schwer, die Auswirkungen eines Zusammenschlusses auszumachen. Die hier vorgestellten Probleme wurden daher meist erst beim Reasoning über den gemeinsame Wissensbestand ausgemacht.

Unterschiedliche Definitionen. Unter der Annahme, dass die Relationen in den zusammenzuführenden Ontologien eindeutig definiert sind, bereiten gleichnamige Relationen mit unterschiedlicher Definition beim Zusammenschluss von Ontologien zwar Probleme, allerdings kann dieser Konflikt, durch das Vorliegen einer Definition, beim Zusammenschluss leicht erkannt und beachtet werden. Die Relation *part_of* etwa, die in der GO eine *lose Verbindung zwischen Teil und Ganzem* beschreibt (siehe 3.1), in der FMA hingegen eine *strikte Existenzabhängigkeit* zwischen Teil und Ganzem vorsieht, konnte vorab erkannt werden. Die Probleme, die sich aus einer unterschiedlichen Definition ergeben, sind, wie dieser beispielhafte Konflikt, recht einfach zu beheben.[13] [14]

Informelle Verwendung von Relationen. Die informelle Verwendung von Relationen bereitet nicht nur Probleme innerhalb einer Ontologie, beim Zusammenschluss von Ontologien entstehen durch vage definierte Relationen große Probleme. Da die Definition fehlt, kann nicht erkannt werden, inwieweit sich Relationen unterscheiden oder ähneln und welche Implikationen dies für das

Reasoning nach sich zieht. Die einfache Annahme, zwei Relationen aus unterschiedlichen Ontologien sind trotz ihrer ungenauen Definition gleich, kann im weiteren Reasoning allerdings große Probleme bereiten. Exemplarisch sei hier auf die vorgestellte Relation *part_of* verwiesen, die oftmals ohne feste Definition verwendet wird und daher beim Zusammenschluss von Ontologien zu Problemen führen kann.

Konzeptionelle Konflikte. Ontologien aus verschiedenen Spezialgebieten besitzen, aufgrund der unterschiedlichen Anforderungen in den Diszplinen, oftmals andere Konzepte und Relationen. Während die GO beispielsweise zelluläre Vorgänge erfasst und daher Wert auf detailreiche Beziehungen zwischen Molekülen und Kausalketten legt, stellt die FMA die makroskopische Anatomie des Menschen dar und definiert daher Relationen und Konzepte anders. In der GO wurde beim Entwurf die zeitliche Komponente beachtet und es können *Entwicklungsprozesse* modelliert werden. Außerdem wird nicht nur der gesunde Zustand beachtet, es können auch *pathologische Merkmale* in der Ontologie dargestellt werden. Diese beiden konzeptionellen Überlegungen wurden in der FMA nicht getroffen und daher gab es in den ersten Versionen der FMA keine Möglichkeit, zeitliche Verläufe oder pathologische Merkmale abzubilden. Durch unterschiedliche Konzepte beim Entwurf der Ontologien, entstehen beim späteren Zusammenlegen also Probleme aufgrund der *unterschiedlichen Ausdruckskraft*. Dies muss beim automatischen Reasoning ebenfalls beachtet werden und es muss unterschieden werden, welches Modell der Realität den Ontologien zugrundegelegt wurde. Eventuell muss eine umfassende Abbildung zwischen den Ontologien ausgearbeitet werden, um die Konzepte miteinander verträglich zu machen.

4 Fazit und Ausblick

Es wurden nun einige der historischen und aktuellen Probleme beim Umgang mit Relationen aus verschiedenen Ontologien dargestellt und die Auswirkungen dieser Fehler wurden beschrieben. Wie zu sehen, ist es bereits beim Entwurf von großer Bedeutung, ein *umfassendes Bild der zu modellierenden Umwelt* zu haben und darauf aufbauend *widerspruchsfreie und eindeutig definierte Relationen* zu entwerfen. Durch diesen Weitblick können Konflikte und fehlerhafte Verwendung innerhalb einer Ontologie vermieden werden. Auch wird durch eine umfassende Definition vermieden, dass die bestehenden Relationen für neue Ideen zweckentfremdet werden.

Darüber hinaus erlaubt eine möglichst *generische Gestaltung der Relationen*, diese in einem breiteren Umfeld einzusetzen und kann den Zusammenschluss mit anderen Domänen erleichtern. Allerdings ist diese Forderung nach generischen Konzepten aufgrund der zunehmenden Spezialisierung der Fachgebiete und der damit einhergehenden Spezialisierung der Ontologien immer schwerer einzuhalten: Spezielle Lösungen erfordern spezielle Ontologien.

Allerdings muss keine komplette Ontologie mit all ihren Relationen für jedes Fachgebiet neu erdacht werden. Der Aufwand bei diesem eher bottom-up-

orientierten Ansatz kann durch das Zurückgreifen auf bestehende Best Practices deutlich verringert werden und die Erfahrung aus dem biomedizinischen Sektor kann beim Entwurf von neuen Ontologien von Nutzen sein. Die biologischen Ontologien befassen sich zumeist mit ähnlichen Problemen, wie etwa die Frage nach korrekter, mereotopologische Darstellung (Beziehungen zwischen Teil, Ganzem und Rändern) oder der Abbildung von pathologischen Merkmalen. Daher können große Teile – konzeptionell wie inhaltlich – wiederverwendet werden und müssen nicht für jede Ontologie neu definiert werden.

Da es in der Bioinformatik mit den „Open Biomedical Ontologies“ ein Konsortium gibt, das sich die Standardisierung und Interoperabilität zwischen den unterschiedlichen Ontologien zum Ziel gesetzt hat, existiert ein ausgearbeites Repertoire an Konzepten und Relationen, auf die beim Entwurf von neuen Ontologien zurückgegriffen werden kann. Die OBO katalogisieren darüber hinaus die bestehenden, bewährten Ontologien und versucht, eine *Upper-Ontology* zu entwerfen, die alle anderen subsummiert.

Ein Blick in den Katalog der OBO kann helfen, doppelte Arbeit zu vermeiden und ermöglicht es, im Vorfeld nach Ontologien suchen, die potentiell zu einem Zusammenschluss mit der neuen Ontologie geeignet sind. Dadurch können die bestehenden Konzepte beim Entwurf der neuen Ontologie beachtet werden und die Kompatibilität kann sichergestellt werden. Diese Vorüberlegungen können zum Erfolg des neuen Entwurfs beitragen und im späteren Verlauf viel Anpassungsarbeit einsparen.

Zu sehen ist, dass die Beachtung der OBO als Referenz beim Entwurf von neuen Ontologien hilfreich sein kann, domänenspezifische Aufgabenstellungen darüber hinaus allerdings beachtet werden müssen. [8]

Fehleranalysen und regelmäßige Reviews sind für den Erfolg einer Ontologie trotz gewissenhaftem Entwurf unerlässlich. Insbesondere um die Kompatibilität mit anderen Ontologien sicherzustellen und den Wissensbestand über die Ontologie hinaus zur Verfügung zu stellen, ist es wichtig, sich über Fehler und Unzulänglichkeiten des eigenen Entwurfs im Klaren zu sein. Da bestehende Fehler durch automatisiertes Reasoning weiter abgeleitet werden und zu mehr fehlerhaftem „Wissen“ führen, ist eine frühe Erkennung unerlässlich.

Literatur

1. United States National Library of Medicine, National Institute of Health: About the UMLS Resources. http://www.nlm.nih.gov/research/umls/about_umls.html.
2. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5**(2) (1993) 199–220
3. Rosse, C., Mejino, J.L.V.: The Foundational Model of Anatomy Ontology. Anatomy Ontologies for Bioinformatics: Principles and Practice (2007) 59–117
4. College of American Pathologists: SNOMED Clinical Terms User Guide. http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/Technical_Docs/snomed_ct_user_guide.pdf.
5. Rector, A., Gangemi, A., Galeazzi, E., Glowinski, A., Rossi-Mori, A.: The Galen Core Model Schemata for Anatomy: Towards a Re-usable Application-Independent Model of Medical Concepts. In: Proceedings of Medical Informatics Europe, MIE 94, Lisabon. (1994) 186–189
6. Smith, B.: The Logic of Biological Classification and the Foundations of Biomedical Ontology. Dag Westerståhl (ed.), Invited Papers from the 10th International Conference in Logic Methodology and Philosophy of Science (2003)
7. Gene Ontology Consortium: An Introduction to the Gene Ontology. http://www.geneontology.org/GO.doc.shtml.
8. Smith, B., Ceusters, W., Klagges, B., Kohler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A., Rosse, C.: Relations in Biomedical Ontologies. Genome Biology **6**(5) (2005) R46
9. Bittner, T., Donnelly, M., Smith, B.: Individuals, Universals, Collections: On the Foundational Relations of Ontology
10. Grenon, P., Smith, B.: SNAP and SPAN, Towards Dynamic Spatial Ontology. Spatial Cognition and Computation
11. Sider, T.: Four-Dimensionalism: an Ontology of Persistence and Time. Oxford University Press (2001)
12. Ceusters, W., Smith, B., Kumar, A., Dhaen, C.: Mistakes in Medical Ontologies: Where Do They Come From and How Can They Be Detected? Stud Health Technol Inform. (2004)
13. Schulz, S., Hahn, U.: Parthood as Spatial Inclusion - Evidence From Biomedical Conceptualizations. In: Principles of Knowledge Representation and Reasoning. Proceedings of the 9th international conference (2004) 55–63
14. Schulz, S., Marko, K., Hahn, U.: Spatial location and its relevance for terminological inferences in bio-ontologies. BMC Bioinformatics **8**(1) (2007) 134

Recommender Systems for Web 2.0 Resource Sharing Platforms

Alexander Grothkast

Technische Universität Kaiserslautern, 67653 Kaiserslautern,
a_grothk@informatik.uni-kl.de

Zusammenfassung Diese Arbeit stellt die Zielsetzung und Klassifikation von Empfehlungssystemem vor. Exemplarisch wird für jede Klasse ein Algorithmus diskutiert und mögliche Probleme erläutert. In Web 2.0 Plattformen, bei denen mehrere Nutzer kollaborativ Inhalte einstellen und mit Tags versehen können, bestehen verschiedene Einsatzmöglichkeiten für Empfehlungssysteme. Die Möglichkeit vorhandene Tags als zusätzliches Hintergrundwissen für Empfehlungen zu nutzen, wurde bislang jedoch nur unzureichend untersucht. Ein generischer Ansatz aus dem aktuellen Jahr erscheint jedoch vielversprechend und wird näher vorgestellt sowie bewertet.

1 Einleitung

In den letzten Jahren hat das Web 2.0 eine immer größere Bedeutung erlangt. Bekannt sind die entsprechenden Plattformen bei denen mehrere Benutzer Inhalte einstellen und darauf zugreifen; so wird ein Austausch von Ressourcen unter den Benutzer möglich. Ein zentrales Merkmal ist die Möglichkeit diese Inhalte zu taggen, also mit frei wählbaren Schlagworten zu versehen. Diese Tags sollen zur besseren Auffindbarkeit von Ressourcen beitragen.

Eine bekannte Web 2.0 Plattform zum Austausch von Bildern ist Flickr[1]. Nach eigenen Angaben wurde bereits am 13. November 2007 das zweimilliardste Bild auf Flickr bereitgestellt [1]. Es stellt sich also immer mehr die Frage, wie Nutzer für sie interessante Inhalte finden können. Welche Möglichkeiten gibt es neben der aktiven Suche nach bestimmten Tags?

Dazu können Empfehlungssysteme dienen. Deren Aufgabe ist gerade die Empfehlung von interessanten Inhalten für Nutzer. Solche Systeme wurden bereits viele Jahre untersucht. Die vorliegende Arbeit stellt daher zunächst Empfehlungssysteme, deren Aufgaben und Klassifizierung vor. In einem zweiten Teil wird untersucht, wie Empfehlungssysteme unter Verwendung von benutzergenerierten Metadaten Inhalte innerhalb von Web 2.0 Plattformen empfehlen können.

2 Empfehlungssysteme

Einen hervorragenden Überblick über Empfehlungssysteme geben Adomavicius und Tuzhilin in [2]. Dort wird ein Überblick über den Stand der Forschung

[1] http://www.flickr.com

im Bereich der Empfehlungssysteme gegeben und die im folgenden Abschnitt wiedergegebene Klassifikation diskutiert.

Einen ähnlichen Überblick geben Höhfeld und Kwiatkowski in [3], bleiben damit jedoch deutlich hinter [2] zurück. So fehlen insbesondere weitergehende Verweise auf anwendbare Methoden für einzelne Ansätze.

Frühe Arbeiten im Themenbereich Empfehlungssysteme gehen auf die 1970er Jahre zurück. In den Kognitionswissenschaften veröffentlichte Rich 1979 eine Arbeit [4]. Dort schlägt sie vor, Nutzer aufgrund unsicheren Wissens durch sogenannte Stereotypen zu modellieren.

In ihrer Einleitung gibt sie eine schöne Motivation: Ein Bibliothekar empfielt Benutzern (den Besuchern der Bibliothek) bestimmte Ressourcen (Bücher) aufgrund der Einordnung des Benutzers in Stereotype: Handelt es sich um einen potentiellen Touristen, ein neugieriges Kind, einen Studenten oder gar eine Person, welche die chinesische Sprache beherrscht? Aufgrund dieser Einordnung können Ressourcen empfohlen werden, die ähnlichen Benutzern gut gefallen haben. Einen solchen Ansatz verfolgen kollaborative Empfehlungssysteme.

Andere Arbeiten, etwa von Salton aus 1989 [5], gehen auf das Information Retrieval zurück. Es werden Ressourcen gesucht, die ähnlich zu bereits bekannten und von bestimmten Benutzern gut bewerteten Ressourcen sind. Diesen Ansatz könnte man mit obigem Beispiel verdeutlichen, wenn man annimmt, dass der Bibliotheksbesucher bereits ein Buch über China gelesen hat. Wenn ihm dieses Buch gefallen hat, kann ein Bibliothekar ähnliche Bücher empfehlen. Diese Sichtweise bildet die Grundlage der inhaltsbasierten Systeme.

Diese Beispiele führen zur folgenden Terminologie für Empfehlungssysteme, die ebenfalls aus dem Überblick von Admonavicius und Tuzhilin [2] entnommen ist:

Empfehlungssystem. *Mathematisch werden einzelnen Personen aus einer Menge an Nutzern C Objekte aus einer Ressourcenmenge S empfohlen. Eine Bewertungsfunktion $u : C \times S \to \mathbb{R}$ bildet jede Nutzer-/Ressourcenkombination auf eine geordnete Menge, wie beispielsweise die reelen Zahlen, ab. Mit diesem Maß wird versucht, die Nützlichkeit einer Ressource für einen bestimmten Benutzer abzubilden. Aufgabe eines Empfehlungssystems ist die Lösung des Optimierungsproblems*

$$s'_c = \max_{s \in S} u(c, s) \tag{1}$$

für einen gegebenen Nutzer $c \in C$.

	Heuristisch	Modellbasiert
Inhaltsbasiert	Term-Frequency/Inverse-Document-Frequency	Bayes-Klassifikator
Kollaborativ	Nearest-Neighbor	Bayes-Klassifikator

Tabelle 1. Eine tabellarische Klassifikation von Empfehlungssystemen angelehnt an die Darstellung in [2]. Es sind, stellvertretend für jeweils eine ganze Anzahl von Algorithmen, die in dieser Arbeit vorgestellten Methoden dargestellt.

Typischerweise liegt die Bewertungsfunktion u jedoch nicht vollständig vor, da Nutzer nur Ressourcen bewerten können, die sie bereits kennen. Empfohlen werden sollen aber bisher dem Nutzer nicht bekannte Objekte. Bevor Gleichung (1) gelöst werden kann, muss ein Empfehlungssystem daher zunächst die fehlenden Funktionswerte für u abschätzen. Dies führt nach [2,3] zu einer Unterscheidung der beiden oben schon genannten Ansätze. Diese Unterscheidung ist in Tabelle 1 dargestellt. Ferner kann noch zwischen heuristischen und modellbasierten Ansätzen unterschieden werden. Erstere nutzen alle vorliegenden Informationen, um aufgrund von Heuristiken direkt Empfehlungen abzugeben. Letztere nutzen die Informationen, um ein Modell zu lernen. Mit Hilfe dieses Modells können dann Empfehlungen gegeben werden.

2.1 Inhaltsbasierte Systeme

Inhaltsbasierte Empfehlungssysteme approximieren eine unbekannte Bewertung $u(c, s)$ durch eine oder mehrere Bewertungen $u(c, s_i)$. Dabei werden möglichst ähnliche Ressourcen $s_i \approx s$ zugrunde gelegt. Dazu müssen die relevanten Inhalte aller Ressourcen auf jeweils ein Inhaltsprofil abgebildet werden. Diese Profile können dann über ein Ähnlichkeitsmaß verglichen werden. Es müssen also ähnliche Ressourcen gefunden werden.

Im *Information Retrieval* werden relevante Dokumente als Ergebnisse zu Suchanfragen gesucht. Es lassen sich Methoden aus dem Information Retrieval adaptieren, um in Empfehlungssystemen die Ähnlichkeit von Ressourcen bewerten zu können.

Term Frequency/Inverse Document Frequency. Eine bekannte Methode zur Gewichtung von Keywords des Information Retrieval ist *Term Frequency/Inverse Document Frequency (TF/IDF)* [5]. In Textdokumenten können solche Keywords – im Gegensatz zu anderen Medien – in der Regel sehr einfach extrahiert werden. Daher kann TF/IDF bei Textdokumenten gut zum Aufbau eines inhaltsbasierten Empfehlungssystems verwendet werden.

Mit der Term Frequency $TF_{i,j}$ wird ein relatives Maß angegeben, wie häufig ein Keyword k_i in einem Dokument r_j vorkommt. Dazu wird die absolute Häufigkeit $f_{i,j}$ gezählt und durch die Anzahl des am häufigsten vorkommenden Keywords geteilt:

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \tag{2}$$

Ein Keyword ist jedoch nur dann geeignet ein Dokument zu beschreiben, wenn es in relativ wenigen Dokumenten überhaupt vorkommt. Dazu wird für ein Keyword die Inverse Document Frequency IDF_i definiert. Adomavicius und Tuzhilin geben in [2]

$$IDF_i = \log \frac{|S|}{n_i} \tag{3}$$

als eine üblicherweise verwendete Definition an. n_i gibt dabei an, in wie vielen Dokumenten das Keyword k_i auftaucht. Als Maß ergibt sich nun die mit der IDF_i gewichte $TF_{i,j}$

$$w_{i,j} = TF_{i,j} \times IDF_i \ . \tag{4}$$

Der Inhalt jeder Ressource wird also durch einen Vektor $\boldsymbol{w_s} = (w_{1,s}, \ldots, w_{k,s})$ beschrieben. Mithilfe eines beliebigen Algorithmus – etwa Rocchios Algorithmus [6] – zur Bildung eines Mittelwertes[2] werden alle von einem Benutzer c bewerteten Ressourcen s_i und deren Gewichte $\boldsymbol{w_{s_i}}$ zu einem Vektor $\boldsymbol{w_c} = (w_{c,1}, \ldots, w_{c,k})$ zusammengefasst. Dieser Vektor repräsentiert die individuelle Gewichtung der einzelnen Keywords durch den Benutzer c.

Damit kann die Bewertungsfunktion durch ein Kosinus-Ähnlichkeits-Maß

$$u(c,s) = \frac{\boldsymbol{w_c} \cdot \boldsymbol{w_s}}{||\boldsymbol{w_c}||_2 \times ||\boldsymbol{w_s}||_2} \tag{5}$$

ausgedrückt werden [5]. Es handelt sich dabei um das im dem Information Retrieval verwendete *Vektorraum-Modell.*

Maschinelles Lernen. Neben heuristischen Verfahren wie TF/IDF können auch *modellbasierte Algorithmen*, etwa aus dem Bereich des *maschinellen Lernens* angewendet werden. Pazzani und Billsus vergleichen in [6] Bayes-Klassifikatoren mit anderen Lernverfahren und Rocchios Algorithmus basierend auf TF/IDF-Gewichten.

Ein *Bayes-Klassifikator* klassifiziert eine Ressource aufgrund der Wahrscheinlichkeit

$$P(C_i | k_{1,j} \& \ldots \& k_{n,j}) \tag{6}$$

in verschiedene Klassen C_i. Bei $k_{1,j}, \ldots, k_{n,j}$ handelt es sich um die in einem Dokument d_j vorkommenden Keywords. In einer naiven Implementation eines solchen *probabilistischen Modells* nimmt man nun statistische Unabhängigkeit der Keywords an. Unter dieser Voraussetzung verhält sich

$$P(C_i) \prod_k P(k_{k,j} | C_i) \tag{7}$$

proportional zu Gleichung (6) [7]. Diese Wahrscheinlichkeiten sind dem Modell bekannt, da sie sich durch einfache Beobachtung bestimmen lassen.

Obwohl im Anwendungsbereich der Empfehlungssysteme die Auftretenswahrscheinlichkeit von Keywords nicht unabhängig von der inhaltlichen Klassifikation des Dokumentes ist, erzielen die Autoren in [6] sehr gute Ergebnisse. Bei einer experimentellen Untersuchung von Webseitenempfehlungen werden mit dem naiven Bayes-Klassifikator keine schlechteren Ergebnisse als mit aufwändigen anderen Methoden des maschinellen Lernens erzielt. Aber auch im Vergleich zur Empfehlung mittels Rocchios Algorithmus und TD/IDF schneidet der Bayes-Klassifikator nicht schlechter ab.

Probleme. In [2] diskutieren die Autoren drei hauptsächliche *Probleme* der inhaltsbasierten Systeme.

Zunächst besteht die Gefahr, den Inhalt einer Ressource nur unzureichend zu beschreiben. Gerade für nicht-textuelle Ressourcen sind in der Regel die Methoden des Information Retrieval nicht anwendbar. Durch eine *unzureichende*

[2] In [2] wird auf eine Vielzahl solcher Algorithmen verwiesen.

Inhaltsanalyse sind eventuell mehrere Ressourcen für das Empfehlungssystem ununterscheidbar; auch wenn eindeutige Qualitätsunterschiede der Ressourcen erkennbar sind.

Zum zweiten besteht die Gefahr der *Überspezialisierung.* Durch die Einschränkungen auf Profile mit interessanten Keywords können potentiell interessante Inhalte nicht empfohlen werden, wenn die dazugehörigen Keywords nicht ein Profil ergeben, welches ähnlich genug zu anderen, bereits vom Nutzer als interessant bewerteten, Ressourcen ist.

Außerdem können *neuen Nutzern* solange überhaupt keine Inhalte empfohlen werden, bis diese eine ausreichende Anzahl an Ressourcen selbst bewertet haben. Erst damit steht eine ausreichende Datengrundlage für ein Profil oder ein Modell der Interessen des Benutzers zur Verfügung.

2.2 Kollaborative Systeme

Im Gegensatz zu inhaltsbasierten versuchen *kollaborative Systeme* fehlende Bewertungen $u(c, s)$ durch Bewertungen $u(c_i, s)$ von anderen Nutzern anzunähern. Dazu werden für einen Benutzer c möglichst ähnliche Benutzer c_i gesucht [2].

Nearest-Neighbor-Methode. Um die fehlende Bewertung $r_{c,s} = u(c, s)$ zu erhalten wird bei der *Nearest-Neighbor-Methode* die Aggregation der Bewertungen einer Menge ähnlicher Benutzer betrachtet:

$$r_{c,s} = \text{aggr}_{c' \in \hat{C}} r_{c',s} \quad . \tag{8}$$

Die Menge $\hat{C}$ umfasst dabei nur eine Anzahl der zu c ähnlichsten Benutzer.

Es stellt sich also die Frage, wie die Ähnlichkeit zweier Benutzer ausgedrückt werden kann. Dazu wird ein Distanzmaß $\text{sim}(c, c')$ eingeführt. Es wird üblicherweise die Menge $S_{c,c'}$ an Ressourcen betrachtet, die sowohl von c als auch c' bewertet wurden. Nach [2] ist ein üblicher Ansatz die Berechnung als *Pearson-Korellations-Koeffizient*

$$\text{sim}(c, c') = \frac{\sum_{s \in S_{c,c'}} (r_{c,s} - \bar{r}_c)(r_{c',s} - \bar{r}_{c'})}{\sqrt{\sum_{s \in S_{c,c'}} (r_{c,s} - \bar{r}_c)^2 \sum_{s \in S_{c,c'}} (r_{c',s} - \bar{r}_{c'})^2}} \quad . \tag{9}$$

Dabei steht $\bar{r}_c$ für das arithmetische Mittel aller von c abgegebenen Bewertungen. Ebenso ist die Berechnung über das in Abschnitt 2.1 vorgestellte *Kosinus-Maß*

$$\text{sim}(c, c') = \cos(\boldsymbol{c}, \boldsymbol{c'}) = \frac{\boldsymbol{c} \cdot \boldsymbol{c'}}{||\boldsymbol{c}||_2 \times ||\boldsymbol{c'}||_2} \tag{10}$$

möglich. Dabei sind $\boldsymbol{c}$ und $\boldsymbol{c'}$ zwei $|S_{c,c'}|$-Vektoren. Diese Vektoren beinhalten die jeweiligen Bewertungen der Benutzer.

Damit können nun die n zu c ähnlichsten Benutzer $\hat{C}$ ausgewählt werden. Die *Aggregatsfunktion* aggr aus Gleichung (8) kann ebenfalls auf unterschiedliche Art und Weise definiert werden. Denkbar ist ein arithmetisches Mittel als

$$\text{aggr}_{c' \in \hat{C}} r_{c',s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s} \quad . \tag{11}$$

Die nach [2] wohl verbreitetste Definition verwendet ein gewichtetes Mittel

$$\text{aggr}_{c' \in \hat{C}} r_{c',s} = \alpha \sum_{c' \in \hat{C}} \text{sim}(c, c') r_{c',s} \quad , \tag{12}$$

mit einem Normalisierungsfaktor α. Der Vorteil dieser Aggregatsfunktion liegt darin, dass zueinander ähnliche Benutzer stärker gewichtet werden. Theoretisch kann damit auch die Menge aller Nutzer $\hat{C} = C$ betrachtet werden kann. Solche Nutzer die keine oder nur geringe Ähnlichkeit zu c haben werden aufgrund eines entsprechenden Maßes sim nicht oder nur mit geringem Einfluss berücksichtigt.

Modellbasiert. Wie auch bei den inhaltsbasierten Systemen gibt es auch bei den kollaborativen Empfehlungssystemen *modellbasierte Ansätze.* Solche Methoden werden etwa von Breese, Heckermann und Kadie untersucht. In [8] schlagen sie dazu ein probabilistisches Modell vor. Dabei wird eine Bewertung in den natürlichen Zahlen zwischen 0 und n vorausgesetzt. Dann liegt die erwartete Bewertung

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \cdot P(r_{c,s} = i | r_{c,s'},\ s' \in S_c) \tag{13}$$

bei der Summe der gewichteten möglichen Bewertungen i. Gewichtet wird mit der Wahrscheinlichkeit, dass diese Ressource s die Bewertung i erhält unter der Bedingung, dass der Benutzer bereits andere Ressourcen bewertet hat.

Ein Ansatz verwendet nun wieder einen *Bayes-Klassifikator.* Dazu wird angenommen, dass die Bewertungswahrscheinlichkeit nur von einem Klassifikationskriterium abhängt und von allen anderen vorliegenden Bewertungen unabhängig ist. Es sind jedoch auch eine Vielzahl anderer Methoden des Maschinellen Lernens einsetzbar.

Probleme. Auch für kollaborative Systeme sehen die Autoren in [2] drei wesentliche *Probleme.* Das Problem der *neuen Benutzer* bleibt das selbe wie bei inhaltsbasierten Systemen. Für einen neuen Benutzer können solange keine ähnlichen Benutzer für fehlende Bewertungen herangezogen werden, bis der neue Benutzer eine ausreichende Anzahl an Ressourcen selbst bewertet hat.

Zum anderen können auch *neue Ressourcen* solange keinen Benutzern empfohlen werden, bis sie ausreichend oft bewertet wurden. Kollaborative Systeme verlassen sich ja gerade auf Empfehlungen von Benutzern, anstatt direkt Inhalte zu vergleichen.

Das dritte Problem ist, dass üblicherweise die *Bewertungsmatrix* der Bewertungen aller Ressourcen von allen Nutzern *dünn besetzt* ist. Es wird von den Nutzern nur ein Bruchteil der tatsächlich vorhandenen Ressourcen selbst bewertet. Gibt es nun einen Benutzer dessen Bewertung dieser Ressourcen auch noch im Vergleich zu anderen Benutzern ungewöhnlich ist, so finden sich keine ausreichend ähnlichen Benutzerprofile und das System kann keine Empfehlungen mehr geben. Pazzani versucht in [9] das Problem durch die Einbeziehung demographischer Merkmale zu umgehen. Nach [2] existieren mehrere verschiedene Ansätze zur Lösung des Problems.

2.3 Hybride Systeme

Um die einzelnen Vorteile unterschiedlicher Ansätze besser zu nutzen, existieren vielfältige Ansätze, verschiedene Empfehlungssysteme zu kombinieren. Neben dem Überblick in [2] gibt BURKE in [10] einen guten Einblick in die Materie. Er klassifiziert *hybride Ansätze* in sieben Kategorien:

- Bei einem *gemischten Ansatz* werden verschiedene Empfehlungssysteme genutzt. Die gesammelten Empfehlungen aller dieser Systeme werden dann gemeinsam ausgegeben.
- *Gewichtete Ansätze* nutzen ebenfalls verschiedene Empfehlungssysteme. Die einzelnen Empfehlungen werden mittels verschiedener Gewichte kombiniert. Im Ergebnis wird eine Empfehlung gegeben.
- *Wechselnde Ansätze* nutzen für eine konkrete Empfehlung immer nur ein Empfehlungssystem. Es existieren jedoch Kriterien anhand derer das befragte Empfehlungssystem festgelegt wird und je nach Situation wechseln kann.
- Daneben ist die *Kaskadierung* von Empfehlungssystemen möglich. Dabei werden die Systeme nacheinander genutzt, um jeweils das Empfehlungsergebnis des vorherigen Systems zu verfeinern.

Die anderen drei Ansätze sind nicht solch generischer Natur:

- Zunächst können einzelne (besonders positive) Eigenschaften von verschiedenen Algorithmen zu einem neuen, *hybriden Algorithmus* vereint werden.
- Zum anderen ist es möglich, die *Ausgabe* von vorhandenen Empfehlungssystemen in Algorithmen eines anderen Systems als *Feature* zu benutzen und so die zugrunde liegende Informationsmenge zu erhöhen.
- Ein letzter Ansatz findet auf *Meta-Ebene* statt. Es wird durch ein erstes Empfehlungssystem ein Modell empfohlen, welches von einem anderen System dann als Eingabe verwendet wird.

Wie sich bereits bei dieser knappen Aufzählung erahnen lässt, unterliegt jeder dieser sieben Ansätze seinen eigenen Beschränkungen und hat wiederum eigene Vor- und Nachteile. In [10] werden diese näher untersucht und veröffentlichte, hybride Ansätze entsprechend der oben skizzierten Einteilung klassifiziert.

3 Web 2.0 Resource Sharing Platforms

Web 2.0 Plattformen wie Del.icio.us[3], Flickr und YouTube[4] sind prominente Beispiele für Plattformen, auf denen verschiedene Ressourcen innerhalb einer Gemeinschaft von Nutzern gesammelt und zugänglich gemacht werden. Diese Ressourcen werden in der Regeln von den Benutzern mit beliebig wählbaren Schlagworten, sogenannten *Tags*, versehen (getaggt) [11].

[3] http://del.icio.us – Plattform zum Speichern und Austauschen von Bookmarks.
[4] http://www.youtube.com – Plattform zum Hochladen, Speichern und Kommentieren von Videos.

Das grundlgende Modell solcher Plattformen hat VANDER WAL in [12] mit dem Begriff *Folksonomie* beschrieben. Aufbauend auf einer entsprechenden mathematischen Definition untersucht der folgende Abschnitt, inwiefern Empfehlungssysteme in solchen Plattformen eine Rolle spielen können. Der Schwerpunkt liegt dabei auf der Empfehlung von dem Nutzer bisher unbekannten Ressourcen.

3.1 Folksonomien

Nach einer Definition von VANDER WAL, die ursprünglich auf 2004 zurückgeht, ist eine Folksonomie „das Ergebnis des persönlichen, freien Taggens von Informationen und Objekten [...] zur persönlichen Suche“ (*engl.:* „the result of personal free tagging of information and objects [...] for one's own retrieval“) [12]. Weitere Aspekte sind ein soziales Umfeld, welches einen gemeinsamen Zugriff und das Taggen durch potenziell jede Person, die auf ein Objekt zugreift, erlaubt.

Problematisch ist eine immer weitere Aufweichung dieser Definition. In [13] beklagt sich VANDER WAL selbst, dass der Begriff Folksonomie vielfach unterschiedlich ausgelegt wird; insbesondere die Definition in Wikipedia[5] gleiche mehr einer Funktionsbeschreibung diverser Tagging-Tools.

SCHMITZ et. al. haben sich in [14] mit einer mathematischen Definition von Folksonomien beschäftigt. Die dortige, vereinfachte Definition wird im Folgenden verwendet:

Definition 1 (Folksonomie). *Eine Folksonomie besteht aus einem Quadrupel*

$$F = (U, T, R, Y) \ . \tag{14}$$

Die endlichen Mengen U, T und R bezeichnen die Mengen der Benutzer, Tags und Ressourcen. $Y \subset U \times T \times R$ ist eine dreistellige Relation, welche Elemente der einzelnen Mengen miteinander verknüpft.

Zumindest die ursprüngliche Definition von VANDER WAL mag an einigen Stellen problematisch sein. Viele prominente Web 2.0 Plattformen beschränken die Folksonomie, indem sie beispielsweise ein Rechtesystem zum Taggen vorsehen. Eine Einordnung von Tagging-Systemen anhand verschiedener Dimensionen mit verschiedenen Ausprägungen nehmen MARLOW et. al. in [11] vor. Für die folgenden Betrachtungen reicht jedoch die mathematische Definition aus Gleichung (14) aus.

3.2 Tag-Empfehlung

Spätestens mit einer mathematischen Definition des Folksonomie-Begriffes wurden in den letzten Jahren entsprechende Web 2.0 Plattformen intensiv in der Forschung untersucht. Ein naheliegender Untersuchungsgegenstand bei solchen Tagging-Systemen liegt in der *Tag-Empfehlung.*

[5] http://www.wikipedia.org

Beispielsweise behandeln JÄSCHKE et. al. in [15] diese Thematik. Sie greifen dabei ebenfalls auf die Definition 1 zurück und verwenden zur Empfehlung einen Ansatz wie in Abschnitt 2.2 mit der Nearest-Neighbor-Methode zum Kollaborativen Filtern vorgestellt wurde.

Die Empfehlung von Tags beschränkt sich aber auf den Prozess des Taggens von Ressourcen, die durch den Nutzer bereits auf anderem Wege vorher gefunden wurden. Die Suche nach interessanten Ressourcen kann für einen Benutzer natürlich ebenfalls durch ein Empfehlungssystem erleichtert werden. Die möglichen Ansätze, eine solche Empfehlung von Ressourcen durch die Einbeziehung der Tags zu verbessern, beschreibt der folgende Abschnitt.

3.3 Ressource-Empfehlung

Die Nutzung von Tags, um eine bessere *Empfehlung von Ressourcen* zu ermöglichen, scheint bislang kaum untersucht worden zu sein. Neben der typischen Beziehung zwischen Benutzern und Ressourcen können zum einen die Benutzer-Tag-Beziehung, also welche Benutzer welche Tags vergeben haben, und die Ressourcen-Tag-Beziehung, also mit welchen Tags interessante Ressourcen getaggt sind, betrachtet werden. Einen solchen generischen Ansatz verfolgen TSO-SUTTER, MARINHO und SCHMIDT-THIEME in [16]. Sie bezeichnen diese Zusammenführung der verschiedenen Relationen als Fusion.

Ausgangspunkt für die Betrachtung in [16] ist wieder eine Folksonomie nach Definition 1. Die dreistellige Relation Y kann in drei zweistellige Relationen für die Beziehungen zwischen *Benutzer-Ressourcen, Tags-Ressourcen und Tags-Benutzer* zerlegt werden. Diese Relationen können durch drei Matrizen R_{UR}, R_{TR} und R_{UT} dargestellt werden. Wir gehen davon aus, dass die Matrizen die Form

$$R_{UR} = \begin{pmatrix} a_{1,1} & \dots & a_{1,n_r} \\ \vdots & & \vdots \\ a_{n_u,1} & \dots & a_{n_u,n_r} \end{pmatrix} \tag{15}$$

$$R_{TR} = \begin{pmatrix} b_{1,1} & \dots & b_{1,n_r} \\ \vdots & & \vdots \\ b_{n_t,1} & \dots & b_{n_t,n_r} \end{pmatrix} \tag{16}$$

$$R_{UT} = \begin{pmatrix} c_{1,1} & \dots & c_{1,n_t} \\ \vdots & & \vdots \\ c_{n_u,1} & \dots & c_{n_u,n_t} \end{pmatrix} \tag{17}$$

haben, wobei n_u, n_r und n_t die Anzahl der Benutzer, Ressourcen und Tags sind.

In einem herkömmlichen Empfehlungssystem, welches kein weiteres Hintergrundwissen verwendet, findet nur die Beziehung zwischen Benutzern und Ressourcen Berücksichtigung. Tags als zusätzliche Informationen werden nicht berücksichtigt. Eine andere Idee liegt darin, ein Benutzerprofil, anstatt mit den Ressourcen, mit verwendeten Tags aufzubauen. Einen solchen Ansatz beschreiben DIEDERICH und IOFCIU in [17].

Der generische Ansatz der *Fusion* aus [16] kombiniert beide Ansätze. Zunächst wird eine *erweiterte User-Matrix*

$$R_{U_{\text{ext}}} = R_{UR} + R_{UT} = \begin{pmatrix} a_{1,1} & \dots & a_{1,n_r} & c_{1,1} & \dots & c_{1,n_t} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{n_u,1} & \dots & a_{n_u,n_r} & c_{n_u,1} & \dots & c_{n_u,n_t} \end{pmatrix} \tag{18}$$

erstellt. Damit ist nun ein kollaboratives Filtern im Sinne von Abschnitt 2.2 möglich. Die Ähnlichkeit der Nutzer durch das sim-Maß bestimmt sich jedoch nicht mehr nur durch Gemeinsamkeiten bei den bewerteten Ressourcen, sondern auch durch Ähnlichkeiten bei den durch die Benutzer vergebenen Tags.

Analog kann eine *erweiterte Ressourcen-Matrix*

$$R_{R_{\text{ext}}} = R_{UR} + R_{TR} = \begin{pmatrix} a_{1,1} & \dots & a_{1,n_r} \\ \vdots & & \vdots \\ a_{n_u,1} & \dots & a_{n_u,n_r} \\ b_{1,1} & \dots & b_{1,n_r} \\ \vdots & & \vdots \\ b_{n_t,1} & \dots & b_{n_t,n_r} \end{pmatrix} \tag{19}$$

aufgestellt werden. Auch hier können nun bekannte Methoden angewendet werden. Tso-Sutter, Marinho und Schmidt-Thieme vereinigen in [16] diese Ergebnisse (auf Basis der beiden erweiterten Matrizen) nun mit einem Algorithmus und erhalten so eine Empfehlung, die auf der Vereinigung zweier kollaborativer Filterungsprozesse beruht.

Sie evaluierten ihren Ansatz auf Basis einer Stichprobe der Plattform last.fm[6]. Dabei greifen sie auf ca. 1800 Ressourcen, 2900 Benutzer und 2000 Tags zurück. In dieser Evaluation zeigen sich bessere Ergebnisse als mit herkömmlichen Methoden. Es liegen also Anhaltspunkte dafür vor, dass der vorgestellte Ansatz auch im Allgemeinen zu deutlich besseren Ergebnissen führen könnte.

3.4 Bewertung

Golder und Huberman analysieren in [18] das Verhalten der Nutzer der Plattform del.icio.us. Eine Erkenntnis liegt darin, dass das Verhältnis der verwendeten Tags pro Nutzer exponentiell verteilt ist. Eine geringe Anzahl an Nutzern trägt einen sehr großen Anteil der Tags bei. Eine große Anzahl von Nutzern taggt nur in sehr geringem Umfang. Obwohl diese Analyse nur auf einer Stichprobe aus del.icio.us beruht, liegt der Schluss nahe, dass auch in anderen Web 2.0 Plattformen eine ähnliche Verteilung vorliegt.

Dieser Umstand kann naive Ansätze, welche Tags zur Empfehlung von Ressourcen nutzen, vor Probleme stellen. Diederich und Iofciu verwenden in [17] Tags zu bibliographischen Angaben, um wissenschaftliche Publikationen den Interessen eines Nutzers entsprechend zu empfehlen. Dabei werden ausschließlich

[6] www.lastfm.de – Eine Plattform für Musikempfehlungen.

die Tags als Features zur Beschreibung von Ressourcen verwendet. Obwohl die Autoren angeben, dass nur zu etwa 20% der Ressourcen Tags vorlagen, erzielen sie überraschend gute Ergebnisse. Dabei fehlen jedoch Angaben zur Verteilung der Tags über die Benutzer und Ressourcen. Der Argumentation der Autoren zufolge scheint es sich jedoch um eine Gleichverteilung zu handeln.

Ein solcher Ansatz, der ausschließlich vergebene Tags als Features verwendet, führt bei einer realistischen Exponentialverteilung jedoch zu neuen Problemen: Die in Abschnitt 2 geschilderten Problematiken der „neuen Benutzer“ oder „neuen Ressourcen“ wandeln sich zu Problemen der „unterdurchschnittlich taggenden Benutzer“ bzw. „unterdurchschnittlich getaggten Ressourcen“. Bei einer Exponentialverteilung, wie in [18] ermittelt, sind also die Mehrzahl der Benutzer und Ressourcen von diesen Problemen betroffen.

Aus diesem Grund erscheint der kombinierte Ansatz von Tso-Sutter, Marinho und Schmidt-Thieme in [16] aussichtsreicher. Durch die dort veröffentlichten experimentellen Ergebnisse übertrifft ein solcher kombinierter Ansatz auch die Ergebnisse von Ansätzen, die das Hintergundwissen, welches durch die Tags zur Verfügung steht, nicht nutzen.

4 Zusammenfassung und Ausblick

Empfehlungssysteme sind in ihren Grundlagen sehr gut verstanden und werden auch vielfach in produktiver Umgebung eingesetzt. Solche Systeme werden üblicherweise in inhaltsbasierte und kollaborative, sowie heuristische und modellbasierte Systeme unterteilt. Für diese Klassen existieren jeweils gut beherrschte Algorithmen, welche in der Regel ausreichende Ergebnisse erzielen und gute Kompromisse bezüglich der Komplexität darstellen. Exemplarisch wurden ausgewählte Algorithmen im zweiten Abschnitt, zusammen mit den jeweiligen möglichen Problemen, vorgestellt. Um spezifische Nachteile zu umgehen und die Vorteile mehrerer Systeme zu vereinen werden hybride Empfehlungssysteme eingesetzt.

In Web 2.0 Plattformenen arbeiten mehrere Benutzer kollaborativ zusammen. Ressourcen werden allen Benutzern gemeinsam zur Verfügung gestellt und können getaggt werden. Die grundlegende Idee hinter diesem Zusammenspiel von Benutzern, Ressourcen und Tags wird durch den Begriff der Folksonomie beschrieben. Wenn auch diese Definition auf die meisten Plattformen nicht genau zutrifft – da meistens zusätzliche Einschränkungen vorliegen –, so ist sie dennoch zur Betrachtung der Möglichkeiten von Empfehlungssystemen in diesem Bereich geeignet.

Der Einsatz von Empfehlungssystem zur Empfehlung von Tags während des konkreten Tagging-Vorgangs zu einer bestimmten Ressource wurde bereits mehrfach untersucht. Weitaus seltener befasst sich die wissenschaftliche Literatur jedoch mit der Fragestellung, inwiefern Tags als zusätzliches Hintergrundwissen zur Empfehlung von Ressourcen genutzt werden können. Ein aktueller Ansatz verwendet die Nutzer-Tag- und die Ressourcen-Tag-Beziehung um die Eingabedaten zu bekannten Algorithmen von Empfehlungssystemen anzureichern.

Nach einer ersten Betrachtung erscheint diese kombinierte Methode sehr erfolgversprechend. Es scheinen damit bessere Ergebnisse erzielbar als ohne die Berücksichtigung der Tag-Daten. Auch scheint es unzureichend, die Tags lediglich als Features zur alleinigen Erstellung von Nutzerprofilen zu verwenden.

Zur weiteren Forschung sollte demnach der beschriebene kombinierte Fusion-Ansatz weiterverfolgt werden. Nächste interessante Schritte könnten die Evaluation verschiedener klassischer Algorithmen aus dem Bereich der Empfehlungssysteme mit diesem Ansatz oder die Evaluation in Bezug auf unterschiedliche Strukturen einer Folksonomie sein.

Literatur

1. Oates, G.: Holy moly! – Flickr Blog. (13. Nov. 2007) http://blog.flickr.net/en/2007/11/13/holy-moly/ vom 8. Aug. 2008
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6) (2005) 734–749
3. Höhfeld, S., Kwiatkowski, M.: Empfehlungssysteme aus informationswissenschaftlicher sicht-state of the art. IWP **58**(5) (2007) 265–276
4. Rich, E.: User modeling via stereotypes. Cognitive Science **3**(4) (1979) 329–354
5. Salton, G.: Automatic Text Processing. Addison-Wesley (1989)
6. Pazzani, M., Billsus, D.: Learning and revising user profiles: The identification of interesting web sites. Machine Learning **27**(3) (1997) 313–331
7. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Zweite Auflage. Prentice Hall, New Jersey (2003)
8. Breese, J.S., Heckermann, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. 14th UAI. (1998) 43–52
9. Pazzani, M.: A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review **13**(5–6) (1999) 393–408
10. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction **12**(4) (2002) 331–370
11. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, To Read. In: Proc. of HYPERTEXT '06. (2006) 31–40
12. Vander Wal, T.: Folksonomy. (2. Feb. 2007) http://vanderwal.net/folksonomy.html vom 17. Juni 2008.
13. Vander Wal, T.: Folksonomy definition and wikipedia. (2. Nov. 2005) http://www.vanderwal.net/random/entrysel.php?blog=1750 vom 17. Juni 2008.
14. Schmitz, C., Hotho, A., Jäschke, R., Stumme, G.: Kollaboratives Wissensmanagement. In Semantic Web - Wege zur vernetzten Wissensgesellschaft. Springer (2006) 273–290
15. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag recommendations in folksonomies. In: Proc. of PKDD 2007. (2007) 506–514
16. Tso-Sutter, K., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: Proc. of the 2008 ACM symposium on Applied computing. (2008) 1995–1999
17. Diederich, J., Ioficu, T.: Finding communities of practice from user profiles based on folksonomies. In: Proc. of TEL-CoPs'06. (2006) 288–297
18. Golder, S., Huberman, B.A.: The structure of collaborative tagging systems. Technical report, HP Labs (2005) http://www.hpl.hp.com/research/idl/papers/tags/.

Supporting Knowledge Creation and Sharing in Social Networks

Markus Rahm

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI)
Integriertes Seminar Sommersemester 2008
markus-rahm@web.de

Betreuer: Dipl.-Inf. Darko Obradovič

05. August 2008

Zusammenfassung Diese Arbeit befasst sich mit Prozessen zur Unterstützung der Wissensgewinnung und Wissensverteilung innerhalb sozialer Netzwerke. Durch ungleichmäßige Informationsverteilung innerhalb sozialer Netze, wie beispielsweise in größeren Unternehmen, kommt es zu Einschränkungen der Effizienz, welche oftmals durch übermäßige Kommunikation verursacht wird. Zur Verbesserung dieser Wissensverteilung wird das Verfahren der sozialen Netzwerkanalyse vorgestellt. Dieses analysiert systematisch die sozialen Strukturen und definiert, beziehungsweise visualisiert, die enthaltenen Informationsflüsse. Anschließend werden die verschiedenen Dimensionen dieser Informationsflüsse in einer kombinierten Sicht zusammengefasst, um daraus geeignete Verbesserungsvorschläge gewinnen zu können.

1 Einleitung

Wenn wir darüber nachdenken, woher Menschen ihre Informationen beziehen, denken wir an Datenbanken, Netzwerke wie Internet/Intranet, Suchportale oder klassische Quellen wie Bücher oder Magazine. Doch eine wesentliche Komponente zur Informationsbeschaffung wird dabei gerne vergessen: Andere Menschen.

Studien der Soziologie und Sozialpsychologie haben gezeigt, dass wen man kennt wesentlichen Einfluss auf das eigene Wissen hat. Persönliche Beziehungen sind ausschlaggebend für die eigene Informationsgewinnung, die Problemlösungsansätze, sowie die eigene Effizienz der Arbeit. Daher ist, durch den stetigen Wandel der Wirtschaft in Richtung zentral organisierter Unternehmensformen, besonderes Augenmerk auf die Beziehungen der Mitarbeiter untereinander zu legen. Gestützt auf diese Problematik wurde eine Studie gestartet, um Manager in den Phasen der Wissenskreierung, Wissensteilung und dem dazugehörigen Lernprozess zu unterstützen. Dazu wurden 40 Manager einer Unternehmung befragt, die

gemäß Experten bereits ein hohes Maß an Aufwand betrieb, um eine gemeinsame, organisierte Wissensbasis zu schaffen. Daraus ergaben sich folgende vier Charakteristika, um effektive von ineffektiven Beziehungen zu unterscheiden:

1. **Wissen**
   Das Bewusstsein darüber, welche Informationen andere Personen zur Verfügung stellen können.
2. **Zugang**
   Kenntnis darüber, was ein anderer weiß, ist nur dann von Nutzen, sofern derjenige auch erreichbar ist. Zugang zu Kenntnissen anderer ist oft durch räumliche Grenzen oder verschiedene Unternehmensformen beschränkt.
3. **Engagement**
   Diejenigen, die den Lerneffekt bedeutend fördern, sind nicht die, die einem mit unstrukturierten Informationen überfluten, sondern jene, die sich in das Problem einfühlen und daraufhin ihr Wissen entsprechend anwenden.
4. **Sicherheit**
   Eine Basis des Vertrauens und der Sicherheit gegenüber anderen Personen führt oft zu einer Verbesserung des Kreativitäts- und des Lernprozesses. Etwaige Wissenslücken werden oftmals erst nach Schaffung einer Vertrauensbasis gegenüber anderen zugegeben.

Die befragten Manager stellten einige Beispiele heraus, in denen der Prozess des Lernens und der Wissensteilung nicht stattfinden konnte, da mindestens ein Kriterium fehlte. Beispielsweise die Existenz eines fähigen Kollegen, der jedoch aufgrund räumlicher Gegebenheiten nicht erreichbar war.

Doch woher wissen die Mitarbeiter, welcher ihrer Kollegen welches Wissen zur Verfügung stellen kann? Welche innerbetrieblichen Beziehungen existieren? Diese Verbindungen müssen definiert und analysiert werden, beispielsweise mit Hilfe der sozialen Netzwerkanalyse.[1]

2 Soziale Netzwerkanalyse

Die Grundidee hinter der sozialen Netzwerkanalyse ist recht einfach. Jeder kennt Netzwerke bestehend aus Knoten und Kanten, wie beispielsweise Verkehrsnetze bestehend aus Orten und Straßen. Ein soziales Netzwerk dagegen definiert sich durch eine Sammlung von Individuen, welche durch Beziehungen miteinander verknüpft werden. Dabei können mehrere Individuen sowie verschiedenartige Beziehungstypen auftreten. Zur Repräsentation und der Analyse solcher Netze ist eine formale Beschreibung nötig, da bereits zur aussagekräftigen Auswertung kleiner sozialer Netzwerke eine Vielzahl an Informationen gesammelt werden muss. Dabei wird häufig auf mathematisch gestützte Werkzeuge zurückgegriffen, da der Umgang mit solch einer Masse von Daten in größeren Netzen schnell sehr komplex werden kann. Dabei wird zur Berechnung von Kennzahlen gerne auf Matrizen zurückgegriffen, sowie zur visuellen Repräsentation Modelle aus der Graphentheorie verwendet. Dadurch ist eine kompakte und systematische Beschreibung der zugrundeliegenden Informationen, sowie eine effiziente

Berechnung komplexester Zusammenhänge möglich. Bevor die eigentliche Analyse beginnen kann, müssen die zugrundeliegenden Daten erhoben werden. Dabei muss darauf geachtet werden, welche Art von Informationen überhaupt benötigt, beziehungsweise Ziel der Analyse sind. Prinzipiell unterscheidet man zwei Typen von Daten[2]:

- **Attributive Daten** beziehen sich auf die Meinungen, Einstellungen und das Verhalten der Individuen. Oft werden die Daten durch Interviews oder Fragebögen erhoben. Der Fokus liegt dabei auf dem Individuum alleine, ohne Interaktionen oder Relationen mit anderen zu definieren. Die Daten lassen sich später einfach mit Hilfe statistischer Verfahren als Attribute in einer **Variablen-Analyse** messen und darstellen.
- **Relationale Daten** hingegen definieren die Zusammenhänge zwischen den Individuen. Beziehungen zueinander, Bindungen oder jegliche Form von Interaktion sind hierbei von Relevanz. Durch diese Relationen entstehen dann, durch Methoden der mathematischen Netzwerkanalyse der Graphentheorie, große, relationale Systeme zur weiteren Betrachtung [3].

Für den Fall der Unterstützung der Wissensschaffung und -verteilung bietet sich eine Analyse der relationalen Verbindungen an. Zur Erhebung der relevanten Daten gibt es verschiedene Verfahren. Einmal durch Beobachtung der einzelnen Personen, durch Auswerten vorhandener statistischer Daten oder durch Befragung der Personen zu ihren Beziehungsnetzen. Bei letzterem stellt sich zusätzlich die Frage, welche individuellen Personen befragt werden sollen? Dabei kann zwischen einer totalen Erhebung über alle Personen bis hin zu einer Stichprobenauswahl entschieden werden. Die Personen sollten jedoch sinnvoll gewählt sein, beispielsweise anhand gemeinsamer Kriterien wie lokaler Nähe innerhalb einer Abteilung.

Danach lassen sich verschiedene Metriken zur Bewertung der Netze und den jeweiligen Knoten in Betracht ziehen[4].

Für das Netzwerk:

- **Density** Über die Dichte eines sozialen Netzwerks lassen sich Aussagen über die Vollständigkeit der möglichen Verbindungen innerhalb eines Netzes treffen. Bei einer hohen Dichte werden weitestgehend alle Verbindungsmöglichkeiten zwischen den Individuen benutzt, sodass keine Lücken oder lange Kommunikationswege bestehen.[2]
- **Centralization** Gibt die Prominenz eines Individuums innerhalb des gesamten Netzwerks wider. Dabei wird die Organisationsstruktur des Netzwerks als Ganzes betrachtet. Um eine hohe, globale Centralization zu bekommen, müssen Individuen eine signifikant strategische Position im gesamten Netzwerk inne haben.[5]
- **Cohesion** Gibt an, wie stark mehrere Individuen direkt miteinander verbunden sind. Damit lassen sich Gruppen gut als **Cliquen** oder soziale Zirkel identifizieren.[2]

Für die Individuen:

- **Betweenness** Der Grad, inwieweit sich ein Individuum zwischen anderen im Netzwerk befindet. Erkennbar an Verbindungen von Dritten, die aufgrund mangelnder Verbindungen über die betroffene Person laufen müssen.[6]
- **Closeness** Der Grad der direkten und indirekten Nähe zu allen anderen Individuen. Personen mit einem geringen Closeness-Wert verfügen über kurze Distanzen zu anderen Personen.[5]
- **Degree** Die Anzahl der direkten Verbindungen zu anderen Individuen.[5]
- **Centrality** Spiegelt die Zentralität eines Individuums innerhalb seiner Umgebung wider, wobei sie nicht direkt messbar ist, sondern durch kombinierte Auswertung der Metriken Betweenness, Closeness und Degree errechnet wird.[5]

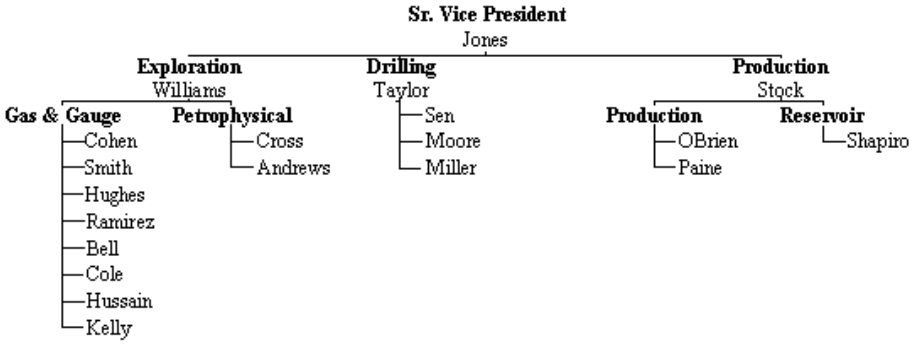

Abbildung 1. Hierarchie einer Unternehmung[1]

Bei der Betrachtung, ob sich solch ein Aufwand zu betreiben überhaupt lohnt, lässt sich der Effekt einer solchen Analyse gut an einem Beispiel erläutern. Abbildung 1 zeigt die Unternehmenshierarchie eines Ölkonzerns. Hierbei interessant sind die Personen Vize-Präsident JONES und Angestellter COLE. Durch eine Mitarbeiterbefragung, wer mit wem Informationen austauscht, ergab sich eine simple Matrix (siehe Abbildung 2).

Dadurch lies sich die informelle Organisationsstruktur ohne weiteres in Form eines Soziogrammes darstellen (siehe Abbildung 3). Hier wird deutlich, dass die Person COLE zu einem zentralen Dreh- und Angelpunkt geworden ist, wohingegen JONES nur eine periphere Rolle wahrnimmt. Nun stellt sich die Frage, wie es zu diesem Umstand kommen konnte, und was man dagegen unternehmen kann.

	Andrews	Bell	Cohen	Cole	Cross	Jones	...
Andrews	∞	0	0	1	0	0	...
Bell	0	∞	0	0	0	0	...
Cohen	0	0	∞	1	1	0	...
Cole	1	0	1	∞	1	1	...
Cross	0	0	1	1	∞	0	...
Jones	0	0	0	1	0	∞	...
...	...	...	...	...	...	...	...

Abbildung 2. Beispiel einer einfachen Beziehungsmatrix

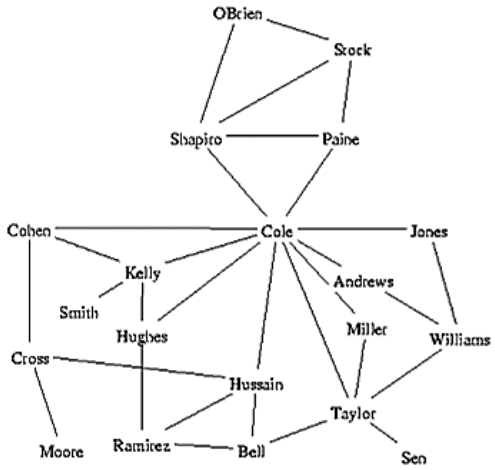

Abbildung 3. Soziogramm der informellen Organisationsstruktur[1]

3 Analyse der Ergebnisse

3.1 Wer weiß was?

Bei der Betrachtung der Kommunikationskanäle des oben genannten Beispiels lässt sich kein direkter Zusammenhang erkennen, wieso gerade Person A mit Person B Informationen austauscht. Die hierarchische Führungsstruktur der Unternehmung hat keinen direkten Einfluss auf das Ergebnis der Analyse. Wie kommt es also zu solch einem Informationsfluss? Das Problem liegt hierbei in dem fehlenden Bewusstsein, wer innerhalb der Unternehmung über welche Kenntnisse verfügt. Die meisten Informationsbeziehungen entstehen oft unsystematisch durch räumliche Gegebenheiten oder gemeinsame Zusammenarbeit in diversen Projekten, sodass sich die Beteiligten persönlich kennen. Hierbei können dann einzelne Personen schnell durch vielfache Mitarbeit in mehreren Projekten zu *Hot Spots* werden, unabhängig von ihrer unternehmenspolitischen Stellung. Die

Tatsache, dass für eine bestimmte Problemstellung gegebenenfalls eine kompetentere Ressource verfügbar sein könnte, wird hierbei vernachlässigt. Hier gilt es nun, durch geeignete Möglichkeiten diesen Mangel an Kenntnissen auszugleichen. Dazu eignen sich diverse Verfahren innerhalb eines Unternehmens, wie beispielsweise [1]:

- Der Aufbau von Fähigkeitsprofilen der Mitarbeiter.
- Die Verwendung von unternehmensinternen gelben Seiten, um Mitarbeiter für verschiedene Aufgaben zu kategorisieren.
- Die Bildung thematischer Gruppen, welche dann durch Help-Desk-Systeme organisiert sind, und diese, je nach Anforderung, zu den richtigen Ansprechpartnern routen.
- Die Schaffung von spezifischen Kommunikationsstätten (*Knowledge Fairs*), in denen Teams oder Abteilungen sich über ihre Kenntnisse bewusst werden und dabei die Ziele der dazugehörigen Projekte besser verinnerlichen. [7]

Durch die Verwendung solcher Methoden können alternative Möglichkeiten zur Problemlösung entwickelt werden. Diese können die vorhandene Kommunikationsstruktur effizienter ausnutzen, ohne dabei einzelne Personen zu über- oder unterbelasten. Doch leider reicht die alleinige Kenntnis darüber, dass jemand anderes eventuell etwas relevantes zu einer Problemlösung beisteuern könnte, nicht aus, wenn man keinen direkten Zugang zu dieser Person hat. [8]

3.2 Wer ist wo?

Die beste Information nützt nichts, wenn sie nicht erreichbar ist. Viele Unternehmen sind so groß, dass persönlicher Kontakt aller Mitarbeiter schier unmöglich ist. Oft existieren viele Zweigstellen, teilweise auf anderen Kontinenten, was eine enge Zusammenarbeit schwierig gestaltet. Um diesem Umstand entgegenzuwirken, greifen Unternehmen oft zu technischen Maßnahmen wie E-Mail, gemeinschaftliche Umgebungen, Videokonferenzen oder Instant Messenger. Dadurch lassen sich einfach räumliche und zeitliche Grenzen ohne besonders großen Aufwand überwinden, jedoch fehlt dabei immer der persönliche Kontakt zu allen Beteiligten. Einige Unternehmen gehen daher noch einen Schritt weiter und gruppieren alle Beteiligten räumlich. Beispielsweise verlegte die Firma Chrysler alle in die Entwicklung eines neuen Automobils involvierten Personen während der Entstehungszeit in ein einziges Gebäude, was sich im Endeffekt positiv auf das Ergebnis auswirkte.[1] Hierbei hängt der Erfolg solcher Methoden oft davon ab, inwieweit sich Einzelpersonen in die Problemstellung einarbeiten können und wollen. Dieses individuelle Engagement gilt es nun zu fördern.

3.3 Eine Frage des Engagements

Wie bereits erwähnt, ist es essentiell wichtig, dass die involvierten Personen ein hohes Maß an Engagement zur Verfügung stellen. Dies ist nicht immer einfach zu gewährleisten. Starke Arbeitsüberlastung, fehlende räumliche Nähe oder mangelndes Vertrauen spielen oft eine große Rolle, so dass Faktoren wie Motivation

und Engagement der Mitarbeiter unterdurchschnittlich stark ausgeprägt sind. Um diesem Umstand entgegenzuwirken nutzen Unternehmen meistens technische Methoden, wie den Einsatz von virtuellen Buddy Systemen, Software die für ortsunabhängige, synchrone Teamarbeit konzipiert ist, oder Whiteboarding-Technologien. Dadurch lässt sich verstreutes Arbeiten an einem gemeinsamen Problem gewährleisten. Sicherlich unterstützen solche Verfahren die Kommunikation und Interaktion zwischen Personen, die sich nicht persönlich gegenüberstehen, dennoch können sie nicht das Wissen wie bei einer Zusammenarbeit von Angesicht zu Angesicht vermitteln. Jegliche Form von Körpersprache, beziehungsweise bestimmte Reaktionen während eines Gespräches, können unterbewusst die Kommunikation beeinflussen. Daher eignen sich Videokonferenzsysteme zur visuellen Interaktion besonders gut, um die interne Kommunikation zu unterstützen.[1]
Einen interessanten Ansatz zur Verbesserung des Engagements zeigte die Firma BRITISH PETROLEUM, die im Bereich des Wissensmanagement viel Ehrgeiz zeigt. Durch Bildung eines *Peer Review* Prozesses wurde eine Möglichkeit gefunden, diejenigen Personen mit dem höchsten Maß an Engagement herauszufiltern. Dazu wurde, bevor jemand eine Aufgabe eines Problemlösungsprozesses übernimmt, dieser gebeten, seine eigenen Vorstellungen der Problematik wiederzugeben. Da der Schwerpunkt dabei auf einer schnellen Problemlösung lag, wurden nur diejenigen ausgewählt, die über das höchste Maß an Wissen und Erfahrung verfügten. Dadurch konnte nicht nur die Effizienz der Problemlösung verbessert werden, auch steigerte dies das Bewusstsein über die Einzelfähigkeiten und das Können der jeweiligen Personen. Ein weiterer Vorteil bei der Verwendung dieser Verfahren ist, dass durch den gesteigerten persönlichen Kontakt der beteiligten Personen eine grundlegende Vertrauensbasis geschaffen wird.[9]

3.4 Der Faktor Mensch

Ein weiterer, überaus wichtiger Punkt bei der Zusammenarbeit während einer Problemlösung ist der Grad an Vertrauen, dem man seinem Gegenüber schenkt, beziehungsweise mit welcher Sicherheit seine Kenntnisse bewertet werden. Oftmals existieren gut funktionierende Kommunikationswege mit einem hohen Maß an Wissensaustausch, guten Zugängen zwischen den Personen, sowie einem ausgeprägter Grad an Engagement, jedoch mit einem gravierenden Mangel an Vertrauen und Sicherheit. Doch welche Erfolge können erzielt werden, wenn die Verlässlichkeit erlangter Informationen nicht gewährleistet werden kann? Fehlende Sicherheit kann somit bewusst, als auch unbewusst, den Lernprozess gerade in kritischen Situationen wesentlich beeinflussen. Der dabei bedeutendste Umstand ist die Tatsache, dass involvierte Personen nicht gerne einen Mangel an Wissen oder Erfahrung zugeben, wenn sie sich aufgrund des geringen Vertrauens unsicher fühlen.[10]

Doch leider gibt es für die Bildung von Vertrauen und Sicherheit keine Patentrezepte. Analysen der Vertrauensbeziehungen innerhalb einiger der FORTUNE 500 Unternehmungen der USA konnten zeigen, dass Beziehungen längere Zeit, sowie physischen, kognitiven und sozialen Raum benötigen, um ein Gefühl

von Sicherheit zu entwickeln. Dabei unterstützen Interaktionen von Angesicht zu Angesicht diesen Prozess immens. [1]

4 Eine kombinierte Sicht der Ergebnisse

Nach einer Einzelbetrachtung der unterschiedlich ausgeprägten Netzwerke gibt eine zusammenfassende Sicht über die verschiedenen Dimensionen einen Überblick darüber, welches Potential letztendlich in der bestehenden Struktur steckt, und wo Verbesserungsvorschläge angebracht werden können. Durch die jeweiligen Analysen der vier Dimensionen Wissen, Zugang, Engagement und Sicherheit ergeben sich vier unabhängige Graphen einer gemeinsamen Menge an Knoten, den Mitarbeitern. Mit dieser Knotenmenge als Basis lässt sich nun ein absoluter Graph erstellen. Dabei wird nur dann eine Kante zwischen zwei Knoten hinzugefügt, wenn die Kante in allen vier Graphen existiert. Dadurch wird direkt ersichtlich, wie stark die verschiedenen Dimensionen Wissen, Zugang, Engagement und Sicherheit im Zusammenhang ausgeprägt sind. Daraus resultierend ist direkt erkennbar, welche Personen innerhalb des Netzwerkes eine kritische Position innehaben und welche Personen, im Bezug auf Wissenskreierung und Wissensteilung, eine untergeordnete Rolle spielen.

Durch das Verständnis, wer eine zentrale Rolle innerhalb des Netzwerks trägt, lassen sich Überbelastungen einzelner, sowie das Bewusstsein darüber, wer eine wertvolle Ressource für die Unternehmung ist, definieren. Dafür müssen jedoch erst die Gründe analysiert werden, weshalb einzelne Personen zu solch einem zentralen Punkt avanciert sind. Hierbei kann es durchaus auch schwarze Schafe geben, die sich bewusst in den Mittelpunkt rücken, beispielsweise durch gezieltes Zurückhalten von Informationen, um sich selbst unabkömmlich für die Gruppe zu machen.[10] Im Kontrast dazu wird oft die Arbeit derjenigen, die legitim eine zentrale Rolle innehaben, nicht gewürdigt. Dieses Problem kritisiert folgendes Zitat eines Mitarbeiters, der wesentlich in den Prozess der Wissensarbeit eingebunden ist:[1]

> *I spend about an hour and a half every day responding to calls and other informational requests. . . [and] . . . none of that time gets seen in my performance metrics.*

Damit durch diese Unzufriedenheit die Motivation und das Engagement der Mitarbeiter nicht negativ beeinflusst werden, wenden Unternehmen Belohnungssysteme gemäß der Anreiz-Beitrags-Theorie der strukturellen Personalführung an. Dabei werden gezielt Mitarbeiter, die besonderes Engagement bei der Wissenskreierung und Wissensteilung zeigen, belohnt. Solche Belohnungen als Würdigung des Arbeitseinsatzes können sein:[1]

- Monetäre Belohnungen für herausragende Leistungen, um den Informationsfluss weiter anzuregen, beispielsweise durch Sonderprämien.
- Kognitive und soziale Freiräume, die sowohl die individuelle, als auch kollektive Kreativität fördern. Möglichkeiten bieten sich in Form von Umgestaltung der Räumlichkeiten oder durch mehr Freiheiten während dem Arbeitsalltag.

– Beförderungen, um denjenigen, die Besonderes geleistet haben, die Möglichkeit zu geben, ihren Erfolg und ihre Bemühungen an andere weiter zu geben.

Nicht nur die zentralen Personen innerhalb des Netzwerkes können definiert werden, sondern auch diejenigen, die an der Kommunikation innerhalb des Unternehmens kaum, beziehungsweise gar nicht partizipieren. Dies kann mehrere Gründe haben:

– Die betroffenen Personen sind neu in der Unternehmung, und verfügen noch nicht über die notwendigen Kenntnisse und Beziehungen, beziehungsweise sind noch nicht richtig in das Firmenumfeld integriert. Hierbei kann jedoch seitens der jeweiligen Abteilungen durch einen optimierten Einarbeitungsprozess Unterstützung geleistet werden.
– Es handelt sich um Führungspositionen des Unternehmens, die im Laufe der Zeit vermehrt administrative Aufgaben wahrnehmen und dadurch eine untergeordnete Rolle bei Prozessen der Problemlösung tragen.
– Bei der Rekrutierung der jeweiligen Personen wurden ihnen mehr Kenntnisse zugetraut, als tatsächlich vorhanden sind. Hier bieten sich dann Weiterbildungen oder eine Neubesetzung der betroffenen Stelle an.

Der entscheidende Vorteil dieser kombinierten Sicht ist nun, dass zentrale und periphere Punkte gefunden, und durch die verschiedenen Sichten der Analyse erklärt werden können. Beispielsweise ist es ein Unterschied ob jemand eine zentrale Rolle aufgrund seines Wissens spielt, oder ob er nur durch seine räumliche Lage bevorzugt wird. So kann nach und nach der Informationsfluss verbessert werden, sowie die Performanz der Wissenskreierung und Wissensteilung qualitativ optimiert werden.

4.1 Weitere Anwendungen

Die bisherigen Betrachtungen der sozialen Netzwerkanalyse bezogen sich nur auf Organisationsstrukturen in Unternehmen. Die gemeinsame Komponente, die zu einer Verbindung führen konnte, war die Mitarbeit im selben Unternehmen. Ein weiterführender Schritt ist es, Gruppen von Personen zu analysieren, die sich durch andere Gemeinsamkeiten gebildet haben. Hierbei bieten sich seit der Entwicklung des *Web 2.0* diverse virtuelle Gemeinschaften an. Dabei kann man diese in zentrale und geschlossene Gruppen, wie beispielsweise *Facebook* oder *MySpace*, beziehungsweise verteilte und offene Gruppen, wie beispielsweise *FOAF*[1] oder *Blogs*, unterscheiden.[11]

Die Betrachtung von solch großen Gruppen stellt eine Herausfoderungen dar. Einerseits ist es schwierig, mehrere Millionen Knoten und Kanten effizient zu verarbeiten und die dabei entstandenen Ergebnisse aussagekräftig darzustellen, andererseits gibt es starke Einschränkungen durch datenschutzrechtliche Schwierigkeiten und Konsequenzen. Jedoch lassen sich aus den Ergebnissen sehr nützliche Informationen gewinnen. So können beispielsweise Unternehmen durch ge-

[1] Friend of a Friend (http://www.foaf-project.org)

zielte, personalisierte Werbemodelle ihr Portfolio optimieren oder etwa Wissenschaftler ein besseres Verständnis über Gruppenstrukturen und deren Dynamik gewinnen.[12]

Eine moderne Anwendung der sozialen Netzwerkanalyse zeigten Forscher der Universität Baltimore. Sie entwickelten 2004 eine Suchmaschine namens *Swoogle*[2] mit der es möglich ist, *RDF*[3]-Dokumente und in HTML-Dokumente eingebetteten *RDF*-Inhalt zu durchsuchen und zu indexieren. Basierend auf diesen Daten konnte über die *foaf:knows* Relation zweier *foaf:Person* Instanzen ein soziales Netz erstellt werden. Durch Anwendung der sozialen Netzwerkanalyse ließen sich danach Aussagen über einzelne Personen treffen, wie beispielsweise wer von vielen Personen gekannt wird, wer gerne bloggt oder wer in vielen Photos annotiert wurde.[11]

5 Fazit

Neben einer Einführung in die Problemfelder der täglichen Informationsbeschaffung innerhalb großer Unternehmen und der Definition der Kerncharakteristika effizienter Wissensbeziehungen wurden in Kapitel 2 die Grundlagen und Eigenschaften der **sozialen Netzwerkanalyse** dargestellt. Dabei wurden verschiedene Datenarten sowie Bewertungsmetriken herausgearbeitet, sowie an einem praktischen Beispiel die allgemeine Hervorgehensweise erörtert. Eine Analyse der vier Netzwerkdimensionen Wissen, Zugang, Engagement und Sicherheit folgte, sowie eine Darstellung potentieller Unterstützungsmöglichkeiten zur Verbesserung der jeweiligen Problemfelder. Die kombinierte Sicht der jeweiligen Analysen brachte dann in Kapitel 4 den wesentlichen Nutzen der angewandten Verfahren näher.

Zusammenfassend lässt sich sagen, dass mit Hilfe der sozialen Netzwerkanalyse hervorragend Schwachstellen innerhalb der Organisationsstruktur moderner Unternehmen aufgedeckt werden können. Die schrittweise Analyse und sukzessive Erweiterung der Sicht auf die jeweiligen Dimensionen verschafft einem dabei einen idealen Überblick über die gegebenen Verhältnisse und deren Engpässe. Das Handhaben von Wissen in Form von Informationen ist eine alltägliche Sache geworden, aber meist eingebettet in den normalen Arbeitsablauf ohne das sich die involvierten Personen dessen wirklich bewusst sind. Dadurch kommt es viel zu oft dazu, das so gut wie gar kein Aufwand betrieben wird, diese Prozesse systematisch zu analysieren, und zu optimieren. Ein interessanter Aspekt ist hierbei, dass es viele Firmen, die sich vermehrt mit modernen Methoden des Wissensmanagement auseinandersetzen, in die FORTUNE 500 geschafft haben, und somit zu den umsatzstärksten Unternehmen weltweit gehören. Zu verstehen wie Wissen innerhalb der Organisation fließt, oder auch nicht fließt, zahlt sich offensichtlich aus. Die Möglichkeit, mit Hilfe der sozialen Netzwerkanalyse, Interaktionen sichtbar und nachvollziehbar zu machen, bietet gerade diesen wissensmanagementbewussten Firmen die Chance, ihre vorhandenen Strukturen noch wirkungsvoller zu gestalten, indem an den richtigen Stellen der jeweils

[2] Swoogle Semantic Web Search Engine (http://swoogle.umbc.edu)
[3] Resource Description Framework (http://www.w3.org/RDF)

nötige Aufwand erbracht wird. Damit zeigt sich eindrucksvoll, wie durch Kombination einfacher Verfahren der Mathematik und der Soziologie herausragende Ergebnisse auf dem Gebiet des Wissensmanagement erzielt werden können.

> *„Eine Investition in Wissen bringt noch immer die besten Zinsen."*
> ***Benjamin Franklin***

Literatur

1. Cross, R., Parker, A., Prusak, L., Borgatti, S.: Knowing what we know: supporting knowledge creation and sharing in social networks. Organizational Dynamics **30**(2) (2001)
2. Scott, J.P.: Social Network Analysis: A Handbook. SAGE Publications (January 2000)
3. Hanneman, R.A., Riddle, M.: Introduction to social network methods. Digitally published (2005)
4. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)
5. Borgatti, S.P., Everett, M.G.: A graph-theoretic perspective on centrality. Social Networks **28** (October 2006)
6. Everett, M., Borgatti, S.P.: Ego network betweenness. Social Networks **27**(1) (January 2005)
7. Donoghue, L.P., Harris, J.G., Weitzman, B.E.: Knowledge management strategies that create value. Outlook Journal **1** (1999)
8. Borgatti, S.P., Cross, R.: A relational view of information seeking and learning in social networks. Manage. Sci. **49**(4) (2003)
9. Collison, C.: Greater than the sum of its parts: Knowledge management in british petroleum. Inside Knowledge (1997)
10. Cross, R., Prusak, L.: The people that make organizations stop — or go. Harvard Business Review **80** (2002)
11. Finin, T., Ding, L., Zhou, L., Joshi, A.: Social Networking on the Semantic Web. The Learning Organization **12**(5) (December 2005) 418–435
12. Dengel, A.: Lecture knowledge management (2008)

Personal Information Management – ein Überblick –

Volker Hudlet

Seminar im Bereich Wissensmanagement an der Technischen Universität Kaiserslautern

Zusammenfassung Im Bereich Personal Information Management (PIM) wird erforscht, wie Benutzer mit den sie umgebenden Informationen umgehen. Man unterscheidet Aktivitäten, die sie durchführen, um sich diese anzueignen, sie zu organisieren, zu verwalten und (wieder) aufzufinden. Diese Seminararbeit gibt einen Überblick über das Forschungsgebiet und die bisher erlangten Erkenntnisse. Außerdem werden mehrere Herausforderungen genannt, die Forschungspotenzial bieten, sowie Ansätze, die sich diesen Herausforderungen stellen und Teilbereiche lösen.

1 Einleitung

Millionen von Menschen kommen täglich damit in Berührung, oft ohne sich darüber überhaupt im Klaren zu sein. Die Rede ist von *Personal Information Management* (PIM). Während man landläufig darunter meist nur das Verwalten von Kontakten, E-Mails und ähnlichem sieht, ist das Gebiet in Wirklichkeit deutlich komplexer und umfasst weit mehr.

Idealerweise haben wir *die richtigen Informationen zur richtigen Zeit, am richtigen Ort, des Weiteren in der richtigen Form, ausreichenden Vollständigkeit und Qualität, um die momentan anstehende Aufgabe durchzuführen* [6]. Somit hätten wir mehr Zeit die Informationen zu verwenden, um Aufgaben effizient zu lösen, anstatt unnötig Zeit investieren zu müssen, um die Informationen zu beschaffen.

Das noch junge Forschungsgebiet PIM hat sich dem Ziel verschrieben, diese Vorgänge zu untersuchen. Bereits in den 1980er [11] und 1990er [1] Jahren wurden Studien durchgeführt, die aufzeigten, wie Benutzer mit den sie betreffenden Informationen umgehen und welche Strategien sie dabei benutzen. Außerdem erkannte man in den 1980er Jahren zunehmend das Potenzial des PCs, den Menschen beim Verarbeiten und Verwalten von Informationen zu unterstützen. Es entstanden erste PIM-Programme, die dem Benutzer grundlegende Hilfe beim Verwalten von Terminen oder Kontakten boten. 1988 wurde dann durch Lansdale [10] erstmals der Begriff Personal Information Management eingeführt.

Dennoch dauerte es weitere 15 Jahre bis 2004 von Bergman, Boardman, Gwizdka und Jones [3] der Aufruf zur Gründung einer PIM-Forschungsgemeinschaft erfolgte. Diese hatten auch schon in den Jahren zuvor Forschung auf diesem Gebiet betrieben. Ein Grund für die beschwerliche Etablierung ist, dass PIM

kein eigenständiger, trennscharfer Bereich ist, sondern dass Aspekte und Erkenntnisse aus den Bereichen Wissensmanagement, Information-Retrieval, Kognitionswissenschaft, Mensch-Computer-Interaktion, künstliche Intelligenz und Datenbankmanagement einfließen.

Im gleichen Jahr kam es daraufhin zu einer Special-Interest-Group-Tagung als Teil der Konferenz über Mensch-Computer-Interaktion (CHI). Im Folgejahr wurde erstmals ein eigenständiger Workshop zum Thema PIM veranstaltet. Auf dieser Konferenz wurde beschlossen, den resultierenden Report [8] weiter zu überarbeiten und als Sonderausgabe von *Communications of the ACM* [14] herauszugeben. Da die Zusammenarbeit innerhalb der Forschungsgemeinschaft positiv verlief und man sich auch persönlich kannte, wurde der Plan gefasst, die Sonderausgabe nochmals zu überarbeiten und zu erweitern, so dass 2007 ein umfassendes Buch [9] erschien, welches den Stand der Forschung und die bisherigen Erkenntnisse umfassend wiedergibt.

Der Rest der Seminararbeit ist wie folgt gegliedert: Zuerst werden grundlegende Begriffe des Personal Information Management erklärt und abgegrenzt. Anschließend werden die drei zentralen PIM-Aktivitäten – Behalten, Finden und Organisieren – sowie die bisherigen Erkenntnisse, die darauf Einfluß haben, vorgestellt. Da die PIM-Forschung noch lange nicht am Ende ist, werden anschließend Herausforderungen genannt, die noch zu meistern sind. Danach werden exemplarisch zwei Ansätze vorgestellt, die jeweils Teilbereiche dieser Herausforderungen lösen. Abschließend erfolgt ein Fazit.

2 Begriffe

Nachdem nun eine geschichtliche Einordnung von PIM erfolgt ist, sollen nachfolgend grundlegende Begriffe erläutern und definiert werden. Damit wird es letztendlich möglich sein, eine präzise Beschreibung für PIM festzulegen.

2.1 Personal Information

Als Grundfrage stellt sich bei persönlichen Informationen zuerst, was eine Information persönlich macht. Die primäre Bedeutung persönlicher Information umfasst alle Informationen, die eine Person besitzt (beispielsweise eine Kopie eines Rundschreibens oder eine E-Mail). Jones führt in [7] darüber hinaus fünf weitere Bedeutungen ein, wie Informationen persönlich sein können. Diese umfassen Informationen über eine Person, an eine Person gerichtete, von einer Person publizierte, durch eine Person wahrgenommene oder für eine Person relevante Informationen.

Wie man sehen kann, gibt es ein breites Spektrum an Bedeutungen der persönlichen Information. Im weiteren Verlauf verbleibt der Fokus auf dem primären Verständnis der Informationen, die eine Person besitzt.

2.2 Informationsobjekt

Im vorangegangen Abschnitt wurde deutlich, wie vielseitig persönliche Informationen sein können. Nun muss geklärt werden, wie Information beschaffen sein muss, um als *Informationsobjekt* für PIM in Betracht zu kommen.

Nach Jones [9, S.3-20] ist ein Informationsobjekt eine *persistente Kapselung von Informationen*. Geforderte Operationen, die man auf einem Informationsobjekt ausführen kann, umfassen: Erwerben, Erstellen, Betrachten, Lagern, Verschieben, Kopieren, Verteilen, Löschen des Informationsobjekts. Zusätzlich können Informationsobjekte einen Namen und andere Attribute erhalten, man kann sie zusammen mit anderen Informationsobjekten gruppieren und anderweitig verarbeiten. Durch diese Definition wird eine allgemeine und maschinelle Verarbeitbarkeit gewährleistet.

Neben computergestützten Informationsobjekten wie beispielsweise Bookmarks, Dateien und Ordnern werden durch die von Jones gegebene Definition auch papiergestützte Medien, z. B. Visitenkarten, Tageszeitungen und Dokumente als Informationsobjekte aufgefasst. Doch nicht alles, was Informationen enthält ist automatisch ein Informationsobjekt. So ist der Inhalt eines Abteilungsmeetings kein Informationsobjekt, ein Protokoll darüber oder ein Mitschnitt davon ist hingegen eines. Webseiten sind auch keine (originären) Informationsobjekte, da manche Bedingungen wie beispielsweise Verschieben oder Löschen nicht möglich sind. Jones kategorisiert Webseiten deswegen als (vom ihm so benannte) Semi-Objekte.

Jedes Informationsobjekt gehört darüberhinaus zu einem Informationstyp. Dieser ist durch die Werkzeuge und Anwendungen bestimmt, mit denen man das Informationsobjekt manipulieren kann. Typen sind u. a. Dokumente, Bookmarks oder E-Mails.

Wie man sehen kann, wird durch die Einführung des Begriffs Informationsobjekt eine *überschaubare Abstraktionsebene für die Berücksichtigung bei PIM* [7] festgelegt.

2.3 Personal Space of Information

Der Begriff Personal Space of Information (PSI) kombiniert die Überlegungen der beiden vorangegangenen Abschnitte. Ein PSI beinhaltet alle Informationsobjekte einer Person, die (mindestens) einer Bedeutung persönlicher Information zuordenbar sind. Außerdem gehören Anwendungen, Werkzeuge und Konzepte dazu, die der Person helfen, diese Informationsobjekte zu verwalten. Per Definition hat jede Person nur einen PSI, der allumfassend und einzigartig ist. Charakteristisch für den PSI ist des Weiteren, dass er *zu großen Teilen unbekannt ist, ungewisse Grenzen hat und dass eine erhebliche Überlappung mit den PSIs anderer Personen besteht* [9, S.3-20]. Das Problem liegt also darin, dass es für eine Person unmöglich ist, ihren kompletten PSI zu kontrollieren und zu verwalten.

2.4 Personal Information Collection

Aufgrund der eben genannten Eigenschaften ist es wünschenswert den PSI in kleinere, handlichere Teilbereiche zu unterteilen. Genau diesen Gedanken verfolgt die Idee der Personal Information Collection (PIC). PICs sind in sich abgeschlossene, persönlich verwaltete Teilmengen des PSI einer Person. Während Boardman [4] Informationsobjekte den Teilmengen jeweils anhand des Formats bzw. der Anwendung, die auf diese Informationsobjekte zugreift, zurechnet, abstrahiert Jones [7] von dieser Idee.

Anders als bei Boardman, bei dem immer Informationsobjekte gleichen (Informations-)Typs eine PIC bilden, können bei ihm auch Informationsobjekte unterschiedlichen Typs eine PIC bilden. Dadurch ist eine Gruppierung nach logischer Zusammengehörigkeit möglich, was den Vorteil bietet, dass beispielsweise projektbezogene Informationsobjekte (u. a. E-Mails und Dokumente) als eine PIC aufgefasst werden können. Daneben ist bei ihm auch weiterhin möglich, gleich typisierte Informationsobjekte zu einer PIC (beispielsweise zu einer Sammlung von Bookmarks) zusammenzufassen. Zu einer PIC gehört neben den enthaltenen Informationsobjekten eine geordnete Darstellung dieser, z. B. durch räumliche Anordnung, eine Ordnerhierarchie oder sonstige Attribute. Dadurch ist eine PIC insbesondere bezüglich der Art ihrer Organisation in sich abgeschlossen.

Während also der komplette PSI schwer beherrschbar ist, sind PICs abgegrenzt und organisiert, was sie als Grundlage für PIM qualifiziert.

2.5 Personal Information Management

Nach diesen Vorüberlegungen soll nun PIM definiert werden. Lansdale [10], der den Begriff zum ersten Mal gebrauchte und einführte, versteht darunter einen Überbegriff für das *Sammeln, Speichern, Organisieren und Wiederabrufen von (digitalen) Informationsobjekten durch eine Person in ihrem persönlichen Computerumfeld*. Seitdem wurde weitere Versuche unternommen, PIM zu definieren, die vom Grundverständnis mit denen Lansdales übereinstimmen.

Gemein haben die Definitionen, dass sie alle stark an die Grundprinzipien des allgemeinen Information Management – also das Speichern von Informationen, um sie zu einem späteren Zeitpunkt wieder abzurufen – angelehnt sind. Während allerdings bei diesem Informationen von Informationsfachleuten (z. B. Bibliothekaren) verwaltet werden und mehreren Benutzern zur Verfügung stehen, liegt die Bürde bei PIM auf dem Einzelnen, wie Bergman et al. [2] kritisch anmerken. Dies kann bei schlecht gestalteten Systemen schnell zu Frust und Unlust führen.

Jones [9, S.3-20] greift den Gedanken von Input, Speichern und Output auf und führt entsprechend drei grundlegende PIM-Aktivitäten ein, die dazu genutzt werden, um mit seinem PSI zu interagieren. Diese Aktivitäten bestehen aus Behalten, Organisieren und Finden und werden im folgenden Kapitel ausgiebig erläutert.

3 Aktivitäten bei Personal Information Management

Wie im vorigen Kapitel erwähnt, sind die drei PIM-Aktivitäten Behalten, Finden und Organisieren die essentiellen Werkzeuge, um PIM zu betreiben. Nach Jones [7] sind PIM-Aktivitäten *der Versuch, ein Mapping zwischen Informationen und dem Bedürfnis nach diesen festzulegen, zu benutzen und zu pflegen.* Aktivitäten des Behaltens bestimmen dabei den Input von Information in den PSI einer Person. Sie wandeln Information in ein (potentielles) Bedürfnis um, z. B. das Speichern der Telefonnummer des Lieblingshotels in Berlin. Dem gegenüber stehen Aktivitäten des Findens, bei denen man von einem Bedürfnis auf Informationen umschwenkt, um ein Ziel zu erfüllen, so z. B. wenn man dringend ein Hotelzimmer in Berlin braucht. Es ist der Output von Information aus dem PSI. Durch Organisationsaktivitäten wird der PSI geordnet und verwaltet und so eine Verbindung zwischen den beiden zuvorgenannten Aktivitäten hergestellt. Nachfolgend werden nun die spezifischen Eigenschaften der jeweiligen Aktivität besprochen.

3.1 Behalten

Am Anfang jeder Aktivität des Behaltens muss zuallererst eine Entscheidung gefällt werden: Erachtet man die vorliegende Information als relevant genug, um sie für ein erwartetes Bedürfnis zu behalten oder verwirft man sie. Dabei steht die Person vor einem Dilemma; neben der richtigen Entscheidung nützliche Information zu behalten bzw. unbrauchbare Information zu verwerfen, kann sie aufgrund von Fehleinschätzungen auch den Fehler begehen, unnütze Information zu behalten bzw. wertvolle Information zu verwerfen. Jones bringt dies in [6] auf den Punkt: *damned if you do; damned if you don't.* Während bei der richtigen Entscheidung Aufwand und Nutzen in einem Verhältnis stehen, ist bei der Fehleinschätzung der Aufwand (durch eventuelle Wiederbeschaffung bzw. unnötige Verwaltung der Information) wesentlich größer als der Nutzen.

Nach dieser initialen Entscheidung stehen im Falle des Behaltens weitere Entscheidungen an: Wie, wo und in welcher Form soll das Informationsobjekt behalten werden. Beispielsweise muss sich eine Person, die eine Visitenkarte überreicht bekam, überlegen, ob sie diese behält, und wenn ja, ob sie sie in einer Kartei hinterlegt oder die Informationen in eine Datenbank überträgt und die Karte selbst wegwirft (aber nicht muss).

Da dieser Entscheidungsbaum für jedes neu aufkommende Informationsobjekt durchwandert werden muss, fühlt sich manche Person überfordert. Um Abhilfe zu schaffen, wäre es denkbar, eine generelle (automatisierte) Strategie einzusetzen. Die beiden Extreme *alles behalten* und *nichts behalten* entlasten den Einzelnen, eignen sich aber nur bedingt, weil diese Probleme mit sich bringen: Wenn alles behalten wird, kann es zwar nicht mehr passieren, dass eine nützliche Information verworfen wird, aber dass sie durch die große Informationsmenge schlicht übersehen wird, wenn sie später gebraucht wird. Zu viel Information kann also genauso schlimm sein wie Fehlende. Nichts behalten könnte

für Teilbereiche denkbar sein, so z. B. für Informationsobjekte, die in einem Unternehmensintranet hinterlegt sind und darüber später wieder gefunden werden können. Nachteilig hierbei ist, dass das Informationsobjekt nicht unter der Kontrolle der Person ist und beispielsweise nach einer Weile nicht mehr verfügbar ist. Weitere Nachteile dieser automatisierten Ansätze werden auch im folgenden Abschnitt ersichtlich.

Was ist also der Ausweg aus der Misere? Jones legt in [9, S.35-56] die Idee des *keep smarter* nahe. Dabei soll der Benutzer bei der Entscheidungsfindung unterstützt werden, aber weiterhin die Entscheidung selbst treffen. Somit behält der Benutzer einen besseren Überblick über die Informationsobjekte, die er in seinen PSI aufnimmt, ohne bei der Entscheidung auf sich allein gestellt zu sein. Eine Möglichkeit dafür ist eine effektive Informationsorganisation, da diese wesentlichen Einfluss auf die Wahrnehmung von neuen Informationsobjekten hat. Nähere Aspekte hierzu werden in Abschnitt 3.3 erläutert.

3.2 Finden

Im Bereich *Information Retrieval* wurden in der Vergangenheit vielfältige Techniken und Ansätze zum Aufspüren und Finden von Informationen entwickelt. Diese lassen sich allerdings nicht ohne Weiteres auf PIM übertragen, da hier mehrere spezielle Aspekte eine Rolle spielen, wie nachfolgend gezeigt wird. Information Retrieval ist im Bereich PIM zwar nützlich, aber nicht ausreichend.

Zuerst muss eine Unterscheidung zwischen (erstmaligem) Finden und Wiederfinden getroffen werden. Wiederfinden setzt voraus, dass sich das gesuchte Informationsobjekt bereits in einer PIC des Benutzer befindet (z. B. die Nummer des Lieblingshotels in Berlin); der Benutzer hatte also schon mit dem Informationsobjekt zu tun (mindestens bei der Behalteentscheidung) und kann dieses Wissen nutzen. Hier wird auch ein Nachteil automatisierten Behaltens klar: Da die Person nie mit dem Informationobjekt in Berührung kam, fehlt für das spätere Finden dieses Wissen. Erstmaliges Finden kann hingegen durch klassisches Information Retrieval abgedeckt werden (z. B. Suchanfrage „Hotel Berlin“). Aus diesem Grund wird im weiteren Verlauf nur Wiederfinden betrachtet.

Barreau und Nardi [1], sowie Boardman und Sasse [5] fanden in ihren Studien heraus, dass Personen *ortsbasiertes Finden* gegenüber *logischem Finden* bevorzugen. Bei Ersterem nutzt die Person kontextuelles Wissen, um den möglichen Ort (z. B. einen Ordner) auszuwählen und diesen dann zu durchsuchen. Dies wird iterativ so lange durchgeführt, bis die Person das gesuchte Informationsobjekt gefunden hat. Bei Letzterem wird eine textbasierte Suche mit Schlüsselwörtern oder anderen Metainformationen durchgeführt, um das Informationsobjekt zu finden. Auch dieser Ansatz muss möglicherweise iterativ durchgeführt werden.

Jedes Finden ist ein Zusammenspiel zwischen des sich Entsinnens und der Wiedererkennung des gewünschten Informationsobjekts. Personen können sich meist noch an ungefähre Angaben (eben beispielsweise den Ort) entsinnen und das Informationsobjekt durch Durchsuchen wiedererkennen. Detaillierte Angaben, wie beispielsweise vergebene Schlüsselwörter für ein Informationsobjekt,

geraten leichter in Vergessenheit. Logisches Finden wird meist nur als letzter Ausweg benutzt, wenn die Person sich nicht mehr an den Ort erinnern kann.

Weiterhin vermittelt ortsbasiertes Finden dem Benutzer ein größeres Gefühl von Kontrolle und bietet gleichzeitig einen Überblick über die durchsuchte PIC. Hier gibt es eine Schnittmenge mit einer Eigenschaft, die bereits Malone [11] in seiner Studie herausfand und als genauso wichtig wie Finden einstufte: *Erinnern.*

Es wurde festgestellt, dass sich viele Personen Informationsobjekte anlegen, die sie beim Sichten der PIC an eine ausstehende Aufgabe erinnern sollen, beispielsweise durch eine an sich selbst gerichtete E-Mail. Ein Nachteil der logischen Suche ist also, dass diese Erinnerungsfunktion verloren geht.

Durch die Präferenz für ortsbasiertes Suchen ist letztlich eine gute und effektive Organisation des PSI und der darin enthaltenen PICs unumgänglich.

3.3 Organisieren

Wie aus den beiden vorangegangen Abschnitten ersichtlich wurde, ist eine effektive Organisation unabdingbar, um die Entscheidungsfindung beim Behalten zu vereinfachen und das Finden zu erleichtern.

Jones versteht *darunter alle Entscheidungen und Vorgänge zur Wahl und Umsetzung eines Repräsentations- und Organisationsschemas für eine PIC.* Im weiteren Sinne gehört auch die nachfolgende Pflege dieses Schemas dazu. Abstrahiert davon ist das übergeordnete Ziel den verwalteten Informationobjekten einen Sinn zu geben, um später als Nutzer bei anderen Aktivitäten von diesem zu profitieren.

Während Aktivitäten des Behaltens und Findens sehr häufig durchgeführt werden, wird deutlich weniger Aufwand in Organisation und Pflege gesteckt. Eine Beurteilung erfolgt nur sporadisch, dann meist in Form eines „Frühjahrsputzes“. Eine grundlegende Problematik liegt darin, dass Personen dazu neigen, ihre Organisation *bottom-up* und *ad hoc* durchzuführen. Wenn beispielsweise ein Ordner oder der Posteingang vollgestopft sind, wird ein neuer Ordner angelegt und Teile dahin verschoben. Diese Neuorganisation geschieht oftmals unreflektiert, was sich oft in nicht aussagekräftigen Namen widerspiegelt. Jones konnte in einer Studie einen Probanden beobachten, der die Ordner „stuff“, „more stuff“ und „still more stuff“ besaß.

Dies lässt sich allerdings nicht verallgemeinern, da jede Person andere Präferenzen und Ansätze hat, ihre PICs zu organisieren. Den Klassiker zur Unterscheidung von verschiedenen Organisationsformen (bei Papierdokumenten) stellte Malone [11] mit der Unterteilung zwischen chaotischer und ordentlicher Organisation vor. Erstere ist durch diverse unklar zusammengestellte Stapel charakterisiert, währende Letztere durch Aktenbildung und klar definierte Stapel glänzt. In Abschnitt 4.1 werden die Unterschiede zwischen Personen näher beleuchtet.

Boardman und Sasse [5], die eine Tool-übergreifende Studie für Dateien, E-Mails und Bookmarks durchführten, stellten darüber hinaus fest, dass selbst bei der gleichen Person die Organisationsstruktur bei den verschiedenen PICs vari-

iert. Die meisten Übereinstimmungen von Ordnern gab es bezüglich der Rollen und Projekte der Person.

Eine Idee, um eine effektive Organisation zu erreichen, wäre es, gute und erfolgreiche Organisationsstrukturen wiederzuverwenden. Durch eine solche „Blaupause“ könnte man beispielsweise für verwandte Projekte eine initiale Struktur anlegen, die eine effektive Organisation mit sich bringt und gewährleistet.

Ein anderer Ansatz ist, die ad-hoc-Organisation durch eine top-down-Organisation zu ersetzen. Dies kann durch Einführung einer Taxonomie, eines Klassifikationschemas, geschehen. Dadurch erreicht man eine einheitliche, Tool-übergreifende Struktur. Jones [6] nennt dies *Personal Unifying Taxonomy*, da diese für eine Person einzigartig und auf seine/ihre Bedürfnisse angepasst ist. Sauermann et al. nennen dies ein Persönliches Informations-Modell (PIMO) [13].

Das übergeordnete Ziel bleibt weiterhin, noch besser zu verstehen, wie Personen ihre PICs strukturieren, um diese dabei zu unterstützen.

4 Herausforderungen

Trotz bisheriger Forschung, die die Eigenschaften und Besonderheiten von PIM ergründet hat, gibt es weiterhin großen Bedarf PIM zu verbessern und noch persönlicher zu machen. Neben den im vorigen Kapitel erläuterten PIM-Aktivitäten gibt es weitere Aspekte, die für die PIM-Forschung eine Herausforderung darstellen. Einige davon werden nachfolgend verdeutlicht.

4.1 PIM-Persönlichkeiten

Die Durchführung von PIM-Aktivitäten wird stets durch das Vier-Tupel Mensch, Aufgabe, Werkzeuge und Kontext beeinflusst. Das Verhalten einer Person wird also stets durch den Aufgabenbezug, die ihr zur Verfügung stehenden Werkzeuge und den Kontext, in der sich die Person befindet, sowie durch sich selbst bzw. ihre kognitiven Fähigkeiten geprägt. Jede Person hat also eine „PIM-Persönlichkeit“ [9, S.217], die ihre PIM-Aktivitäten bestimmt. Je besser die Einflussfaktoren zueinander passen, desto effektiver sind die PIM-Aktivitäten.

Gwizdka und Chignell geben in [9, S.206-220] einen Überblick über verschiedene Studien zu individuellen Unterschieden von Personen im Umgang mit PIM. Sie kommen zu dem Schluss, dass Personen von der generellen Strategie zwar in Gruppen unterschieden werden können (z. B. filing vs. piling [11]), aber jede Person diese Strategie oftmals variiert (z. B. abhängig von der spezifischen PIC [5]). Jede Person ist dabei auch unbewusst von innerlichen Fähigkeiten (z. B. Erfahrung, Arbeitsgedächtnis) beeinflusst.

Um den Einzelnen bei der Durchführung seiner PIM-Aktivitäten zu unterstützen, bleibt es eine Herausforderung, Werkzeuge zur Verfügung stellen, die ihn möglichst optimal unterstützen, indem sie sich an seine PIM-Persönlichkeit anpassen. Dies erfordert ein benutzerzentriertes Design der Werkzeuge, die Möglichkeiten zur Anpassung bieten und/oder adaptiv den Kontext berücksichtigen und sich dementsprechend verhalten.

4.2 Informationsfragmentierung

Informationsobjekte sind über verschiedene PICs verteilt, haben verschiedene Typen (z. B. E-Mail oder Datei) und Formate und sind auch oft über verschiedene Orte verstreut (z. B. Arbeitsplatz, Wohnung, PDA oder Handy). Diese Informationsfragmentierung erschwert und *behindert die effiziente Durchführung von PIM-Aktivitäten.* Es kann leichter vorkommen, das Informationsobjekte falsch abgelegt werden und dadurch übersehen oder vergessen werden können.

Meist muss eine Person mehrere Organisationsschemata für verschiedene Typen und Anwendungen pflegen, die oftmals schwer vergleichbar und unkonsistent zueinander sind. Zwar gibt es bereits PIM-Werkzeuge; diese sind aber häufig auf einen speziellen Aspekt ausgelegt und erhöhen durch Einführung eigener Organisationsschemata die Informationsfragmentierung zusätzlich. Das manuelle Pflegen der Schemata überfordert viele Personen und führt zu Frust.

Wie man sehen kann, ist Informationsfragmentierung und deren Überwindung eine große Herausforderung für die PIM-Forschung. Boardman und Sasse [5] stellten fest, dass gerade bezüglich Rollen und Projekten der Benutzer große Überlappung zwischen den verschiedenen PICs bestand. Sie empfehlen zur Integration zwei verschiedene Ansätze: Werkzeuge, die verschiedene Informationstypen unterstützen oder eine vereinheitlichte Sicht auf alle Informationsobjekte, die die Werkzeuge benutzen können.

5 Ansätze

Nachdem nun klar ist, welchen Herausforderungen sich die PIM-Forschung unter anderem stellen muss, werden nachfolgend aus der Fülle von verfügbaren Ansätzen beispielhaft zwei Ansätze vorgestellt, die versuchen, dies zu erfüllen.

5.1 Semantic Desktop und PIMO

Der Ansatz des Semantic Desktops ist eine *semantische Schicht, die als Middleware dient, um Anwendungen und ihre Daten zu integrieren* [12]. Er stellt sich damit der Herausforderung der Informationsfragmentierung als ein Framework, dass die Anwendungsgrenzen hinter sich lässt und dem Benutzer eine vereinheitlichte Repräsentation seiner Informationsobjekte zur Verfügung stellt.

Ziel ist es, einen Knowledge Space bereitzustellen, der *unabhängig davon ist, in welcher Weise der Benutzer auf die darin enthaltenen Daten zugreift und auch unabhängig von Quelle, Format und Autor der Daten ist* [12]. Erreicht wird dies dadurch, dass Informationsobjekte durch den W3C-Standard Ressource Descriptor Framework (RDF) beschrieben werden. Dadurch erhalten die Daten über die Informationsobjekte eine einheitliche Repräsentation und können über einen Uniform Resource Identifier (URI) identifiziert werden. Ein Knowledge Space stellt eine Wissensdomäne in Form einer Menge von Konzepten dar. Beim Semantic Desktop wird diese Aufgabe durch das Personal Information Model (PIMO) übernommen, welches es erlaubt, eine persönliche, subjektive Sicht

auf diese einheitliche Repräsentation zu ermöglichen. Das PIMO einer Person ist eine *formale Repräsentation der Strukturen und Konzepte in Anlehnung an sein mentales Modell* [13]. Da es sich um eine applikationunabhängige Repräsentation handelt, sind die Konzepte, um Informationsobjekte zu kategorisieren auch applikationsübergreifend sicht- und benutzbar. Ein weiterer Vorteil besteht darin, dass jede Applikation mit dem PIMO integriert wird und nicht jede Applikation mit jeder anderen, was den Umsetzungsaufwand deutlich verringert.

Der Ansatz realisiert in groben Zügen Jones' Idee der Personal Unifying Taxonomie (vgl. Abschnitt 3.3), ist aber wesentlich mächtiger, da statt Taxonomien Ontologien zum Einsatz kommen.

Das PIMO ist mehrschichtig aufgebaut und besteht aus den Ontologien PIMO-Basic, PIMO-Upper, PIMO-Mid und PIMO-User sowie domänenspezifischen Ontologien. Diese Einteilung erfolgte aus der Motivation heraus, dass das persönliche mentale Modell zwar nicht vorgeschrieben werden kann, man dem Benutzer den Einstieg durch vorgegebene (Grund-)Strukturen aber erleichtert. PIMO-Basic und PIMO-Upper stellen die grundlegenden Sprachkonstrukte bzw. die abstrakten und domänenunabhängigen Konzepte dar. Darauf aufbauend findet man die domänenspezifischen Ontologien, die konkrete Interessensgebiete des Benutzers beschreiben, sowie PIMO-Mid, die dazu dient, die einzelnen domänenspezifischen Ontologien zu integrieren. Mithilfe von PIMO-User kann der Benutzer letztlich die Vorgaben so anpassen und erweitern, dass sie seinem mentalen Modell am ehesten entsprechen. Ein Vorteil des gemeinsamen Überbaus besteht darin, dass es so möglich ist, für eine Gruppe gemeinsame Konzepte einzuführen und so den Informationsaustausch untereinander zu erleichtern.

Die Umsetzung und Implementierung von Semantic Desktop und PIMO erfolgt im Rahmen des Projektes NEPOMUK[1].

Wie man sehen kann, stellt sich der Ansatz mehreren Herausforderungen und bietet eine solide Grundlage für PIM, insbesondere für Aktivitäten des Organisierens, erleichtert aber damit auch Behalten und Finden wie in Abschnitt 3.3 gezeigt.

5.2 User-subjective Approach

Wie bereits in Abschnitt 2.5 erwähnt, zeichnet sich PIM gegenüber allgemeinem Information Management dadurch aus, dass der Einzelne bei allen Aktivitäten auf sich allein gestellt ist und selbst Entscheidungen bezüglich Organisation, Behalten und Finden treffen muss. Bergman et al. [2] versuchen mit dem User-subjective Approach diesen Nachteil in einen Vorteil umkehren, indem sie sich die benutzerabhängigen, subjektiven Attribute zu Nutze machen, die bei der Interaktion mit Informationsobjekten entstehen.

Allgemein sind Attribute eines Informationsobjekts Variablen, die das Objekt beschreiben, also Metadaten. *Objektive* Attribute sind benutzerunabhängig und ohne Wissen über den konkrete Benutzer aus dem Informationsobjekt ableitbar,

[1] `http://nepomuk.semanticdesktop.org/`

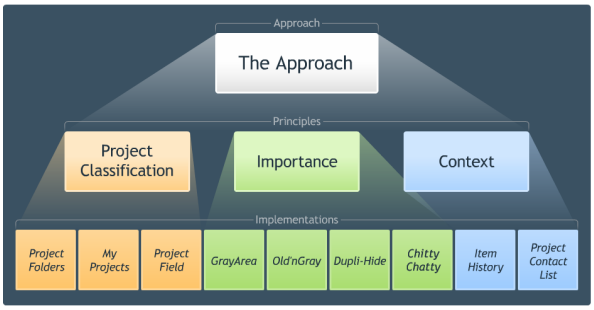

Abbildung 1. Übersicht über den User-subjective Approach [2]
(mit freundlicher Genehmigung des Autors)

wie beispielsweise das Format oder die Größe des Informationsobjekts. *Subjektive* Attribute sind dagegen an die konkrete Person gebunden. Deswegen kann man hier das Informationsobjekt nicht zu Hilfe nehmen, sondern muss die Attribute aus den Interaktionen der Person ableiten. Das PIM-Werkzeug sollte diese Attribute automatisch erfassen oder dies über *direct manipulation* ermöglichen.

Bergman et al. fokussieren bei ihrem Ansatz, wie in Abbildung 1 ersichtlich, auf drei Prinzipien, die die von ihnen identifizierten subjektiven Attribute – *Projektbezug*, *Wichtigkeit* und *Kontext* – berücksichtigen. Das Prinzip der Projektzuordung besagt, dass alle Informationsobjekte, die sich unabhängig von Typ und Format auf das gleiche Projekt beziehen unter einer gemeinsamen Kategorie klassifiziert werden. Das Prinzip der subjektiven Wichtigkeit bestimmt das visuelle Verhalten und die Zugänglichkeit eines Informationsobjekts abhängig von dessen Wichtigkeit. Weniger wichtige Objekte werden visuell zurückgestuft, um den Benutzer nicht abzulenken. Das Prinzip des subjektiven Kontexts legt fest, dass Informationsobjekte im gleichen Kontext erscheinen wie beim letzten Zugriff.

Für jedes dieser Prizipien gibt es mehrere Ideen der Umsetzung und Implementierung. Beispielhaft soll hier mit *GrayArea* eine Umsetzung des Prinzips der subjektiven Wichtigkeit aufgezeigt werden. GrayArea erweitert Ordner um einen grauen Bereich am unteren Ende, der weniger wichtige Informationsobjekte beherbergt. Die Objekte sind weiterhin verfügbar, werden aber kleiner dargestellt und stören so nicht die Sicht des Benutzers. Eine Implementierung davon existiert allerdings noch nicht.

Insgesamt stellen Bergman et al. mit dem User-subjective Approach ein Modell und Ideen zur Verfügung, die es erleichtern, benutzerbezogene PIM-Werkzeuge zu entwickeln. Der Ansatz stellt sich also der Herausforderung der PIM-Persönlichkeiten.

6 Fazit

Es wurde gezeigt, dass PIM hauptsächlich durch die drei Aktivitäten Behalten, Finden und Organisiere bestimmt ist. Es wurden bereits viele Studien zu die-

sen verschiedenen PIM-Aktivitäten durchgeführt und damit ihre Eigenschaften ergründet. Darüberhinaus wurden bereits mehrere Herausforderungen (u. a. Informationsfragmentierung) identifiziert, die noch zu lösen sind. Mit dem Semantic Desktop und dem User-subjective Approach wurden in dieser Seminararbeit Ansätze präsentiert, die den Benutzer bei seinen PIM-Aktivitäten unterstützen sollen und sich gleichzeitig diesen Herausforderungen stellen.

Insgesamt ist und bleibt PIM ein spannendes und herausforderndes Forschungsfeld. Trotzdem bleibt das anfangs erwähnte Ideal der richtigen Information zur richtigen Zeit am richtigen Ort auch weiterhin eine schwer erreichbare Vision.

Literatur

1. Deborah Barreau and Bonnie A. Nardi. Finding and Reminding: File Organization from the Desktop. *SIGCHI Bull.*, 27(3):39–43, 1995.
2. Ofer Bergman, Ruth Beyth-Marom, Rafi Nachmias, and Steve Whittaker. The User-Subjective Approach: A New Direction for PIM Systems Design. In Jaime Teevan and William Jones, editors, *Proceedings of the Personal Information Management Workshop at the CHI 2008*, 2008.
3. Ofer Bergman, Richard P. Boardman, Jacek Gwizdka, and William Jones. Personal Information Management. In *CHI 2004 Extended Abstracts on Human Factors in Computing Systems*, pages 1598–1599, 2004.
4. Richard Boardman. *Improving Tool Support for Personal Information Management.* PhD thesis, Imperial College, London, September 2004.
5. Richard Boardman and M Angela Sasse. "Stuff Goes into the Computer and Doesn't Come Out": A Cross-tool Study of Personal Information Management. In *Proceedings of the 2004 Conference on Human Factors in Computing Systems, CHI 2004*, pages 583–590, 2004.
6. William Jones. Finders, keepers? The present and future perfect in support of personal information management. *First Monday*, 9(3), 2004.
7. William Jones. *Keeping Found Things Found: The Study and Practice of Personal Information Management.* Academic Press, December 2007.
8. William Jones and Harry Bruce. A Report on the NSF-Sponsored Workshop on Personal Information Management. Technical report, NSF, 2005.
9. William P. Jones and Jaime Teevan. *Personal Information Management.* University of Washington Press, October 2007.
10. M. Lansdale. The psychology of personal information management. *Applied Ergonomics*, 19(1):55–66, March 1988.
11. Thomas W. Malone. How Do People Organize Their Desks? Implications for the Design of Office Information Systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, 1983.
12. Leo Sauermann, Gunnar Grimnes, and Thomas Roth-Berghofer. The Semantic Desktop as a foundation for PIM research. In Jaime Teevan and William Jones, editors, *Proceedings of the Personal Information Management Workshop at the CHI 2008*, 2008.
13. Leo Sauermann, Ludger van Elst, and Andreas Dengel. PIMO - a Framework for Representing Personal Information Models. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics' 07*, pages pp. 270–277. JUCS, 2007.
14. J. Teevan, W. Jones, and B. Benderson, editors. *Communications of the ACM: A Special Issue on Personal Information Management*, volume 49. ACM Press, 2006.

An Overview on Ontology Learning from Web Documents

Jörn Hees

TU Kaiserslautern

Abstract This seminar paper gives an overview of the field of ontology learning from Web documents. Ontologies provide an explicit semantic which is the basis for machine reasoning techniques. Nevertheless, machine processible ontologies are rare in comparison to the documents available through the WWW. Hence it would be desirable to learn ontologies from Web documents. After a short recap of ontologies and the ontology learning layer cake, the main ontology learning paradigms are presented. Some of them work on plaintext, others on semi-structured documents such as HTML documents. The methods presented include lexico-syntactic patterns, Harris' distributional hypothesis, term subsumption, table head extraction, group by markup context and XHTML tree mining.

1 Introduction

Today's World Wide Web is a huge decentralized collection of information, but its documents are designed for human readers in nearly all cases and thereby difficult to interpret for machines. During the last two decades and in parallel to the astonishing growth of the WWW, many different approaches have been developed to search for information in the Web. Nevertheless, machines are still not able to interpret the content of the documents they are working with. A user of a search engine retrieves a list of Web pages, which contain the specified words and are sorted by some metric, in most cases. Nevertheless, the human reader has to search for the actual information in the result list.

In 2001 Berners-Lee, Hendler and Lassila proposed what they called the Semantic Web [1]. The Semantic Web is an extension of the WWW, so that Webpages not only fulfill a human reader's need for layout and navigation but also provide well-defined information for machine manipulation in form of ontologies. In their vision these extensions would allow user searches to be answered with real information instead of a list of Web pages that might contain the desired information, e.g., a searched phone number.

In spite of ongoing research, the spread of the Semantic Web is still negligible in comparison to the WWW. A possible reason for this is that machine processible ontologies are rare because most authors do not know or recognize their use when creating a simple Web page. Hence it would be desirable to learn ontologies from Web documents in order to support the propagation of the Semantic Web.

The next section will give a short description of ontologies. The most important ontology learning fundamentals are addressed in Sect. 3. Afterwards several ontology learning approaches for plaintext and semi-structured documents are described in Sect. 4 and 5. Section 6 briefly presents methods for ontology population and finally Sect. 7 concludes this paper.

2 Ontologies – an Overview

The shortest and most often cited definition of an ontology is the following one by Gruber:

> "An ontology is an explicit specification of a conceptualization." [2]

Even though this is the shortest definition, it might not be the most descriptive one.[1] When Berners-Lee, Hendler and Lassila made up the idea of the Semantic Web, they also included a short and very comprehensible definition:

> "An ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules." [1]

An ontology represents the knowledge of a specific and always limited domain of discourse in an explicit (namely written down) and formal way. It formally defines terms and all kinds of relations among them, so that all disputants can agree on and talk about them. Such relations especially include taxonomic relations, although they are not limited to them, as explained in the following. An ontology may furthermore contain inference rules allowing more advanced reasoning (see Sect. 3 for examples).

2.1 Taxonomic Relations

Alltogether taxonomies group things (terms) to build hierarchical, tree-like structures. Two common taxonomic relations are:

Hypernym & Hyponym (is-a) The hyponym is a sub-class of the hypernym, e.g., "golden retriever" is a hyponym of the hypernym "dog". As this relation can be expressed by a phrase like "a hyponym is-a hypernym", e.g., "a golden retriever is a dog", it is also called *is-a* relation.

Holonym & Meronym (is-part-of) The meronym is part of the holonym, e.g., "paw" is a meronym of the hypernym "dog". This relation is also called *is-part-of* relation as it can be expressed by a phrase like "meronym is-part-of holonym", e.g., "a paw is-part-of a dog"

[1] Another verbose and recent definition by Gruber can be found in [3].

2.2 Other Relations

Besides taxonomic relations an ontology may contain any other relation, as shown in WordNet[2]. WordNet was manually created and lists a word's hypernym(s), hyponyms, holonym(s) and meronyms. Furthermore WordNet makes heavy use of synonyms grouped into synonym-sets:

Synonym & Antonym Synonyms are semantically equivalent words, e.g., the words "dog" and "canine". Antonyms are semantically contrary words, e.g., "dark" and "bright".

Other frequently addressed relations include semantic siblings, translations, homonyms and qualia structures:

Semantic Sibling Two terms sharing the same hypernym are also called *co-hyponyms*, e.g., "golden retriever" and "dalmatian".

Translation Dictionaries consist of these relations, e.g., "dog" and "Hund" (German for dog).

Homonym A word which has different meanings depending on the context, e.g., "bank" (park vs. finance) or "board" (chalk vs. conference).

Qualia Structures Qualia structures[3] describe the meaning of a term by these four roles (examples below for "dog"):

Formal Describes properties distinguishing the object ("animal", "hairy").

Constitutive Describes physical properties of the object ("fur", "paw").

Telic Describes the purpose or function of the object ("bark", "play").

Agentive Describes how the object is generated ("bear", "give birth").

3 Ontology Learning

After the recap of ontologies in the previous section, this section will introduce some ontology learning fundamentals shared by nearly all ontology learning methods.

Ontology learning is a process which tries to extract knowledge in form of various relations between concepts and instances. While the number of Web documents is growing, the availability of explicit knowledge is still limited. Ontology learning methods try to close this gap.

The ontology learning layer cake shown in Fig. 1 divides the ontology learning process into 6 layers.

The first layer tries to identify the existing *terms* in the examined documents. While whitespaces can help to find word boundaries[4] and identify terms like "dog", many terms consist of multiple words, such as "golden retriever". To identify such multi-word terms correctly and furthermore to identify the kind

[2] `http://wordnet.princeton.edu/`

[3] [4]

[4] Notice that this is only true in most western languages, while it does not hold for Finnish, Japanese, etc.

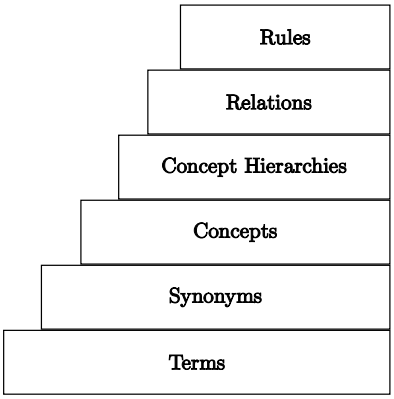

Figure 1. Ontology Learning Layer Cake [5]

of term (e.g., a noun, adjective, etc.), natural language processing (NLP) and part-of-speech tagging (POS-tagging) techniques are frequently used.

After the identification of terms, the ontology learning methods try to group *synonymous* terms together, e.g., "dog" and "canine", in the second layer.

The third layer consists of finding *concepts* for the identified synonym groups. Such a concept consists of an intension, an extension and a lexicon. The intension is a semantic definition of the concept. The lexicon lists direct instances of this concept. The extension includes all elements belonging to the concept, which includes terms listed by the lexicon as well as other inferred relations. For example the intension for the concept DOG[5] would be a description like "a member of the genus Canis [...] domesticated by man [...] occurs in many breeds"[6]. The lexicon could include the terms "dog" and "canine" and the extension could include the terms "dog", "canine", "golden retriever", "collie" and "Lassie".

Afterwards the ontology learning methods try to build *concept hierarchies* by identifying taxonomic relations in the fourth step, e.g., GOLDEN RETRIEVER is-a DOG.

Other relations are extracted in the fifth step, such as DOG does-not-like CAT.

The sixth step aims at finding inference rules, such as:

$$\forall x, \exists y.\ \text{is-a}(x, \text{DOG}) \wedge \text{is-a}(y, \text{HUMAN}) \rightarrow \text{belongs-to}(x, y)$$

As the sixth step is computationally cost-intensive and relies upon all previous layers, none of the presented methods implements it.

After presenting the different layers of ontology learning approaches, several such methods shall be presented in the next sections. Nearly all of them tackle the term acquisition in the same way as described above, but they differ a lot in the layers above. None of the presented approaches implements all layers.

[5] Concepts are written in small caps to distinguish them from instances.

[6] `http://wordnet.princeton.edu/perl/webwn?s=dog`

Nowadays a vast number of different ontology learning approaches exists. They can roughly be categorized depending on the type of information they rely upon. The next section will present ontology learning methods which are applicable to text in general. They treat Web documents as plaintext documents and make use of purely textual information only. Afterwards approaches are presented which exploit HTML-tags included in Web documents. Section 6 then briefly presents ontology population, which is a subclass of ontology learning.

4 Ontology Learning from Text

Several ontology learning approaches exist which were originally designed to work on plaintext documents from limited text corpora and were adapted to use the Web as a huge text corpus. These approaches are applicable to documents which can be converted into a plaintext form.

In the next subsections three ontology learning paradigms of this kind are presented. For further readings see [6].

4.1 Lexico-Syntactic Patterns

A frequently used strategy to extract embedded relations from natural language texts is based on the use of language style patterns. Such patterns are called lexico-syntactic patterns, sometimes also referred to as Hearst-patterns [7].

Lexico-syntactic patterns make use of NLP techniques to define patterns as these:

(1) NP_0 such as $\{\mathrm{NP}_1,\ \mathrm{NP}_2, \ldots (\text{ and}|\text{ or})\}\ \mathrm{NP}_n$
(2) $\mathrm{NP}_1\{,\ \mathrm{NP}_2,\ \mathrm{NP}_3, \ldots\}$ and other NP_0

Such patterns match phrases as in the following two examples:

(1) ... *dog breeds* **such as** *golden retriever*, *boxer* **and** *dalmatian* ...
(2) ... *golden retriever*, *boxer*, *dalmatian* **and** ~~many~~ **other** ~~popular~~ *dog breeds* ...

To a reader such expressions imply that NP_0 is a hypernym of NP_i, e.g., that the *hypernym*(*dog breed*, *golden retriever*) relation holds. The second example also illustrates the importance of NLP and POS-tagging to discard undesired words and flections.

Even though Hearst-patterns are intriguing by their simplicity and low computational costs, they traditionally suffer from their low coverage. Patterns which reliably indicate the relation of interest occur rarely while frequent patterns are not reliable enough. Fortunately this disadvantage is negligible when applying lexico-syntactic patterns to huge document collections as the Web.

Besides, Hearst presented more patterns for hypernym-/hyponym-detection as well as a method for finding such patterns, which can increase the overall coverage [7]. He also postulated that many other lexical relations would be acquirable in the same way.

An example for this is shown by Cimiano and Wenderoth. They use lexico-syntactic patterns to learn qualia structures and present patterns for the formal, constitutive and telic role, but detect that the agentive role cannot be described reliably by hearst-patterns [4].

Pasça shows another application of lexico-syntactic patterns [8], where he uses Hearst-patterns in combination with POS-tagging to extract hypernym relations from Web documents. He also uses the extracted relations to identify more Hearst-patterns and iterates this process, as described by Hearst, to increase the coverage. Furthermore he groups the extracted relations by their hypernyms and then infers the strength of relation between two hypernyms by the amounts of hyponyms they share.

Thereby lexico-syntactic patterns are one of the easiest and most frequently used ontology learning approaches.

4.2 Harris' Distributional Hypothesis

Another ontology learning approach is based on Harris' distributional hypothesis, which basically states that terms occurring in the same contexts have a similar meaning [9]. This hypothesis allows to investigate typical contexts, also called *second order associations* or *second order co-occurrences*, of terms with statistical methods and to apply clustering techniques on such syntactic contexts. See [10] for a comprehensive introduction.

For example let A be a term of interest. Then it is possible to determine a set of contextual terms $B_1, \ldots, B_n$ which show most significant co-occurrences[7] with A. A search for terms with a similar context $B_1, \ldots, B_n$ then reveals the words $C_1, \ldots, C_m$ which have a high probability for being synonyms to A.

Another application for such clustering techniques can be found in the identification of homonyms. Such terms are characterized by their membership in two or more clusters of synonyms which are only connected by a single term, the homonym.

4.3 Term Subsumption

In 1999 Sanderson and Croft presented another now widely used approach [12], which focuses on exploiting statistical conspicuousnesses. They formulate that a term A *subsumes* a term B if term B almost always occurs together with term A while one can observe that A only rarely occurs together with B but with many other terms. Hence, the subsumption can be formulated like this: B implies A while A does not imply B, or expressed with conditional probabilities: $P(A|B)$ is high while $P(B|A)$ is low.

In an example which again focuses on the domain of animals and dog breeds, the sub-concept *golden retriever* will almost certainly occur in combination with the super-concept term *dog* while on the other side the term *dog* will most

[7] For an extensive collection of statistical association measures see [11].

probably also occur in contexts without *golden retriever* and instead other breeds such as *boxer*, *German shepherd*, *dalmatian*, etc.

From the statistics of the occurrences and co-occurrences of two terms it is therefore concludable that one term is hierarchically above the other term, but for example it is not possible to distinguish between *is-a* and *is-part-of* relations with this approach. Furthermore the results from the statistical process depend on how the underlying co-occurrence of two terms is defined, e.g., if sentence or document wide co-occurrences are used, and whether only significantly strong co-occurrences are used.

5 Ontology Learning from Semi-Structured Documents

After the plaintext ontology learning paradigms presented in the last chapter this chapter focuses on ontology learning methods which also make use of (semi-)structured data. While highly-structured information is relatively rare on the Web, the vast majority of Web documents consists of HTML documents. HTML documents are semi-structured as the authors use markup (meta-data) in form of HTML-tags. Even though the HTML-tags are used for layout purposes in most cases, several approaches exist which try to exploit the additional information such markup provides.

5.1 Table Head Extraction

Tables in Web documents are a good example of such additional information provided by HTML-tags. They can provide a lot of information without unnecessary text. For example a table with the two columns "country" and "capital" can list hundreds of valuable relations of the type *is-capital-of* and thus would be very interesting for ontology population. Furthermore, tables which contain overlapping headers with multiple levels of abstraction can provide taxonomic relations. More general header fields would be expressed by the `<colspan>`-/`<rowspan>`-tags summarizing more specific columns or rows.

Even though it is very interesting to use the data inherent to tables, serious problems are that the `<table>`-tag is frequently abused for layout purposes and that the use of the `<th>`-tag for table headers is unreliable. Jung and Kwon hence provide an approach which tries to distinguish meaningful and decorative tables[8] and tries to extract the table head [13].

They observe that decorative tables often contain many links and pictures, many different cell sizes, empty rows or columns, highly customized borders, intermediate cell spans, etc. In contrast to this, meaningful tables often contain textual information and numeric columns or rows. They also observe that missing `<th>`-tags are often compensated by `<b>`- and `<font>`-tags in the first row or column.

[8] Meaningful tables include valuable information, in contrast to decorative tables which for example split the browser window into a navigational and textual part.

From their observations Jung and Kwon generate heuristics and apply machine learning techniques to build a table classifier which decides whether a table is meaningful or not and extracts the identified table head for further usage.

5.2 Group by Markup Context

Another method which exploits Web document markup (HTML) is presented by Kruschwitz [14]. He introduces what he calls *concept terms* as keywords occurring in more than one of the following HTML markup contexts: `<meta>`, `<head>`, `<title>` or emphasizing tags as `<b>` or `<i>`. From the occurrence of a term in several contexts he concludes that this term is more important than others in the document. Hence, he discards everything except the identified concept terms and by this greatly reduces the data examined in further steps.

Kruschwitz then defines what *concept terms of order n* and *related concept terms of order n* areand introduces an *importance* relation on concept terms: In an analyzed document collection a *concept term c of order n* for document d is a term which occurs in at least n different markup contexts of d. Two concept terms c_1 and c_2 are *related of order n* if there exists a document in which both terms are concept terms of order n. And finally c_1 is assumed to be more *important* than c_2 if in every document d c_1 is a concept term of higher order than c_2.

By this the approach does not share the typical first level of the ontology learning layer cake, but identifies terms with the help of whitespace and markup boundaries. Furthermore it allows to infer hierarchical relations between identified concept terms, which were found to be related, even though it is not easy to distinguish which kind of relation exists between the terms.

Kruschwitz uses the extracted relations to build a set of hierarchies as a data driven world model, which is then used to support users of a search engine with additional options to refine their search queries. To achieve this goal, related concept terms which are proposed are sorted descending by the *order* of their relation, so that stronger relations are suggested first. For further readings also see [15].

5.3 XHTML Tree Mining

While Kruschwitz's approach only uses a limited amount and only a flat view of markup contexts, Brunzel presents a more general ontology learning approach called XHTML Tree Mining (XTREEM) [16], which actually consists of a bundle of methods that heavily exploit XHTML structures.

Preprocessing: First of all XTREEM converts its input possibly consisting of invalid or even not well-formed HTML documents (e.g., unclosed HTML-tags as `<br>` or `<li>`, etc.) into well-formed and valid XHTML documents (`<br />`, `<li>...</li>`, etc.). By this, all documents can be interpreted as XML trees.

XTREEM-T: After these preprocessing steps, one of the first deliverables is another method for the extraction of terms (layer one of the ontology learning

```
<html>
    <head>...</head>
    <body>
        <h1>Dog breeds</h1>
        <p>A <b>dog</b> is a domesticated <a>animal</a>...
            <b>breeds</b>...</p>
        <h2>Golden Retriever</h2><p>...</p>
        <h2>Boxer</h2><p>...</p>
        <h2>Dalmatian</h2><p>...</p>
    </body>
</html>
```

Listing 1. Exemplary HTML document about dogs.

layer cake). While Kruschwitz's approach was only focused on a few explicit markup contexts Brunzel's method generalizes the idea to all text spans and extracts sophisticated terms by listing the text spans according to the frequency of their occurrence in an analyzed document collection.

Again notice that in contrast to traditional methods, which rely on NLP- and POS-tagging techniques, this method is mining information in form of HTML-tags which were put into the documents by their authors. It is thus greatly independent from domain knowledge or the language used.

Group-By-Path: In the next step XTREEM introduces a method called Group-By-Path which collects all belonging text spans for every unique path in a document tree. For example consider List. 1, in which the group for the path `<html><body><h2>` would consist of {`Golden Retriever,Boxer,Dalmatian`}. Every document can now be represented as a collection of its text span groups.

Two text spans, resp. identified terms, can be regarded as being related if they share many text span groups, which means that they share many paths. As with Kruschwitz's method before, again the problem occurs that it is not easy to distinguish which relation exists between two terms even in cases where it is observable that they are highly related.

XTREEM-S: Brunzel interprets the identified relation between two terms as a kind of a co-occurrence. By this it is possible to search for synonyms with the help of second order co-occurrences as described in Sect. 4.2. Even though his results are promising, an evaluation shows that precision and especially recall are still very low, so that this method can only find a small amount of synonyms with high probability.

Semantic Siblings (XTREEM-SP & XTREEM-SG): Another application of the Group-By-Path preprocessing is the identification of *semantic siblings*. In contrast to most other ontology learning techniques, which focus on learning hierarchical relations, Brunzel's approach can immediately identify semantic siblings. The Group-By-Path method correlates text spans reachable by the same path, which means that they are on the same level of abstraction.

The method of finding semantic siblings can be split into finding pair-wise or group-wise semantic sibling relations. For a selected pair of terms a sibling

pair relation (XTREEM-SP) can be defined by the significance value of the co-occurrences of both terms. Again the co-occurrence of two terms is the number of text span groups, resp. paths, in which both terms co-occur. For sibling groups (XTREEM-SG) promising results can be achieved by applying different clustering techniques as K-means or hierarchical clustering [16].

6 Ontology Population

As mentioned in Sect. 3, ontology population is closely related to the field of ontology learning. While ontology learning approaches try to construct an ontology out of a (Web) corpus, ontology population methods focus on enriching such ontologies with instances. Therefore they are provided with an *existing ontology* and do not try to modify the given concepts, concept hierarchies or relations but only try to populate the ontology with instances of the given concepts and relations found in the documents.

Ontology population methods are often similar to the presented ontology learning approaches but restricted to the identification of terms and the extraction of *instance-of* relations.

An example for such an ontology population method by de Boer, van Someren and Wielinga can be found in [17]. They extract instances of relations with focus on the art domain, e.g., ("Expressionism" *has_artist* "August Macke") is an instance of the relation *has_artist* with domain Art Style and range Artist. This goal is achieved by searching the Web for documents with the phrase "has artist" and identifying all terms in the documents by matching them to a large art domain dictionary and any names by their capitalization[9]. Afterwards they apply a weak form of lexico-syntactic patterns to extract possible instances of the *has_artist* relation. Identified relations which exceed a desired significance value are then used to populate the initial ontology.

7 Conclusion

Nearly seven years after the proposition of the Semantic Web it is still difficult to find ontologies of the desired level of detail and domain of interest. Nevertheless, the need for such ontologies grows as their use for machine reasoning could greatly simplify everyday tasks as the search for information on the Web.

The ontology learning approaches presented in this seminar paper try to extract valuable information from (Web) documents to narrow this gap and most of them show promising results. Nevertheless, all of the methods have their own advantages and disadvantages.

Even though the application of well known and often used ontology learning methods for plaintext often seems tempting, this also causes various disadvantages. For example a conversion of a Web document into a plaintext representation nearly always looses valuable information in form of stylistic markup which

[9] Notice that this is only possible in few languages like English.

was probably entered by the author manually. This loss of information is very illustrative when considering HTML tables as described in Sect. 5.1. A textual representation of navigational elements might also cause problems.

Methods exploiting such document markup seem very promising. While many big Web search engines heavily rely on markup, the use of such (semi-)structured data appears to be exceptional in current research.

Still none of the presented approaches is perfect. None of the presented methods can extract a complete and exact ontology from a domain corpus and most likely this will never be possible. Nevertheless, every presented method is able to contribute a valuable part of extracted information.

Hence, besides finding new approaches which are able to extract further information from the analyzed documents, one should also focus on how to combine several existing ontology learning approaches. This can help to extract more comprehensive ontologies as well as it can help to compensate for method-specific disadvantages. One such work shall be mentioned here: Popescu, Yates and Etzioni successfully combine the usage of ontology population methods with Hearst-patterns and similarity measures based on co-occurrences to extend an initial ontology with newly discovered instances [18].

Except the combination of presented ontology learning approaches, it can be reasonable to integrate freely available "general purpose" knowledge provided by numerous Web-based information resources. Such resources can provide valuable additions to models derivable from the domain corpus only. Examples can be found in [6] and [19] where additional information from Wikipedia, Wiktionary, DWDS[10] and Wordnet is used.

Last but not least one should also take into consideration that enormous amounts of meta-data in form of tags are generated by millions of users in Web 2.0 platforms every day. The learning of ontologies from such Folksonomies (e.g., [20]) is a highly interesting research field, especially due to the possibilities of nearly immediate interaction with millions of users.

References

[1] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American **284**(5) (May 2001) 34–43

[2] Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition **5** (Jun 1993) 199–220

[3] Gruber, T.R.: Ontology. In Liu, L., Özsu, M.T., eds.: Encyclopedia of Database Systems. Springer-Verlag (2008)

[4] Cimiano, P., Wenderoth, J.: Automatically Learning Qualia Structures from the Web. In: Proc. of the ACL-SIGLEX Workshop on Deep Lexical Acquisition, Ann Arbor, Michigan, Association for Computational Linguistics (June 2005) 28–37

[5] Buitelaar, P., Cimiano, P., Magnini, B.: Ontology Learning from Text: An Overview. In Buitelaar, P., Cimiano, P., Magnini, B., eds.: Ontology Learning

[10] `http://www.dwds.de`

from Text: Methods, Evaluation and Applications. Volume 123 of Frontiers in Artificial Intelligence and Applications. IOS Press (July 2005)
[6] Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: Learning Taxonomic Relations from Heterogeneous Sources of Evidence. In Buitelaar, P., Cimiano, P., Magnini, B., eds.: Ontology Learning from Text: Methods, Evaluation and Applications. Volume 123 of Frontiers in Artificial Intelligence. IOS Press (July 2005) 59–73
[7] Hearst, M.A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proc. of the 14th Internat. Conf. on Computational Linguistics, Nantes, France (July 1992)
[8] Pasça, M.: Acquisition of Categorized Named Entities for Web Search. In: CIKM '04: Proc. of the 13th ACM internat. conf. on Information and knowledge management, New York, NY, USA, ACM (2004) 137–145
[9] Harris, Z.S.: Distributional structure. Word **10(23)** (1954) 146–162
[10] Heyer, G., Quasthoff, U., Wittig, T.: Text Mining: Wissensrohstoff Text. W3L (2006)
[11] Evert, S.: The statistics of word cooccurrences - word pairs and collocations. PhD thesis, Uni Stuttgart, Philosophisch-historische Fakultät, Institut für Maschinelle Sprachverarbeitung, Fachrichtung Computerlingustik (2005)
[12] Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: SIGIR '99: Proc. of the 22nd annual internat. ACM SIGIR conf. on Research and development in information retrieval, New York, NY, USA, ACM Press (August 1999) 206–213
[13] Jung, S.W., Kwon, H.C.: A Scalable Hybrid Approach for Extracting Head Components from Web Tables. IEEE Transactions on Knowledge and Data Engineering **18**(2) (2006) 174–187
[14] Kruschwitz, U.: A Rapidly Acquired Domain Model Derived from Markup Structure. In: Proc. of the ESSLLI'01 Workshop on Semantic Knowledge Acquisition and Categorization, Helsinki, Finland (2001)
[15] Kruschwitz, U.: Automatically Acquired Domain Knowledge for ad hoc Search: Evaluation Results. In: In Proc. of the Internat. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE'03), Beijing, IEEE (2003)
[16] Brunzel, M.: The XTREEM Methods for Ontology Learning from Web Documents. In Buitelaar, P., Cimiano, P., eds.: Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Volume 167 of Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam (2008) 3–26
[17] de Boer, V., van Someren, M., Wielinga, B.J.: Extracting Instances of Relations from Web Documents Using Redundancy. In: The Semantic Web: Research and Applications. Volume 4011 of Lecture Notes in Computer Science. Springer, Berlin / Heidelberg (2006) 245–258
[18] Popescu, A.M., Yates, A., Etzioni, O.: Class Extraction from the World Wide Web. In: Proc. of the AAAI 2004 Workshop on Adaptive Text Extraction and Mining. (2004)
[19] Weber, N., Buitelaar, P.: Web-based Ontology Learning with ISOLDE. In: In Proc. of the Internat. Semantic Web Conf. Workshop. (November 2006)
[20] Schmitz, C., Hotho, A., Jäschke, R., Stumme, G.: Mining Association Rules in Folksonomies. In: Proc. IFCS 2006 Conf., Ljubljana (July 2006) 261–270

Herausforderung der Wissensarbeit:
Bewältigen von Unterbrechungen bei nebenläufigen Arbeiten.

Jens Göddel

Technische Universität Kaiserslautern/ Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI GmbH) Trippstadter Strasse 122, 67663 Kaiserslautern, Deutschland

Zusammenfassung Diese Ausarbeitung beschäftigt sich mit Problemen, deren Ursprüngen und Auswirkungen, die bei der Wissensarbeit auftreten können. Hierbei wird speziell auf die Nebenläufigkeit der Wissensarbeit eingegangen, bei der ein häufiger Wechsel bzw. Unterbrechungen der verschiedenen Tätigkeiten bzw. Aufgaben wahrscheinlich ist. Diese Wechsel zwischen den Tätigkeiten bzw. Aufgaben und die Unterbrechung längerer Tätigkeiten sollen durch speziell entworfene Softwaresystem unterstützt werden, um den Wissensarbeiter bei seinen Aufgaben und Planung zu entlasten und somit auch seine Produktivität zu steigern. Hierzu werden mögliche Softwaresysteme vorgestellt.

1 Einleitung

> "Intellectual capital is far more important than money." (Walter Wriston, quoted in Cortada and Woods, 1999)

Wissensarbeit wird für das betriebliche Überleben und den Erfolg immer wichtiger. Aus diesem Grund ist die Verbesserung der Arbeitsbedingungen und Abläufe der Wissensarbeit in das Blickfeld vieler Forscher und Unternehmen gerückt [1].

Obwohl Wissensarbeit nicht genau definiert werden kann, ohne Kritik und Diskussionen loszutreten, werden Versuche unternommen, sie zu charakterisieren. Dies geschieht aus dem trivialen Grund, dass man das Potential des Wissensarbeiters nur ausschöpfen kann, wenn man die Charakteristiken der Wissensarbeit kennt, die sich auf die Produktivität auswirken. Nur so kann man die geeigneten Bedingungen schaffen und die Prozesse der Wissensarbeit optimieren.

Dazu wird in [1] versucht die wichtigsten Charakteristiken aus einer Vielzahl von Publikationen zu filtern, woraus sich 12 Hauptcharakteristiken von Wissensarbeitern und Wissensarbeit ergeben. Besonders relevant ist, dass durch diese Charakteristiken der erhöhte *Kommunikationsbedarf* und die Wichtigkeit des *Befindens* des Wissensarbeiters für die Produktivität in den Vordergrund gerückt wird, was auch von [2,3,4] als Ursachen für Probleme und Ansatzpunkte für Verbesserung angegeben werden. Zusätzlich wird von [2] noch die Charakteristik *Multitasking*[1] als Ursache gesehen. Auslöser für das Wechseln können

[1] Nebenläufige Aufgaben zwischen denen gewechselt werden muss, um sie zeitgerecht zu erledigen

innerer oder externer Natur sein. Unter anderem werden diese externen Unterbrechungen wiederum von [2,3,4] als Hauptursachen angeführt. In Abschnitt 2 wird auf diese Probleme näher eingegangen und beschrieben, wodurch sie entstehen.

Um den Wissensarbeiter bei diesen Problemen besser unterstützen zu können und vielleicht sogar Teile dieser selbständig oder semi-selbständig zu übernehmen, soll speziell entwickelte Software zum Einsatz kommen. In Abschnitt 3 werden mögliche Softwaresysteme zu dieser Thematik vorgestellt, die diese Funktionalität erfüllen können oder könnten.

Abschließend wird zu den möglichen Lösungen aus Abschnitt 3 Stellung genommen und hinterfragt, ob sie den Funktionsanforderungen bzgl. der Probleme im Bereich Wissensarbeit gerecht werden können.

2 Probleme der Wissensarbeit

2.1 Multitasking

Wissensarbeiter sind meist mit vielen Aufgaben gleichzeitig betraut [2,3,4,5], die meist bis zu einem bestimmten Zeitpunkt parallel erledigt werden müssen. Da diese Aufgaben natürlich nicht gleichzeitig erledigt werden können, muss der Wissensarbeiter sehr oft zwischen den verschiedenen Tätigkeiten, die zum Erreichen der verschiedenen Aufgaben führen, wechseln, um diese termingerecht zu erfüllen. Dies führt zu einem Mehraufwand an Aufmerksamkeit und Zeit. Der Wissensarbeiter muss die verschiedenen Termine und Aufgaben so koordinieren, dass sie in einer bestimmten Zeit effizient erledigt werden können. Er muss diesen so aufgestellten Zeitplan auf dem neusten Stand halten und auch darauf achten, dass er alle Aufgaben im richtigen Zeitabschnitt beginnt und diese auch zum richtigen Zeitpunkt erledigt hat. Alle mit diesen Terminen in Beziehung stehenden Aufgaben, Tätigkeiten und Personen müssen dafür gefunden und eingebunden werden. Weiterhin muss die Planung aber auch auf Unvorhergesehenes wie Software-/Hardwareprobleme (Computerabsturz, Drucker defekt), Änderung der Prioritäten durch übergeordnete Instanzen, aber auch unerwartete Zusammenarbeiten mit Anderen (Gespräche, Diskussionen, E-Mail) neu eingestellt und angepasst werden.

Der Wissensarbeiter ist also meist damit konfrontiert, zwischen verschiedenen Arbeitssituation zu wechseln, diese Wechsel bestmöglich zu koordinieren und dennoch seine Ziele im Blick zu behalten. Dadurch bedingt, bedeutet Multitasking nicht nur einen höheren Zeitaufwand für den Wissensarbeiter, sondern es gilt auch eine immer größere Komplexität (bedingt durch die Anzahl der Aufgaben und assoziierten Tätigkeiten und Personen) zu überblicken und zu beherrschen. In Folge dessen kann es dazu kommen, dass vergessen wird, Tätigkeiten zum richtigen Zeitpunkt zu beginnen und diese dadurch nicht zum gewünschten oder erforderlichen Zeitpunkt fertig gestellt werden können. Dies wiederum kann erhebliche Auswirkungen auf den Zeitplan, das Stressempfinden und die Produktivität des Wissensarbeiters haben. Solch ein Fehler wird auch

als *zukünftiger Erinnerungsfehler* bezeichnet [2,6] und als wichtiger Aspekt bei der Wissensarbeit angesehen, der im hohen Maße zum Problem werden kann. Darum, haben sich Wissensarbeiter eine Vielzahl unterschiedlicher Strategien und Werkzeuge zu Nutze gemacht. Bei diesen Lösungen ist es aber nicht direkt ersichtlich ob, wie weit und in welchem Ausmaß sie zur Handhabung dieses Problems beitragen, besonders weil sie auch meist selbst wiederum einen Wechsel zwischen unterschiedlichen Medien [2] beinhalten (Post-it, Zu-Erledigen-Listen, Kalender usw.). Dies führt dazu, dass nicht die Gesamtheit aller Tätigkeiten der unterschiedlichen Aufgaben beinhaltet ist. Somit ist kein Überblick gewährleistet. Die Integration verschiedener Aufgaben muss meist getrennt betrachtet werden, was wiederum zu den gleichen Problemen führen kann.

Das Hauptaugenmerk liegt hierbei auf der effizienten Neuaufnahme der richtigen Tätigkeit zum richtigen Zeitpunkt und damit einhergehend, auf ein flexibles und zugeschnittenes Koordinieren und Managen der verschiedenen Aufgaben und deren Termine, sowie damit verbundener Tätigkeiten (z.B. Gespräche mit Personen, Vorbereitung der geeigneten Arbeitsumgebung, Auffinden zugehöriger Informationen usw.). An dieser Stelle sollten Softwarelösungen ansetzen können, um eine Verbesserung bei Wissensarbeit herbeizuführen.

Wechsel zwischen Aufgaben oder zugehörigen Tätigkeiten können von unterschiedlicher Natur sein und durch unterschiedliche Geschehnisse ausgelöst werden [2,3,4,5]. Der Wechsel kann zwischen den Tätigkeiten einer übergeordneten Aufgabe oder zwischen größeren Aufgaben selbst stattfinden. Diese stellen die genannten Wechsel dar und sind meistens vom Wissensarbeiter geplant.

Eine andere Art des Wechsels ist es, wenn eine Tätigkeit pausiert wird, um sich etwas anderem zu widmen und zu einem späteren Zeitpunkt die gestoppte Tätigkeit wieder aufzunehmen. Dies wird als Unterbrechung [2,3,4] bezeichnet, welche wiederum andere Probleme verursacht. Auf Unterbrechungen und deren Folgen geht der folgende Abschnitt näher ein.

2.2 Unterbrechungen

Wie schon zuvor erwähnt, ist eine Unterbrechung das Pausieren einer Tätigkeit, um sich möglicherweise etwas Anderem zuzuwenden, mit der wahrscheinlichen Absicht, die pausierte Tätigkeit zu einem späteren Zeitpunkt wieder aufzunehmen. Die Absicht eine solche Unterbrechung vorzunehmen kann aus ganz unterschiedlichen Ursachen her rühren. Der Wissenarbeiter selbst kann die Ursache sein. Er hat dies so geplant oder plant neu(Mittagspause, unerwartete Zusatz-Recherche, neuer Lösungsansatz), dann wird dies als eine *Innere Unterbrechung* bezeichnet. Die Unterbrechung kann aber auch von außen her rühren (Kollegen stellen Fragen, E-Mail, Messenger blinkt auf usw.), dann wird diese Unterbrechung als eine *Externe Unterbrechung* bezeichnet und ist somit meist nicht vom Wissensarbeiter selbst eingeplant. Die Folgen der ersten Art der Unterbrechung sind ähnlich zu den Folgen des Multitaskings, dessen Auswirkungen schon in Abschnitt 2.1 erläutert wurden. Obgleich auch hier einige der Kosten entstehen können, die auch für die externen Unterbrechung entstehen, die folgend beschrieben werden.

Die größeren Auswirkungen haben jedoch die externen Unterbrechungen, die auch von [2,3,4] als Hauptkostentreiber bei Unterbrechungen in der Wissensarbeit gesehen werden. Dabei treten Unterbrechungen bei der Wissensarbeit in relativ hohem Maße auf. Nach Untersuchungen aus [7] geschieht das über viermal pro Stunde, was auch durch den von [1] herausgestellten erhöhten Kommunikationsbedarf eines Wissensarbeiters erklärt wird. Diese Zahl der Unterbrechungen ist sehr beachtlich, wenn man bedenkt, dass die in Abschnitt 2.1 beschriebenen *zukünftigen Erinnerungsfehler* auch von Unterbrechungen ausgelöst werden können und mitunter als ein Hauptgrund für diese Fehler benannt werden [2,7].

Mehrere Untersuchungen zeigten aber auch, dass Unterbrechung sehr unterschiedlich sind und sie damit auch unterschiedliche Auswirkungen haben können.

In den Untersuchungen von [2,8] wurde herausgefunden, dass die Folgen der Unterbrechungen größer sind, je länger und komplexer die unterbrochene Tätigkeit ist. Auch die Anzahl assoziierter Dokumente wirkte sich im gleichen Verhältnis auf die Folgen aus. Folglich haben Komplexität, Länge und zugehörige Dokumente direkte Auswirkungen auf die Kosten einer Unterbrechung.

Außerdem hat der Zusammenhang der unterbrochenen Tätigkeit mit den Tätigkeiten der Unterbrechungen eine indirekte Auswirkung auf die Kosten. Dieser Zusammenhang bezieht sich besonders auf die inhaltliche Korrelation der Tätigkeiten, also inwieweit sie sich inhaltlich nahe stehen [2,3,4]. Zwar sind die Auswirkungen nicht direkt ersichtlich, doch ergeben sich für den Wissensarbeiter, allem Anschein nach, positive Effekte. Er empfindet die Unterbrechung als hilfreich und nicht als störend. Der Einfluss auf sein Stress- und Frustrationsempfinden ist geringer und dadurch wird auch die Arbeitsbelastung als niedriger empfunden. Zusätzlich wird aus [4] ersichtlich, dass die benötigte Zeit bis zur Wiederaufnahme bei stärker korrelierenden Tätigkeiten verkürzt wird. Somit zeigt sich, dass der Wissensarbeiter selbst starken Einfluss auf die Folgen einer Unterbrechung haben kann. Nicht nur die Art der Unterbrechung selbst, auch die Einstellung zu neuen Erfahrungen und der Grad an persönlichen Strukturen, die benötigt werden, spielen eine Rolle bei der Kostenverursachung einer Unterbrechung. Dieser Aspekt wird in [3] dadurch ersichtlich, dass diese zwei Faktoren im inversen Verhältnis zur zeitlichen Bearbeitung der Unterbrechung stehen und damit die Einstellung einen direkten Einfluss auf die Produktivität des Wissenarbeiters hat. Was verdeutlicht, dass die Folgen von Unterbrechungen somit nicht nur zeitlicher Natur sind, sondern sich nach [2,3,4,8] auch stark auf den Wissensarbeiter selbst auswirken. Dies wird auch dadurch erklärt, dass die Wissensarbeiter versuchten, die verlorene Zeit durch schnelleres Arbeiten zu kompensieren [3]. Es verursacht zwar keine großen Qualitätseinbußen, schlägt sich aber verstärkt auf das Stress- und Frustrationsempfinden des Wissensarbeiters nieder, was nach [1] ein wichtiger Aspekt bei der Produktivität ist. Dies wird auch von [2,3,4,8] hervorgehoben.

Zusammenfassend ergeben sich die folgenden Faktoren bei der Kostenentwicklung:

- Komplexität
- Dauer der Tätigkeiten

- Assoziationen zu den Tätigkeiten
- Stress- und Frustrationsempfinden
- Persönlichkeit des Wissensarbeiters (Einstellung und Arbeitsmethoden)

Diese Faktoren wirken sich somit auf die Kosten einer Unterbrechung aus und bestimmen, in welchem Grad die Unterbrechung sich auf die Produktivität auswirkt. Die hierbei *beeinflussten* Kosten sind hauptsächlich zeitliche Aspekte, was die Gesamtzeit für reguläre Tätigkeiten sowie die Bearbeitungszeit der Unterbrechung selbst angeht. Insbesondere schlägt sich die Zeit, die benötigt wird, um die unterbrochene Tätigkeit wieder aufzunehmen (in den Kontext zurück finden, zusätzliche Ablenkungsaktivitäten [4]) hier nieder und zwar umso mehr, je länger die Unterbrechungszeit[2] ist. Weiter werden die gefühlte Arbeitsbelastung und der gefühlte Zeitdruck stark beeinträchtigt, welche von den verschiedenen Studien immer mehr als produktivitätsentscheidend in den Vordergrund gerückt werden [1,2,3,4].

Zusätzlich zu den genannten Faktoren spielt die Frage, *wann* unterbrochen wird, ein wichtige Rolle, was zuvor auch schon durch die Typisierung der Unterbrechung und der Abgrenzung derer angedeutet wurde und von [4,8] weiter heraus gestellt wird. Es wird hierbei versucht, die zuvor beschriebenen Auswirkungen dadurch zu verringern, dass die Unterbrechung auf einen geeigneteren Zeitpunkt verschoben wird, zu dem die unterbrechende Tätigkeit dann stattfindet [4,8]. Diese Zeitpunkte werden *Breakpoints* genannt. Es hat sich gezeigt, dass Unterbrechungen an diesen Breakpoints weniger Auswirkungen haben. Die Zeit, die benötigt wurde, um zur unterbrochenen Tätigkeit zurückzukehren, sowie die schnellere Bearbeitung der Unterbrechungstätigkeit selbst und die Auswirkungen auf das Befinden des Wissensarbeiters, wurden stark gesenkt (Stress, Frustration und Arbeitsbelastung). Es werden hierbei drei Typen von Breakpoints unterschieden:

- Coarse
- Medium
- Fine

Als *Coarse* bezeichnete Unterbrechungen existieren zwischen größeren Aufgaben. Die Kosten für eine Unterbrechung zu solch einem Zeitpunkt sind am geringsten [4,8]. Als *Fine* bezeichnete Unterbrechungen existieren zwischen den kleinstmöglichen Tätigkeiten. Dieses spiegelt sich auch in den oben beschrieben Arten einer Unterbrechung wieder. Die Art und Weise der richtigen Momentfindung ist aber nur für die Verwendung von festgelegten Tätigkeits- und Aufgabenabläufen positiv erforscht. Bei frei wählbaren Abläufen sind positive Auswirkung noch nicht erwiesen.

Eine weitere Möglichkeit der zeitlich geschickten Unterbrechung bietet sich durch die Aufstellung von statistischen Modellen an. Mittels Sammeln und Analysieren von Indikatoren am Desktop oder anderen messbaren physikalischen

[2] Gesamt verstrichene Zeit, seit die Tätigkeit unterbrochen wurde bis zum Beginn der Wiederaufnahme.

Hinweisen in der Umgebung des Wissensarbeiters wird es ermöglicht den Wissensarbeiter zu einem geschickten Moment zu unterbrechen (z.B. Tippen, scrollen, browsen → Aufmerksamkeit des Wissensarbeiters). Diese Art der Ermittlung könnte dann auch die erwähnten Breakpoints subsumieren und zusammen ein so genanntes *composite model* ergeben. Die Ergebnisse dieser Modelle [4] sind, bzgl. einem korrektem Auffinden der *Breakpoints* mit einer Wahrscheinlichkeit von 70-90%, sehr ermutigend.

Durch die Annahme einer effizienten und ausreichend korrekten Auffindung solch zeitlich geschickter Unterbrechungsmomente stellt sich die Frage, wann welche Unterbrechung am sinnvollsten bzgl. ihrer Kosten ist. Hierbei spielt die zuvor erwähnte Korrelation der Tätigkeiten die Hauptrolle. Nach [4] verhält es sich so, dass Unterbrechungen, die an Breakpoints geschehen, zwar meist alle geringere Kosten aufweisen. Diese Kosten können aber noch verbessert werden, indem eine Unterbrechung, die inhaltlich mit der gerade ausgeführten Tätigkeit korreliert, zu *Fine* oder *Medium Breakpoints* stattfindet. Die Ursachen und Auswirkungen hierzu sind die gleichen, wie zuvor oben schon bei der Korrelation erwähnt. Wenn aber umgekehrt solch eine korrelierende Unterbrechung auf einen *Coarse Breakpoint* verlagert wird, können dadurch mögliche negative Folgen für den Wissensarbeiter entstehen (z.B. Wiederaufarbeitung des Kontextes aus der schon abgeschlossenen Aufgabe). Für nicht zur Aufgabe passende Unterbrechungen ist aber das Gegenteil der Fall, hier werden *Coarse Breakpoints* bevorzugt, welche zu höherer Zufriedenheit und Produktivität führen.

Für die in diesem Abschnitt beschriebenen Probleme ist es also wichtig, dass mögliche Systemlösungen dem Wissensarbeiter bei der effizienten Wiederaufnahme der Tätigkeit helfen. Das heißt die Rückführung in den Kontext der unterbrochenen Aufgabe oder Tätigkeit ermöglichen und das so, dass kein zu hoher Mehraufwand vom Wissensarbeiter betrieben werden muss. Wichtig ist es hierbei auch, die für die Tätigkeit wichtigen Assoziationen zu behalten und dem Wissensarbeiter zu einem bestimmten Zeitpunkt wieder leicht zugänglich zu machen. Zusätzlich sollte versucht werden den Blick des Wissensarbeiters für die noch zu erledigenden Arbeiten zu schärfen, damit dieser nicht zwischendurch seine Aufmerksamkeit mit unnötigen oder momentan falschen Beschäftigungen zerstreut. Der Wissensarbeiter sollte somit auch an unterbrochene Tätigkeiten erinnert werden, möglichst zu einem guten Zeitpunkt. Mögliche Auswirkungen auf die Einstellung des Wissensarbeiters sollten in besonderem Maße berücksichtigt werden, da diese wie zuvor beschrieben sich direkt oder indirekt stark auf die Produktivität auswirken. Hierzu sollte versucht werden, die verschiedenen inneren oder externen Unterbrechungen, ähnlich wie beim Multitasking schon erwähnt, zu koordinieren, um so die Komplexität für den Wissensarbeiter zu reduzieren. Ebenfalls sollte verstärkt auch auf die Art der Tätigkeit und dabei konkret auf die Dauer dieser geschaut werden. Außerdem sollte darauf geachtet werden, dass diese Funktionalitäten möglichst flexibel gehalten werden, das heißt auf die persönlichen Strukturen des Wissenarbeiters einstellbar sind. Diese persönlichen Einstellungen und Strukturen des Wissensarbeiters spielen auch eine entscheidende Rolle, wenn man versucht die Kosten für eine Unterbrechung

durch eine zeitlich geschickte Verzögerung zu reduzieren, wie es im Detail weiter oben beschrieben wurde.

3 Softwaresysteme für die Wissensarbeit

Die vorhergehenden Ausführungen zeigen, dass der Wissensarbeiter bedingt durch die Charakteristika seines Tätigkeitsbereiches mit einigen Schwierigkeiten zurecht kommen muss. Mögliche Unterstützung der Wissensarbeit sollte bei den Ursachen und Auswirkungen von Multitasking und Unterbrechungen ansetzen, die zuvor beschrieben wurden. Im folgenden werden dazu einige Softwaresysteme bezüglich ihrer Ansätze und Funktionen vorgestellt.

3.1 Multitasking Assistenten

GroupBar ist ein Programm, das von der Microsoft Forschungsgruppe VIBE (Visualization and Interaction for Business and Entertainment) entwickelt wurde. Es versucht den Wissensarbeiter bei dem Wechseln zwischen verschiedenen Tätigkeiten zu unterstützen. Hierbei wird besonderen Wert auf das *Managen* vieler verschiedener offener Fenster gelegt (Window-Management). Da es sich nach [9] gezeigt hat, dass durch größere Bildschirme und Arbeitsflächen immer mehr Fenster vom Nutzer offen gelassen werden und in eine von ihm bevorzugte Position gebracht werden, ist dies eine Kernaufgabe von GroupBar. Weiter soll dem Nutzer ermöglicht werden, seine Fenster in einen Kontext zu bringen, einem bestimmten Projekt zuzuordnen und zu gruppieren, um einen besseren Überblick zu ermöglichen. Dies spiegelt sich auch im Namen der Anwendung wider. Durch die Anlehnung an die original Windows-Taskbar wird dem Nutzer eine möglichst einfache und intuitive Benutzung ermöglicht.

GroupBar stellt hierfür eine Drag&Drop-Funktion zur Verfügung, mit der es möglich wird, die offenen Fenster und Tasks zu ordnen (arrangieren) oder in Gruppen zusammenzufassen.

Weiter soll durch Überwachen des Nutzerverhaltens ein automatisches Gruppieren der Fenster ermöglicht werden. Dabei werden Fenster zusammengefasst, die basierend auf der Frequenz der gemeinsamen Nutzung oder der zeitlichen Nähe für im hohem Maße zusammengehörig gehalten werden. Die so auf die ein oder andere Art arrangierten und gruppierten Fenster können dann als ein Ganzes gesehen werden und in einem gemeinsamen Kontext verwaltet werden. GroupBar bietet für die so gruppierten Fenster eine Reihe von Funktionen.

Es ist möglich, einzelnen Gruppen ein Layout zuzuordnen, das einen leichteren Einstieg in den Kontext ermöglicht und somit ein schnelleres Zurechtfinden garantiert. Dieses Layout kann vom Nutzer durch die Snapshot-Funktion selbst bestimmt werden, die die Anordnung und die Form der momentanen Gruppe aufzeichnet und dauerhaft speichert. Der Nutzer kann unter bewährten Layouts wählen, die von GroupBar unter Berücksichtigung von Erkenntnissen des *Window-Managements* bereitgestellt werden.

Mit einem Linksklick auf die Gruppe können die Fenster in einer Gruppe gleichzeitig aufgerufen werden oder, wenn sie schon offen sind, mit der gleichen Aktion auch wieder alle geschlossen werden. Dies ermöglicht einen schnellen und intuitiven Übergang zwischen Tätigkeiten in verschiedenen Kontexten.

Eine weitere herausragende Funktionalität für Gruppen ist die *Restore Funktion*. Diese macht es möglich, persistierte Gruppen mit Layout und Positionierung wiederherzustellen, selbst wenn diese nicht geöffnet sind. Hierfür muss der Nutzer nur eine Gruppe mittels der Snapshot-Funktion persistieren (mit Tag--Funktion), um diese dann später aus einer Liste mit Vorschau und Beschreibung der einzelnen Gruppen wieder auszuwählen. So gespeicherten Gruppen werden von GroupBar dann genauso wieder hergestellt, wie sie aufgezeichnet wurden.

GroupBar hat sich in Studien [2,9] als sehr hilfreich erwiesen, um sich effizient zwischen verschieden Tätigkeiten zu bewegen und die Fülle an Aufgaben zu organisieren und zu überblicken. Die als intuitiv bezeichneten Funktionen wurden von den verschieden Nutzern hierbei ohne großen Zeitaufwand erkannt und auch für das weitere Arbeiten angenommen. Hierbei wurde vor allem die zeitsparende Restore-Funktion in den Vordergrund gestellt, die dem Benutzer ermüdende und frustrierende Aufgaben abnimmt, sowie das schnelle Zurechtfinden durch richtig arrangierte Fenster (personalisiert) erleichtert. Weiterhin sollen die vorgegebenen Layouts durch das GroupBar-Team für die verschiedenen Ausgabegeräte weiter entwickelt und angepasst werden, was ein schnelleres und adäquates Anordnen ermöglichen soll.

Scalable Fabric ist ein Programm, das in seiner Funktionalität ähnlich der von GroupBar ist. Es wird dem Wissensarbeiter auch ermöglicht, seine Fenster per Drag&Drop anzuordnen und zu gruppieren. Eine Speicherung dieser Anordnung und das automatische Öffnen wird aber nicht angeboten. Die Darstellung der Gruppierung und die Anordnung unterscheidet sich aber von GroupBar. Im Unterschied zur GroupBar ordnet Scalable Fabric die Gruppen in der Peripherie des Desktops an. Weiterhin wird das Originalfenster durch Bewegen an den Desktoprand immer weiter verkleinert (shrinking), je näher man dem Desktoprand kommt [10]. Fenster die auf die gleiche Position am Desktoprand verschoben werden, werden zusammen gruppiert.

Die Vorteile sind für den Wissensarbeiter etwa gleich denen von GroupBar, was das Gruppieren angeht. Doch ergeben sich durch die kleineren Bilder Schwierigkeiten beim Orientieren zwischen den verschieden Gruppen, sowie in der Gruppe selbst und somit auch für das Wechseln zwischen den Aufgaben [11].

TaskTracer ist ein Programm, das auf der Idee aufbaut, dass Wissensarbeiter ihre Arbeit und Abläufe in einzelne Aufgaben unterteilen. Mit diesen Aufgaben werden diverse Prozesse verbunden, die zur Erledigung der Aufgabe führen sollen [12]. Weiter werden diesen Aufgaben und Prozessen verschiedene Informationen/Ressourcen (Dokumente, Nachrichten, Kontakte mit Kollegen, Internetadressen, etc.) sowie zugehörige Hilfsmittel (Anwendungen,Telefone, etc.), die zum Zugriff auf die Ressourcen und Bearbeiten dieser dienen, zugeordnet.

TaskTracer ordnet diesen Aufgaben somit mittels umfangreicher Datensammlung über Ablauf, Ressourcen, Hilfsmittel und das Benutzerverhalten ein Profil zu. Hierbei greift TaskTracer unter anderem auf das Microsoft .NET Framework, das Betriebssystem selbst und Visual Basic for Applications (VBA) Compiler zurück [12]. Diese Profile können aber auch durch den Nutzer verändert und angepasst werden, was im initialen Zustand von TaskTracer auch erforderlich ist.

Mittels dieser Profile will TaskTracer den Wissensarbeiter unterstützen, indem die momentane Aufgabe erkannt und mögliche Wechsel zuverlässig vorhergesehen werden. Auch soll durch Zuhilfenahme dieser Informationssammlung und der Aufbereitung der Daten eine komfortable Wiederaufnahme von unterbrochenen Aufgaben und Tätigkeiten ermöglicht werden. TaskTracer bietet z.B. hierfür die Funktionalität, alle Anwendungen und Ressourcen, die mit den wiederaufgenommenen Aufgaben/Prozessen assoziiert werden, wiederherzustellen (neu starten/in den Vordergrund zurückholen) und bei Dokumenten, die bearbeitet wurden, die Stellen hervorzuheben, die zuletzt verändert wurden. Zusätzlich ist es TaskTracer durch Analyse des momentanen Prozesses und der kumulierten vergangenen Prozesse möglich, die Arbeitsoberfläche des Benutzers adäquat zu ordnen und mit den richtigen Inhalten zu füllen (FileDialoge anpassen, Quickstartbar mit geeigneten Programmen füllen und im Speicher vorladen, etc.). Ein weiterer Vorteil dieses Vorgehens ist die Möglichkeit, anderen Wissensarbeitern diese Daten zur Verfügung zu stellen, um diese zu unterstützen und anzuleiten. Somit ist hierdurch auch ein potentieller Wissenstransfer ermöglicht. TaskTracer ist durch den modularen Aufbau sehr gut geeignet, die Funktionalität des Programms auch in andere Software einfließen zu lassen und auch andere UI (UserInterfaces), als die schon vorhandenen, zu verwenden. Durch diese Eigenschaften ist ein Zusammenspiel von GroupBar und TaskTracer durchaus denkbar und auch sinnvoll, da sie sich in ihren Funktionalitäten und Benutzeroberflächen sehr gut ergänzen würden.

Es hat sich gezeigt, dass es unter Verwendung von TaskTracer dem Wissensarbeiter erleichtert wird, seine Abläufe und Aufgaben zu planen und zu strukturieren [12]. Vor allem die gute Sammlung und Darstellung der assoziierten Ressourcen und Hilfsmitteln erleichtern die Orientierung und helfen den Überblick zu bewahren. Außerdem wird durch die Funktionalitäten des Programms das Wechseln und Wiederaufnehmen der Aufgaben und Tätigkeiten erleichtert. Die flexible Anpassung der Prozesse, aber vor allem auch das selbstständige Erlernen der Abläufe und somit auch das Erkennen der aktuellen Aufgabe, werden sich in einem durch Multitasking geprägten Arbeitsumfeld als nützlich erweisen, da die Software den Wissensarbeiter durch diese Kontext-Informationen besser unterstützen kann. Des Weiteren hilft es dem Wissensarbeiter, wenn er seine Aufgaben und Ziele ständig vor Augen hat, was auch durch das UI von TaskTracer unterstützt wird.

Clipping List/ChangeBorders möchte den Wissensarbeiter bei den drei Hauptproblemen unterstützen und bei aktuellen Prozessen und Abfolgen von

Aufgaben den Überblick wahren. Weiter soll der Wissensarbeiter sensibilisiert werden, besser zu erkennen, wann er zu unterbrochenen Aufgaben zurückkehren muss. Dies ist vor allem bei unterbrochenen Tätigkeiten, die bei Eintritt eines bestimmten Ereignisses wieder aufgenommen werden sollen, wichtig. Außerdem soll es dem Wissensarbeiter erleichtert werden, seine Aufmerksamkeit wieder auf die unterbrochene Tätigkeit zu lenken und effizient zu ihr zurückzufinden. Clipping List setzt hierbei, anders als die zuvor vorgestellten Systeme, auf das Aufbereiten und Darstellen von relevanten Informationen der Tätigkeiten/Fenster, die in den Hintergrund gerückt werden. Hierbei wird versucht den Benutzer so wenig wie möglich abzulenken, ihm aber so viele Information wie nötig zukommen zu lassen.

Clipping List geht dabei so vor, dass der Nutzer wichtige Stellen in Dokumenten oder Programmen, welche er in den Hintergrund versetzen möchte, markiert. Diese Informationen werden ihm dann in einer vertikalen Liste peripher im Desktop dargestellt, daher auch die Namensgebung. Zu diesen Ausschnitten wird auch noch ein Piktogramm dargestellt, da es sich gezeigt hat, dass dies die Orientierung enorm erleichtert [11]. Dieses Vorgehen bei der Infromationsbeschaffung ist angelehnt an das System WinCuts [13]. Die so gewonnenen Ausschnitte mit relevanten Informationen ersetzen damit praktisch die minimierten Fenster von Scalable Fabric und bieten dem Benutzer eine bessere Möglichkeit sich zurechtzufinden. Dies wird in [11] als herausragender Vorteil angesehen. Weiterhin ist es dem Benutzer aber auch möglich, diese Ausschnitte auf verschiedene Arten zu gruppieren und zu ordnen (Wichtigkeit/Abfolge). Kombiniert mit *ChangeBorders* zeigt sich Clipping List noch effektiver bei der Unterstützung des Wissensarbeiters [11]. Tritt eine Veränderung in einem Programm oder Dokument auf, das in die Clipping List aufgenommen wurde, wird der Ausschnitt in der Clipping List mit einer farblichen Veränderung seines Rahmens kenntlich gemacht (hierbei ist auch ein Kenntlichmachen unterschiedlicher Veränderung durch verschieden Farben angedacht). Es hat sich gezeigt, dass durch *flexibles* Darstellen von *relevanten* Information in Form von Clipping List in der Peripherie des Desktops die Performance des Wissensarbeiters erheblich gesteigert wurde. Außerdem wurden die Informationen der Ausschnitte durchweg als ausreichend empfunden [11]. Was Wichtigkeit und Notwendigkeit angeht, wurde das zeitliche Verhalten durch Signalisieren und besseres Orientieren erheblich verbessert. Auch der Wechsel zwischen Tätigkeiten wurde als einfacher empfunden und auch beschleunigt. Diese Umstände führten in der Studie von [11] dazu, dass die Benutzer sich leicht damit taten, Clipping List in ihre Arbeit einzubinden, wobei die flexiblen Informationen als wichtigster Aspekt gesehen wurden. Zusätzlich wurde die Zeit minimiert, die zwischen dem Wechseln von Tätigkeiten durch Ablenkung oder Zerstreuung vergeudet wird.

3.2 Unterbrechungs-Assistent

Wegen des begrenzten Rahmens, wird exemplarisch für Systeme, die an Hand von *composite models* (siehe Abschnitt 2.2) Unterbrechungen auf Breakpoints verzögern, hier nur auf das OASIS System eingegangen.

OASIS ist ein System, das sich aus zwei Hauptbestandteilen zusammensetzt, dem *Breakpoint Detector* und dem *Scheduler*. Letzter fängt/erhält Unterbrechungsanfragen (Requests). Diese müssen dem Scheduler mitteilen, zu welchem Breakpoint sie übermittelt werden sollen und welchen Zeitspanne sie maximal warten können (Request-Policy). Mit diesen Informationen kann der Scheduler nun die richtigen Weichen stellen, indem er sie mit den momentanen Ergebnissen des Breakpoint Detectors abgleicht. Dieser teilt dem Scheduler mit, welcher Breakpoint momentan vorliegt. Hierfür wird das Benutzerverhalten gesammelt und mittels composite models (siehe Abschnitt 2.2) analysiert. OASIS ermöglicht es diese Modelle noch anzupassen und anzulernen, um die Ergebnisse weiter zu verbessern. Das kann manuell oder mittels *Toolkits*, wie z.B. *Subtle* [14], geschehen. In [4] wurde hierbei festgestellt, dass das System bei der Erkennung von Breakpoints gute Ergebnisse liefert, aber leider nicht immer mit dem Benutzer bei den *Breakpoint-Typen* übereinstimmt. Dieser Umstand wurde auf die Schwierigkeit der Kontextbestimmung und der wenigen Informationen zurückgeführt.

Trotz allem hat sich gezeigt, dass durch Verwendung von OASIS die Frustration der Benutzer, sowie die Bearbeitungszeit der eigentlichen Aufgaben und die der Unterbrechung, signifikant gesunken ist. Auswirkungen auf die Zeit, die zwischen Unterbrechung und Aufgabe an Verstreuung verloren geht, konnten nicht festgestellt werden.

4 Zusammenfassung

In dieser Ausarbeitung wurde gezeigt, dass Wissensarbeiter durch Probleme des Multitaskings und durch Unterbrechungen stark beeinflusst werden. Diese Probleme wirken sich direkt oder indirekt auf die Produktivität und die Performance des Wissensarbeiters aus. Die in Abschnitt 3 exemplarisch erläuterten Softwaresysteme sind mit ihren teilweise unterschiedlichen Ansätzen alle auf dem richtigen Weg, den Wissensarbeiter bei diesen Problemen zu unterstützen und erzielen auch brauchbare Ergebnisse damit. Durch Kombination verschiedener Ansätzen würden sich vielleicht noch weitaus bessere Ergebnisse ergeben. So könnte es zum Beispiel hilfreich sein, den composite models eines OASIS Systems die Kontextinformationen, die mit GroupBar, TaskTracer und vor allem *Clipping List* erhoben werden, zugänglich zu machen, um die Typen der Breakpoints besser zu erkennen. Die in [15,16] beschriebenen Ansätze könnten auch dazu beitragen, automatisch für den Benutzer relevante Informationen aus Ressourcen zu extrahieren und diese zum Beispiel Clipping List zur Verfügung zu stellen. Es wurde hier zwar noch kein Königsweg gefunden, aber die Probleme sind erkannt und die Lösungen weisen durchweg in die richtige Richtung.

Literatur

1. Yau, J.W.: Defining knowledge work: A british and hispanic cross-cultural study (March 2003)

2. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2004) 175–182
3. Mark, G., Gudith, D., Klocke, U.: The cost of interrupted work: more speed , stress. In: CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2008) 107–110
4. Iqbal, S.T., Bailey, B.P.: Effects of intelligent notification management on users and their tasks. In: CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2008) 93–102
5. González, V.M., Mark, G.: Managing currents of work: multi-tasking among multiple collaborations. In: ECSCW'05: Proceedings of the ninth conference on European Conference on Computer Supported Cooperative Work, New York, NY, USA, Springer-Verlag New York, Inc. (2005) 143–162
6. Ellis, J., Kvavilashvili, L.: Applied cognitive psychology (2000)
7. O'Conaill, B., Frohlich, D.: Timespace in the workplace: dealing with interruptions. In: CHI '95: Conference companion on Human factors in computing systems, New York, NY, USA, ACM (1995) 262–263
8. Adamczyk, P.D., Bailey, B.P.: If not now, when?: the effects of interruption at different moments within task execution. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2004) 271–278
9. Smith, G., Baudisch, P., Robertson, G., Czerwinski, M., Meyers, B., Robbins, D., Andrews, D.: Groupbar: The taskbar evolved. In: Proc. OZCHI, Microsoft Research (2003) 34–43
10. Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D.R.: Scalable fabric: flexible task management. In: AVI '04: Proceedings of the working conference on Advanced visual interfaces, New York, NY, USA, ACM (2004) 85–89
11. Matthews, T., Czerwinski, M., Robertson, G., Tan, D.: Clipping lists , change borders: improving multitasking efficiency with peripheral information design. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 989–998
12. Dragunov, A.N., Dietterich, T.G.: Tasktracer: a desktop environment to support multi-tasking knowledge workers. In: IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces, New York, NY, USA, ACM (2005) 75–82
13. Tan, D.S., Meyers, B., Czerwinski, M.: Wincuts: manipulating arbitrary window regions for more effective use of screen space. In: CHI '04: CHI '04 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM (2004) 1525–1528
14. Fogarty, J., Hudson, S.E.: Toolkit support for developing and deploying sensor-based statistical models of human situations. In: CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2007) 135–144
15. Buscher, G., Dengel, A., van Elst, L., Mittag, F.: Generating and using gaze-based document annotations. In: CHI '08: CHI '08 extended abstracts on Human factors in computing systems, New York, NY, USA, ACM (2008) 3045–3050
16. Miller, T., Agne, S.: Attention-based information retrieval using eye tracker data. In: K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture, New York, NY, USA, ACM (2005) 209–210