Fakultät für Informatik
der Technischen Universität München

User-Centered Social Software – Model and Characteristics of a Software Family for Social Information Management

Vanda Lehel

Fakultät für Informatik
der Technischen Universität München

Lehrstuhl für Software Engineering
betrieblicher Informationssysteme

User-Centered Social Software – Model and Characteristics of a Software Family for Social Information Management

Vanda Lehel

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Helmut Krcmar
Prüfer der Dissertation:	1. Univ.-Prof. Dr. Florian Matthes
	2. Univ.-Prof. Dr. Johann Schlichter

Die Dissertation wurde am 18.04.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 09.08.2007 angenommen.

Abstract

In recent years, web community platforms that pertain to the emerging class of *social software* have gained considerable popularity among lead users on the Web. Social software platforms focus on supporting information exchange in online social networks by providing services for information search, publication and sharing, and collaborative classification of personal information in individual users' social contexts. Hence, individual users can link together and exchange information with selected contacts like co-workers, friends or family members by means of social software.

However, functional features and information models of social software platforms reveal limitations from the individual user's perspective with regard to managing distributed personal information, that is, most significantly lack of service integration, flexible metadata management and cross-platform relation management of personal information.

Within the scope of this thesis, the concept of *user-centered social software* as an innovative approach for personal information management support in online social networks is developed. The basis for this concept is established by the characterization of different kinds of social software and the extension of personal information management activities to information exchange in online social networks referred to as *social information management*.

The user-centered social software model therefore proposes an integrated view on all available information from the individual user's perspective and encompasses services for the acquisition of relevant information, controlled information dissemination to selected contacts and flexible metadata and semantic information relation management concerning the organization of distributed personal information.

The characteristics of user-centered social software are further substantiated by means of a software architecture design and are illustrated on a user interface prototype, termed *Social Organizer*.

Acknowledgements

This work emerged from my activities as a research assistant at the Chair Software Engineering for Business Information Systems (Prof. Dr. Florian Matthes) at the Department of Informatics of the Technische Universität München.

First, I would like to thank my doctoral advisor, Prof. Dr. Florian Matthes, for providing the opportunity to work on this interesting research topic and his suggestions and advice that greatly contributed to the success of this work. I would also like to thank my second referee, Prof. Dr. Johann Schlichter, for valuable suggestions to enhance essential parts of the thesis.

I have enjoyed collaborating with my colleague Sheng Wei and thank him for fruitful discussions and cooperation within our joint research project.

I would like to thank my father, Dr. Csaba Lehel, for helpful suggestions that greatly helped me improving this work. I also thank Patrick Hupe for helpful discussions and comments to enhance parts of this thesis.

I am very grateful to my family and Patrick for their support, constant encouragement and understanding during my doctoral work.

Contents

Index of Figures

Index of Tables

Index of Acronyms

BLOB	Binary Large Object
CMC	Computer-Mediated Communication
DHTML	Dynamic HTML
ECA	Event Condition Action
FOAF	Friend of a Friend
GDI	General Definition of Information
HFC	Hierarchical Faceted Categories
HTML	Hypertext Markup Language
IM	Information Management
IRC	Internet Relay Chat
KM	Knowledge Management
MXML	Multimedia Extensible Markup Language
OCSS	Object-Centered Social Software
OODBMS	Object-Oriented Database Management System
OPML	Outline Processor Markup Language
OSAF	Open Source Applications Foundation
PIM	Personal Information Management
PKM	Personal Knowledge Management
RBAC	Role-Based Access Control Model
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RIA	Rich Internet Application
RSS	RDF Site Summary (RSS 0.9, 1.0), Really Simple Syndication (RSS 2.0)
SNA	Social Network Analysis
SOA	Service-Oriented Architecture
SSP	Social Software Platform
UCSS	User-Centered Social Software
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XFN	XHTML Friends Network
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language

Chapter 1

Introduction

Social software has been arising as a new class of web community platforms focusing on individual users and enabling them to organize, acquire and disseminate information in their respective social contexts. This chapter motivates the need for the extension of personal information management support to online social networks – referred to as social information management – by the family of user-centered social software.

1.1 Motivation

Over the last decade, the World Wide Web has evolved into the primary information acquisition and dissemination medium since it enables a standardized presentation and networking of commercial, official and private content. Due to significant technological progress concerning the miniaturization of computing devices, web content becomes more and more ubiquitous being accessible from anywhere by anyone from several kinds of stationary and mobile devices [JoBr05]. Thus, a fast-growing number of services provide access to content available at multiple information sources. Moreover, machine-readable semantic enrichment of web content is an emerging practice that facilitates the development of higher-order services for information access [BHL01].

In general, multiple services are integrated on web platforms to meet specific requirements. Significant solutions for such multi-user platforms include (enterprise) information portals and community platforms. The former are centralized systems that enable personalized access to a certain part of available content and offer their users services like unified storage, classification, search, and networking mechanisms for content [We02]. By contrast, community platforms primarily focus on content publication and sharing support as well as interpersonal and group communication services for their members. An essential part of both kinds of web platforms is therefore user management providing authentication and authorization mechanisms based on user groups and user roles.

Web community platforms with further services enabling the representation of real-world and virtual social relationships and social interaction are an emerging class of social software. Examples for such platforms include on the one hand object-centered social software like Flickr [Flic07] to

publish photographs and images, CiteULike [Cite07] for publication and sharing of bibliographic references or del.icio.us [Deli07] to manage bookmark collections online. On the other hand, social networking services provide the opportunity to explicitly establish and manage social relationships between individuals. Well-established examples of social networking services are Linkedln [Link07] in the United States and XING [Xing07] in Europe, both widely used for managing business contacts.

Social software platforms make a significant contribution to the further development of the web which is denoted by the emerging term *Web 2.0* in compliance with O'Reilly [Or05]. This transformation of the World Wide Web results in a further logical network that is established on top of the physically connected devices of the Internet and the logical network of hyperlinked and semantically enriched hypertext content. Web 2.0 evolves as a combination of Semantic Web concepts [BHL01] with the strength of social networks. This integration is achieved by services like social software.

Due to quasi-permanent access to online services in the private environment the location and management of information is not restricted to personal devices. Hence, individuals act in a multitude of roles as social software platform users and thus in different social contexts both private and professional with the objective of completing a task. Examples for an individual's tasks include assorting educational material for a given topic from trusted sources, sharing photographs with family members or completing steps of a business process in the role of an employee.

Before a particular task can be accomplished in any social context, acquisition and/or dissemination of specific content is required from web and local information sources. In addition, content needs to be organized for later retrieval and (re-)use. The aforementioned three processes of information acquisition, dissemination and organization represent the primary processes of social information management as an extension of personal information management based on [TJB06].

The above examples point out the essential characteristics that need to be taken into consideration when trying to support the three processes of social information management mentioned before. In the case of information acquisition, the relevance of information and trustworthiness of the information source play an important role. The process of information organization can be enhanced by flexible metadata and cross-platform relation management of information objects. In order to address security concerns in the case of information dissemination, users need to be supported in sharing information exactly with the people they intend to by appropriate authorization mechanisms.

Social information management is thus viewed as the enrichment of corresponding personal information management processes by social factors due to an extension of its scope to social networks that are established at social software platforms on the web.

So far, a multitude of so-called personal applications exist that address the support of personal information management, i.e. organization and exchange of specific types of information stored locally on the personal device like information about contacts, e-mails or RSS feeds. Examples for such applications are personal information managers like Microsoft Outlook [Outl07], feed readers (e.g. RSSOwl [RSSO07]) and mind mapping tools like the PersonalBrain [Pers07]. Tools used on stationary devices are still more appropriate for this purpose than those available for mobile devices due to the field of application and technological restrictions. In contrast to social software platforms, these personal applications do not focus on services supporting the organization and exchange of information in the several social contexts of a user, though.

The considerations above point out an increasing need for supporting users in managing distributed personal information on their personal devices and the social software platforms they are members of as stated in [LeMa05]. From the user's perspective, an application is required that provides a unified

view on all available information including services for information organization, acquisition and dissemination. On the one hand, unifying approaches exist for personal information management support across personal applications like Haystack [Ka05] developed in the context of a research project and the commercial product JetBrains Omea Pro [Omea07]. On the other hand, integration of several social software platforms on a single platform by customized aggregation of content and corresponding services is increasingly present on the web, Netvibes [Netv07] and Pageflakes [Page07] being representative examples.

However, an appropriate model for a social information management tool integrating support of personal information management and information exchange in online social networks – referred to as *user-centered social software* (*UCSS*) throughout this thesis – does not yet exist.

1.2 Main Contributions of this Thesis

The objective of this thesis is to characterize the newly-emerging family of user-centered social software and present a reference model for the development of applications that belong to this software family.

The application scenario of user-centered social software for social information management support is shown in Figure 1.1.

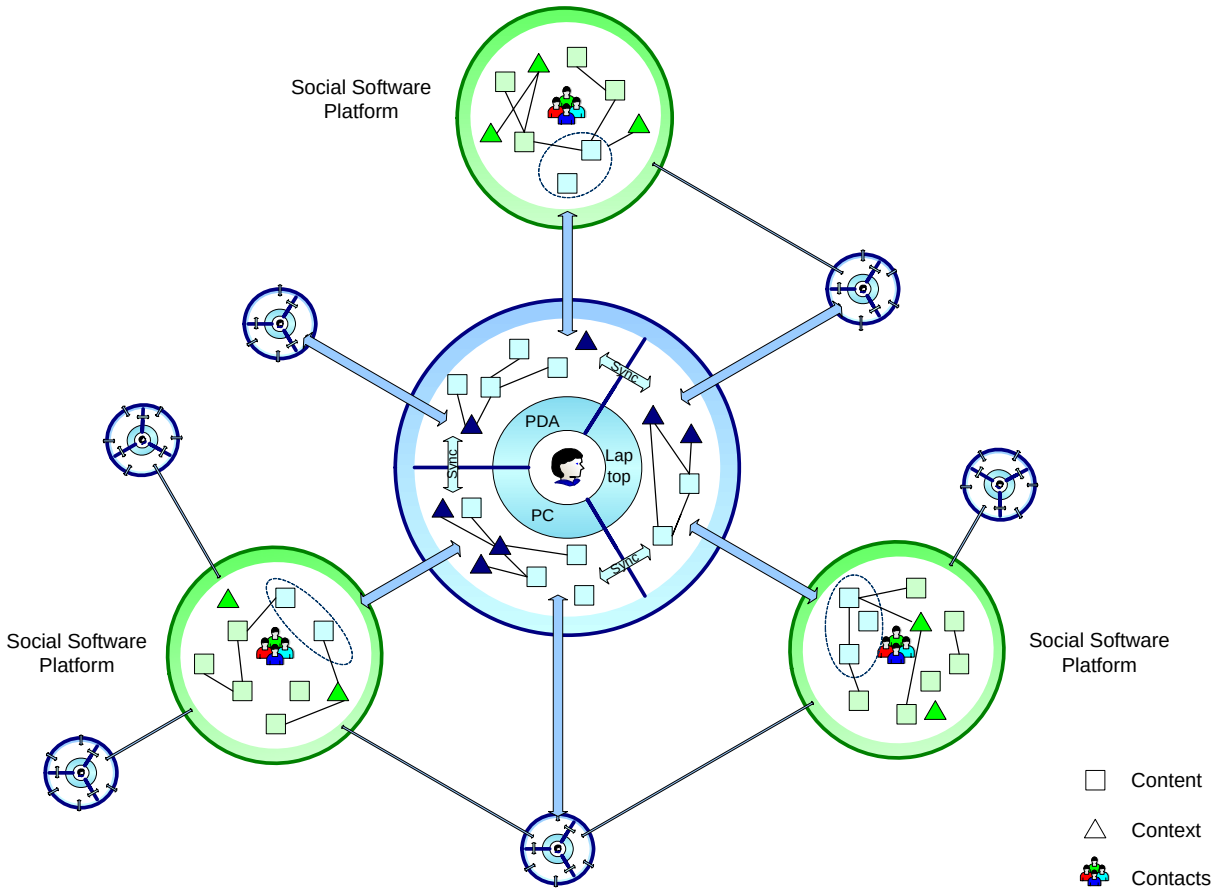


FIGURE 1.1. *Application scenario of user-centered social software*

In the center of the diagram, the individual user's view on all available information from both personal applications and social software is illustrated. Information acquisition and dissemination as specific cases of bidirectional information exchange in the case of social information management occur between the user and his contacts. Social software platforms are represented by circles with the according contacts illustrated by user group icons whereas peer-to-peer communication services are shown as direct connections to the contacts' personal devices. Bidirectional arrows depict the use of open protocols for information exchange.

On each stationary and mobile device, like PCs, Laptops or PDAs, personal information is stored and managed by appropriate personal applications in the form of content objects (shown as squares in Fig. 1.1) like documents, e-mails, weblogs or RSS feeds. These are linked together directly or by means of contexts indicated by triangles in Fig. 1.1. A context represents an abstraction of classification schemes such as categories, keywords and directories (see Section 4.1 for details) in the scope of this thesis.

On social software platforms, the individual user's content objects are managed in his social contexts – e.g. for information organization and sharing purposes or in order to obtain feedback – and are networked with the content of other platform members, that is, the user's contacts, by means of shared contexts. In the case of social software platforms, contexts are often specified in the form of freely chosen and shared keywords, referred to as tags (see Section 3.2 for details), that are created by the platform members themselves.

The scenario in Figure 1.1 further illustrates that the user can access personal information either on his different personal devices or the content published by himself or his contacts at social software platforms.

With the proposal of user-centered social software, the thesis first of all addresses the lack of an integrated view on the available content in the personal and the several social contexts of a user. For this purpose, services of existing personal applications and social software (e.g. search, classification, publication and sharing, commenting) are integrated for supporting simultaneous information acquisition, organization and dissemination processes regardless where the information originates from.

The basis for the concept of user-centered social software is established by the characterization of different kinds of social software and the extension of personal information management activities to information exchange in online social networks. Use cases are extracted from the processes of social information management in order to elaborate requirements for the specification of business rules that facilitate acquisition of relevant information and controlled information dissemination to selected contacts.

Further value-adding services considering information organization are the management of flexible user-defined metadata and cross-platform semantic relation management among content objects. By means of the latter, links between content objects located at different social software platforms or managed by different personal applications are established manually or automatically and visualized in an integrated view.

Within the scope of this thesis, an example of a user-centered social software, termed *Social Organizer*, is designed that illustrates social information management support by means of a user interface prototype.

1.3 Thesis Structure

The subsequent chapters of this thesis are structured according to the following description while each chapter concludes with an intermediate summary.

In Chapter 2, personal information management and relevant supporting solutions are presented. Section 2.1 is devoted to the classification of information and introduces the concept of information objects that are representable in information systems. In Section 2.2, the objectives, primary processes and requirements of personal information management support are expounded. Based upon this background, related solutions for personal information management support in the scope of this thesis are reviewed in Section 2.3.

Chapter 3 is concerned with social software. Section 3.1 provides the fundamentals of information exchange relationships in social networks. The subsequent Section 3.2 is devoted to a characterization of social software platforms based on their distinct purposes. Further, static and dynamic views on social software platforms are developed in Section 3.3. From an analysis of information models of special-purpose social software platforms a generic content provider model is deduced and social software service are categorized that also satisfy the model and services of personal applications as a special case.

Based on these preliminary considerations, in Chapter 4, the characteristics and a model of user-centered social software based on the three primary concepts content, context and contacts are discussed in detail in Section 4.1. This chapter also includes further considerations on social information management as an extension of personal information management, in particular, which use cases of information acquisition, dissemination and organization can be identified which are discussed in Section 4.2. The subsequent Section 4.3 introduces the role-based authorization model of user-centered social software that is essential for the support of information dissemination use cases. Section 4.4 is finally devoted to the discussion of rule-based support of the identified use cases by user-centered social software.

Chapter 5 is devoted to the design of a user-centered social software instance with significant use cases, termed the Social Organizer. Section 5.1 presents the information model of the Social Organizer that relies on the user-centered social software model of Chapter 4. In Section 5.2, the specification of rules is explained considering predefined rule schemata on the one hand and rule schemata defined by the individual user on the other hand. Section 5.3 discusses social information management support in the Social Organizer including user interface metaphors for significant concepts of the user-centered social software model. Implementation aspects concerning the prototyping of the Social Organizer are given in Section 5.4.

The thesis concludes in Chapter 6 with a summary of contributions and an outlook on future research directions.

Chapter 2

Personal Information Management

This chapter aims at introducing personal information management (PIM) in order to establish the basis for its extension to online social networks, termed social information management. Relevant terminology on fundamentals of the concept information and characteristics of information objects are described for this purpose. Based upon this background, the main processes of personal information management as auxiliary activities for the accomplishment of personal tasks are detailed and requirements for unified PIM support are elaborated. The chapter concludes with a review on significant supporting solutions for personal information management within the scope of this thesis.

2.1 Terminology

Individuals increasingly have access to digitally stored information from diverse areas of applications that are managed on stationary and mobile personal devices. Considering the entirety of personal information, not merely digital metadata exist but digital documents in the form of text, image, audio or video files are managed. Frequently, brief notes or commented bookmarks can be found on personal devices in addition to complete documents [MaLe02b]. Hence, individual users manage vast amounts of personal information and need to be supported in the related activities by appropriate tools.

A definition of the term *personal information management* within the scope of this thesis is given in this section. For this purpose, the prerequisites of classifying *information* and characterizing *information objects* are initially expounded in the following.

2.1.1 Classification of Information

The concept of information is interpreted in several ways depending on the respective field [Fl05]. From the perspective of information systems in the field of computer science, the concept of information is based on information theory founded by Shannon [Sh48].

In information theory, information is considered as data communication. Information is encoded and stored in the form of *data* which is transmitted in terms of *messages* during human-machine or

machine-machine communication respectively. Information is encoded following a specific *syntax* that determines the rules of form and composition of data from symbols of the underlying alphabet (e.g. binary encoding). *Well-formed* data is thus correct according to the given syntactical rules [Fl05]. According to Shannon "frequently the messages have *meaning*" ([Sh48], p. 5, emphasis in original), that is, the data incorporate *semantics* interpreted by the recipient of the message. Further, *pragmatics* is concerned with the actual interpretation of the transmitted information in a particular context.

The General Definition of Information (GDI): σ is an instance of information, understood as semantic content, if and only if: (GDI.1) σ consists of one or more *data*; (GDI.2) the data in σ are *well-formed*; (GDI.3) the well-formed data in σ are *meaningful.*

TABLE 2.1. *The General Definition of Information (source: [Fl05])*

Based on the aforementioned considerations, Floridi [Fl05] formulates the *General Definition of Information* (*GDI*) applied in sciences where information is viewed as semantic content. The GDI as a tripartite definition is given in Table 2.1. From this, it follows that information as semantic content is equivalent to well-formed and meaningful data.

Data, Information and Knowledge

Besides the concept of data, information is usually correlated with the concept of *knowledge* in relevant literature. Based on their meaning, often a hierarchical ordering of data, information and knowledge is presumed.

Watson [Wa02] establishes a hierarchy of meanings which is depicted in Figure 2.1. The hierarchy is on the one hand related to context dependence. On the other hand, understanding the meaning of the aforementioned three concepts through a specific level of interpretation by an individual is essential. The process of understanding by an individual thus determines transitions between data and information as well as information and knowledge (see Figure 2.1).

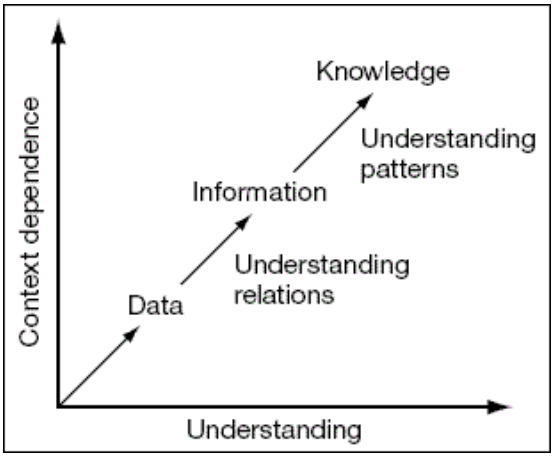


FIGURE 2.1. *Data, information and knowledge as a hierarchy of meanings (source: [Wa02], p. 7)*

According to Figure 2.1, data is established on the lowest level with marginal context dependence since data is seen as a representation of information residing on the next level in the hierarchy. For transition between the first two levels, understanding of inherent relations is required as depicted in Figure 2.1. In the hierarchy of meanings, knowledge emerges from information by understanding specific patterns in information. Knowledge is a pragmatic concept since it also highly depends on the individual's context.

By contrast, Callahan [Ca06] describes the interrelations of data, information and knowledge as a system as depicted in Figure 2.2. Transition processes occur in a specific context that determines the meaning of these concepts. Similar to the abovementioned hierarchical approach, interpretation of the three concepts by an individual is required. Hence, data is viewed as a pre-stage and thus formal representation of information which is deduced from the underlying data. Transition from information to knowledge is referred to as the process of *sensemaking*, though, as depicted in Figure 2.2. Further, a feedback loop is incorporated in the diagram in order to denote that existing knowledge has impact on how information is understood from data in the given context.

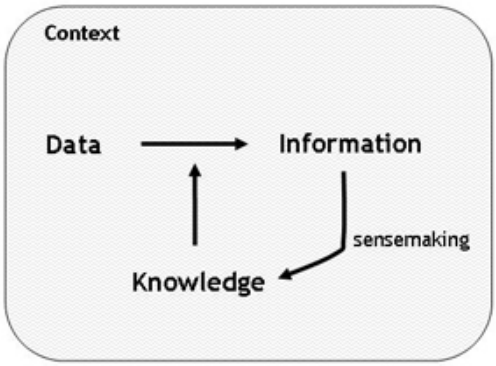


FIGURE 2.2. *Data, information and knowledge as a system (source: [Ca06])*

Information as Explicit Knowledge

Polanyi [Pola58] identified two different kinds of knowledge that are termed *explicit knowledge* and *tacit knowledge* respectively. Explicit knowledge indicates knowledge that can be expressed and transmitted and is thus representable in information systems. Hence, explicit knowledge is considered equivalent to the concept of information.

The distinction between explicit and tacit knowledge is revealed by Polanyi in his statement "We can know more than we can tell." ([Po66], p. 4). Polanyi expresses by this means that the human brain comprehends noticeably more (tacit knowledge) than it is able to impart (explicit knowledge). Tacit knowledge therefore refers to expert knowledge of individuals that cannot be immediately made useful for other persons or represented in information systems.

In order to harness knowledge within an enterprise Nonaka and Takeuchi [NoTa95] identified four significant modes of knowledge conversion. These describe potential transformation processes between explicit and tacit knowledge depicted in Figure 2.3. *Externalization* denotes that explicit knowledge evolves from tacit knowledge. By contrast, *internalization* suggests the conversion of explicit into tacit knowledge. The transformation of explicit knowledge into another form of explicit knowledge is referred to as *combination*. During *socialization* no information is created but merely tacit knowledge is shared among individuals.

Since expressed tacit knowledge becomes information through the process of externalization it can be handled accordingly like information [JoBr05].

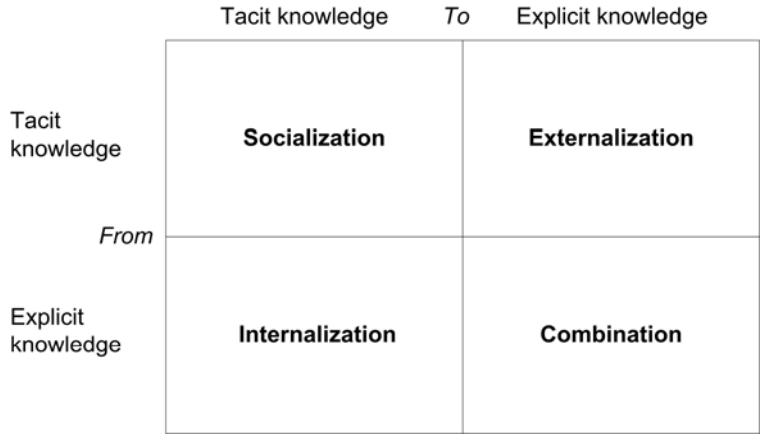


FIGURE 2.3. *Four modes of knowledge conversion (source: [NoTa95], p. 62)*

2.1.2 Information Objects

Semantic entities of digitally stored information are termed *information objects*. An information object is usually typed according to the represented semantic entity. Examples for *types* of information objects that can be represented in repositories of information systems and applications include contacts, e-mails, calendar items, as well as textual, image, audio and video files and annotations. In a broader sense, all information objects of the aforementioned types can be considered as *documents* [Bu97].

Information objects are made persistent in *information repositories*. An information object is either comprised of completely structured, unstructured or semi-structured data. Structured data signifies that an information object encapsulates *metadata* representing its attributes by typed values [St04]. Information objects with underlying unstructured data do not have any associated metadata but are merely composed of unstructured content in corresponding files. Semi-structured data indicates that an information object is comprised of a combination of metadata and unstructured content. Typically, textual documents, images, audio and video files are composed of unstructured or semi-structured data [BlAt03].

In Figure 2.4, a semi-structured document object and a structured contact object are exemplified. The document object *d* encapsulates three attributes or *properties* as metadata: *title* of type string, *size* of a numerical type and creation date represented by the date attribute *created*. Since document *d* also refers to its unstructured content (see Figure 2.4) it is considered as semi-structured. The contact object *c* illustrates a structured information object with associated metadata representing contact and personal information about an individual as depicted in Figure 2.4.

Applications on the personal device (e.g. e-mail client, word processing tool, image editing tool) visualize specific types of information objects and/or provide lifecycle management – create, edit, transform and delete operations – for metadata (and content) and facilitate organization of information objects [JoBr05]. However, the representation format of identical types of information objects usually varies depending on the given database schema in the information repository of the respective applications. Management of untyped information objects is more general and flexible which results in more complex requirements for maintaining and handling them, though. Schema-free management of

information objects can be achieved by dynamically associating attributes to a general information object.

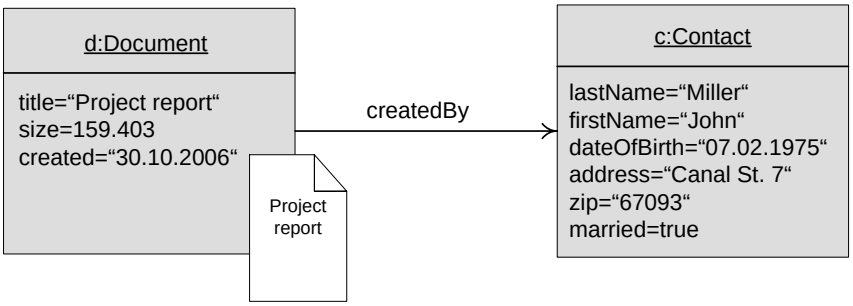


FIGURE 2.4. *Information objects composed of structured and semi-structured data*

Throughout this thesis, *content objects* are synonyms for information objects and the term *content type* is hence also used to denote the type of an information object.

Relations of Information Objects

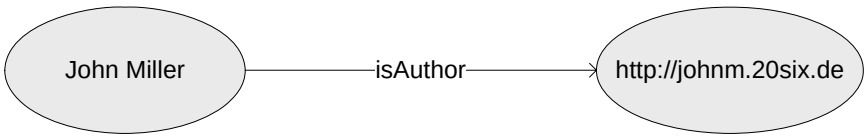
Besides the type and structure of an information object, relations to other information objects play a significant role. In the scope of this thesis, the notation 'linked to' is applied to denote that two information objects are related by any relation.

Thus, a *link* is established between two information objects based on a *link type* while the latter represents the relation. A pair of information objects can either be related by a *unidirectional link* (*directed link*) or a *bidirectional link* (*undirected link*).

Figure 2.4 exemplifies that the document object *d* and the contact object *c* are related by the relation *createdBy* which is indicated by the directed arrow within the object diagram.

Different representations of related information objects exist that include graphical representations and human-readable and/or machine-readable textual statements.

As illustrated by the example in Figure 2.5, the relation *isAuthor* is established between an information object of type contact and an information object of type weblog which is an online personal diary (see Section 3.2.2 for details).




```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ucss="http://wwwmatthes.in.tum.de/ucss">
  <rdf:Description rdf:about="http://johnm.20six.de">
    <ucss:isAuthor>John Miller</ucss:isAuthor>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 2.5. *Representation of a binary relation between information objects in RDF*

The relation is depicted graphically by ellipses representing information objects and a directed arrow showing the direction of the relation. In addition, the relation is expressed XML-based using RDF statements [KlCa04] at the bottom part of Figure 2.5.

The relation in RDF is considered as a triple of subject, predicate and object where an RDF description states that the subject – information object from the domain – is related to the object – information object from the range – by the predicate which denotes the relation.

Further, the inverse relation of *createdBy* in Figure 2.4 is thus *isAuthor* which is depicted in Figure 2.5.

Networks of Information Objects

Networks of information objects emerge from one or more binary relations over a set of information objects that are linked together. A network of information objects as nodes can be represented as a graph.

Information objects in a network are connected by *direct links* if they are linked by directed or undirected links, that is, a path of length 1 exists in the graph that connects those objects. Further, information objects are considered as connected by *indirect links* if a path of length greater than 1 exists in the network [Ha69].

In Figure 2.6 a network of information objects of different content types is visualized as a digraph [Eu02]. The labels of the unidirectional links indicate the underlying relations or link types respectively.

For example, an object of type person – representing a contact object in previous examples (see Figures 2.4 and 2.5) – is linked to an information object of type place by the link type *lives at*. Further, the same person object is related to a document by the link type *has published*.

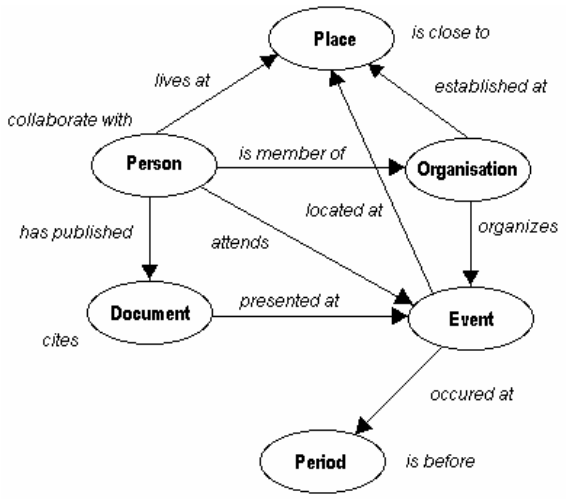


FIGURE 2.6. *Visualization of a network of information objects as a digraph (source: [Eu02])*

Personal Information Objects

Personal information encompasses all information available to or about an individual user. According to [JoBr05] and [MaLe02b], several categories of *personal information objects* exist.

Information objects exist that are kept and maintained by the individual for personal use. A further distinction is made in [MaLe02b] resulting in two categories of personally kept information objects since information objects are handled that have either been personally created or merely acquired and are personally owned by the individual user.

Examples for personally created and personally owned information objects are given in the first two columns of Table 2.2. For instance, information about personal contacts, appointments, notes, photos and recommendations of references and bookmarks fall into the first category. By contrast, personally owned information objects encompass journals, lecture notes, e-mails from contacts and collected photos and video files.

The third category of personal information objects comprises information about a person which is maintained by others. Examples for personal information objects of this category are shown in the third column of Table 2.2 including certificates, health information, contracts and user profile information. Considering this category, ensuring privacy of personal information is an essential issue.

The entirety of these personal information objects maintained and controlled by an individual is subsumed under the term *personal space of information* (*PSI*) according to [JoBr05].

Additionally, Jones and Bruce [JoBr05] define a fourth category of personal information which has been experienced by a person. For instance, information encountered by the individual when browsing books or web pages fall into the category of personally experienced information which is immediately transformed into knowledge. The category of personally experienced information is thus not further considered within the scope of this thesis.

Personally created information	Personally owned information	Information about a person
• Personal contacts, appointments and tasks linked to individuals, and organizations • Recommended references and bookmarks • Ideas, annotations and personal notes • Curriculum vitae, personal data sheet • Personal publications and presentations • Personal photos, audio and video recordings • Personal correspondence (mail, e-mail, fax, chat)	• Books, journals, lecture notes, manuals • Correspondence and project documents from contacts • Collected photos, audio and video recordings • Information to real-world entities (finance, property, ...)	• Certificates • Press releases • Health information, medical records • Contracts • Memberships • User profiles and account information

TABLE 2.2. *Examples of personal information objects (source: [MaLe02b])*

Based on these preliminary considerations, personal information management is characterized in the following section.

2.2 Characterization of Personal Information Management

The term *personal information management* (*PIM*) was coined by Lansdale [La88] in order to subsume "the methods and procedures by which we handle, categorize, and retrieve information on a day-to-day basis". Thus, personal information management refers to personal information owned and managed by an individual user in contrast to *information management* in the context of enterprises.

According to Krcmar [Kr04] the latter refers to management tasks concerning information supply and demand, underlying information systems and information and communication technologies within an enterprise. In order to achieve effective information management, however, individuals' use of information needs to be taken into consideration [Da94] that results in increasing significance of personal information management in the corporate environment.

Personal information management can further be distinguished from *knowledge management* and *personal knowledge management* based on the interpretation level of the concepts information and knowledge discussed in Section 2.1.1.

Drucker [Dr93] recognized the significance of knowledge management as in today's society the basic economic resource is intellectual capital, i.e. knowledge, handled by an increasing number of employees in enterprises. Drucker emphasized their field of activity by the term *knowledge worker*.

In compliance with Nonaka and Takeuchi [NoTa95], knowledge management focuses on harnessing tacit as well as explicit knowledge in enterprises (see Section 2.1.1). Therefore, knowledge management extends information management of externalized, i.e. representable knowledge, to supporting the four modes of knowledge conversion introduced in Section 2.1.1.

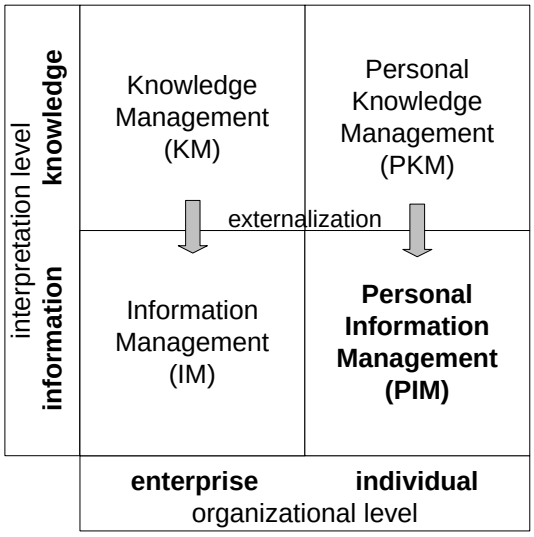


FIGURE 2.7. *Classification of personal information management*

Figure 2.7 (left hand side) depicts the related concepts of information management (IM) and knowledge management (KM) at the enterprise level. An analogous relationship between personal information management (PIM) and personal knowledge management (PKM) at the individual level is further illustrated in Figure 2.7 (right hand side). The externalization of knowledge is indicated by gray vertical arrows (see Figure 2.7).

Dorsey [Do01] refers to personal knowledge management as a set of human problem-solving skills of retrieving, evaluating, organizing, analyzing, presenting, securing and collaborating information

which are required for knowledge work. Davenport [Da05] holds a similar view with regard to personal knowledge management by stating that its focus lies in enhancing the individual's capabilities of acquiring, processing and using knowledge in an enterprise and thus PKM support plays an essential role.

2.2.1 Personal Information Management versus Personal Knowledge Management

The comprehension of personal information management and its tool-based support varies in relevant literature. Available definitions usually agree upon explicit knowledge or information representing the central component of PIM, though [JoBr05].

Personal information management often refers to a special case of information management in enterprises focusing on an individual user in the role of an employee (Barreau [Ba95], Rautenstrauch [Ra97]).

In compliance with Rautenstrauch [Ra97], personal information management denotes the efficient design of the workplace in an enterprise according to the individual user's needs. The objective of personal information management is thus to ensure an adequate information supply at the workplace concerning the individual user's tasks.

This definition, however, can be extended to several roles of an individual user, not merely his role as an employee in the context of enterprises. In the private context, individual users are also concerned with managing several kinds of personal information objects as illustrated in Table 2.1 that are not restricted to their workplace.

According to Spurgin [Sp06], personal information management encompasses storing and organizing information in order to retrieve it for later reuse whereas Graham [Gr04] refers to personal knowledge management as organizing, storing and sharing thoughts. Hence, there is often merely a slight distinction between the perception of personal information management and personal knowledge management.

Personal information management addresses explicit knowledge, that is, information, comprising information organization and exchange. According to Jones and Bruce [JoBr05], expressed knowledge is equivalent to information which is addressed by personal information management. This view is adopted in the scope of this thesis and *organizing thoughts* is considered as a process of personal information management to be supported by PIM tools.

Teevan et al. ([TJB06], p. 40) provide the following definition for personal information management:

> "Personal information management (PIM) is intended to support the activities we, as individuals, perform to order our daily lives through the acquisition, organization, maintenance, retrieval and sharing of information."

This definition extends Barreau's [Ba95] view on PIM which does not include information dissemination as a PIM activity. The above definition is appropriate and adopted here since this thesis focuses on acquiring, organizing and exchanging personal information objects in the individual user's several social contexts.

2.2.2 Objectives of Personal Information Management

Personal information management aims at providing the individual "the right information at the right time, in the right place, in the right form, and of sufficient completeness and quality to perform the current activity" ([Jo04], p.1). These requirements refer to digital information handled today. However, electronic documents partly replace large amounts of paper documents (e.g. books, contracts and letters) which have been maintained and organized by individual users and led to the necessity for personal information management from the beginning of the 20th century [JoBr05]. Today, individual users manage personal information in paper and digital form which often exist in multiple versions.

Processes of Personal Information Management

Individual users manage their personal information in order to fulfill the aforementioned requirements and hence obtain appropriate information for completing a specific *personal task*. Thus, "*PIM activities are an effort to establish, use and maintain a mapping between information and need.*" ([JoBr05], p. 10, emphasis in original). Relevant information being available on time may be critical for particular tasks in the professional and private context of individual users [JoBr05]. An approach to support collaborative task completion, i.e. support of business processes like ordering, integrated into an enterprise information portal is described in [Le01].

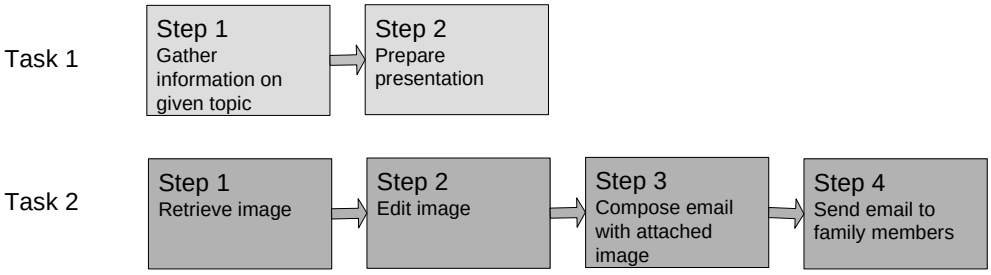


FIGURE 2.8. *Examples for personal tasks involving information handling*

By contrast, personal tasks for example include gathering information on a given topic to prepare a presentation or editing an image and sending it by e-mail to family members (see Figure 2.8). Hence, a task is composed of multiple *steps* as depicted in Figure 2.8. According to [JoBr05], tasks differ in length of time for accomplishment and the frequency the individual user is concerned with a particular *type of task*. In contrast to task-specific or production activities respectively, personal information management encompasses supporting activities [Bo04].

Thus, personal information is *acquired* and *organized* for later reuse before accomplishing a task. Further, steps of a task often involve PIM processes of information *acquisition* and *dissemination*. Concerning the abovementioned examples, information acquisition from local and remote information sources is applied in order to retrieve existing and gather new material on a topic (see Step 1 of Task 1 in Figure 2.8). The last step of the Task 2 in Figure 2.8 includes information dissemination when the e-mail is sent in order to successfully complete the task.

The three recurring processes of personal information management based on the definition in Section 2.2.1 are information acquisition, organization and dissemination as depicted in Figure 2.9. Information maintenance is indicated by the information repository the processes operate on (see Figure 2.9). Information retrieval and use are considered as task-specific activities in the scope of this thesis.

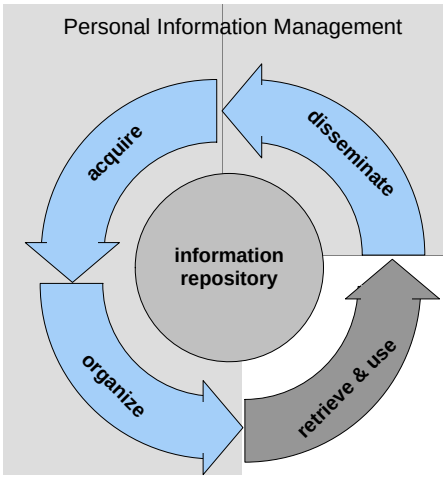


FIGURE 2.9. *Processes of personal information management*

The cycle starts with information acquisition which involves activities to search, browse, encounter and decide to keep or not found information objects. While searching and browsing are usually goal-oriented activities – especially when related to a particular task – the individual user has to anticipate how encountered information objects may be useful at a later date in order to decide to maintain and manage them [JoBr05] (see Section 2.2.3).

Information organization as the succeeding step encompasses classifying (e.g. by directory, folder and keyword), linking together personal information objects and extending them by metadata (e.g. annotation) which are accomplished independently of tasks. The individual user retrieves and uses information at a later point in time in order to accomplish a personal task. Task-specific steps involve creating, editing, transforming and deleting information objects, i.e. lifecycle management, accomplished by associated applications on the personal device.

Typically, information dissemination, that is, publication and sharing of information objects, is associated with a step of a personal task (see Step 4 of Task 2 in Figure 2.8) and occurs during information use. Information dissemination is thus shown as the following step of the cycle.

Figure 2.10 exemplifies the way personal information management processes are involved in the completion of tasks during a time interval as illustrated by the examples in Figure 2.8. The example tasks in Figure 2.10 are composed of two and three steps respectively. The steps of Task 1 are shaded in light gray while the steps of Task 2 are indicated by dark gray shading. Further, the typical scenario at an individual user's workplace of accomplishing tasks in parallel is depicted (*multitasking*).

Initially, it is assumed that information objects have been acquired and organized by the individual user and are thus maintained in the information repository/ies of the personal device before starting to work on a task. Task 1 then starts with acquiring new information. Before accomplishing the next task-specific step of Task 1, the acquired information is organized task-independently within the repository (see Figure 2.10). The succeeding step the individual user performs is the first step of Task 2 that again involves information acquisition. Following this step, the last step of Task 1 is completed which requires information dissemination. Task 2 is then completed by a task-specific step.

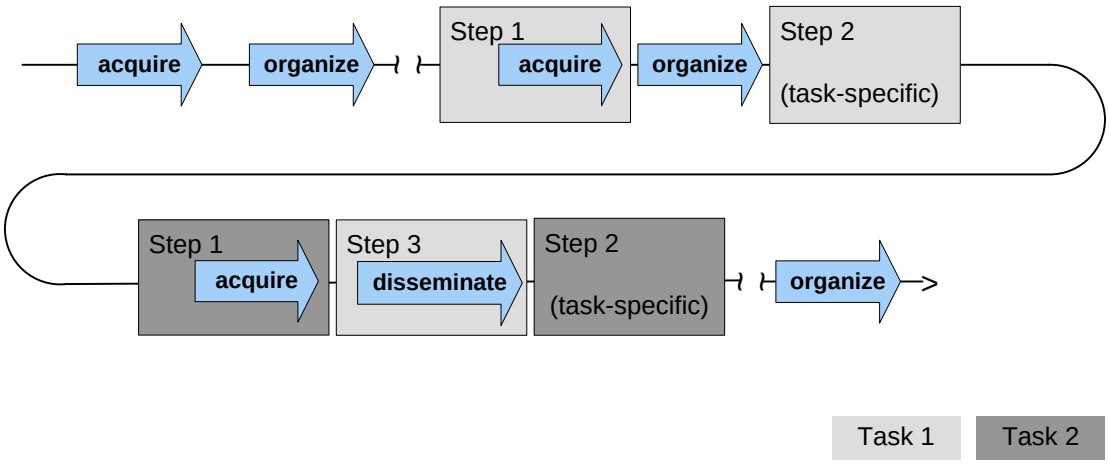


FIGURE 2.10. *Significance of PIM processes for task completion*

After the exemplified sequence of performing tasks, the individual user may organize information in the repositories and thus the scenario periodically restarts with a new (different) sequence of task completion.

2.2.3 Acquisition and Retrieval Strategies of Personal Information

The process of information acquisition involves interpreting information and deciding to store it or not for future use. Personal information maintained and organized in the information repositories of personal devices can hence be retrieved more efficiently when required for performing a particular task. Therefore, individual users develop strategies for balancing costs and values of maintaining, organizing and retrieving information objects to meet current needs.

Individual users' information-seeking behavior to meet information needs is subject to study across several disciplines including information science, computer science and cognitive psychology [Wi94] as discussed in the following subsections.

Information Acquisition Strategies

As described in Section 2.2.2, new information objects are either found when searched for on purpose or coincidentally encountered during information acquisition by the individual user. Remote information sources to acquire digital information from are usually found on the web by using search engines like Google [Goog07a] or accessing information portals or web community platforms (see Chapter 3).

According to Kirsh, however, in recent years "the amount of information has been rising exponentially while the amount of usable or high quality information has been rising only linearly." ([Ki00], p. 26). The estimation by Lyman and Varian [LyVa03] in their survey about the World Wide Web containing 167 terabytes of information on static web pages and about 450 times more dynamically generated web content in the year 2003 explains that cognitive overload of individual users is caused by uncertain quality and relevance of the huge information supply.

On the one hand, *oversupply of information* results from pushed information from remote information sources including e-mails, subscriptions and real-time content like news, stock quotes or weather forecasts over which the individual user has limited control.

Oversupply of pulled information on the other hand occurs when the individual user keeps searching assuming that still more appropriate information exists on the web to meet current or future needs. This kind of *exploratory search* or *orienteering* is characterized by continuous keyword search and browsing of available information sources ([Wh06], [Te07]).

Due to a variety of information sources information acquisition may be affected by frequently obtaining conflicting information that demands the individual user's careful attention [Go99].

Searching for relevant information with sufficient quality is hence a time-consuming auxiliary task for the individual user. *Information anxiety* may arise from the abovementioned oversupply of information that leads to being unable to find relevant information. The primary concern for the individual user is how to decide that no more relevant information can be acquired and cease searching activity [Ki00]. Hence, according to Jones [Jo04], the *keeping decision* of the individual user is significant since it implies whether information objects are kept, in what form and the way they are maintained and classified in the information repository.

In contrast to information objects that have been intentionally searched for to accomplish a task, individuals often encounter information they consider useful for a future task [MaJo06]. In this case, individual users first need to determine whether to keep the encountered information and often immediately organize it for retrieval. A difficulty arises from not yet maintaining an adequate organizational structure for encountered information objects which are often placed into a 'miscellaneous' folder [JoBr05] or a portfolio [We02]. The current task is hence interrupted due to handling encountered information [MaJo06].

Kirsh [Ki00] observed that individuals apply four *information inventory control strategies* in order to *filter relevant information* for present or future use, that is, to decide which information objects should be maintained and organized on personal devices. Figure 2.11 illustrates how the respective information inventory control strategies affect the information repository on the personal device.

In Figure 2.11, the simplified network of information objects that are already maintained by the individual user are depicted as small squares along with associated classification schemes represented by triangles. Further, the information objects related to the current task context are pointed at by thin arrows. Information objects located at remote information sources and acquired by means of a particular strategy are shown as larger squares outside the personal device.

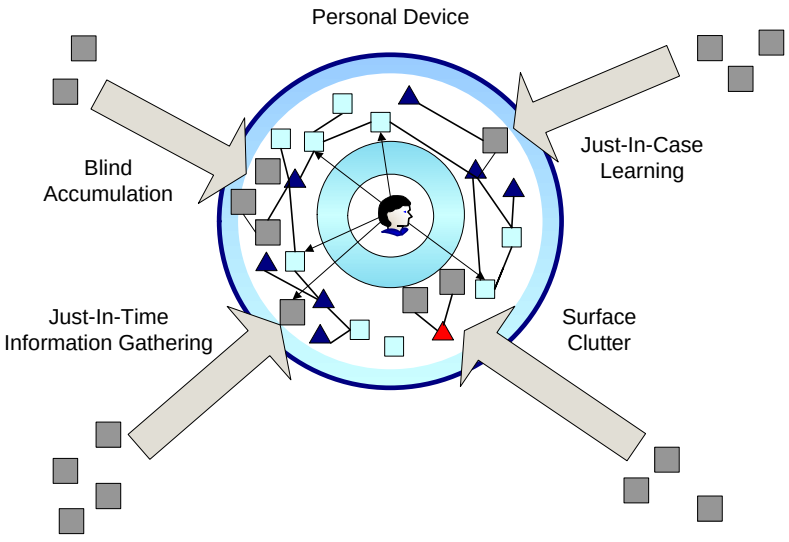


FIGURE 2.11. *Information inventory control strategies based on [Ki00]*

The four information inventory control strategies referring to Kirsh [Ki00] are discussed in the following based on Figure 2.11:

Blind Accumulation: Gathering of all information that might be needed for future tasks (see Figure 2.11, top left). This strategy is rational if the information can later be retrieved when needed, e.g. through immediate application of organizational schemes. Depending merely on search to retrieve information objects is not sufficient according to [JoBr05] since interpreting and organizing information after acquisition are essential for the refinding process.

Just-In-Case Learning: Deliberate accumulation of information that have a certain probability of being useful at a later date. When applying this strategy, individual users spend a considerable amount of time acquiring and organizing information instead of focusing on a task although the value of the acquired information objects cannot be exactly predicted. As shown in Figure 2.11 at the top right, newly acquired information objects are classified based on already existing (or new) organizational schemes.

Surface Clutter: Gathering of information expected to be useful and keeping it easily accessible without sophisticated organizing. A special classification object (e.g. 'miscellaneous' folder, portfolio) can be used for this purpose which is indicated by the emphasized triangle at the bottom right in Figure 2.11. However, this strategy may result in information overload since large amounts of information are not properly organized and just prepared to be used.

Just-In-Time Information Gathering: Ignoring of all information that is not related to the individual user's current task. This strategy is shown at the bottom left of Figure 2.11 by an arrow pointing at the gray square and indicating its relevance for the current task context. Although this strategy ensures high average value for each information object, crucial information may not be found in time and thus the strategy may lead to lack of information for the current task.

Further, trashing strategies also play a significant role in order to eliminate information overload. Disposing information soon after it has been used is typical for the strategy of *Just-In-Time Information Gathering* while individual users who apply *Just-in-case Learning* are less effective trashers [Ki00].

Information Retrieval Strategies

Acquired information needs to be retrieved later by the individual user when required for the completion of the current task. Usually, a combination of organizing strategies comprised of classifying and extending information objects by metadata and linking associated objects together respectively and searching strategies such as browsing over classification schemes, searching by metadata and full-text search over content are applied on the personal device.

Cutrell and Dumais [CuDu06] distinguish three forms of metadata that are essential for the retrieval of personal information. *Inherent properties* of an information object are usually not manipulable by the individual user and are thus read-only, for instance the creation date, size and format of an image or audio file.

By contrast, *user-defined metadata* such as classification schemes and annotations are explicitly added by the individual user in order to facilitate later retrieval of information objects, especially in the case of media content types such as images [SBD06]. The individual user may also add metadata that indicates subjective relevance of the information object – referred to as *relevance feedback* by Marchionini [Ma06] – in order to facilitate the retrieval of relevant information.

Activity-based metadata represent data implicitly collected about the individual user's interaction with information objects like the frequency of access, number of user-defined metadata and for which types of tasks it was retrieved. Teevan [Te07] observed that implicit relevance feedback about the individual user's interaction improves information retrieval (e.g. by providing personalized search results).

Ravasio [Ra04] observed that searching by metadata for information objects is often applied by individual users for information retrieval purposes. In order to refind information, the individual hence has to recall significant characteristics of information objects (e.g. name, author, topic, creation date) and apply search strategies accordingly – referred to as rules by [Ra04]. Otherwise, the required information object eventually cannot be retrieved despite the fact that the individual remembers its existence.

With regard to searching for information objects, individual users are less satisfied when unable to refind information they remember in contrast to finding information in the first place – by accomplishing information acquisition – without targeting at one particular information object [JoBr05]. Since information retrieval usually implies searching for known information objects, precisely specified queries can be applied by the individual user, though [Ma06].

According to Cutrell et al. [CDT06], a significant difference between finding and refinding information lies in the fact that individuals more easily recognize previously found information objects by also recalling task contexts in which they were encountered. Hence, "task contexts can be used to support proactive information gathering" ([CDT06], p. 63).

Concerning the issue of information retrieval when completing a task, a major challenge lies in *information fragmentation* of personal information across applications on a personal device but also across various personal devices [KaJo06]. Personal information that is required to perform the current task may thus be located at several stationary (e.g. workstation at home and at work) and mobile devices.

Thus, individual users need to establish several distinct organizational schemes depending on the applications that manage particular content types [JoBr05]. Information fragmentation also leads to higher costs and difficulties of organizational schemes [Jo04].

According to Boardman ([Bo01] and [Bo02]), hierarchies for managing different content types like folder hierarchies for e-mails, directory structures for textual documents and multimedia files as well as structures for maintaining bookmarks of web pages for example only show slight differences within an individual user's workplace.

Individuals who manage large amounts of information thus develop their own ways of organizing information in order to rapidly retrieve relevant information [Ki00]. Inconsistencies arise from redundant specification of organizational schemes within several applications on personal devices, however. Individuals also tend to maintain multiple copies of information objects at different locations for more efficient information retrieval which may lead to inconsistent versions of information objects. Further, oversupply of information for a task occurs when the classification scheme (e.g. categories and folders) is not effective [Ki00].

Further, the form in which the information is available may not be the one that is required for completing the current task (e.g. inappropriate image compression format, paper instead of digital document). In this case, transformation of information objects is necessary as an additional task-specific step.

Information Acquisition and Retrieval in the Mobile Scenario

According to Mikalsen and Kofod-Petersen [MiKo04], the *user context* plays a significant role for information acquisition and retrieval in the mobile scenario, i.e. on mobile personal devices. The user context is composed of five categories shown in Figure 2.12: the *task context*, *social context*, *personal context*, *spatio-temporal context* and *environmental context*.

The relevance of the task context as described above and the social context indicating the individual's private and professional social roles are not limited to the mobile scenario, but are also significant during information acquisition and retrieval activities using stationary personal devices. The aspect of the user's environmental context referring to available information sources has also impact on the individual user's PIM activities and may differ depending on the personal device at hand.

Personal tasks in the mobile scenario may extend those accomplished on personal devices (e.g. acquiring information on a sight in a foreign city) due to significance of the spatio-temporal context including movement, however.

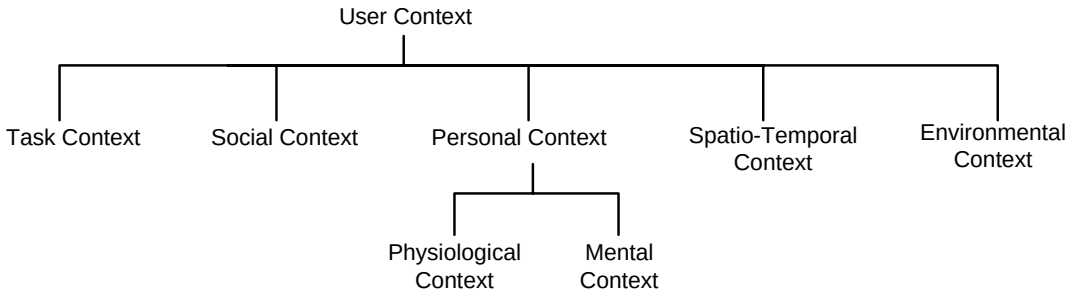


FIGURE 2.12. *Categories of user context (source: [MiKo04], p. 4)*

The personal context that refers to the mental and physical state the individual user is in when completing a task is as a psychological aspect and thus beyond the scope of the model presented within this thesis.

2.2.4 Requirements for Personal Information Management Support

Individuals benefit from personal information management support by adequate tools since valuable resources indicated by the individual user's time, energy, attention, and money are better used for the accomplishment of important tasks [JoBr05].

As early as 1945, Bush [Bu45] recognized that individuals need support in managing increasing quantities of information and characterized a memex as "a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility." ([Bu45], p. 43). The memex is thus considered as the first concept for an adequate personal information management tool providing fast and flexible access to personal information that can be annotated and linked together. Stored items are selected associatively rather than index-based which is adapted to the capabilities of the human mind.

Today, workplace environments should provide a means for appropriate personal information management by facilitating the abovementioned processes of information acquisition, organization, retrieval and dissemination.

Requirements for unified PIM support and usability issues of well-designed workplace environments are discussed in the following subsections.

Requirements for Unified PIM Support

A unification of PIM support – instead of several specialized tools that can only handle particular content types (see Section 2.1.2) and support selected PIM processes – is essential from the individual user's perspective in order to achieve efficient accomplishment of tasks [KaJo06].

According to [KaJo06] and [De03], the unification of information across physical locations, that is, several personal devices, applications and web information sources is significant in order to address the issue of information fragmentation. Thus, support for the management of multiple distributed versions of information objects which represent the same semantic entity by propagating updates is required.

Further, unification of the management of various content types (see Section 2.1.2) is necessary to avoid switching between specialized tools within a single task context. This includes on the one hand unified access, manipulation (i.e. editing functions of specialized tools) and integrative views of arbitrary content types [KaJo06]. On the other hand, services for information acquisition (searching, browsing and keeping information objects), organization (i.e. classification, flexible metadata management and linking of information objects across applications) and retrieval (searching and refinding information objects) need to be integrated in a unified PIM supporting environment.

Support of information organization is essential in order to enhance personal information objects and prepare them appropriately for retrieval and reuse for future tasks while the main challenge lies in combining automatic and human organization [JoBr05].

Retrieving information through user-defined classification schemes is a frequently applied strategy as explained in Section 2.2.3. Further, personal information can be extended by metadata like annotations, marking of relevance and additional structured data. By means of link management individuals are able to link together arbitrary types and distributed information objects by various user-defined relations – or remove extraneous relations respectively – in order to refind related and similar information later.

According to [JoBr05], the aforementioned enhancements to personal information should be applied to a single information object or clustered collections of information objects of specified characteristics either manually by the individual user or (semi-)automatically within the workplace environment. Jones and Bruce [JoBr05] argue that added metadata and links need to be optional and without affecting original metadata, yet searchable, and available with the respective information object regardless of relocating it.

Functional and Usability Requirements for Workplace Environments

Considering functional aspects, well-designed workplace environments postulated by Kirsh [Ki00] represent adequate personal information management tools since they facilitate the accomplishment of tasks and also provide unified PIM support. Since personal information management encompasses auxiliary activities with the objective of later information retrieval required for task completion, tools

for PIM support should focus on individual users' contexts and tasks. Integration of task-specific steps can thus be achieved by integration of specialized PIM tools (see Section 2.3.1) in the workplace environment.

With regard to usability issues, a workplace environment should conform to the objectives of the ISO 9241-11 standard specified in terms of *effectiveness*, *efficiency* and *user satisfaction* [Da06].

	High-Level Requirement	**Requirements for Realization**
(R_1)	Unified PIM support	• **Integrated services for search, classification, flexible metadata management, and linking of arbitrary types of information objects**
(R_2)	Effectiveness	• Providing relevant and trustworthy information for task completion
(R_3)	Efficiency	• **Efficient information retrieval** • Seamless task switching • Information timing • Selection of reasonable services for steps of a task • Efficient user interface
(R_4)	User satisfaction	• **Intuitive user interface** • **Support of information inventory control strategies**

TABLE 2.3. *Functional and usability requirements for adequate workplace environments*

Concerning effectiveness, the workplace environment is responsible for enabling the completion of all steps of a task by providing *relevant* and *trustworthy* information. The aspect of relevance implies that both information overload and a lack of information need to be prevented. Subjective importance of suggested information and the task context in which it was previously used by the individual should also be taken into consideration [BBN06].

The aspect of trustworthiness denotes that it is essential for the individual user to make a distinction between information representing *factual data* or *subjective opinions* [Wi88]. Thus, it heavily depends on the user's social contexts whether an information source is considered reliable (e.g. author is known by the individual user).

The issue of efficiency is on the one hand achieved by enhanced information retrieval from the perspective of PIM support. On the other hand, multitasking (see Figure 2.10) and seamless task switching are significant while the costs of exiting and reentering a task as well as the number of steps required to complete a task should be minimized [Ki00]. Further, *information timing* by suggesting relevant – previously encountered or newly acquired – information objects to the user when it is required for performing a task instead of switching to information retrieval activity and interrupting the current task enhances the individual user's performance [JoBr05].

Efficient task completion is also facilitated by merely presenting services (e.g. search, classification, transformation, modification of information objects) on the user interface of the workplace environment which are reasonable for the current step of the task [Ki00].

Further, an efficient user interface is required for task completion according to Raskin [Ra00] indicated on the one hand by the *information efficiency* of the user interface defined as the minimal

amount of information necessary to achieve a goal related to the amount of information to be supplied by the individual user. On the other hand, an efficient user interface minimizes the individual's learning time and error rate [Ra00] The *GOMS model* introduced by Card et al. [CMN83] can be applied for keystroke-level modeling of GUIs in order to analyze their efficiency along with *Fitt's law* (average time for the user in getting the cursor to a target) and *Hick's law* (average time it takes the user to choose an action from a specified number of alternatives) [Ra00].

User satisfaction results from intuitive interfaces that "help users specify their information needs and understand the results returned" ([JoBr05], p. 35). It is also essential to individually support information acquisition by taking into consideration the several information inventory control strategies [Ki00] as described in Section 2.2.3 while especially the strategies of Just-In-Case Learning and Surface Clutter need to be enabled by adequate mechanisms for information organization. User satisfaction can further be achieved by avoiding cognitive overload caused by an oversupply of information and constant multi-tasking and interruption [Ki00].

The functional and usability requirements for adequate workplace environments as discussed above are summarized in Table 2.3. With regard to workplace environments that merely focus on integration of PIM support only the requirements printed in bold are relevant and considered further on in Section 2.3.2 in the scope of this thesis.

2.3 Supporting Solutions

Personal information management is supported by several tools that pertain to the family of *personal applications*. Such *PIM tools* can be distinguished depending on whether PIM support is their primary objective [Bo04]. The *directory structure* provided by the operating system of the respective personal device underlies all personal applications and facilitates the organization of information objects represented as files of different formats including textual documents, images, audio and video files.

Personal Application	Supported Content Types	Supported PIM Processes
Personal information manager	e-mail contact calendar item task note	acquisition (push) organization (classify) dissemination
Web browser	web page bookmark	acquisition (pull) organization (classify)
Feed reader	news feed	acquisition (push) organization (classify)
Media tool	audio file video file image file	acquisition (push and pull) organization (classify)
Mind mapping tool	concept note relation	organization (classify, link, extend by annotation)

TABLE 2.4. *PIM processes and content types supported by personal applications*

On the one hand, personal applications encompass a large number of special-purpose solutions that merely support the management of particular content types and thus do not fulfill requirement (R_1) of

unified PIM support from Table 2.3. On the other hand, desktop search engines and integrated approaches for workplace environments are available that partly provide unified management of information objects across personal applications.

In the following, solutions for PIM support on stationary personal devices are evaluated against the requirements developed in Section 2.2.4. Mobile counterparts of personal applications are not further distinguished since functional differences increasingly diminish between applications on stationary and mobile devices.

2.3.1 Personal Applications

Individual users apply so-called *personal productivity tools* that facilitate the accomplishment of tasks and/or the management of personal information.

On the one hand, *office tools* like word processing software, spreadsheets and image editing tools primarily focus on accomplishing task-specific steps by providing lifecycle management services such as creating and editing information objects of particular content types.

On the other hand, *personal information managers*, *web browsers*, *feed readers*, *media tools* and *mind mapping tools* establish a category of personal applications that are designated for personal information management support purposes. The aforementioned PIM tools differ in supported information object types and PIM processes, though, as summarized in Table 2.4.

Often desktop and web-based versions – i.e. accessible by means of a web browser – of these personal applications are available (e.g. e-mail client vs. webmail service). A variety of the abovementioned personal applications along with communication tools (see Section 3.2.1) are further provided on online personal desktops such as Desktoptwo [Desk07] and Goowy [Goow07].

Personal Information Managers

Personal information managers focus on supporting information exchange by means of an integrated e-mail client to send and receive e-mail messages with optionally attached files and an address book for managing contacts. Further, time management and the scheduling of personal tasks are facilitated by calendar functionality, notes and the maintenance of task lists while the context (see Section 4.1 for a definition) is given through classified e-mails [WBG06].

The usual supported content types are e-mails, contacts, calendar items, tasks and notes (see Table 2.4). Thus, personal information managers only partly achieve unified PIM support (R_1) by integrating different content types and according services for search and classification.

Widely adopted examples for personal information managers include Microsoft Outlook [Outl07] – depicted in Figure 2.13 – and IBM Lotus Organizer [Lotu07]. In Figure 2.13, a view on the management of e-mails is illustrated including the folder hierarchy, metadata of e-mails in the currently selected folder and details on the selected e-mail. Time and task management are further optionally integrated in this view (see right side of Figure 2.13).

Concerning personal information managers, information acquisition from and dissemination to contacts are supported by means of received and sent e-mail messages respectively. E-mail messages representing pushed information are organized in a hierarchical folder structure and are classified manually by the individual user or by the email-client using user-defined classification rules that determine the destination folder of incoming and outgoing e-mail messages. Further, information retrieval is usually facilitated by search mechanisms.

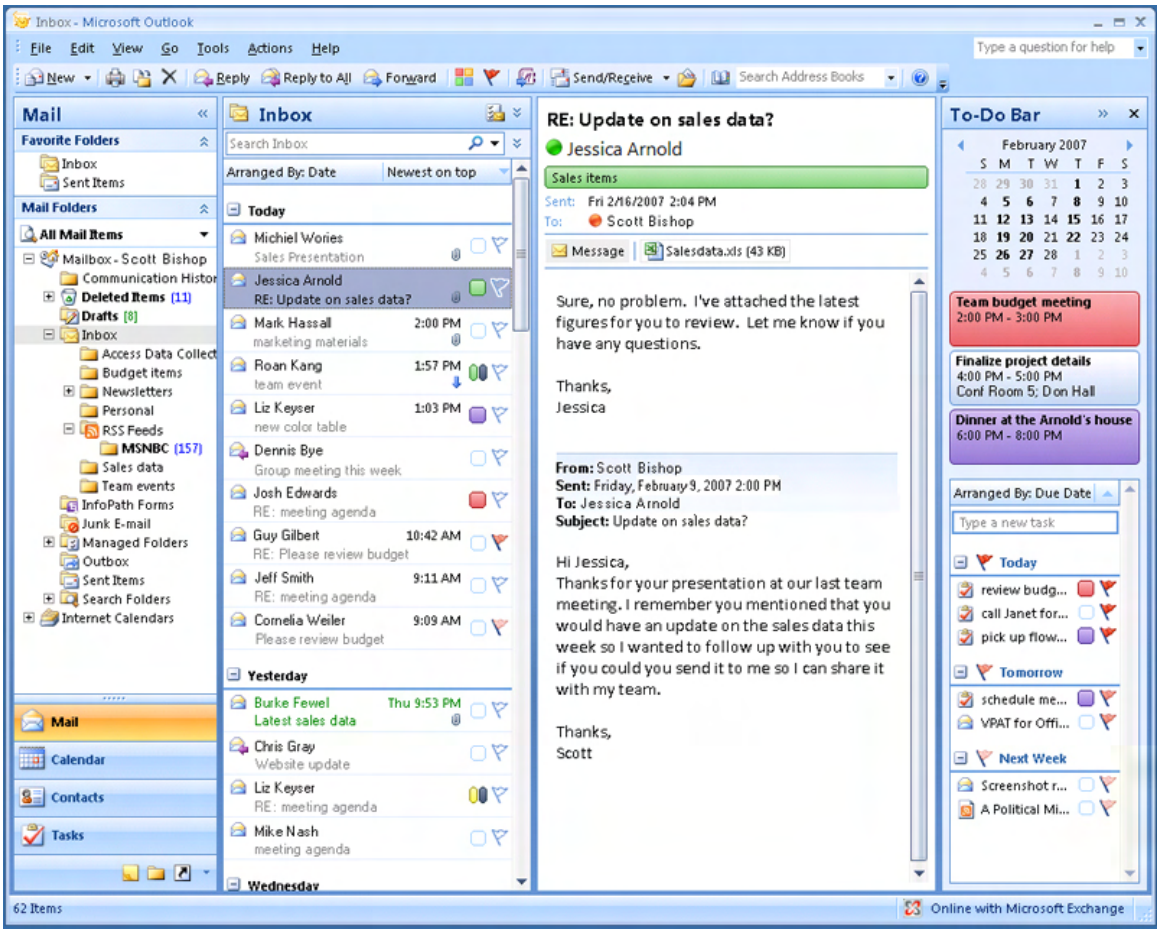


FIGURE 2.13. *Personal information manager (Microsoft Outlook, source: [Outl07])*

PIM Tools for Information Acquisition and Organization

In order to mediate information acquisition from the World Wide Web, individual users apply web browsers like Microsoft Internet Explorer [IE07] and Mozilla Firefox [Mozi07] which are the mostly adopted examples worldwide. These personal applications facilitate visualizing web pages – identified by a URL – and navigating among them by hyperlinks. Information is hence manually pulled by the individual user from static or dynamically generated web pages.

Web pages can either be directly accessed by a known URL in order to obtain the required information. A prevalent practice to acquire information from the web is by using web search engines like Google [Goog07a], however. For media content, a large number of specialized search engines are available (e.g. Picsearch [Pics07] to search images). Instead of navigating to the web page of a search engine, web browsers often integrate search functionality by add-ons.

Further, web browser add-ons (e.g. Mozilla Firefox [Mozi07] extensions) are available to automatically acquire real-time content like news, weather forecasts and stock quotes. A special extension of Mozilla Firefox is the web browser Flock [Floc07] that integrates the individual user's accounts on particular social software platforms (see Chapter 3) for information acquisition and dissemination purposes. A further approach for the realization of a Semantic Web browser that facilitates automatic acquisition of semantically annotated data (i.e. RDF metadata [KlCa04]) from the web is described by Quan and Karger [QuKa04].

With regard to information organization, web browsers usually support the hierarchical organization of *bookmarks* to web pages in a folder structure for later reuse [Bo04].

In contrast to information acquisition mediated by web browsers, feed readers or aggregators like RSSBandit [RSSB07] and RSSOwl [RSSO07] provide the individual user with pushed information. These personal applications aggregate feeds from web sites that consist of a chronologically sorted list of feed items (e.g. news, bookmarks) and are conform to common XML-based formats like RSS [Wi05] and Atom [GrHo06].

By means of a feed reader, the individual user is able to subscribe to feeds on a web site in order to automatically acquire recent information. Similar to bookmarks in a web browser, feeds can be hierarchically organized by feed readers.

Further personal applications are represented by media tools that primarily support the management of media content by folder structures (see Table 2.4). Information objects can further be sorted chronologically to enhance information retrieval. The popular image organizing tool Picasa [Pica07] and iTunes [Itun07] that facilitates the organization and acquisition (i.e. by purchase) of audio and video files pertain to the category of media tools.

Mind Mapping Tools

Mind mapping tools support the management of annotated concepts – often referred to as *thoughts* – and their relations which are not representable in other kinds of personal applications (see Table 2.4). Further, specific content types (e.g. documents) can be attached to concepts. The relations of concepts represent a network of information objects and are visualized as a graph (see Section 2.1.2) in order to establish a two-dimensional map. Popular examples of mind mapping tools include the PersonalBrain [Pers07] – depicted in Figure 2.14 – and VisualMind [Visu07].

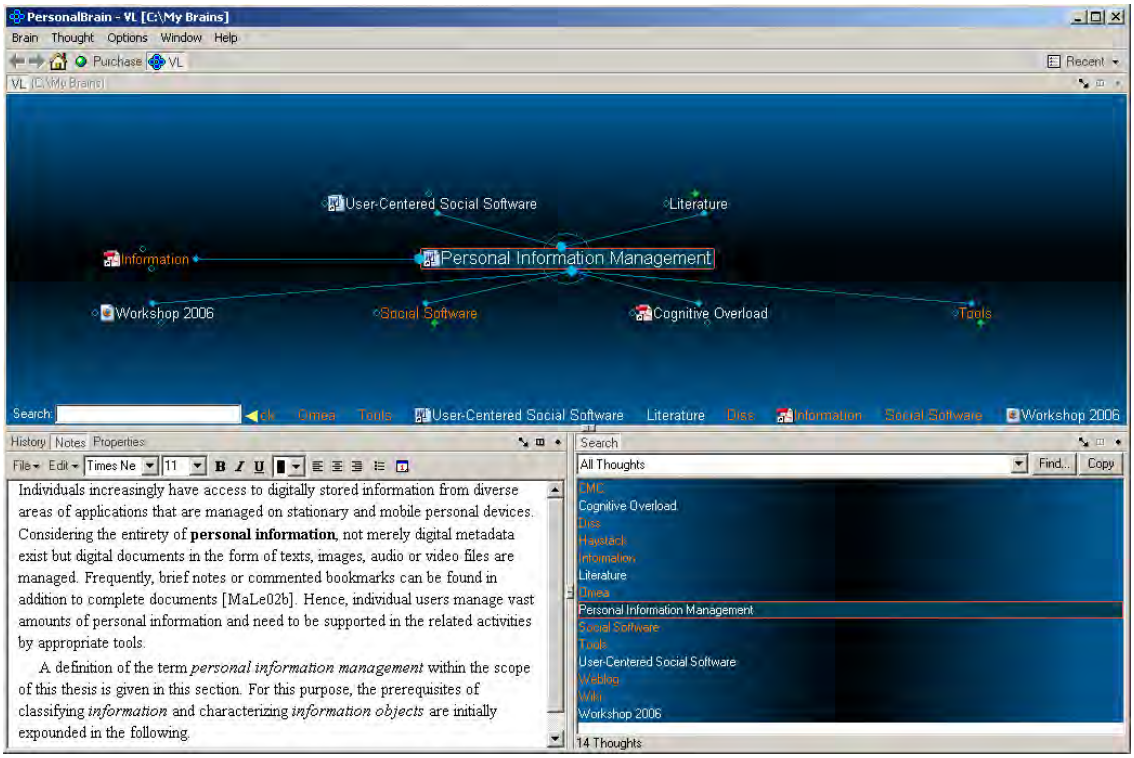


FIGURE 2.14. *Mind mapping tool (PersonalBrain [Pers07])*

The top part of Figure 2.14 illustrates the network of concepts which is adapted to the selected concept ('Personal Information Management' with an attached textual document) to be positioned in the center of the map. At the bottom left, a note as formatted text is associated with the concept. Thoughts can further be searched for by metadata (bottom right of Figure 2.14).

Relations of different kinds (e.g. generalization, cross-reference) among concepts can be specified in contrast to conventional, hierarchical classification schemes of personal applications. Mind mapping tools on the one hand enhance information organization by enabling linking together concepts in a way that reflects the individual user's thinking. On the other hand, information retrieval is also improved from the individual's perspective by navigating in the graph.

A related approach for a map-based information organization and retrieval is described in [Mo04] where a two-dimensional map is integrated into an information portal in order to visualize directory content by topic such as the distance of two documents indicates their similarity of content. In this case, the map is automatically generated, though.

Since each personal application focuses on special content types and distinct organizational schemes there is a lack of unified PIM support on the personal device indicated by requirement (R_1) in Table 2.3. The following section introduces solutions for unified personal information management support in order to address the limitations of common personal applications.

2.3.2 Approaches to Unified PIM Support

Several approaches to unified PIM support exist comprising both research prototypes and commercial applications. On the one hand, tools exist that unify a particular PIM activity across multiple content types and formats. Desktop search engines are devoted to this purpose, for instance, by unifying search functionality in order to enhance information acquisition and retrieval.

On the other hand, workplace approaches address the issue of a more extensive unified PIM support by integrating different services to support PIM processes across personal applications.

Unified Approaches for Information Acquisition and Retrieval

Unified information acquisition and retrieval can be achieved by integrating search functionality across personal applications by means of *desktop search engines*. The repositories of several personal applications which manage a variety of content types according to Section 2.3.1 are indexed for these purposes.

Dumais et al. [Du03] introduce a search-based prototype system termed Stuff I've Seen (SIS) in order to unify retrieval of personal information. In contrast to web search engines for information acquisition that apply rank-based ordering of the search result, SIS provides sorting of items in the search result based on the individual user's personal context (e.g. time-based and author-based ordering) which proved to be more appropriate with regard to personal information according to a user study [Du03].

A further approach for enhanced information retrieval by search mechanisms is the organizational structure of *hierarchical faceted categories* (*HFC*) described by Hearst [He06]. In contrast to grouping of items in the search result by clustering based on distinct topics, a set of category hierarchies corresponding to different semantic aspects of a domain termed facets are established semi-

automatically. Thus, items of the search result are associated to multiple categories and can be browsed more efficiently by the individual user.

Popular examples of commercial desktop search engines include Copernic Desktop Search [Cope07] (see Figure 2.15), Google Desktop Search [Goog07c], and Yahoo Desktop Search [Yaho07].

Desktop search engines are often integrated with web search engines to simultaneously facilitate information retrieval from repositories of personal applications and the acquisition of new information from the web if not yet maintained on the personal device.

Figure 2.15 illustrates a desktop search engine where the result of keyword search by metadata and content may encompass several content types. The number of items of information objects in the search result is indicated and can additionally be filtered by type, though. Below the search result a preview of the selected information object is shown that provides more efficient evaluation whether an item in the search result is relevant to current needs.

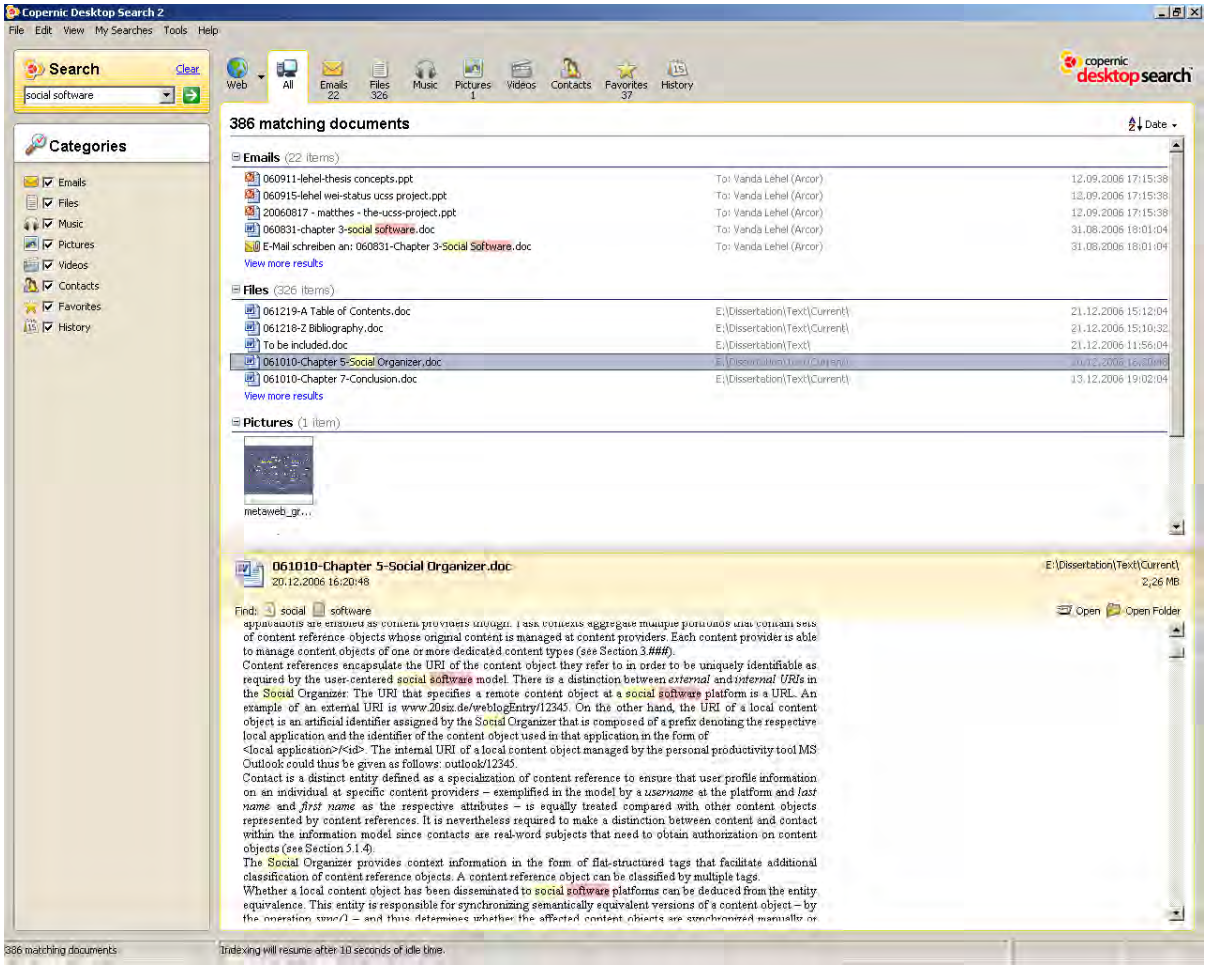


FIGURE 2.15. *Desktop search engine (Copernic Desktop Search [Cope07])*

Efficiency of information retrieval is enhanced by means of desktop search engines from the individual user's perspective since merely one personal application is used to obtain the results, and the user is able to dedicate more time to the current task. Although desktop search engines support significant types of personal information, unification of PIM support is limited to search functionality for information acquisition and retrieval.

Unified Approaches for Information Organization

Several solutions for unified information organization have been proposed in relevant literature as opposed to traditional hierarchical classification of personal information in order to ameliorate information retrieval and reuse for the accomplishment of tasks.

Fertig et al. [FFG96] suggest the replacement of the directory structure by dynamical organization of personal information using a time-ordered stream of documents. The according system described in [FrGe96] is denoted as Lifestreams and provides a linear chronological stream of heterogeneous information objects like personal documents and e–mails such as most recent information objects appear on top of the desktop. Newly created or acquired information objects are dynamically sorted within the stream. The individual user is able to further define susbstreams that are established according to specified search criteria and result in dynamic collections of documents.

By contrast, ContactMap proposed by Nardi et al. [Na02] focuses on the visualization of the individual user's contacts on a map and facilitates grouping them according to social contexts (e.g. current and previous co-workers) which also reflect personal relationships among contacts. Personal information like documents, e-mails and web pages are organized by associating them to contacts and can be thus retrieved contact-based. Further, communication services (e.g. e-mail services, instant messaging services, see Section 3.2.1) for information exchange with contacts are also integrated in ContactMap.

WebTop introduced by Wolber et al. [WKR02] establishes a network of personal documents and bookmarked web pages based on direct links (see Section 2.1.2) which map relations of a document contained in a directory, a bookmark referencing a web page, a hyperlink between two web pages and similar content relations among personal documents and web pages. When viewing a document, the (automatically determined) available relations are presented to enable the individual user to navigate in the network of information objects and retrieve relevant information.

Approaches for unified information organization show a lack of integrating arbitrary content types. Further, comparable to search-based solutions for information acquisition and retrieval described above, only one view on a collection of information objects is available per solution which may not ever represent an efficient information retrieval mechanism and be appropriate for the individual user's needs in any case.

Workplace Environments for PIM Support

The workplace environments presented here are designated for unified PIM support purposes while integrated support for task completion is not considered (see Section 2.2.4).

Figure 2.16 depicts the Haystack universal information client described by Karger et al. [Ka05] which is a workplace environment based on the Eclipse Rich Client Platform [Ecli07]. Haystack aims at individually supporting users in their PIM activities that meet their specific needs. Since each user has different preferences considering significant information objects, associated properties and relations among them, Haystack incorporates a flexible semi-structured data model represented in RDF [KlCa04] to achieve this goal and uses a URI for identifying information objects of arbitrary content types [Ka05].

With regard to information organization, this schema-light approach facilitates flexible metadata management by user-defined properties and supports linking together information objects by arbitrary relations in addition to multiple categories for information objects ([Ka05], [Qu03]). Classification of information objects is also enabled by task-specific collections specified by the individual user.

Further, multiple views for an information object exist that are customizable per content type. Services are provided for information acquisition and dissemination by e-mail and for search-based information retrieval. The network of information objects established by links is navigated by the individual user for finding relevant information by orienteering (see Section 2.2.3).

The view in Figure 2.16 focuses on the individual user's inbox of e-mails and includes the management of collections on the left side and detail views on single information objects, i.e. a contact and an e-mail, on the right side. The context menu associated with the selected contact in the inbox view only offers operations that can be applied to the current object (e.g. annotate, send e-mail or add to address book in the case of a contact) and thus improves the individual user's efficiency in PIM-related activities [Ka05].

Concerning the requirements summarized in Table 2.3, Haystack provides an adequate model for unified PIM support presented on a well-structured and intuitive user interface. Efficient information retrieval is achieved by unified classification, search mechanisms and the alternative of navigating personal information based on links. Further, Haystack meets the requirement of supporting individual information inventory control strategies. Categories for information objects enable the strategy of Just-In-Case Learning while the strategy of Surface Clutter is facilitated by user-defined ad-hoc collections of information objects.

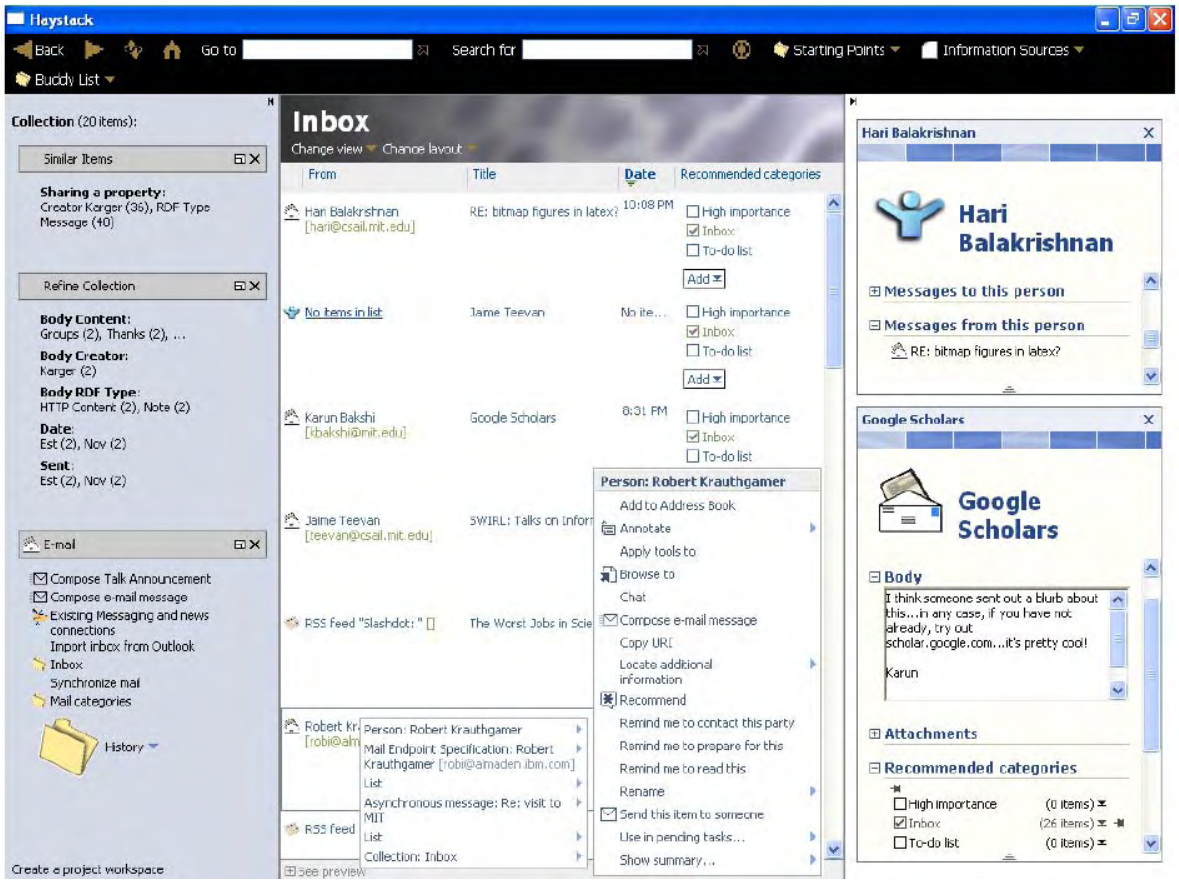


FIGURE 2.16. *Workplace environment for PIM support (Haystack, source: [Ka05], p. 15)*

The Open Source Applications Foundation (OSAF) follows a similar approach by designing a desktop application for PIM support termed Chandler which is under current development [Du07]. Chandler on the one hand aims at providing PIM services comparable to a personal information manager (see

Section 2.3.1) but also integrating the management of documents on the personal device. On the other hand, PIM services should also be extensible to further content types on the personal device and on web platforms [Du07]. So far, Chandler merely provides rudimentary functionality based on a calendar view.

JetBrains Omea Pro [Omea07] is a commercial desktop application for unified PIM support which is also similar to a personal information manager by integrating services for the organization of e-mails, contacts and tasks, but further includes the aggregation of news feeds and the unified management of documents and web bookmarks. Information objects are imported from personal applications like e-mail clients and web browsers for this purpose.

Omea Pro as a workplace environment provides different views for distinct content types with according previews and enhances information retrieval by unified classification and search services within a well-structured user interface. In contrast to the Haystack universal information client discussed above, Omea Pro does not support flexible metadata management and the establishment of links between arbitrary information objects.

Essential characteristics of the aforementioned workplace environments for PIM support are typically unified maintenance, organization and retrieval of information objects across personal applications while Haystack incorporates the most flexible data model. However, individual users are increasingly members of social software platforms representing web communities with the intention of publishing personal information objects and sharing them with contacts within their established online personal networks which is discussed in the subsequent Chapter 3.

Although the introduced environments address the primary concerns of PIM unification by addressing information acquisition, organization and retrieval, the major limitation lies in not sufficiently supporting information acquisition from and dissemination to contacts in the several social contexts of an individual user.

2.4 Summary

Personal information management (PIM) comprises activities for information handling by individual users on personal devices that pertain to the three main processes of information acquisition, organization and dissemination. PIM processes represent auxiliary activities facilitating the primary objective of accomplishing personal tasks.

Relevant information from the individual user’s perspective is usually acquired from web information sources by push or pull mechanisms. Individuals thus need to apply several strategies in order to decide how to keep, organize and retrieve previously found information objects for later reuse that require appropriate tool support.

Essential requirements for unified PIM support based on the individual user’s information handling strategies encompass integrated services for searching, classifying, linking together arbitrary types of information objects and extending them by user-defined metadata. Links among information objects based on binary relations establish a network that facilitates navigating personal information and results in more efficient information retrieval.

The review of supporting solutions for PIM shows that traditional personal applications are special-purpose tools addressing particular content types and thus show a lack of integrated PIM support, while solutions exist that partly integrate suitable services across personal applications. Adequate workplace environments further exist that fulfill the requirements of unified PIM support.

Since individual users increasingly become members of web communities that pertain to the emerging class of social software these solutions are no longer sufficient for adequate PIM support. The aforementioned approaches neither integrate the individual user's personal information maintained and organized on those platforms nor support information acquisition from and dissemination to contacts in the established social networks.

Hence, Chapter 3 is concerned with complementary issues of organizing, publishing and sharing of personal information at social software platforms. The scope of personal information management is thus further extended to online social networks.

Chapter 3

Social Software

This chapter presents the characteristics of different kinds of social software platforms based on representative examples as a background for the approach of user-centered social software and social information management developed within this thesis. Since social software relies on online representation of social relationships and networks fundamentals on these concepts are given in advance. Further, from domain-specific information models and services of social software a generic content provider model and service model are deduced that establish the basis for the user-centered social software model explained in Chapter 4.

3.1 Social Networks

The study of real-world networks which are present in diverse application areas is described by Barabási in [Ba03]. Real-world networks are established from nodes and directed or undirected links between pairs of nodes.

The *degree* of a node indicates the number of links connected to that node in the case of undirected links whereas the concepts of *in-degree* and *out-degree* describe the same characteristic for a node in a network composed of directed links. A *path* in the network determines a sequence of nodes being connected by links. The *distance* between two nodes is defined as the number of nodes in a shortest path connecting these nodes relying on graph-theoretic fundamentals [Ha69].

Hence, the characteristics of real-world networks can be formalized by *random graphs* that emerge from a set of nodes by randomly adding links to them. This early approach to formalizing the characteristics of large real-world networks as *random networks* has first been proposed by Erdős and Rényi in [ErRe59] assuming an equal distribution of links over the nodes.

Studying the structure of the World Wide Web, Barabási and Albert [BaAl99] observed that the underlying random graph of hyperlinked pages follows power law distribution implying different characteristics for the network structure than in the case of random networks. Barabási et al. used the term *scale-free network* in order to denote this kind of large real-world network. Most significantly, scale-free networks consist of a few nodes with comparatively high degree termed *hubs* and a large

number of nodes having a small degree resulting from preferential attachment of links to nodes that already show a large number of in-coming links.

Besides hyperlinked web pages, scale-free networks are present in miscellaneous real-world domains [Ba03]. The types of possible nodes and links in a real-world network vary depending on the application area, however. Significant examples for scale-free networks are social networks (e.g. based on links between individuals within a community who know each other, are co-authoring a publication or act in the same movie), computer networks (e.g. stationary and mobile devices connected over the Internet) and transport networks (e.g. based on air connections between cities).

Within the scope of social software, the study of *social networks* – the term was coined by the anthropologist J. A. Barnes (1954, referred to by Wassermann and Faust [WaFa94]) – is relevant. Nodes in social networks represent *social entities* (e.g. individuals or departments of an enterprise) that are termed *actors*. Links are termed *ties* and represent social relationships of different kinds including interactional, political and economical relationships between pairs of actors. Thus, a social network emerges from the actors and the ties they are linked together with. Social networks involving only actors of the same type are referred to as *one-mode networks* [WaFa94].

Throughout this thesis, *individuals* being community members at social software platforms are considered as actors of the underlying social networks, that is, the focus lies on the special case of *interpersonal networks*. Large interpersonal networks are scale-free since there are particular individuals in the network who are more favored by others to connect to by means of specific social relationships.

In addition, interpersonal networks incorporate the characteristics of the *small-world network* model based on the *small world phenomenon* – also referred to as *six degrees of separation* [Hu01]. This phenomenon has first been observed by Milgram who empirically verified that the average distance between two randomly chosen actors in a large interpersonal network (US American citizens) does not exceed the value six [Mi67]. Further analysis of small world networks is discussed by Watts [Wa99].

Online interpersonal networks established at social software platforms are considered further on from the individual user's standpoint. Such interpersonal networks focusing on one individual, the *focal person*, and his relationships with other individuals are termed *egocentric networks* or *personal networks* respectively (e.g. [Ha96], [WaFa94], and [We99]).

Hill and Dunbar argue in [HiDu03] that the size of personal networks in general does not exceed 150 individuals in accordance with the *Dunbar number* (Dunbar 1993, referred to by Hill and Dunbar [HiDu03]) which predicts the maximal cognitive group size in order to maintain stable relationships based on anthropological considerations. The aforementioned discovery is applicable for online personal networks which are significant in the scope of this thesis in order to predict the number of contacts an individual is interacting with for information exchange purposes in different social contexts facilitated by social software platforms (see Section 3.2).

3.1.1 Social Network Analysis

Social network analysis (SNA) is an interdisciplinary research area that focuses on the analysis of patterns in relationships among social entities where regular patterns in relationships are referred to as structure [WaFa94]. A historical outline on the development of social network analysis is provided by Freeman [Fr04].

SNA represents an alternative research method in social sciences with primary focus on relationships as the unit of analysis. The main objective is to study the evolution and impact of structures arguing that social entities influence each other by means of being linked together (Wasserman and Faust [WaFa94], Wellman and Berkowitz [WeBe97], Scott [Sc00]). Therefore, SNA comprises available information on relationships among social entities and does not restrict study to attributes of autonomous individual units [WaFa94].

According to Wasserman and Faust, SNA can further be characterized as a "generalization of standard data analytic techniques and applied statistics" ([WaFa94], p. 21) since mathematical models are applied to formalize metaphorical terms – such as social position, popularity or isolation – and to establish the basis for studying social networks. Graph theoretical models on the one hand provide an adequate representation of social networks including related concepts that facilitate describing formal properties of social networks. On the other hand, statistical and probability theory are adopted in order to test propositions about structure in social networks. Finally, algebraic models provide a means to study combinations of relationships in a social network such as 'friend of a friend'.

The unit of analysis in the scope of SNA is usually a *dyad* composed of a pair of actors and the tie(s) by which they are linked together whereas for some application scenarios *triads* encompassing three actors and the respective ties are more appropriate (e.g. study of ties representing transitive relations, see below). Accordingly, a *subgroup* is defined as the subset of a certain number of actors and all ties among them. The concept of *group* refers to a finite set of actors on which network analysis is performed. Concerning ties, collections of ties between actors being of a specific kind are defined as a *relation* with determined basic algebraic properties like reflexivity, transitivity, symmetry, asymmetry and antisymmetry [WaFa94].

Hence, in the context of SNA a social network is composed of at least one group of actors and particular relation(s) defined on them that are relevant to the respective measurement objectives [WaFa94]. This definition implies that networks of social networks can also be subject to analysis as described by Garton et al. [GHW97].

Sociograms

The fundamentals of SNA originate from sociometry founded by the social psychologist J. L. Moreno ([Mo37] and further discussed in [Mo53]) who first studied interpersonal relationships in small groups [Fr04].

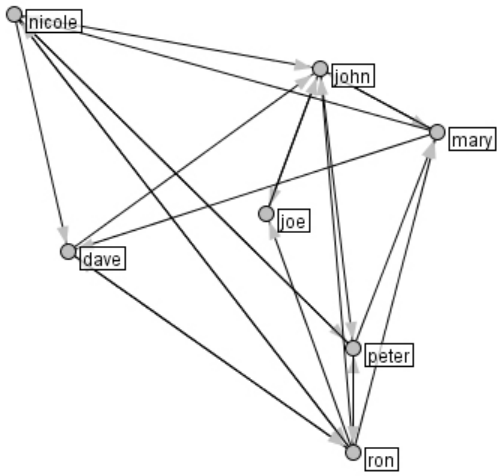


FIGURE 3.1. *Example for a sociogram (created with NetVis [Cu07])*

The visualization of social relationships in interpersonal networks by a sociogram also traces back to Moreno [Mo37]. This visualization technique for graphs is feasible in the case of small social networks while larger networks are usually represented by data matrices, however [Ha96]. Sociograms intuitively visualize characteristics of social networks corresponding to formal measurement results.

In a sociogram, vertices represent actors and edges represent ties between them. An example of a two-dimensionally visualized sociogram is shown in Figure 3.1 illustrating social relationships among a group of seven individuals in an interpersonal network while the respective ties are indicated by directed arrows.

Originally, sociograms were drawn by hand. Resulting from technological development, several alternatives of computer-generated two-dimensional and three-dimensional illustrations of sociograms exist that are outlined by Freeman [Fr00].

3.1.2 Interpersonal Relationships

Personal networks emphasize an individual's different social environments or social contexts respectively [MiGa04]. Interpersonal relationships originating from social contexts and represented by ties between individuals are subject to analysis in interpersonal and thus personal networks.

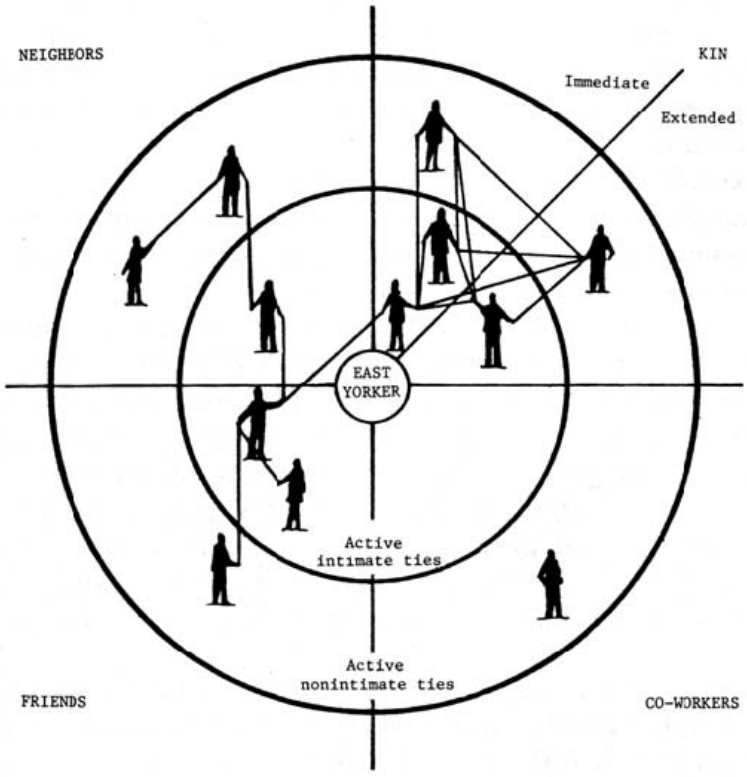


FIGURE 3.2. *Example of a typical personal network (source: [WeBe97], p. 27)*

Wellman [We99] examined the typical structure of a personal network as shown in the diagram in Figure 3.2 exemplified by the personal network of a North American (from East York, Canada). The focal person is depicted in the center of the diagram as 'East Yorker'. By definition of the personal network, the focal person is directly tied to each individual who is a member of the network; hence, the representation of these ties is omitted from the diagram [We99].

The diagram in Figure 3.2 further shows that members of the given personal network are also tied together among themselves. For example, the focal person's relatives are highly interconnected within the personal network. By contrast, the co-worker illustrated in the diagram is isolated since he has no connections to any other individual in the network – "his isolation reflecting a separation of work and social life in this focal person's life" ([We99], p. 19).

Hence, ties in a personal network may represent *role-based relationships* in particular social contexts including close relationships such as kinship (immediate and extended relatives), friendship, affiliation (e.g. neighbors, members in the same club) and formal relationships (e.g. co-workers) pointed out in Figure 3.2. Mitchell [Mi69] specifies the characteristics of *frequency*, *intensity* and *durability* that describe the quality of such interpersonal relationships and are also subject to analysis in interpersonal networks.

Active ties between actors are further characterized by *social interaction* including behavioral interaction (e.g. written communication), exchange of resources (e.g. information) and evaluation of other individuals (e.g. feedback).

The aforementioned typical examples for social relationships from Wellman [We99] and Wassermann and Faust [WaFa94] are relevant for the application scenarios of user-centered social software this thesis is concerned with (see Chapter 4). Information exchange relationships between individuals are primarily focused concerning the use cases of social information management, however. As explained in Chapter 2, different kinds of information are available to the individual and may be exchanged with others in various social contexts.

Information Exchange Relationships

According to Haythornthwaite [Ha96] interpersonal relationships based on the exchange of resources including information can be characterized by the three attributes content, direction and strength.

Content representing the kind of resource is a significant characteristic since "relationships can cover the sharing, delivery, or exchange of a wide variety of resources, including information" ([Ha96], p. 326). In the scope of this thesis, information is considered as content.

The direction of information exchange relationships describes the way information is transferred between pairs of individuals. The underlying relation can either be *undirected* – also referred to as *symmetric* – if the considered information flows in both directions. Otherwise the relationship is referred to as *directed* or *asymmetric* respectively. The sociogram in Figure 3.1 shows directed relationships between individuals while the semantics of the respective ties may for instance indicate which actors have sent a message to other individuals in the group. Further examples of directed and undirected information exchange relationships are given in [Ha96].

Strength indicates the intensity of a relationship, that is, a relationship with frequent transfer of information is considered stronger than a relationship where information is rarely exchanged. Further, *tie strength* refers to the cumulative strength of all social relationships between two individuals which is also affected by the continuity of the relationships [Ha96].

A close friendship with frequent information exchange denotes a *strong tie* implying trust between the respective individuals for example. By contrast, a loose contact in general implies a *weak tie*. With regard to the personal network shown in Figure 3.2, active intimate ties represent strong ties between a member of the network and the focal person while active nonintimate ties constitute weak ties.

Granovetter observed that individuals who are linked to an individual by strong ties are likely to share a large number of strong ties to other individuals [Gr73]. Concerning information exchange

relationships, individuals linked by strong ties to the focal person tend to have access to the same information and thus do not represent sources for new information. Hence, Granovetter [Gr73] found that weak ties are essential in order to establish connections between different interpersonal networks and thus obtain new information. Due to the small world phenomenon, though, receiving information indirectly over more than two steps is insignificant.

3.1.3 Information Exchange in Social Networks

The measurement techniques of social network analysis (see Section 3.1.1) can be applied to examine the structure of information exchange relationships between actors in interpersonal networks and thus reveal whom actors interact with in order to receive and forward information. Exhaustive analysis of patterns in information exchange relationships in the network further shows "who controls, facilitates, or inhibits the flow of information, and who has similar information needs or uses" ([Ha96], p. 339).

Haythornthwaite [Ha96] outlines appropriate measurement techniques for the five primary characteristics *cohesion*, *structural equivalence*, *prominence*, *range* and *brokerage* that should be examined in order to discover how structures in an interpersonal network have impact on information exchange.

The arrangement of the network (e.g. large number of clusters or centralized graph [Ha96]) measured by *cohesion* affects how easily information can be exchanged among actors. *Cohesion* in a social network describes the extent to which actors are interconnected and are likely to have access to the same information. Cohesion is indicated by the *density* of the network, that is, the number of links present related to the number of possible links. Highly interconnected actors in an area of the social network form a *cluster*; in case the actors are fully interconnected and can directly interact with one another a *clique* is established. *Centralization* of the whole network measures whether actors are organized at a central point.

Structural equivalence refers to actors who have "identical ties to and from all other actors in the network" ([WaFa94], p. 356). With regard to information exchange, the analysis of structural equivalence in the social network may lead to identifying actors who have resembling *information needs* or maintain similarly significant roles in information distribution.

Prominence indicates the influence of actors in the social network measured by the *centrality* of an actor. Usually, the actor with the highest degree is considered most central in the network whereas an *isolate* is an actor with a degree of zero, that is, no connections to other actors.

However, centrality can also be measured by examining the *closeness* of an actor which refers to an alternative specification of calculating the shortest path between an actor and every other actor in the network. In this case, the actor with the minimal sum of the calculated distances is considered most central. Concerning information exchange, the *information exposure* of an actor for accessing relevant information and his opportunity of forwarding information is indicated by the actor's centrality.

Brokerage examines whether an actor has the role of an intermediary connecting clusters in the network which is measured by the actor's *betweenness*. According to Burt ([Bu92]), these actors maintain an essential role in filling *structural holes* between clusters in the social network if these areas of the network have not yet been connected. With regard to information exchange, an actor in the role of a *broker* is able to deliver information from one cluster to another and thus maintains control over the information flow. Nevertheless, an actor needs to be able to legitimate the information he disseminates to others referred to as the aspect of *information legitimation* [Ha96].

Range indicates the number of sources the actor has access to and accordingly depends on the size of an actor's personal network on the one hand and his access to other interpersonal networks by means of bridging ties on the other hand. The range of an actor is determined by direct and indirect ties to other actors in the network. The range implies that an actor has access to more information of different kinds provided that he is connected to a larger number of other actors. However, concerning information legitimation, an individual is most likely to consider the information he receives trustful and using it in case the information originates from an actor connected to him directly by a strong tie [Ha96].

Analyzing information exchange relationships based on measurements of the aforementioned characteristics of social networks further reveals *information routes* where information flows among actors [Ha96].

Hence, study of information exchange in social networks reveals from the individual's perspective whom an individual is interacting with in order to acquire and disseminate information in his various social contexts. In the scope of this thesis, information exchange of individuals based on online personal networks established at social software platforms is relevant and thus subject to further considerations.

3.2 Characterization of Social Software

Social interaction over the World Wide Web as a global medium facilitating interpersonal communication and information exchange increasingly gains importance. Social software represent a significant class of Web 2.0 applications that are mediators for these purposes.

Several kinds of communication services and community platforms pertain to the newly emerging class of social software [LMW05] as illustrated in Figure 3.3. According to Tepper [Te03], the aforementioned development is enhanced by the steep increase of individuals willing to use social software.

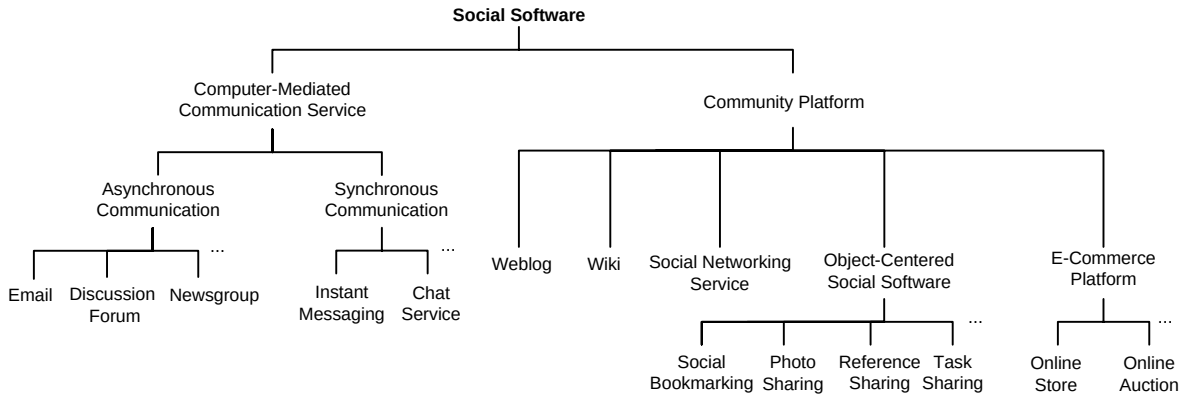


FIGURE 3.3. *Classification of social software*

Widely adopted social software includes discussion forums on different topics (e.g. forums for product support provided by Oracle Technology Network [OTN07]), weblog communities (e.g. Blogger [Blog07]), social networking services (e.g. XING [Xing07]), social bookmarking services (e.g. del.icio.us [Deli07]), reference sharing services (e.g. CiteULike [Cite07]) and photo sharing services (e.g. Flickr [Flic07]).

The most significant distinctive characteristic of social software is the formation of online social networks among their users. The community at a social software platform is thus based on the corresponding interpersonal network emerging from the individual users' personal networks. The social relationships that establish a particular interpersonal network at a social software platform depend on the kind of social software and the provided services, though.

From a social software user's perspective, personal networks are established at each social software platform representing the individual user's different social contexts (see Section 3.1.2) by maintaining a 'Web of Trust'. Members of a personal network are hence other platform users who are tied to the focal person and constitute a particular social context (e.g. private or professional context). Individuals being members of any social context of the focal person are referred to as that person's *contacts* in the following.

On the one hand, an online personal network may represent a *real-world social context*, that is, the individual user is engaging in social interaction with his contacts not merely by means of the respective social software but also directly in the physical world. On the other hand, social software facilitates the formation of *virtual social contexts* without real-world social interaction since the individual user may establish entirely new social relationships online. The structure of personal networks established by means of social software conforms to personal networks underlying real-world interpersonal relationships in social contexts as exemplified by Figure 3.2.

Within personal networks, information acquisition from and dissemination to selected contacts are significant. Social software platforms usually provide services for interpersonal and group communication and additionally support metadata management, information search, publication and sharing, subscription, commenting and collaborative classification in social contexts based on the various underlying interpersonal networks. Hence, individual users can organize, link together and exchange information in the form of content objects (see Chapter 2.1.2) with selected contacts by means of social software.

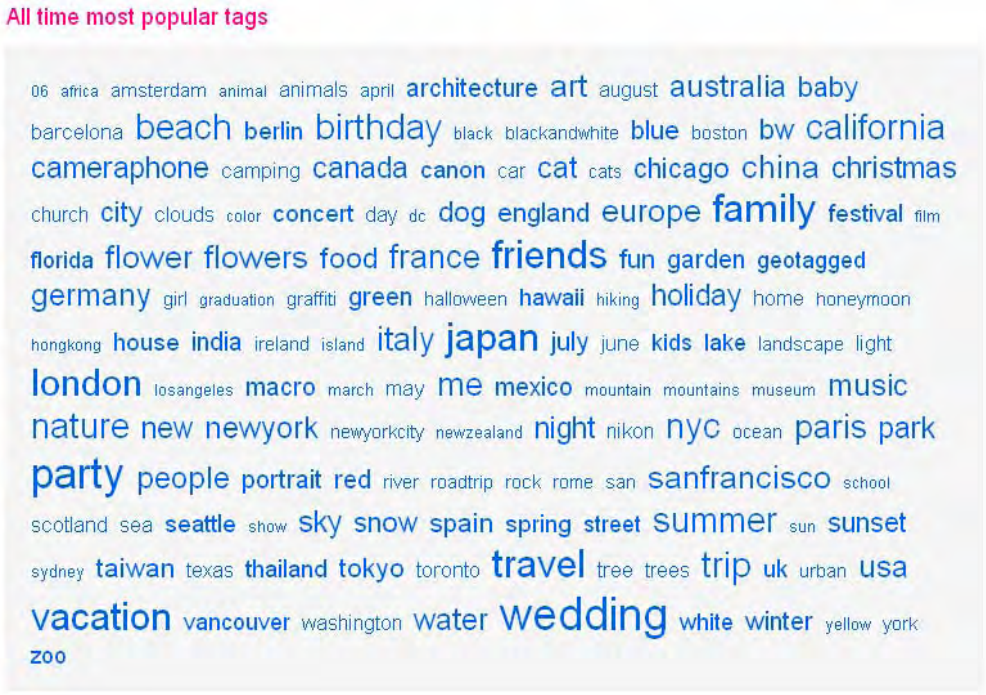


FIGURE 3.4. *Collaborative tagging at a photo sharing service (Flickr [Flic07])*

A content object managed at a specific social software platform usually has a content type while a particular social software platform provides services for the management of selected content types (e.g. bookmarks, references, images or weblogs). For instance, social networking services focus on explicitly establishing online personal networks and thus primarily manage detailed user profile information. By contrast, concerning photo sharing services the main objective is the management, publication and sharing of images.

Collaborative tagging is a common practice at social software platforms. It represents a means of classifying information by freely chosen and shared keywords and is also referred to as *folksonomy* contrasting to traditional formal methods of taxonomic classification but similar to hierarchical faceted categories (HFC) explained in Section 2.3.2. According to Huberman [Hu05], this practice is efficient since previous tags applied by other users have impact on an individual user's decision and thus collaborative tagging results in a stable classification of content objects. Further, social networks are established around shared tags, that is, common interests [BiZa05].

Figure 3.4 demonstrates collaborative tagging at the photo sharing platform Flickr [Flic07] where tags are shown in a flat structured *tag cloud* that is sorted alphabetically. The font size of tags indicates the frequency of the respective tags, i.e. the tags' popularity by the entire user community at that social software platform. For example, Figure 3.4 shows that 'japan' is a popular tag at the Flickr platform which means that users have already tagged a large number of photos by this keyword.

User management including user registration, authentication and authorization services is an essential part of social software. The applied authorization mechanisms rest on the role-based access control model (RBAC) as introduced in [FCK95]. Besides roles that are globally valid throughout a specific social software platform – such as the role of an administrator – two kinds of so-called *user groups* exist that facilitate selective access to content objects managed by an individual user.

Predefined groups are related to general roles in the social contexts of an individual such as *family members*, *friends* and *co-worker*s in correspondence with Figure 3.2. Several social software platforms additionally provide management of *user-defined groups* that are established and labeled by the individual user (e.g. at the social networking service iWiW [Iwiw07]). At social software platforms the individual user may assign contacts to multiple of these user groups if they are users of the respective platform. By means of user groups the individual user is supported in managing access control to content objects throughout his online personal networks.

The application scenarios of social software imply that the individual user establishes several online personal networks at different social software platforms.

Galla (see [Ga04]) addresses this issue by proposing an approach to the management of distributed social relationships established in several personal networks in the context of shared information spaces. Galla's approach can be adopted for social software platforms since they represent specific kinds of web communities.

Mika and Galimberti [MiGa04] further propose a slight extension of such representations of social relationships like FOAF [BrMi05] in order to treat relationships as first-class objects for enhanced automatic relationship management and further integrate concepts from social network analysis.

The following subsections introduce different kinds of communication services and community platforms referring to Figure 3.3 that conform to the abovementioned characteristics of social software.

3.2.1 Computer-Mediated Communication Services

A wide variety of computer-mediated communication (CMC) services pertain to the class of social software since they facilitate online social interaction in the form of interpersonal communication and also effect the formation of personal networks based on the respective communication relationships of the interlocutors.

According to Riva and Galimberti [RiGa98] two distinct kinds of computer-mediated communication services exist. The difference lies in the fact whether individuals are able to communicate simultaneously (synchronous CMC) or merely delayed (asynchronous CMC) by using the respective services (see Figure 3.3).

The most significant communication services that fall into the category of social software are on the one hand e-mail, newsgroups and discussion forums representing asynchronous communication services [Do03]. E-mails represent the origins of social interaction over the Internet supported by e-mail clients or personal information managers respectively according to Section 2.3.1.

On the other hand, chat services and instant messaging services facilitating synchronous communication also pertain to the class of social software [LMW05].

Social interaction via CMC most commonly occurs in textual form implying limitations in expressiveness compared to real-world social interaction since in general only written communication is present “to reproduce the metacommunicative features (emotions, illocutionary force) of face-to-face conversation” ([RiGa98], p. 19).

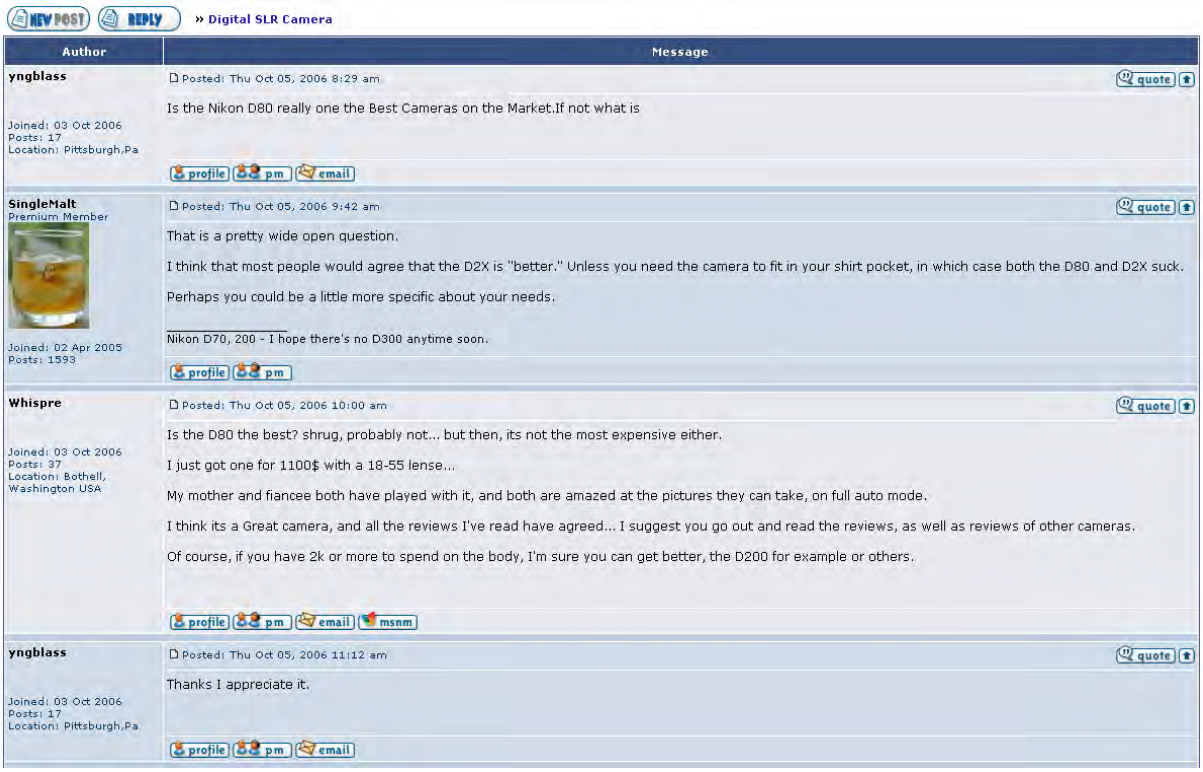


FIGURE 3.5. *Example of a discussion forum (Phototakers [Phot07])*

Riva and Galimberti [RiGa98] introduce three psychosocial aspects that determine the way computer-mediated social interaction is established between individuals. The fact that the network of relationships represents the space for social interaction is denoted by the aspect of *networked reality*

with regard to CMC. *Virtual conversation* is inherent to computer-mediated communication since "interaction no longer has to depend on the physical co-presence of interlocutors" ([RiGa98], p. 5). A further characteristic of computer-mediated communication is the way individuals present their identities, that is *identity construction*, also discussed by Döring [Do03].

In order to use the aforementioned communication services individual users are required to adopt pseudonyms termed nicknames that anonymize their real-world social identities. Individuals often modify their characteristics (e.g. outward appearance) or construct false virtual identities that follow stereotypical behavior patterns in order to gain social acceptance from other users and establish virtual social contexts online ([RiGa98], [Do03]).

Due to anonymity, computer-mediated communication services are often moderated by registered users in the role of a *moderator* who may inhibit other users from engaging in conversation when they infringe upon the conventions of politeness – termed *netiquette* – specified for the respective service [Do03].

Newsgroups, discussion forums, chat and instant messaging services represent social software that can be applied in both professional and private social contexts of individual users depending on the topics they focus on. CMC services are hence often provided by enterprise software on the Intranet of enterprises and organizations in order to facilitate communication among employees, partners and customers.

Newsgroups and Discussion Forums

Newsgroups and discussion forums represent asynchronous communication services usually focusing on different topics that are hierarchically organized [Do03]. Although newsgroups and discussion forums differ in technical realization both provide equivalent functionality: Several discussion threads representing written conversation in a newsgroup or forum are associated with a particular topic. In the example of a discussion forum in Figure 3.5 a thread is illustrated that is composed of a list of messages from individuals who are using nicknames.

Usually, users are able to post textual messages – *posts* or *articles* respectively – to a thread in order to engage in the discussion. However, a distinctive characteristic of this kind of asynchronous communication mechanism lies in the fact that the group of participants consists of a large number of individuals who are only reading the posts whereas the number of individuals making an actual contribution to the conversation is relatively small [Do03]. While newsgroups are publicly available discussion forums are often restricted to registered users as members.

Considering technical realization, newsgroups are repositories provided at news servers in the Usenet system first introduced by Daniel et al. [DET80] and require a client application called *news aggregator* to let their users read articles posted to a newsgroup. Topics in newsgroups are based on the Usenet hierarchy (for instance the newsgroup 'comp.software-eng' on software engineering topics). By contrast, discussion forums are provided at web platforms and can be accessed by means of a web browser (e.g. Oracle Technology Network forums [OTN07] and forums on topics in photography [Phot07] which is shown in Figure 3.5).

Newsgroups and discussion forums are based on the *pull mechanism* which means that members need to fetch the articles or posts themselves by means of a newsreader or visiting the web site respectively instead of being automatically notified about new content (*push mechanism*) [Do03].

Chat Services and Instant Messaging Services

Chat and instant messaging services facilitate synchronous (real-time) interpersonal and group communication.

While chat services mainly support broadcasting of textual messages instant messaging services provide point-to-point (or point-to-multipoint) communication of individuals by means of text or audio messages (e.g. using VoIP [VoIP07]) that may also be accompanied by video transmission for video conferencing purposes.

Chat services provide *chat rooms* that are often designated to specific topics similar to newsgroups and discussion forums. Several registered users can enter a chat room in order to discuss the respective topic in real-time among themselves. By contrast, instant messaging services focus on communication between peers [Do03].

Similar to newsgroups and discussion forums, specific implementations of these services may require each user to run a client application on the personal device as described in [Do03] in contrast to the majority of web-based social software platforms.

A multitude of chat services exist that are realized based on variations of the Internet Relay Chat (IRC) protocol first standardized by [OiRe93]. *Web chat* is offered at web platforms such as 2channel examined by Matsumura et al. [Ma05] which is popular in the Japanese area.

Widely-used examples of instant messaging services with several millions of active users are ICQ [ICQ07], Windows Live Messenger [WLM07] and the Internet telephony service Skype [Skyp07] that all require the use of client-side applications. Further, Trillian [Tril07] is an example for a unifying instant messaging service which integrates decentralized user accounts from other instant messaging services.

Formation of Personal Networks

The abovementioned communication services vary in their popularity, indicated by the number of their members and the frequency of interaction among those. Thus, the popularity has great impact on the size of interpersonal networks that are established at these communication services and show characteristics of the scale-free and small-world network models [RaRa04] (see Section 3.1).

Individual users are supported in managing their contacts in *buddy lists* which represent predefined user groups in the scope of computer-mediated communication services. Favorite members can be specified as contacts by the individual user by adding them to such lists. Further, members can be indicated with whom the individual user does not intend to engage in social interaction at all.

Xiong and Donath [XiDo99] propose a graphical representation for user interaction (i.e. frequency of posts and replies) by means of CMC services that does not reflect relationships among users, though. By contrast, Müller [Mu99] examined the structure of personal networks users establish by applying newsgroup services and chat services.

The result of this study among users in Switzerland shows that the structures of personal networks that are implicitly established by using newsgroup and chat services significantly differ. According to Müller [Mu99], the personal network of a chat user is usually more densely knit, contains less isolates and more probably represents a real-world social context than the personal network established by a newsgroup user.

On the one hand, in case of an asynchronous communication service – exemplified by a newsgroup in [Mu99] – the individual user engages in discussion with several contacts who rarely know each other since they are related to him merely by weak ties within a virtual social context.

On the other hand, the user of a synchronous communication service – exemplified by a chat service in [Mu99] – tends to communicate with contacts from the same virtual (and often also real-world) social context who maintain social relationships among each other since chat users have simultaneous virtual conversation with several users by broadcasting messages in a chat room.

The aforementioned communication services are intended to support peer-to-peer communication scenarios that are beyond the scope of this thesis. Nevertheless, they are often provided as an additional means of communication support at social software platforms that primarily focus on facilitating information publication and sharing.

Hence, the subsequent sections are concerned with the characteristics of popular social software platforms in more detail which are designated for information exchange purposes among their users.

3.2.2 Weblogs

In the following, the main characteristics of weblogs and weblog communities are introduced that fall into the category of social software platforms according to Figure 3.3.

The primary purpose of weblogs or blogs – the term was coined 1997 by Jorn Barger – is facilitating simple publication of content on the web [Ba99].

Blogging is a common practice in order to easily publish and share personal knowledge and information on the web in the form of mainly textual *weblog entries* – or *posts* / *postings* respectively – usually displayed in reverse chronological order on a web site referred to as a *weblog* [LMW05]. Weblog entries are composed using a WYSIWYG editor.

An individual is in general the weblog owner who is the author of and responsible for the published content in weblog entries. Thus, the term weblog usually refers to a *personal weblog* whereas a *teamlog* represents a multi-author weblog for collaboration of several individual users or a group [LMS03]. In fact, users intend to share more personal information if they can have similar control over their published content like if they managed it on their personal device. Hence, weblogs also provide support for personal information management [Ef03].

The content of a weblog often constitutes a personal diary including image, audio and video content embedded in weblog entries besides formatted text. Further, weblog entries are also enriched by annotated hyperlinks to web sites the author encountered while searching the web. Hence, content in weblog entries does not only represent anecdotes or the owner's insight on a specific topic but may also contain remarks and recommendation on content of others that leads to networking effects among individuals.

The popularity of weblogs thus results from the immediate reflection of ownership and the quality content that is found in the weblog entries in spite of the absence of centralized control [Pa03]. Individuals get to know each other by reading each other's insights and develop trust [Ro04].

Weblog entries are commented on by Internet users who provide the author of the weblog entry with feedback by adding a textual comment and optionally rating the published content depending on whether rating of content is provided. The flat list of comments for a weblog entry is similar to a thread in a discussion forum emerging from the contributions of several individuals.

Further, a list of favorite weblogs owned by other individuals is usually managed in a *blogroll* displayed on the weblog [Av06].

The aforementioned weblog-related services foster social networking among weblog owners and individuals – identified by their e-mail address – who merely visit weblogs and occasionally comment on weblog entries. Thus, the personal network of a weblog owner is on the one hand established from individuals providing feedback for his weblog entries. On the other hand, directed links between weblogs also connect the weblog owner to his contacts and results in the formation of an interpersonal network.

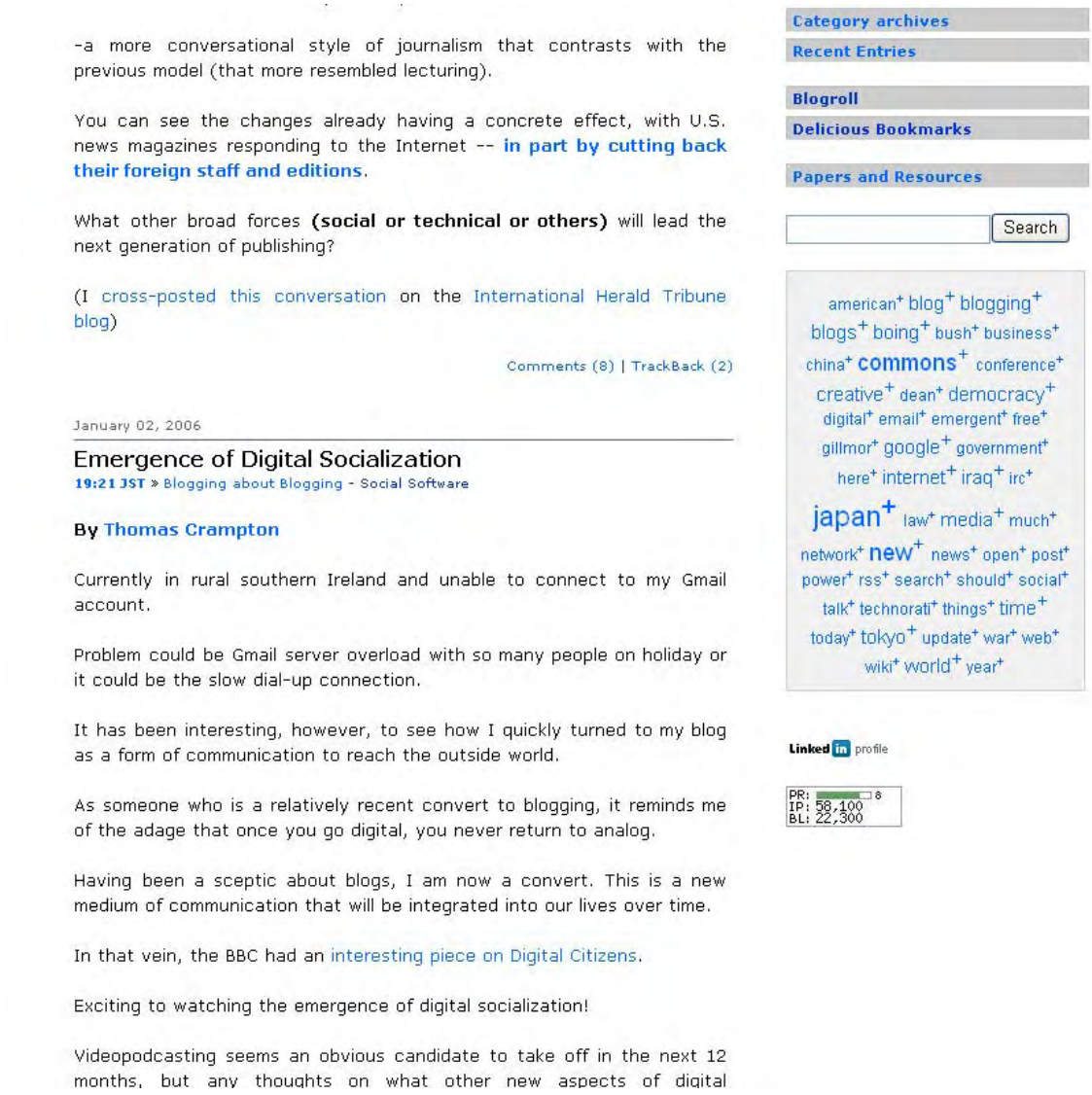


FIGURE 3.6. *High-end weblog (Joi Ito's Web [Ito07])*

A representative example for a weblog is 'Joi Ito's Web' [Ito07] shown in Figure 3.6 which is a high-end weblog providing a wide variety of additional services besides commenting and a blogroll including the management of categories, archiving, tagging of weblog content and displaying remote feedback for a weblog entry. In order to obtain such kind of extensive functionality weblogs are often established using special weblog software like WordPress [Word07].

Weblog Communities and the Blogosphere

A large number of weblogs are not maintained individually at a web site but belong to a public weblog community or are hosted on the Intranet of organizations and enterprises. Examples for international weblog communities are Blogger [Blog07], LiveJournal [Live07], Xanga [Xang07] and 20six [Twen07].

Members of a weblog community can directly link together their content among each other – by managing a blogroll and providing feedback for other users' content.

Further, weblog communities often provide users with role-based content sharing support in the form of predefined groups or additionally user-defined groups in order to define who may read weblog entries and/or provide feedback.

Weblog communities also provide search services over content managed at the community, that is, over weblog entries, feedback and user information. The weblog search platform Technorati [Tech07] unifies these services by offering search for weblog entries by popular tags or specified phrases over a large number of weblogs and weblog communities on the web. Similarly, on PubSub [Pubs07] users are able to subscribe to keywords in order to get notified whenever those appear in recent weblog entries.

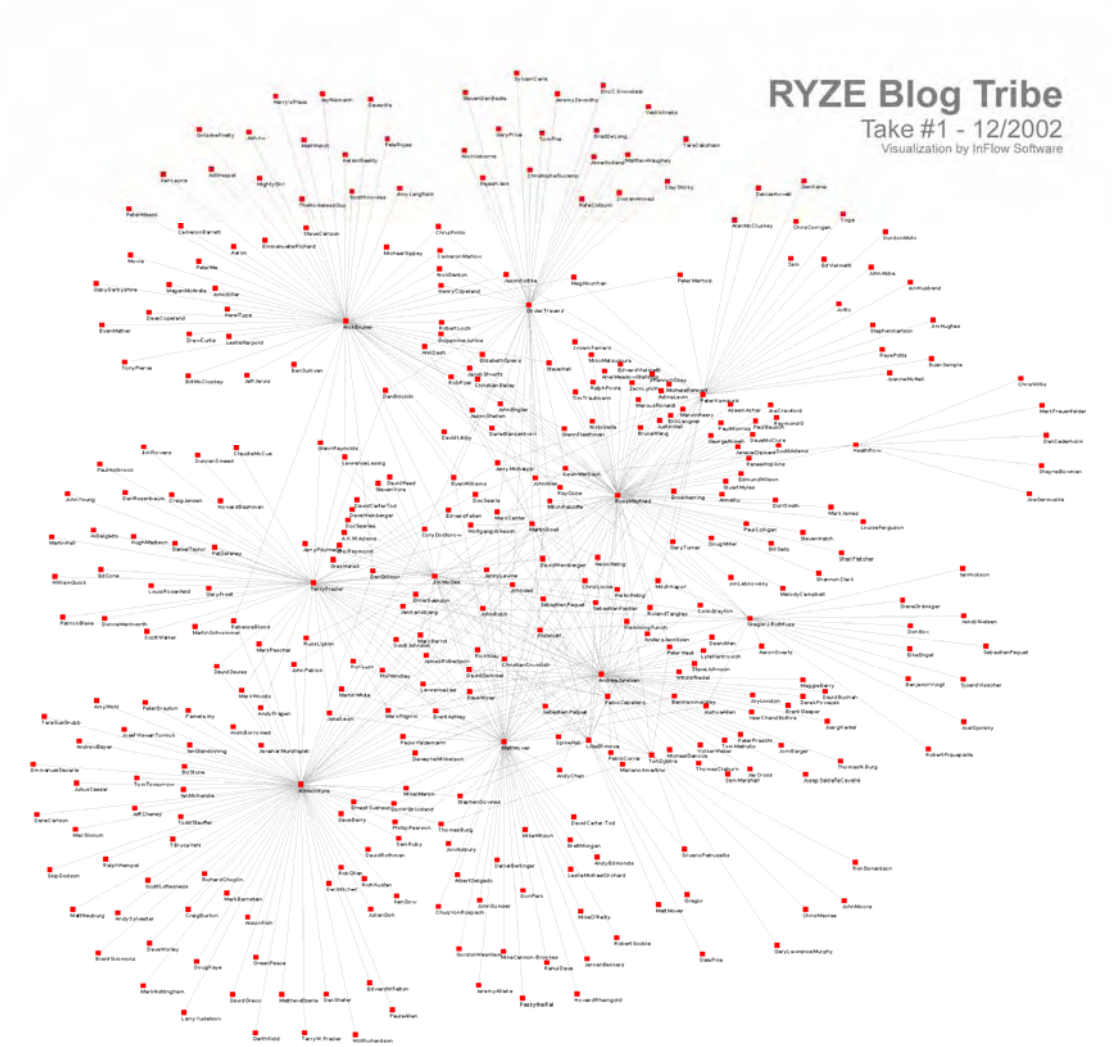


FIGURE 3.7. *Blogroll network of members at the community Ryze [Ryze07] (source: [Ma03])*

In the context of weblogs, the term *blogosphere* denotes the social network that is comprised of all weblogs on the web. Herring et al. [He05] applied SNA methods (see Section 3.1.1) to investigate a sample of randomly-selected weblogs and hence predict the structure of the blogosphere. The analysis showed that parts of the blogosphere were densely interconnected forming clusters and cliques resulting in a small-world network. A corresponding example for the structure of a small part of the blogosphere (weblog owners at the social networking platform Ryze [Ryze07]) is depicted in Figure 3.7.

The popularity of weblogs is indicated by the growth of the blogosphere. Statistics from all weblogs indexed by Technorati [Tech07] show that the number of weblogs has doubled every six months in the last three years [Si06].

Weblog-Related Technologies

Since weblogs represent a widely-adopted form of social software several weblog-related technologies and standards emerged that also became popular in other kinds of social software and are thus outlined in the following.

An essential mechanism for the identification of weblog entries is provided by *permalinks* [Co03] that represent immutable URLs for web resources. By this means, weblog entries are ensured to be available permanently in *chronological archives*.

TrackBack [Trac04] provides a mechanism for the owner of a weblog to be notified about feedback referring to his weblog on external websites (remote feedback) and thus links together related content from different systems. Originally implemented in the weblog publishing software Movable Type [Mova07], TrackBack uses an XML-based representation format including details about the remote comment and a permalink of the referencing weblog entry or feedback.

Further weblog-related technologies foster selective syndication of published content by individuals [LMW03]. Collections of weblog entries and comments in a weblog can be made available as items of RSS feeds. Originally, RSS is an XML-based news syndication format (see [Wi05]]) that can also be used for the exchange of other content types, like weblog entries in this case. Another popular content syndication format for weblogs is Atom, see [NoSa05] for details.

Using *RSS enclosures*, multimedia content can also be embedded in the items in order to enable *podcasting*, the distribution of audio and video files in feeds. Thus, other users can subscribe to the RSS feeds of their favorite weblogs (e.g. using feed readers introduced in Section 2.3.1 or web-based news aggregators like Google Reader [Goog07d]) and remain up-to-date about the latest content by notification mechanisms. For the exchange of hierarchically structured resources like categories in a weblog XML-based formats like *OPML* [OPML00] exist.

The individual user's relationships to authors of referenced weblogs in the blogroll can further be embedded into the HTML page of the weblog by using the microformat *XFN* (XHTML Friends Network, [XFN07], [Micr07]).

In addition, several formats for *geotagging* weblogs exist such as *GeoTags* [GeoT07] and *GeoURL* [GeoU07] in order to add geographical identification metadata to weblogs and thus facilitate detection by location-based search services.

Weblog community systems usually offer Weblog APIs to allow users posting to their weblogs using client tools instead of creating each weblog entry manually by visiting the weblog. Popular examples are the Atom Publishing Protocol (Atom API) [GrHo06] and the Blogger Data API [Blog07a].

3.2.3 Wikis

Wikis as collaborative authoring environments on the web pertain to the category of community platforms as shown in Figure 3.3. A wiki was first introduced by Ward Cunningham, termed the WikiWikiWeb [Wiki07a].

On a wiki platform, individual users incrementally establish a web site and keep the published content up-to-date by directly editing parts of web pages. Editing comprises composing new web pages (termed *articles*) and modifying content on exiting web pages using an editor for special wiki markup [LeCu01].

An essential concept in wikis is networking of content by linking together articles through hyperlinks since they represent web pages usually in a markup language like HTML [Av06]. Further, categories can be used to establish a higher level navigation for the wiki platform.

Each article contains a version history in order that editors are able to see all modifications on an article and to reverse changes if required.

In order to establish a wiki system on a web site, wiki software like the popular free license MediaWiki [Medi07] is required. Figure 3.8 depicts the Microformats Wiki [Micr07] – based on the aforementioned MediaWiki [Medi07] software – that is an essential part of the web site microformats.org used for documentation purposes. In Figure 3.8, an article is shown whose parts can also be edited independently instead of editing the whole web page. The navigation bar on the left side of the wiki article further includes search functionality over the wiki content.

On the one hand, several wikis are operated on the Intranet of enterprises and organizations and only authorized users are able to contribute to those wikis or even read the published articles. On the other hand, public wiki platforms exist where (registered) Internet users are permitted to add and modify content.

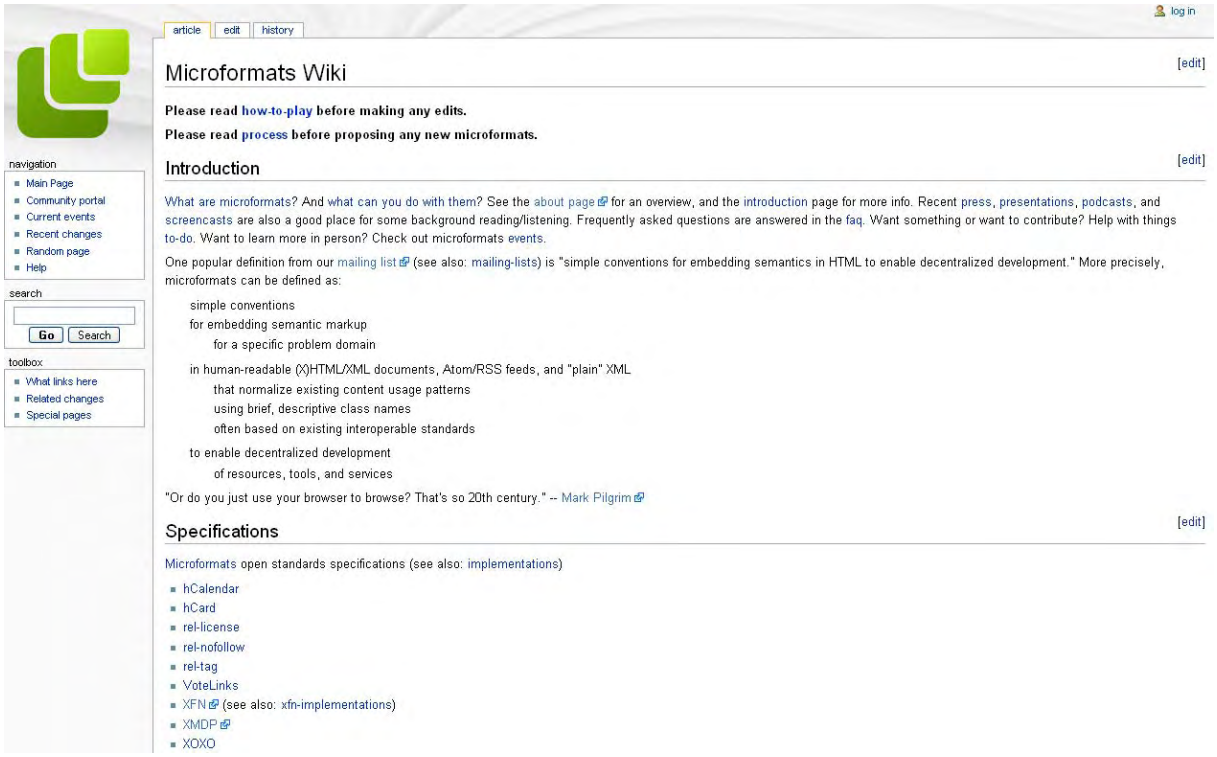


FIGURE 3.8. *Example of a wiki article (Microformats Wiki [Micr07])*

A widely-used example of a publicly available wiki is the multilingual online encyclopedia Wikipedia [Wiki07] that encompasses a large knowledge base and is open to global collaboration. Wikipedia content is provided and revised by a large number of Internet users which ensures that the articles in general comprise trustful information. Additionally, the authorization concept includes the role of an administrator to detain ordinary Wikipedia users from editing selected essential articles (e.g. main page).

Since Wikipedia articles are not machine-readable despite of the wiki's enforced structure, Völkel et al. [Vo06] proposed an extension by Semantic Web technologies – termed Semantic Wikipedia – in order to enhance content retrieval.

Comparable to a wiki, the LEO online dictionary [LEO07] (originally providing German ⇔ English translation) also emerges by collaborative authoring of its users. Although Internet users do not directly edit entries of the dictionary they may either send in terms and expressions they have not yet found in the dictionary or discuss possible translations of such terms with other people in a discussion forum (see Section 3.2.1).

Collaboratively editing a web page leads to online social relationships among wiki editors. With regard to wikis, formation of interpersonal networks is also based on information exchange relationships among individual users. Editors disseminate their knowledge in the form of wiki articles while the majority of users acquire information from those wikis.

Information and knowledge sharing within organizations can also be enhanced by integrating functionality of social software platforms in enterprise software. Since weblogs and wikis provide collaborative authoring services for informal communities of practice they are appropriate as a supportive tool for information and knowledge management purposes [Ro04].

The aforementioned integration of weblogs into enterprise information portals (EIP) is discussed in [LeMa04]. Similar approaches exist to integrate weblogs and wikis into the information management infrastructure of organizations [Av06].

In the scope of this thesis, however, wikis as collaborative authoring environments are omitted from further considerations on social software platforms.

3.2.4 Social Networking Services

Social networking services provide the opportunity to explicitly build up and manage contact relationships between individuals on a community platform [MiGa04]. Therefore they represent a significant class of social software platforms according to Figure 3.3.

Members of a social networking service are registered users who are supported in managing their own user profiles on the platform by adding business and/or private contact information as well as information on their professional career, professional and/or private interests, memberships in organizations and personal photos. Platform members may also control information sharing with other registered users or unregistered web users depending whether user-defined authorization mechanisms are provided by the respective social networking service.

Further, the primary purpose for registered users lies in mapping real-world social relationships on the social networking platform. New contacts are added by either inviting them to the community or navigating to their user profile if existent. Thus, links signify professional or social ties between users.

A further value for members of a social networking service lies in viewing the path of indirect contacts leading to any other community member and thus establishing interesting new social relationships online.

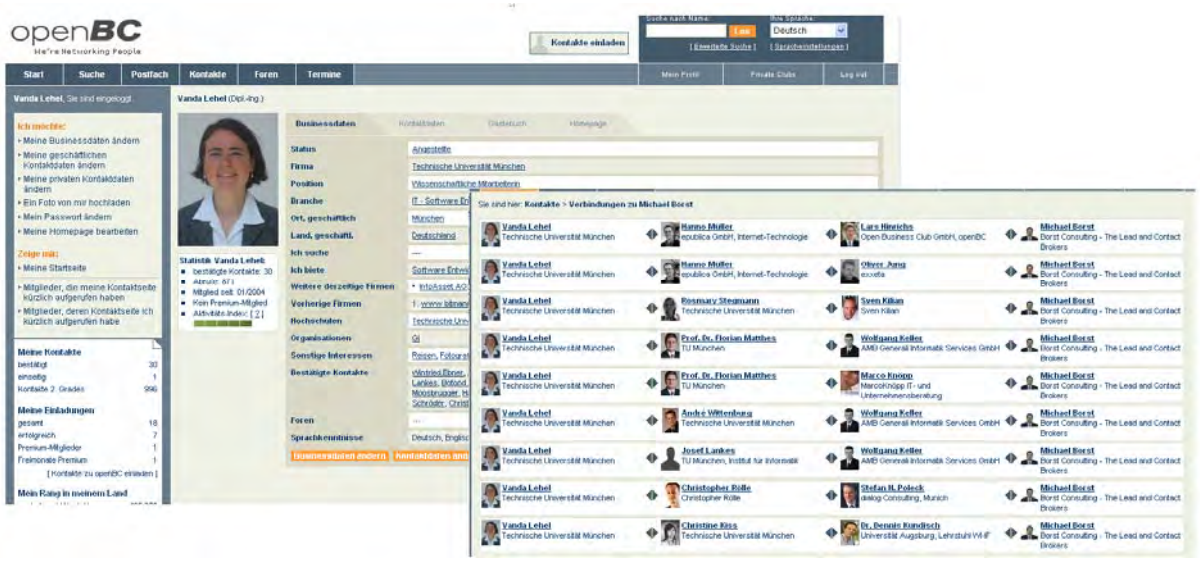


FIGURE 3.9. *Social networking service (XING [Xing07])*

Well-established examples of social networking services are Linkedln [Link07] in the USA and XING [Xing07] in Europe, both widely used for managing business contacts. Further examples for social networking services in the private context (family and friends) are Friendster [Frie07] and MySpace [MySp07] which is the largest community worldwide. Often social networking services can merely be joined by invitation and the number of forwarded invitations per user may also be limited (e.g. iWiW [Iwiw07]).

Figure 3.9 illustrates the primary purpose of social networking services exemplified by the representative platform XING [Xing07]. The figure thus displays a user profile page (Fig. 3.9, background) and existing shortest paths of length three from the current session user to another community member (Fig. 3.9, foreground).

Social networking services often facilitate tagging, i.e. members are enabled to add tags to their own contacts like on the platform XING [Xing07]. Further, social networking services support their members with additional communication services, e.g. simple mailing lists or discussion forums. The social software platform MySpace [MySp07] also offers a webmail service and facilitates blogging besides the objective of social relationship management.

Social network analysis (see Section 3.1.1) applied for social networking services focuses on measuring social relationships and information flows between community members and other information/knowledge processing entities [WaFa94].

The limitation of social networking services as contact management tools becomes apparent if individuals are considered as members of different communities. In this case, information exchange and collaboration between different networks are difficult to achieve. An approach to address this problem is the XML-based FOAF [BrMi05] format to represent relationships between contacts and thereby describe the structure of the social network of an individual.

A further issue lies in the fact that available user profile information is based on rigidly defined metadata depending on the respective social networking service.

3.2.5 Object-Centered Social Software

Object-centered social software (OCSS) represent another class of online community platforms (see Figure 3.3) that focus on allowing users for storing, categorizing and sharing personal collections of content objects. Rheingold [Rh05] coined the term *object-centered social software* in order to denote that these systems group their members around shared content published and managed online. Thus, in contrast to social networking services, social networks (i.e. interpersonal networks) are implicitly established at these community platforms.

Unlike social networking services that are primarily devoted to the management of user profiles, that is contact information, object-centered social software platforms focus on selected content types each besides user profile information on registered users. Content types include bookmarks of web pages (e.g. del.icio.us [Deli07]), bibliographic references (e.g. CiteULike [Cite07]), videos (e.g. YouTube [Yout07]), slides (e.g. SlideShare [Slid07]), files (e.g. Box.net [Boxn07]), news (e.g. Digg [Digg07]) and goals (e.g. 43things [Fort07]). The available metadata for content types depend on the respective community platform, however.

Further, collaborative tagging (see Section 3.2) is a key characteristic concerning object-centered social software. Tags chosen by a member represent shared metadata and thus context information about that user in the community. In Hammond et al. [Ha05] an overview is given about the motivations for tagging: it is essential to consider who is the tag creator and who is the tag user. For example, Flickr [Flic07] users normally manage their collection of photos for private use, while members of community platforms like del.icio.us [Deli07] tend to use collections of references from other users. From these usage patterns of shared context in a community, social networks in different social contexts are built implicitly around these content objects.

Notification of newly created content or feedback from other platform users for an individual user's content is often provided by object-centered social software. For this purpose, in general Atom [NoSa05] or RSS [Wi05] feeds are offered for subscription like in the case of weblogs. Other weblog-related technologies like TrackBack [Trac04] for tracking remote feedback are also supported by such community platforms (see Section 3.2.2).

For administrative purposes, object-centered social software platforms also offer proprietary APIs similar to APIs provided by weblog communities (see Section 3.2.2) like the Flickr API [Flic07a].

In the following, two categories of object-centered social software are distinguished depending upon whether a reference to content located elsewhere (with extended metadata) or full content is managed at that social software platform.

Social Bookmarking Services

Social bookmarking services like del.icio.us [Deli07] (see Figure 3.10), Furl [Furl07], Linkroll [Link07a] and Simpy [Simp07] facilitate managing online collections of bookmarked web sites thus replacing local browser bookmarking functionality introduced in Section 2.3.1.

As shown by a representative example in Figure 3.10, collections of bookmarks are displayed on web pages of social bookmarking services including the user who bookmarked the page, his description, all associated tags and the date of reference.

The number of users adding a bookmark to their collection determines its popularity at the social bookmarking service. In Figure 3.10, the popularity of a bookmark is indicated by showing the

number of members who added that bookmark highlighted in different shades of red corresponding to the number of references.

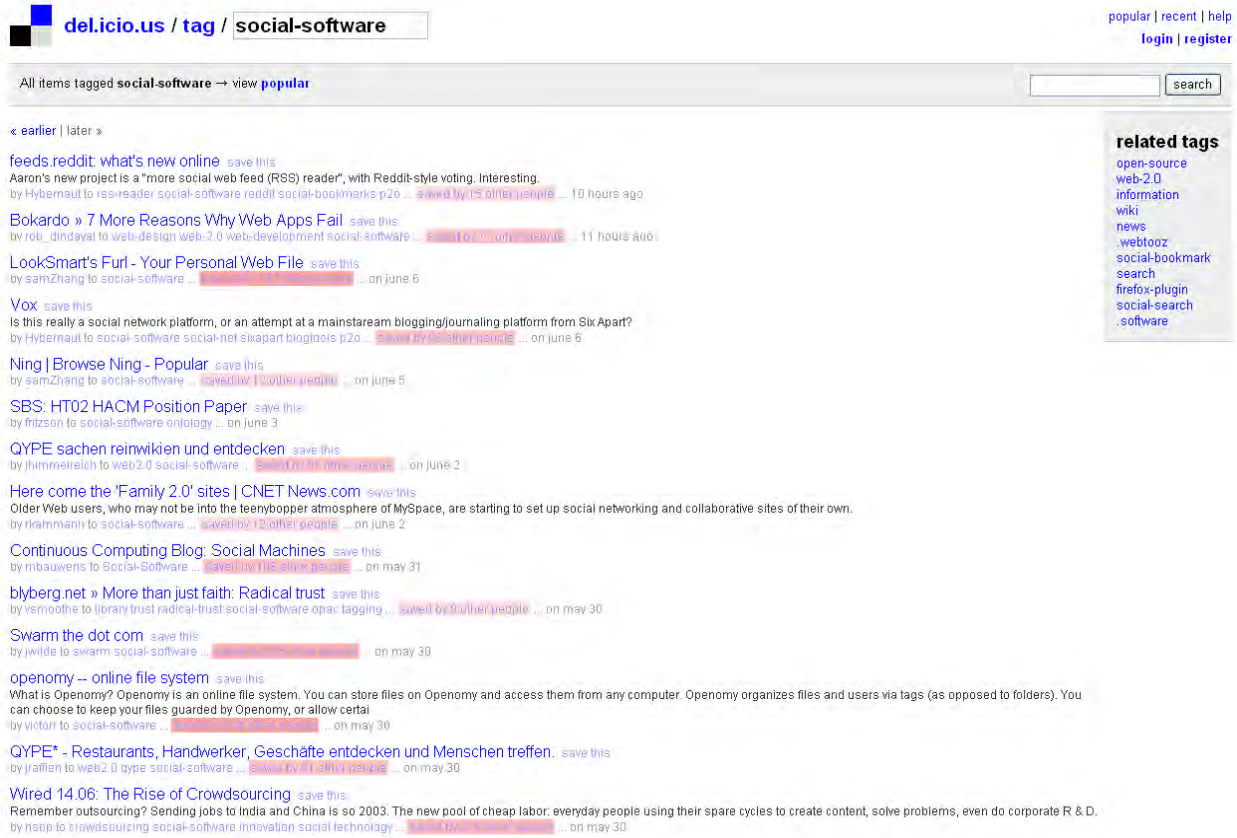


FIGURE 3.10. *Social bookmarking service (del.icio.us [Deli07])*

Typically, recent or popular bookmarks associated with a certain tag – which is 'social-software' in the example in Figure 3.10 – or a particular user's bookmarks can be viewed. Related tags are offered (Figure 3.10, on the right) in order to navigate to bookmarks of web pages with similar topics.

Thus, social bookmarking services as object-centered social software platforms connect individual users around shared bookmarks by using tags for collaboratively classifying the published content.

The unifying social bookmarking service Socializer [Soci07] provides a solution for the issue of individuals being members at several platforms by allowing the simultaneous publication of a link to multiple social bookmarking services.

Bibliographic reference sharing platforms (e.g. CiteULike [Cite07] and Connotea [Conn07]) can be considered as a special form of social bookmarking services since bookmarks to web resources representing electronic publications are supported in particular by bibliographic metadata management.

In addition, annotated bibliographic references of non-digital resources like books and journals are also managed at those platforms. With regard to export services particular bibliographic reference sharing platforms – CiteULike [Cite07] for instance – also provide further bibliographic reference formats like the text-based format BibTex [Pa88] in addition to RSS [Wi05].

Content Publication and Sharing Services

Content publication and sharing services that also pertain to the class of object-centered social software focus on one ore more selected content types each depending on the purpose of the respective community platform.

Examples for this kind of object-centered social software communities are photo sharing services like Flickr [Flic07], video sharing platforms (e.g. YouTube [Yout07]) and goal sharing platforms like 43things [Fort07]. The supported content types in the aforementioned cases are images, videos and goals respectively.

Unlike in the case of social bookmarking services, however, the original content is managed and published on these platforms instead of merely metadata-enriched references to content objects.

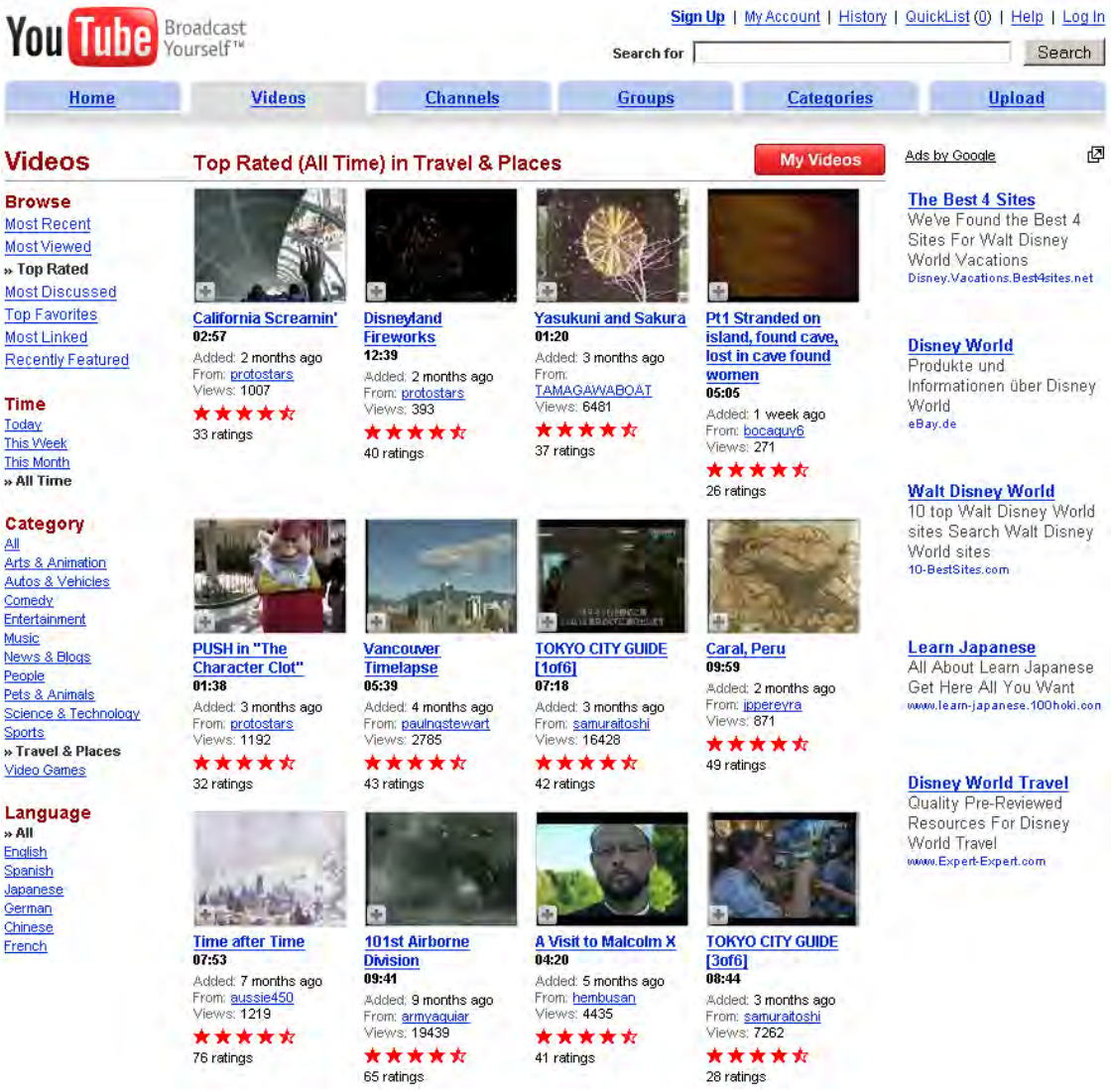


FIGURE 3.11. *Video sharing platform (YouTube [Yout07])*

Figure 3.11 illustrates a representative web page of the video sharing platform YouTube [Yout07] where 'Top Rated' videos are displayed from the category of 'Travel & Places'. On the left side browsing of videos by time and over featured lists, categories or languages are provided. Videos can be uploaded by platform users in order to manage and publish them online. Thus, collaborative

tagging of published videos is provided. Besides the title, owner, and uploaded date of a video the number of users who viewed the video and the average rating are displayed – indicated by stars on a scale from zero to five (see Figure 3.11).

The aforementioned example points out that content publication and sharing services typically provide metadata management, collaborative tagging and feedback services for objects of selected content types. Further, search services and browsing over categories are available.

Management of predefined and/or user-defined groups for role-based sharing of the individual user's content is also an essential service.

The above considerations show that content managed by several kinds of object-centered social software is usually based on rigidly defined metadata which depend on the specific platform.

3.2.6 E-Commerce Platforms

An increasing number of e-commerce platforms including the online auction platform eBay [Ebay07] and the online store amazon [Amaz07] show characteristics of social software (see Figure 3.3). These platforms are representative since they attract a large number of customers. In contrast to social software platforms introduced in the previous sections, the primary purpose of these platforms is sales of specific products.

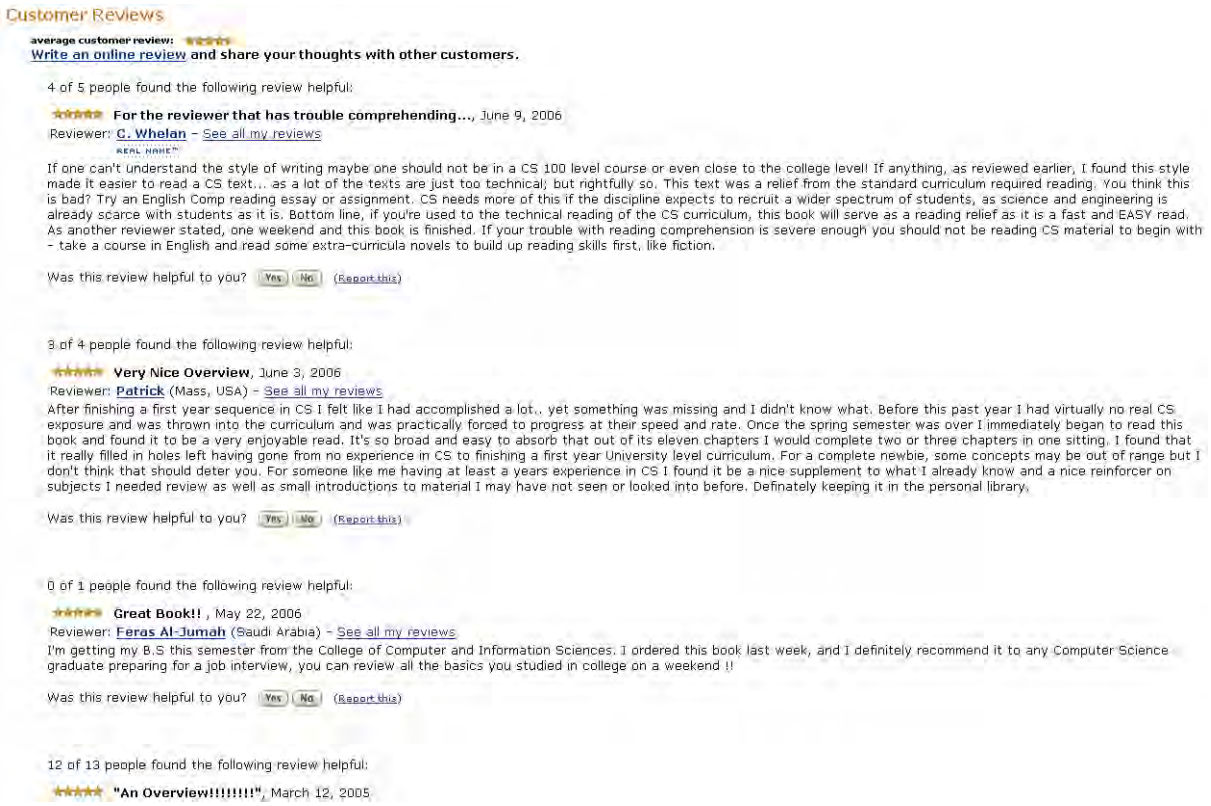


FIGURE 3.12. *Recommendation services at an online store (amazon [Amaz07])*

Novel e-commerce platforms fall into the category of social software since interpersonal networks are established through communication relationships. Recommendation services play a significant role as shown in Figure 3.12 when products are commented on and recommended by customers. Feedback

(textual comment and rating) to products provides other individuals decision support on purchasing the respective product.

In the example of Figure 3.12, customer reviews on a book at the online store amazon [Amaz07] are shown. The reviews represent feedback on the book and include a rating indicated by stars that ranges from zero to five in this case. Further, customers who are interested in that book are also able to appraise whether a particular review was helpful to them (see Figure 3.12).

Therefore, providing feedback at e-commerce platforms is similar to feedback in weblogs (see Section 3.2.2) and at object-centered social software communities (see Section 3.2.5).

Product and price comparison platforms like Shopping.com [Shop07] provide recommendation services independently of a specific vendor similarly to unifying services in the context of object-centered social software (see Section 3.2.5).

The subsequent section examines characteristics of the information models of social software platforms, discusses their limitations and introduces the generic content provider model. Further, social software services are classified from a user-centered perspective.

3.3 Social Software Model

In this section, a generic social software model of the abovementioned highly specialized social software platforms is introduced that encompasses the content provider model and social software services.

Within the scope of this thesis, the focus lies on the one hand on information organization such as metadata management, information networking and classification. On the other hand, information exchange including information search, subscription, publication and sharing by means of these community platforms, most notably object-centered social software, play a significant role.

3.3.1 Information Models of Social Software Platforms

Depending on the purpose of a social software platform (e.g. weblog community, social networking service, social bookmarking service or video sharing platform) the domain-specific information model represents the essential entities, encapsulated metadata and relations among those entities.

Figures 3.13 and 3.14 illustrate the simplified information models of a weblog community and a social bookmarking service respectively.

Concerning the sample weblog community in Figure 3.13, registered users may own multiple weblogs. Each weblog consists of weblog categories which eventually comprise weblog entries. A weblog entry further encapsulates associated feedback. In addition, fine-grained access control is included in the sample information model of a weblog community by restricting user groups whose members can read (*readers*), create (*authors*) or edit and activate (*moderators*) weblog entries in a weblog (see Figure 3.13). In case that no user group is specified for read, write or activation authorization, the owner's completely private content can be represented at those authorization levels within the weblog.

Figure 3.14 illustrates relations of the main domain-specific concepts of a social bookmarking service: registered users own bookmarks which refer to web resources using URLs and can be tagged by multiple keywords. An individual user's tags applied to bookmarks can further be organized in

bundles representing overlapping collections of tags. In contrast to the access control model of the weblog community in Figure 3.13, the bookmarks in this sample social bookmarking service are publicly visible to any web user but merely editable by the bookmarks' owner.

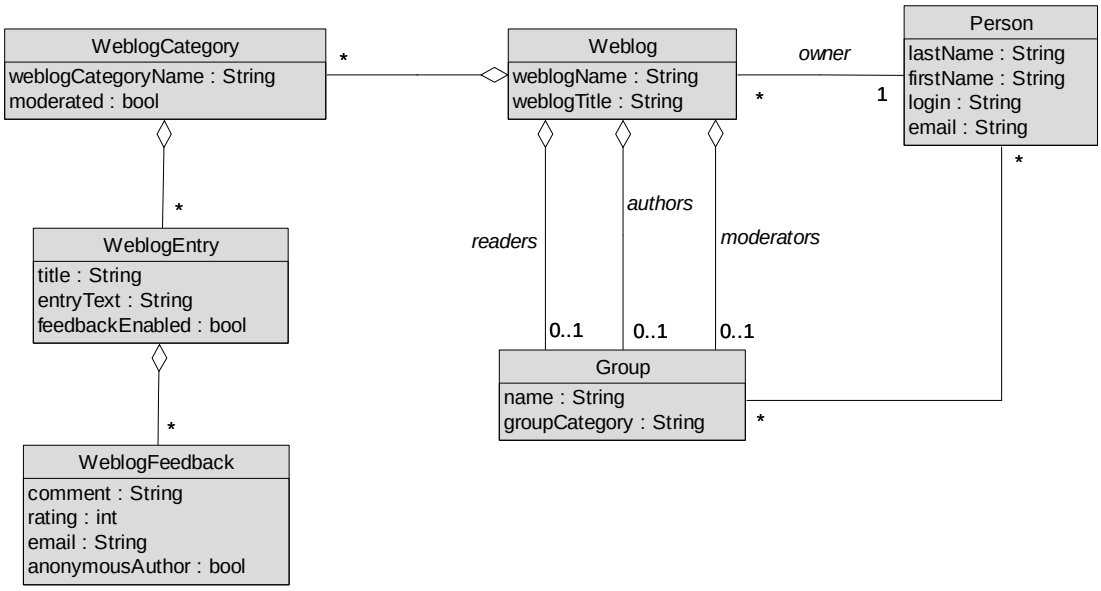


FIGURE 3.13. *Simplified information model of a weblog community (based on [LeMa04])*

Besides differences of entities and relations due to the distinct designations of these social software platforms, discrepancies further arise from the provided metadata for identical content types, in this case user profile information.

For instance, the corresponding concepts *Person* (Fig. 3.13) and *User* (Fig. 3.14) refer to the same entity of a registered user representing it by different structured metadata, i.e. attribute identifier and/or type, though. For instance, an individual user's name is represented by the attributes *lastName* and *firstName* in the case of the weblog community whereas the single attribute *name* is available at the sample social bookmarking service.

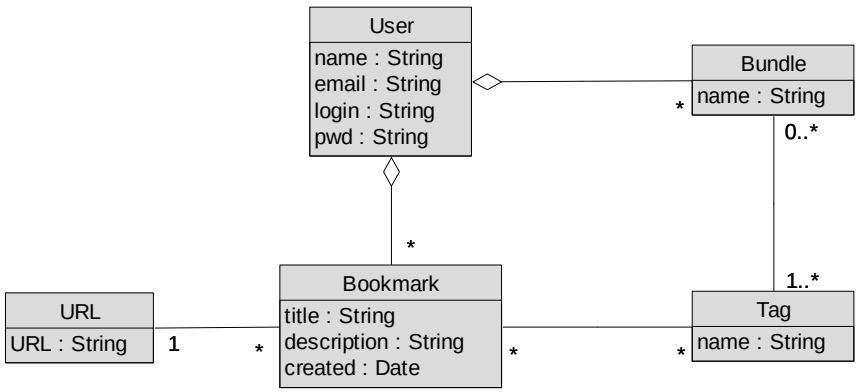


FIGURE 3.14. *Simplified information model of a social bookmarking service (based on del.icio.us[Deli07])*

Hence, two social software platforms designated to the same purpose (e.g. two photo sharing platforms) may also differ in metadata representation of corresponding entities and the specified entity relations despite necessary similarities in their information models.

Limitations of Social Software Platforms

From a user-centered perspective, social software platforms show several limitations implied by their domain-specific information models.

Limitations of social software communities arise from merely focussing on one content type. Unified services are available, though, but only per content type (see Section 3.2) instead of an integrated view on all content objects an individual user intends to give attention to at any one time independently of their content types.

Thus, the same functionality (search, publishing, tagging, access control, feedback and rating of content objects) is realized with different user interfaces that depend on the respective social software platform and the content type.

Concerning content types, rigidly defined metadata is specified according to proprietary schemas as shown Figures 3.13 and 3.14. Thus, no means exists in order to define and manage content objects regardless of content types in a schema-free manner or to extend content objects by flexible user-defined metadata.

Due to the distribution of content objects across social software platforms, the individual user has no opportunity of linking together semantically associated content objects that reside at different platforms.

In addition, the majority of social software platforms are noncommercial thus providing no guarantee for long-term availability of the managed content objects and quality of service. From the individual user's perspective, lack of offline access to content on the personal device unlike in the case of personal applications is therefore a significant limitation.

Ownership of content also plays an essential role from the individual user's point of view. Access control mechanisms and their expressiveness differ on social software platforms as pointed out by the sample information models in Figures 3.13 and 3.14. Both examples illustrate that write access is usually limited to the content's owner by default. A further distinction concerning read access to content is not provided in either case (for instance, only within the weblog community in the above examples). Hence, social software platforms exist where private, public and selectively shared content cannot be managed separately by means of role-based access control mechanisms.

A further issue lies in the fact that the identity of content objects is limited to a particular social software platform. This also results in decentralized identity management for individual users who are often members of multiple social software communities.

From the popularity of social software an increasing need arises to address the aforementioned limitations of social software platforms from a user-centered perspective since individual users are frequently members of a multitude of communities.

Several so-called Web 2.0 desktops like Netvibes [Netv07] and Pageflakes [Page07] attempt to overcome the limitation of an integrated view by providing unified access to the individual user's accounts on selected social software platforms and feed-based aggregation of relevant content from available information sources. However, these services only offer separate modules presented in a

single view which are often limited to read access with regard to the individual's content on integrated platforms.

In order to suggest a more appropriate user-centered model, the communalities in information models and services of social software platforms are analyzed in the following two subsections.

3.3.2 Content Provider Model

From the above considerations on various kinds of social software platforms the generic content provider model is deduced according to Figure 3.15. The content provider model subsumes general characteristics of social software platforms by abstracting from the special purpose of several community platforms. However, the model abstracts from domain-specific information models concerning representation of metadata and relations among entities at social software platforms as given in Section 3.3.1.

As depicted in Figure 3.15, social software platforms are viewed as *content providers* from a user-centered perspective. Since the primary objective is the management of content objects of selected types this characteristic is reflected by the general entity *content* in the model. Examples for objects of concrete content types are therefore indicated by inheritance in the content provider model – for instance, weblog, document, e-mail and bookmark shown in Figure 3.15.

The content provider model further encompasses *classification schemes* of social software platforms (see Figure 3.15). According to Section 3.2, on social software platforms, typically tags (flat classification) or categories (hierarchical classification) for content objects are used. Hierarchical classification schemes are often termed folders like in the case of personal applications (see Chapter 2). Content objects are thus linked together by means of the available classification schemes at content providers.

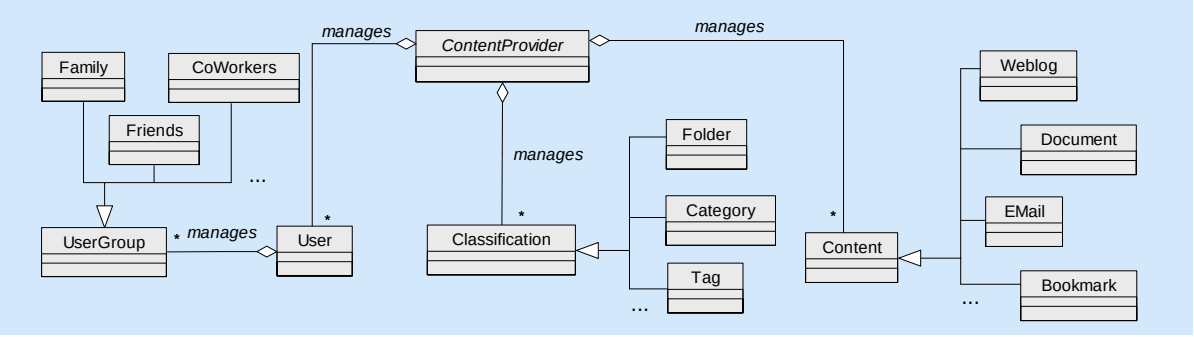


FIGURE 3.15. *Content provider model*

Social software platforms also comprise user management for authentication, authorization and personalization purposes represented by the entity *user* in the content provider model (see Figure 3.15). Users of social software platforms are able to manage their contacts in user groups corresponding to private and professional social contexts (see Section 3.1.2) such as family, friends and co-workers.

The content provider model is further applicable to personal applications supporting personal information management as described in Chapter 2. Personal applications form a special case of content providers considering the model since they are single-user applications thus not involving user management unless they are designated to manage user profile information in the form of contacts as

content objects. An example for the latter are e-mail clients that manage at least e-mail and user profile information (contact objects) as described in Section 2.3.1.

According to Section 3.2, each social software platform provides management of rigidly defined metadata for selected content types usually specified in a proprietary manner. Metadata representing identical content types may thus differ depending on the platform.

Access Control to Content Objects

Figure 3.16 illustrates access to personally managed content objects on social software platforms from the individual user's perspective. Each individual user maintains an area on the social software platform – referred to as the *individual user's area* throughout this thesis – comprised of (overlapping) logical scopes that determine the accessibility of content objects by other platform users.

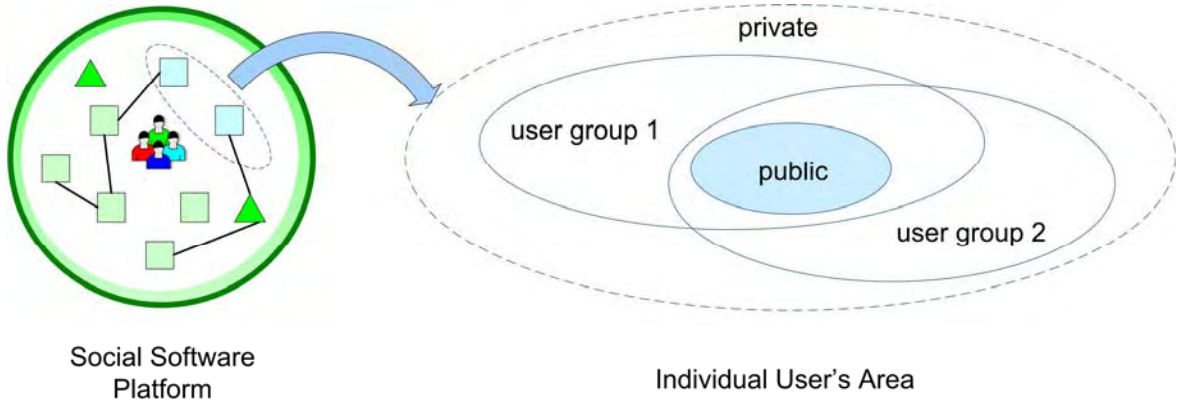


FIGURE 3.16. *Logical scopes of access to content objects in the individual user's area*

In addition to the private and public scope, predefined and/or user-defined groups exemplified in Figure 3.15 can be managed by the individual user at several social software platforms in order to achieve fine-grained access control to content objects with regard to their contacts. Figure 3.16 exemplifies the aforementioned logical scopes in the individual user's area on a social software platform which is marked by a dashed border.

The *private scope* includes all other scopes, that is, individual users have access to all content objects that are stored and managed in their own area. By contrast, the *public scope* refers to content objects that are publicly accessible by any platform user. Further scopes on content objects are determined by the specification of user groups on a social software platform, i.e. contacts that are members of a particular user group may access content objects which are included in the scope of that user group. As further depicted in Figure 3.16, the intersection of all defined scopes includes the public scope.

3.3.3 Social Software Services

Social software platforms provide a multitude of services termed social software services in our approach [LMW05] that can be classified into the five functional categories from the individual user's perspective, that are, *retrieval services*, *publication services*, *subscription services*, *feedback services* and *lifecycle management services* according to Figure 3.17.

Retrieval services provide read access to content objects managed at a social software platform. On the one hand, a set of content objects can be filtered by the specified search criteria (‘Get content by criteria’). Special cases of collection-based retrieval services shown in Figure 3.17 offer querying content from a social software platform authored by particular users (‘Get content by user’) or associated with a specific tag or from a category (‘Get content by tag/category’). On the other hand, a further retrieval service facilitates obtaining metadata of a particular content object (‘Get metadata of content’).

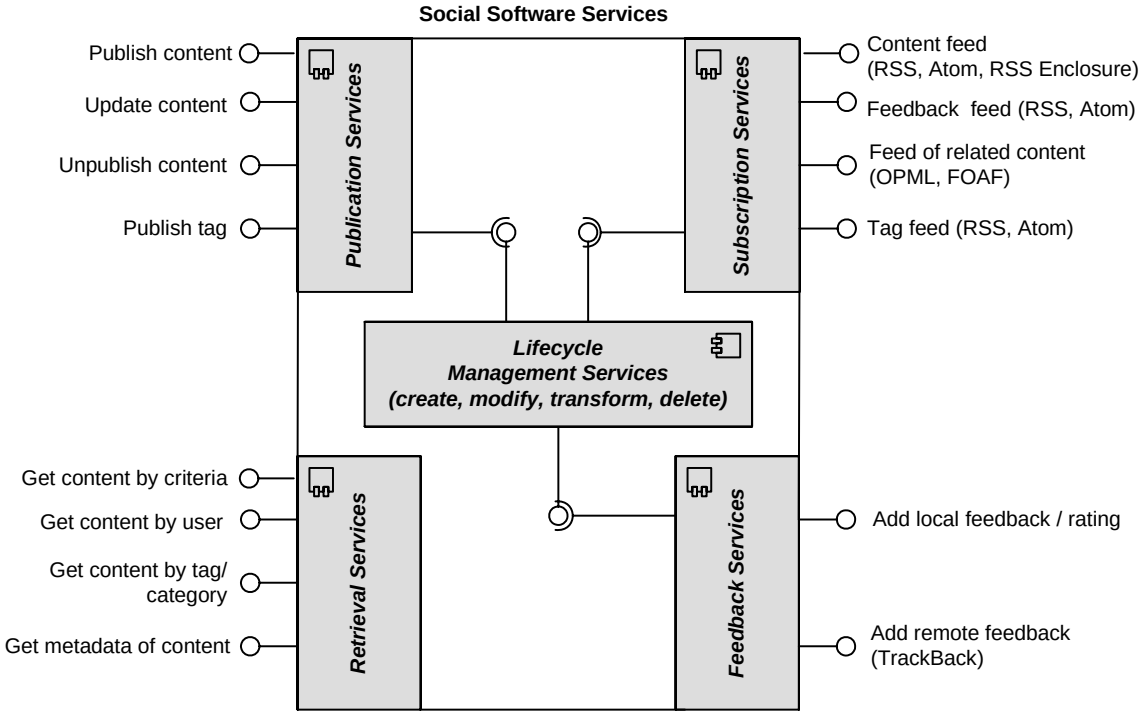


FIGURE 3.17. *Classification of social software services (based on [LMW05])*

Publication services facilitate publishing (uploading) of content objects to social software platforms in order to manage them online and share them with selected contacts who are community members (‘Publish content’). Usually, services also exist to update (‘Update content’) or remove (‘Unpublish content’) already published content. In addition, ‘Publish tag’ denotes a service for publishing tags as classification information for content objects and is therefore separated from the ‘Publish content’ service in Figure 3.17.

As depicted in Figure 3.17, subscription services encompass feeds of content objects – mainly in RSS [Wi05] and Atom [NoSa05] formats – that can be subscribed to in order to get notified of newly created content (‘Content feed’) of a platform user or recent feedback to the individual user’s own content (‘Feedback feed’). Additionally, feeds of related content from a user may be offered, usually in further XML-formats (see Section 3.2) like OPML [OPML00] for hierarchically structured content and FOAF [BrMi05] representing information about the user and his maintained social relationships (‘Feed of related content’).‘Tag feeds’ are distinct from the aforementioned subscription services since they provide subscription to recent content for a specified tag that originate from multiple platform members.

Feedback services (see Figure 3.17) subsume the publication of feedback objects which are associated with other content objects and depend on their lifecycle. Social software platforms on the

one hand often support the publication of local feedback or rating for a specific content object (‘Add local feedback/rating’) according to Section 3.2. On the other hand, remote feedback services (‘Add remote feedback’) may additionally be offered at particular platforms in order to track feedback to a content object in other systems (TrackBack [Trac04]) according to Section 3.2.2.

Services of the aforementioned four kinds that imply write access to a social software platform rely on lifecycle management services of that community platform. These are applied in order to create a new content object (e.g. ‘Publish content’ or ‘Add local feedback’), modify (‘Update content’) or transform (‘Content feed’) an already published content object or to delete (‘Unpublish content’) content objects from a specific social software platform.

Personal applications represent a special case with regard to social software services similar to the circumstances explained for the content provider model (see Section 3.3.2). This results from personal applications merely providing retrieval services and life cycle management services for the content objects they manage in their respective repositories. Since information exchange in online social networks is not an issue concerning personal applications, publication, subscription and feedback services are usually not present in their models.

In case an appropriate API is provided at the respective social software platform retrieval, publication, subscription and feedback services are made available for external access. This establishes the basis for an application that offers an integrative view on the content and services of several social software platforms and personal applications from a user-centered perspective.

3.4 Summary

The emerging class of social software represent innovative Web 2.0 applications that facilitate communication and information exchange in online social networks. Several kinds of computer-mediated communication services and community platforms pertain to the class of social software that do not merely connect information on the web but also connect individuals around shared content thus implicitly or explicitly establishing online social networks.

Social software platforms have gained considerable popularity among lead users on the web since they provide communication services and additionally support metadata management, information search, publication and sharing, subscription, commenting and collaborative classification in social contexts based on various underlying social networks. Hence, individual users can link together and exchange information with selected contacts by means of social software in contrast to conventional personal applications.

However, functional features and information models of social software platforms reveal limitations of these community platforms regarding the individual user’s perspective, most significantly lack of service integration, flexible metadata management and cross-platform relation management of content objects.

The generic content provider model developed in this chapter incorporates common characteristics of those models and can also be applied to describe personal applications. In contrast to personal applications, social software platform additionally provide publication, subscription and feedback services that enable information exchange in the individual user’s social contexts.

Based on considerations from Chapters 2 and 3, the user-centered social software model addresses the limitations of personal applications and social software platforms by facilitating an integrated view on personal information management in the different social contexts from the individual user's perspective. The proposed model encapsulates the characterization of social information management as an extension of personal information management by information exchange in the social contexts of an individual user.

Chapter 4

User-Centered Social Software

This chapter introduces static and dynamic aspects of the user-centered social software model. The conceptual model of user-centered social software which is based on the generic content provider model of social software platforms and personal applications developed in Chapter 3 relates the core concepts of content, context and contacts. Further, social information management is characterized as an extension of personal information management by describing significant sub processes and use cases of information acquisition, organization, and dissemination. Essential characteristics of the user-centered social software model are explained comprising schema-free metadata management, cross-platform relation management, a role-based authorization model and rule-based support of social information management.

4.1 Terminology and Conceptual Model

Following the term object-centered social software coined by Rheingold [Rh05] (see Section 3.2.5) *user-centered social software* (*UCSS*) denotes the alteration of the view on social software platforms to the individual user's perspective instead of emphasizing shared content objects. The term user-centered social software thus refers to a new software family of personal applications targeting the individual user who intends to discover, manage, publish and share personal content in a consistent, customizable and secure way in social networks established at social software platforms as an extension to personal information management activities.

User-centered social software runs on end user devices like workstations, laptops or mobile devices and facilitates social information management, that is, information acquisition, organization and dissemination in the social contexts of an individual user by providing a unified view on the available content managed on social software platforms and by personal applications. Therefore, the family of user-centered social software is designed to support *organizing information in a social way* as stated in [LMW05], i.e. managing information within personal networks and interacting with public and shared content owned and/or authored by contacts.

The top part of Figure 4.1 illustrates the primary abstractions of the content provider model as introduced in Section 3.3.2. Within the scope of user-centered social software, the content provider is

either a social software platform that provides user management or a personal application on the personal device.

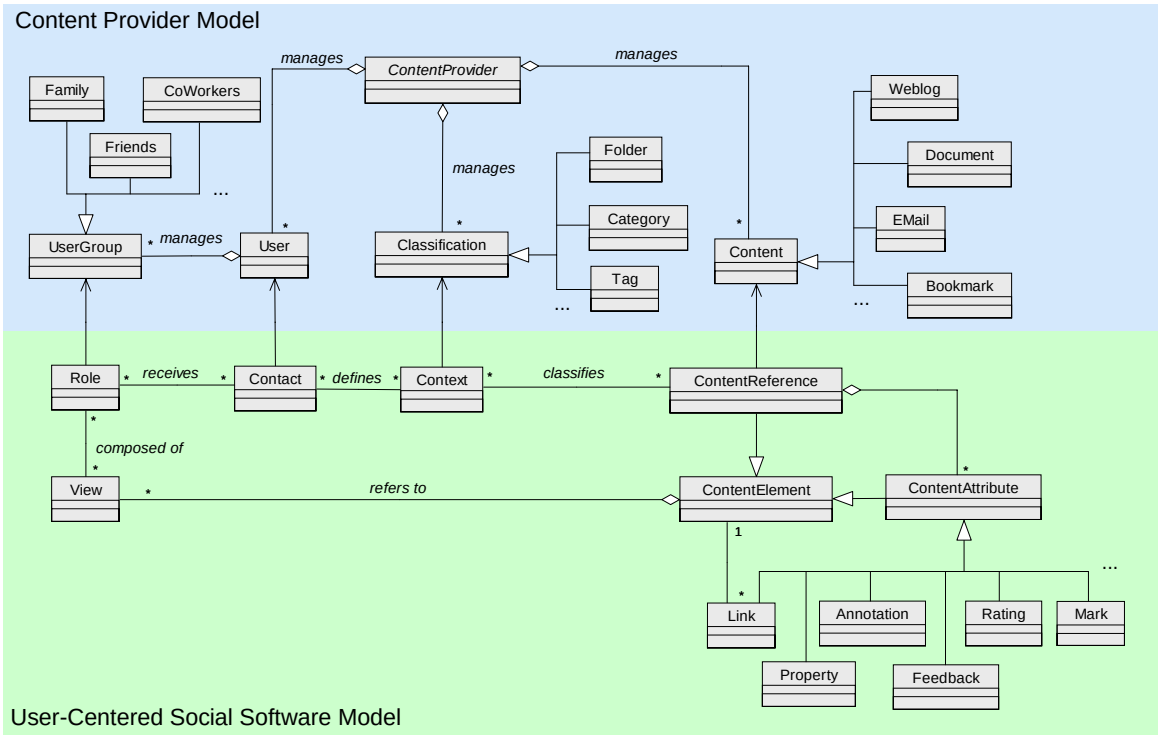


FIGURE 4.1. *User-centered social software model*

Content objects of specific content types – for example type weblog, weblog entry, e-mail, calendar item, document or bookmark – represent information being managed at one or more content providers and organized by the individual user according to Chapters 2 and 3. Content refers to all information that is available to the user and is uniquely identifiable by a URI. Therefore, considering social software platforms, content not merely relates to the user's own personal information published on a platform but also includes information managed there by other platform users. Content objects managed by a personal application on the personal device are referred to as *local content objects* whereas content objects managed on social software platforms are termed *remote content objects*.

Content can further be classified in multiple ways depending on the available classification mechanisms at the respective content provider and the content type of an object, for example folders in the case of an e-mail client, categories at an online store or tags at a social bookmarking service (see Section 3.2). The according classification schemes are either flat or hierarchical.

The user-centered social software model is characterized by three core concepts *contact*, *content reference* and *context* defined in the following that are essential for the static representation of information acquisition and dissemination in online social networks.

An individual user's *contacts* on a social software platform form a subgroup of all users on such a platform and represent one or more of the user's social contexts. According to Section 3.2, contacts are those platform users with whom the individual user is engaging in information acquisition and dissemination processes. Contacts are assigned to *roles* that correspond to predefined or user-defined user groups at the content provider and managed by the individual user (see Figure 4.1). Examples for roles in the scope of user-centered social software include family member, friend, acquaintance, co-

worker and schoolfellow that correspond to user groups on social software platforms as described in Section 3.2.

Content reference objects represent references to specific content objects at the content provider and pertain to a certain content type each including weblog, document, e-mail and bookmark that are given as examples in Figure 4.1. Metadata on contacts, i.e. user profile information, is also represented by content reference objects. The terms *content object* and *content reference object* are used as synonyms in the scope of user-centered social software in the following.

Context is defined as an abstract representation of classification schemes within the scope of user-centered social software. A further characteristic of context is semantically relating content from several contacts according to the user-centered social software model in Figure 4.1. Thus, individual users can post content to social software platforms and share it with their contacts by means of contexts they define themselves. Therefore, contexts facilitate a semantic grouping of content objects with regard to the specified characteristics.

The bottom part of Figure 4.1 depicts the correspondence of the concepts of the content provider model to the abstractions of user-centered social software.

Concerning information organization, content objects – represented by content references – need to be extended platform-independently by metadata in a flexible way. Newly added metadata to content objects and binary relations between content objects are thus represented by a certain kind of *content attribute*. The *content element* entity in the model (see Figure 4.1) denotes a single concept for content references and content attributes in order to enable the unified handling of both. Thus, content attributes become first class objects according to the model and can be organized and exchanged in the same way as content references.

In the user-centered social software model, four different kinds of content attributes exist. Content attributes termed *properties* pertain to only one content reference object. The name and value range of a property are determined by the *property type* as shown in the diagram in Figure 4.2. The value range assigned to a property type is either defined by a simple data type like boolean, integer, floating-point number, character string or date. Further, the value range can be specified by a domain with specific predefined values according to a nominal or ordinal scale like academic title, country calling code or weekday. Examples of properties include the weekday an e-mail was sent, the URL a bookmark refers to or the number of pages of a document.

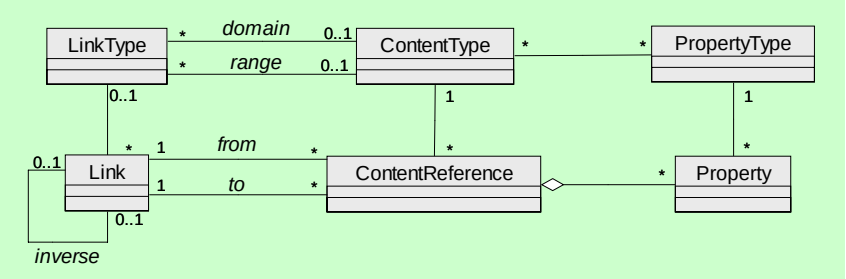


FIGURE 4.2. *Model of links and properties*

The second kind of content attributes are *links* that represent binary relations between content elements according to Section 2.1.2. From the viewpoint of one content element, several other content elements are related to it. The links that can be established between two content elements depend on the content types or the kinds of content attributes respectively. This means, the *link type* determines the domain

and range in terms of appropriate content types as depicted in Figure 4.2. An example for a link is the relation *createdAt* between a specific document (e.g. a report) and a calendar item (e.g. a meeting). According to this example, links of type *createdAt* can only be established between document objects on the one hand and calendar items on the other hand. Links are discussed further in Section 4.1.1.

The third kind of content attributes is composed of content type-independent metadata as an extension of content with regard to personal opinion and social interaction of the individual user. The following significant examples (see Figure 4.1) fall into this category: *annotations*, *feedbacks*, and *ratings*. An annotation is in general a textual note attached by individual users to their own content whereas a feedback is a textual comment or review provided by the user for the content of a contact. A rating is a value based on a defined ordinal scale that can be added to a certain content object to indicate its quality or relevance.

The fourth kind of content attributes are *marks* that represent a specific value termed *label* according to a nominal or ordinal scale. The label can be a textual or numerical value or a color code respectively with specified semantics. By using an ordinal scale, such labels can be applied for instance to prioritize content objects. Predefined semantics of labels are for example used to denote specific characteristics of a content object like accessibility (see Section 4.3) or automatic information organization according to Section 4.4.3.

Further, the concept of *views* also contributes to the representation of information dissemination in the user-centered social software model (see Figure 4.1). The view on a content element defines its designated accessibility with regard to read access (i.e. public, private and role-based) as explained in detail in Section 4.3.

Views as an elementary concept establish the basis for role-based authorization presented in Section 4.3. A set of views on content objects form a role that can then be assigned to several contacts. In general, views and roles are not defined individually for the respective content elements that the individual user intends to share with his contacts but the specification of rules determines a role which is discussed in Section 4.4.5.

4.1.1 Link Management

User-centered social software facilitates cross-platform relation management of content objects. Hence, content objects that are located at different content providers can be related whereas the relation is managed by the user-centered social software. By contrast, related solutions for personal information management support are merely able to relate content objects that are managed by personal applications (see Chapter 2.3). Management of relations between content objects located at different social software platforms is therefore a significant extension to personal information management capabilities of existing tools.

Characteristics of Link Types and Links

Within the scope of user-centered social software, binary relations between content elements are termed links as introduced above. Polyadic relations involve an arbitrary number of content objects though, but can be represented by multiple binary links and are thus not taken into further consideration.

In order to establish relations between content objects, it is necessary to define link types in advance. According to Section 2.1.2, a particular link type determines the domain and range of content

objects that can be related. The domain and range are specified by a single content type or a set of content types. A special case is not to restrict the content type but specify the set of all defined content types in the user-centered social software for domain or range respectively. The semantics of a link type can be specified arbitrarily by the individual user and is in general described by a label.

Additional characteristics of a relation with regard to algebraic properties are also specified when the link type is established. These form the basis of the capability of inferring further links in a network of content objects based on existing links (see Section 4.4.3).

Elementary characteristics of a relation include the algebraic properties symmetry, asymmetry and transitivity and the declaration of the *inverse link type* when appropriate.

A link type specified as a *generalization* (*specialization*) is used to transitively establish a hierarchy of content objects. For example, the hierarchy of directories can be described by this means.

The characteristic of an *order relation* denotes the capability of comparing and sequentially ordering content objects according to a specified ordinal scale. The following special cases of order relations are significant in the scope of user-centered social software: *chronological*, *alphabetical* and *numerical order*. Chronological order is for instance applied to relate content objects according to their last modification date, alphabetical order is applied in order to relate textual metadata and numerical order is able to relate content objects based on their rating. As pointed out by these examples, such order relations for content objects are in general implicitly inferred from original attributes or other content attributes like ratings on the one hand or properties whose value range corresponds to one or more intervals on an ordinal scale on the other hand (see Section 4.4.2) which also might depend on the content type.

A special link type with predefined and content type-independent semantics is *equivalence* which is used to relate semantically equivalent versions of a content object located at different content providers and managed by the user-centered social software. Content objects are automatically related by *equivalence* (see Sections 4.2.2 and 4.4.3) with a link type labeled *equivalentTo*.

Relations with particular semantics may be specified by the individual user by establishing further link types manually in addition to predefined link types in the user-centered social software. For user-defined link types, the characteristics discussed above can also be declared in addition.

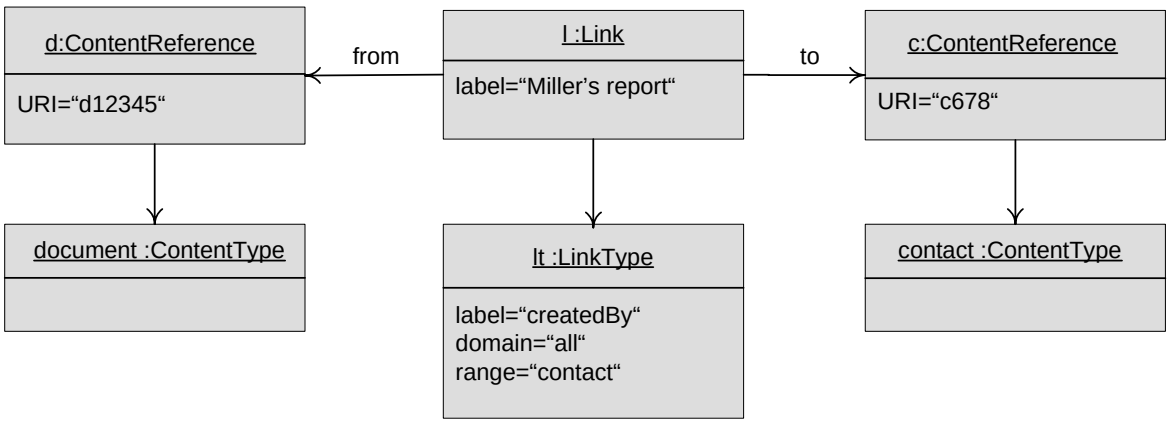


FIGURE 4.3. *Links as first-class objects*

Links between two content objects can be created based on a link type. Therefore, such a link at least constitutes of a content object that corresponds to the range and a content object that corresponds to the domain of the respective link type. By contrast, an *ad-hoc link* established manually by the

individual user between two content objects is merely defined by the semantics of its label. The semantics of an ad-hoc link has to be interpreted by the individual user.

An essential characteristic of a link is its direction as introduced in Section 2.1.2. Links of link types that are specified as symmetric relations are represented by *bidirectional links*. In case of an inverse link being specified for the content objects concerned, the two corresponding links can also be considered forming a bidirectional link. All other relations are represented by *unidirectional links*.

Links are established according to link types either manually by the individual user or automatically by the user-centered social software when performing related use cases of information organization (see Section 4.2.2). Manual creation of links is either based on predefined or user-defined link types. Figure 4.3 illustrates links as first-class objects in the user-centered social software model applied to the example given in Figure 2.4.

However, predefined link types are required for automatic creation of links between content objects. Automatic links are created rule-based as described in Section 4.4.3.

Network of Content Objects

Content objects managed by the user-centered social software form a network that is depicted in Figure 4.4 in compliance with Section 2.1.2. Content objects are represented by their content types and illustrated by an appropriate icon. Both unidirectional and bidirectional links are depicted as solid arrows and lines respectively in the given example. Dashed lines by contrast refer to relations of content reference objects to those objects that are not (yet) referenced and managed by the user-centered social software.

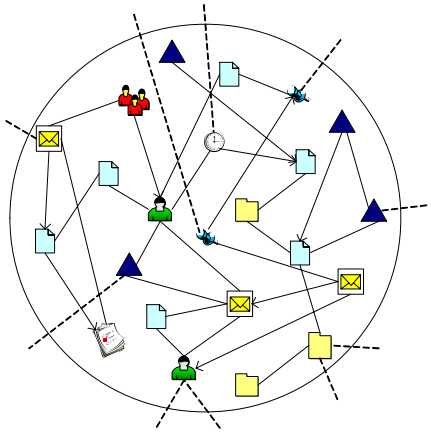


FIGURE 4.4. *Network of content objects managed by the user-centered social software*

Hence, a link is established by means of the user-centered social software between content objects that are not necessarily managed at the same content provider. By contrast, a relation represented by a dashed line is out of scope of the user-centered social software while one of the content objects involved in that relation is already referenced.

4.1.2 Task Contexts and Portfolios

The individual user applies an instance of the user-centered social software in order to acquire and disseminate information while coping with a certain task. In order to facilitate the accomplishment of a task, the user can organize the available information in advance but also during that task (see Section 2.2.2). However, task-independent information organization by means of the user-centered social

software as a PIM tool enhances the reuse of information for later tasks due to flexible metadata and link management.

For effective use of the user-centered social software, *task contexts* are introduced (see Figure 4.5) that represent the individual user's workspaces related to specific tasks according to Section 2.2.2. For this purpose, the user can define task contexts that represent an integrated view on content providers that are relevant to a specific personal task such as those exemplified in Figure 2.8. Therefore, only content located at the assigned content providers is visible for information acquisition, organization and dissemination.

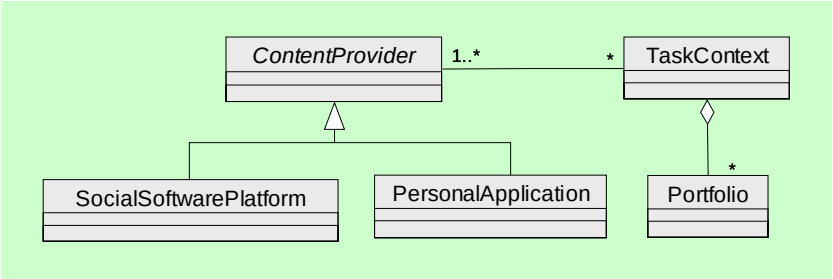


FIGURE 4.5. *Task contexts and portfolios*

A further characteristic of user-centered social software is that social information management is performed on networks of content objects. This implies on the one hand, that the majority of information acquisition and dissemination processes are considered to be operations on sets of content objects. On the other hand, these sets are not merely flat lists of objects but the content objects are strongly connected by binary links.

The workspace concept of user-centered social software hence includes *portfolios* that are user-defined collections of content references following [We02]. Portfolios enable grouping of content references for later reuse in the respective task context and thus represent a means for applying the information inventory control strategy of Surface Clutter introduced in Section 2.2.3. Using portfolios, for example search results (see Section 4.2.1) can be made persistent. Compared with contexts, portfolios represent a mechanism for temporary classification of content objects.

In the subsequent section, the processes and use cases of social information management are explained that constitute the dynamic aspect of the user-centered social software model.

4.2 Social Information Management

Based on the processes of personal information management introduced in Chapter 2, the model for social information management comprising social interaction is described in the following.

Social information management encompasses the sub processes of information acquisition, organization and dissemination in compliance with personal information management. Hence, user-centered social software supports this kind of *information-based processes* where explicit information is taken into account. By contrast, *knowledge-based processes* describe the handling of implicit information, i.e. knowledge, by individuals that cannot be represented in information systems (see Section 2.1.1).

Social information management can therefore be considered as an extension of personal information management (focusing also explicit information) since information is not only organized by personal applications but information organization is extended to social software platforms facilitating information exchange in the social contexts of the individual user.

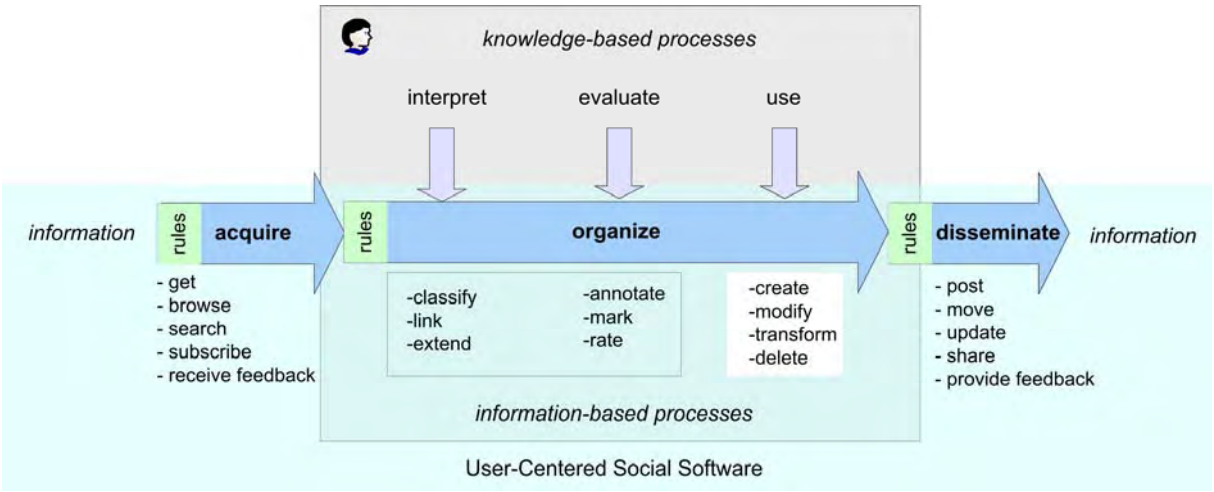


FIGURE 4.6. *Processes of social information management*

Figure 4.6 presents the processes and sub processes of social information management. The three primary information-based processes of social information management, *acquire*, *organize* and *disseminate* are illustrated in bold type in the figure. The processes and their sub processes are related to one cycle of the continuous information flow from the individual user's point of view.

The box in the center of the figure represents on the one hand knowledge processing by the user himself who is depicted by an appropriate icon at the top of the box. On the other hand, the bottom part of the box represents one of the personal devices the user interacts with at a given time for information processing. Transformation of knowledge into explicit information during information processing is illustrated by vertical arrows. The individual user is supported in social and thus personal information management by the user-centered social software in addition to other personal applications like personal productivity tools.

One cycle of the information flow in Figure 4.6 can be considered as starting with information acquisition (horizontal arrow on the left). The incoming information in the form of content objects is first *interpreted* by the user. This knowledge-based process implies learning form incoming information especially from feedback of contacts.

Thereafter, the user-centered social software can be used to organize the content objects in different ways that are represented by the sub processes of information organization (horizontal arrow within the box). Further handling enables the user to *evaluate* the content and make a note of his insights by the supported organization processes of the user-centered social software. In this step, the user's knowledge is transformed into explicit information. The user's intention to *use* information is supported by the appropriate applications that are designated to provide lifecycle management for content objects of certain formats of content types (see white box in Figure 4.6) like personal productivity tools and is out of the scope of user-centered social software.

Finally, in the step of information dissemination (horizontal arrow on the right) content objects are published on social software platforms to let contacts access parts of personal information. By means of the outgoing information, one cycle of the information flow ends.

The subsequent sections are devoted to the elaboration of the sub processes and use cases of the three primary social information management processes.

4.2.1 Information Acquisition

According to the information flow, information acquisition subsumes the several ways for incoming information as its sub processes as depicted in Figure 4.6 below the arrow on the left. These can be applied either manually by the user or semi-automatically by the user-centered social software on previously specified user requirements as explained further in the following sections.

Manual Information Acquisition

The *get* process refers to directly accessing and obtaining a single content object when its URI is known. In general, this is a manual process accomplished by the user since no criteria are specified to be matched by the content objects compared to the use cases of the search process as explained below. In contrast to the remaining information acquisition processes described below, only one content object is handled. Hence, merely the use case *get by URI* is associated with the get process.

The *browse* process refers to the activity of orienteering introduced in Section 2.2.3 and thus to navigation over the classification schemes of a content provider. In this case, the user can search manually for content at a specific content provider in contrast to the search process. The use cases defined for this process are exemplified by *browse categories*, *browse folders* or *browse tags* according to the respective classification scheme. Therefore, it depends on the content provider which use cases can be applied.

Information Search

During the *search* process, a set of content objects is obtained that correspond to the criteria specified by the individual user. This set of content objects represents the search result. The search criteria are in general composed of content-related restrictions such as the content type on the one hand and/or restrictions on the content, its original metadata from the content provider, the context and further content attributes managed by the user-centered social software on the other hand. The search result may be influenced by additional rules that apply for specific use cases of the search process (see Section 4.4 for details).

The search process defines two separate scopes that correspond to finding information in the first place by means of information acquisition and refinding information referred to as information retrieval respectively according to Section 2.2.3. Thus, the scope is either the repository of the user-centered social software, referred to as the *search inside scope* in order to support retrieval of previously acquired information. In this case, search is applied to content reference objects managed by the user-centered social software.

By contrast, the *search outside scope* is concerned with content objects that are located at the respective content providers and are not yet referenced and managed by the user-centered social software. The search outside scope both refers to searching for local content objects managed by personal applications on the end user device and for remote content objects located at social software platforms. Therefore, a special case of a local search is achieved by defining a task context to which only personal applications as content providers are assigned.

In the event of a *keyword search* as a use case of the search process, the aforementioned restrictions are expressed by means of keywords specified by the individual user. Examples for a keyword search include obtaining all content objects that are tagged by a certain keyword or searching for all content objects of a specific content type, created not later than the given date and their content attributes matching several textual keywords.

Figure 4.7 illustrates the scenario of a *keyword search* by the individual user for information acquisition purposes which implies the search outside scope. Application of the keyword search use case results in an integration of retrieval services of different content providers indicated by the 'getContent' request according to Section 3.3.3 and final composition of several obtained search results. Thus, the user-centered social software is a mediator in order to unify information search.

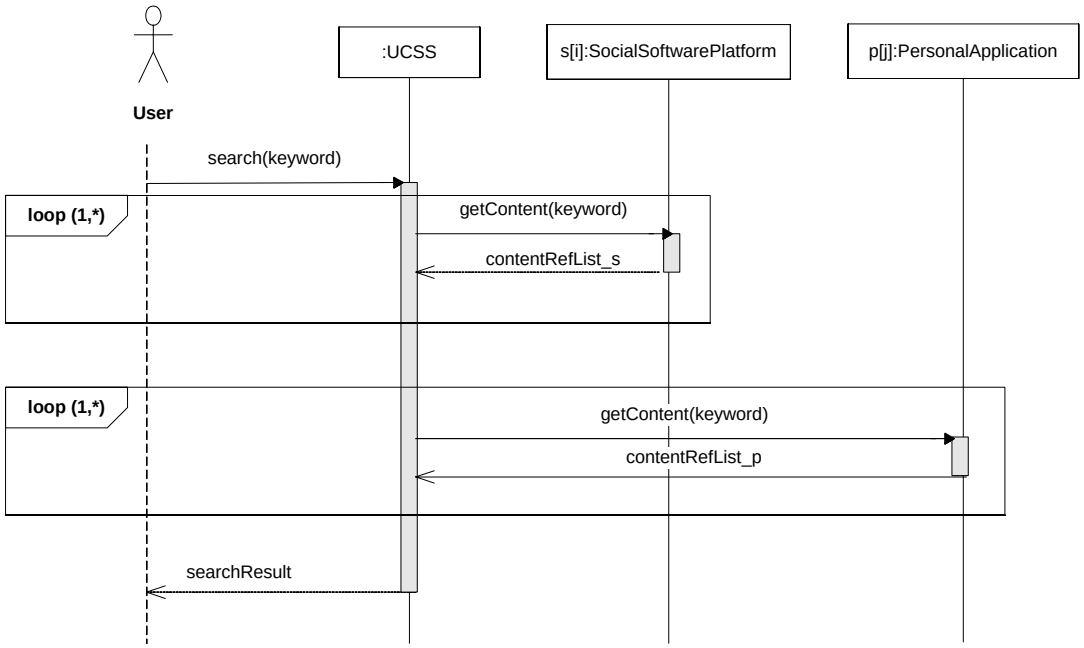


FIGURE 4.7. *Typical scenario for unified information acquisition*

Context search is a special case of the keyword search that returns all content objects classified by the specified keyword according to the available classification schemes. With *contact search* as another specialization, user profile information can be searched for based on keywords. For this kind of search use case, contacts are considered content objects with respect to personal metadata representation which is user profile information at social software platforms. The abovementioned two use cases require separate mention due to their significance for social information management with context and contact being core concepts of user-centered social software.

A further use case is *full text search* being a further modification of the keyword search that can be applied to unstructured or semi-structured content objects containing textual content. This means, not merely the metadata of the content objects is included in the search but in addition their textual content. Whether this use case is available, therefore strongly depends on the content type. Valid content types for this use case include documents, weblog entries or e-mails.

For *relation search*, the criteria are determined by either a given content object or a set of content objects respectively. The search result contains all objects that are related to all specified content objects by links in the case of the *search related to all* or to at least one of the given content objects that holds for the *search related to any* use case.

It can be specified, in which way the obtained objects in the search result should be related to the given ones: either by direct links – defined as path length of 1 in the network of content objects – or indirect links up to an arbitrary path length (see Chapter 2.1.2). Indirect links require intermediate content objects and define transitive relations. Finally, if the path length is high enough or without any restriction respectively, the search result constitutes the transitive closure of the initial set of content objects. The *relation search* use case can further be restricted to the possible link types between any two content objects that are also taken into consideration to find suitable relations.

Similarity search represents a use case where the search result has to correspond to the criteria that are specified by an initial set of content objects analogous to the abovementioned relation search. In this case, content objects are obtained that are similar to the set of given content objects. Similarity is defined by restrictions on *content-related element*s such as the content itself, its content type, original metadata and further content attributes as well as the content provider and – in the case of remote content objects – the contact the content object originates from.

Fully automatic acquisition of content can then be achieved by automatically processing *similarity search* in the search outside scope and suggesting content objects to be added to the repository of the user-centered social software. The initial set of local content objects can be chosen for example according to explicit ratings that represent the individual user's opinion. Further aspects include user behavior monitoring to infer implicit ratings on content objects as future research directions (see Chapter 6).

Content objects that should be used in such initial sets of similarity search for comparison purposes are obtained for example considering the task contexts the content object was used in or the relative frequency of read and write access. Negative examples are recognized by low attention of the individual user on the respective content object: no annotations and rating, sparse connections to other content objects or a relatively low frequency of access.

Information Subscription

The *subscribe* process represents a semi-automatic mechanism of information acquisition where the user specifies subscribable content objects (e.g. content feed within a weblog). The items associated with a subscribed content object are then acquired automatically by the user-centered social software. Initialization of the subscribe process can be accomplished by the individual user in order to get notified about up-to-date content of contacts. Hence, one use case of this process is *subscribe to new content*. Another use case is mentioned below related to the receive feedback process.

The *receive feedback* process indicates the social aspect that the incoming information has been provided by one of the individual user's contacts and refers to a content that originates from the user. The feedback may contain a textual comment and optionally a rating of the content it is associated with. This process can be performed manually by the use case *get feedback*, where the user explicitly requests (new) feedback for a specific content object.

More efficiently, the receive feedback process can be combined with the subscribe process in order to automatically get notified about recent feedback from contacts. This combination is an automatic process represented by the use case *subscribe to feedback* which is available for any content object that is located on a social software platform and can be commented there.

In order to support the use cases *subscribe to new content* and *subscribe to feedback*, the predefined *subscription mark* is used on content objects with the two values *subscribed* and *not subscribed*. Whenever a single content object is subscribed to manually by the individual user, the subscribe mark

of the content object is set. The automatic application of the subscription mark is explained in Section 4.4.3.

A use case that can be applied to each information acquisition process is to create a local copy of a remote content object obtained by manual information acquisition, information search or subscription respectively which is maintained by the individual user or a contact on a social software platform. This use case is termed *copy content*. The local copy of the content object has to be managed by an appropriate personal application with regard to its content type and needs to be integrated into the repository of that application while the *copy content* use case is accomplished. For redundant storage of multiple versions of the content objects, synchronization is required in order to identify inconsistencies and support the propagation of updates (see Section 4.2.4).

4.2.2 Information Organization

Information organization encompasses two different kinds of sub processes, depicted in Figure 4.6 enclosed in two separate boxes. With regard to one cycle of the information flow, one part of the first kind of sub processes – depicted within a border in Figure 4.6 – can be completed on incoming information that has been interpreted by the individual user. The other part of these sub processes can be accomplished after evaluating the respective incoming information. Information use that involves lifecycle management of content objects encompasses the sub processes shown within the white box in Figure 4.6 and is further supported by content providers (see Section 2.2.2).

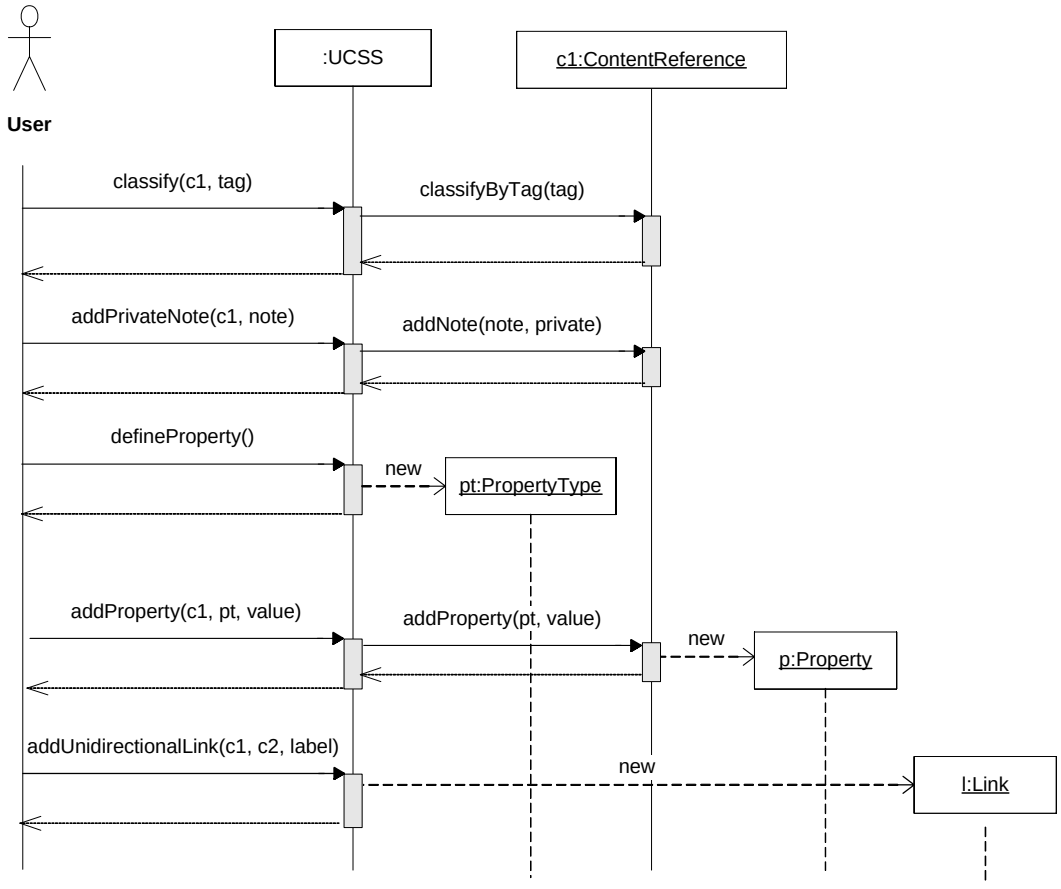


FIGURE 4.8. *Typical scenario of enhanced information organization*

Figure 4.8 depicts a typical scenario of information organization with selected use cases and aims at illustrating support of user-defined metadata for content objects and links between content objects. The individual user first classifies a content object by a tag which is identified by its content reference in the user-centered social software. The content object is further annotated for private use and extended by a property whose property type has been defined in advance. Finally, an ad-hoc link between two content objects is added.

The use cases of information organization that are described in detail in the following subsections can be accomplished either manually as shown in Figure 4.8 or automatically. In order to automatically apply these use cases, rule-based support is necessary which is discussed in Section 4.4.3.

Information Interpretation

Information interpretation is a knowledge-based process that can be supported by the following appropriate information-based processes and their use cases.

The *classify* process refers to the assignment of content objects to designated contexts also shown in the example scenario of Figure 4.8. Hence, this abstract process is referred to as *classify by context* and its concrete use cases include *categorize*, *classify by tag*, *put into folder* or *put into directory* respectively. Other use cases result from the available classification schemes of the associated content providers.

Content objects can further be classified by means of the user-centered social software in order to be extended by additional context information. The latter classification may also take effect at the content provider if the related contexts of a content object are posted there (see Section 4.2.3) – provided that a representation of the respective classification scheme is available at the content provider. For example, a content object located at a social software platform (where tagging is enabled) could be extended by additional tags in the user-centered social software. The individual user can then choose to post these tags to the social software platform along with the content object. The *remove context* use case facilitates the removal of contexts from a content object.

During the *link* process, a relation between two content objects is established either manually or automatically. The benefit of this process lies in the fact that content objects located at different content providers can be related to each other with user-centered social software support. Additionally, in the scenario of a personal application that does not provide flexible enough networking of the managed content objects the required links can also be managed by the user-centered social software. Manual creation of links along with the definition of specific link types allows the user to establish user-defined semantic relations between content objects. The application of the link process results in a network of content objects that is managed by the user-centered social software as described in Section 4.1.1.

The use cases of this process are *add unidirectional link* or *add bidirectional link* between two content objects that determine how the established link can be traversed in the network of content objects (see example in Figure 4.8). Bidirectional links are reasonable if the semantics of the link are equal in both directions using symmetric link types. The *add inverse link* use case can be applied to specify the reverse direction of a formerly created unidirectional link. Links can be specified with regard to a link type (see Section 4.1.1 for details). The definition of a link type is denoted by the use case *define link*. As an alternative, user-defined ad-hoc links can be established between content

objects that are merely defined by the semantics of their label. Links are removed by applying the *remove link* use case.

During the *extend* process, properties managed by the user-centered social software are added to a content object that originates from any associated content provider which is exemplified in the scenario in Figure 4.8. This process represents an extended metadata management by the *add property* use case since properties that lack an appropriate representation at the content provider can nevertheless be added to content objects. It is necessary to define property types in advance – with regard to their value ranges – that can then be added to content objects of a specific type which results in the use case *define property*. In case of an extension of the original metadata representation, the extend process can be combined with the update process (see Section 4.2.3). *Remove property* refers to the use case of removing a property from a content object.

Information Evaluation

Information evaluation as a knowledge-based process is supported on the level of information flow by the processes and use cases introduced below.

The *annotate* process refers to adding textual notes to a content object after evaluation. The notes are either private and only visible for the individual user or otherwise accessible for selected contacts at content providers. One special use case of this process is therefore *add private note* depicted in the example in Figure 4.8 whereas the use case *add note accessible to contacts* depends on the specified contacts. The annotate process can thus be effectively combined with the share and provide feedback processes (see Section 4.2.3). Notes can be removed by applying the *remove note* use case.

In the *mark* process, a content object is labeled with a mark that corresponds to a specific characteristic of content objects. An essential use case is *define mark* where the scale and values of a mark are determined along with the specification of content objects – in general by content type – that can be labeled by that mark, which then enables the application of the *add mark* use case. Furthermore, marks can be removed by applying the *remove mark* use case.

The *rate* process assigns a value – that is a rating, as explained in Section 4.1 – based on a defined ordinal scale to a content object. The rating is therefore a special mark that directly relates to the social aspects of evaluating a content object by the individual user. This process is composed of the three corresponding use cases *define rating*, *add rating* and *remove rating* in analogous manner to the mark process. The only restriction for the *define rating* use case lies in the fact that an ordinal scale has to be used in order to define appropriate rating values. The combination of the rate process with the share process (see Section 4.2.3) results in a use case that facilitates the user to share his opinion with his contacts.

Information Use

The (re-)use of information is addressed by the information-based processes constituting lifecycle management of content objects. The associated processes are *create*, *modify* or *delete* a content object and *transform* it into a different format. In contrast to the processes and use cases described above, these processes are not necessarily supported by the user-centered social software but need to be accomplished by personal applications and information organization services of social software platforms that are devoted to lifecycle management of content objects.

Specialized services to handle different content types – like for example services for image editing and transformation of images into different formats – are not integrated into user-centered social

software. The scope of this software family is rather a flexible cross-platform management of content objects and integration of services for information organization encompassing the social contexts of the individual user.

Besides the aforementioned processes, information use involves continuous application of the use cases that are part of information interpretation and evaluation also later in time when it becomes appropriate in the user's task context (see Section 4.1.1).

4.2.3 Information Dissemination

In Figure 4.6, the sub processes of information dissemination are illustrated below the arrow on the right. These processes represent the handling of outgoing information to social software platforms, since information managed by personal applications is not intended to be accessed by contacts in the user-centered social software model.

Figure 4.9 illustrates a typical scenario of information dissemination. The individual user applies the *keyword search* use case in order to retrieve a content object managed by the user-centered social software (see Section 4.2.1). The content object is then published at social software platform s, i.e. a new content object is created on the platform, and the individual user subscribes to notification of new feedback for that content (see *subscribe to feedback* use case in Section 4.2.1). When a contact provides feedback to the content object the individual user is immediately notified.

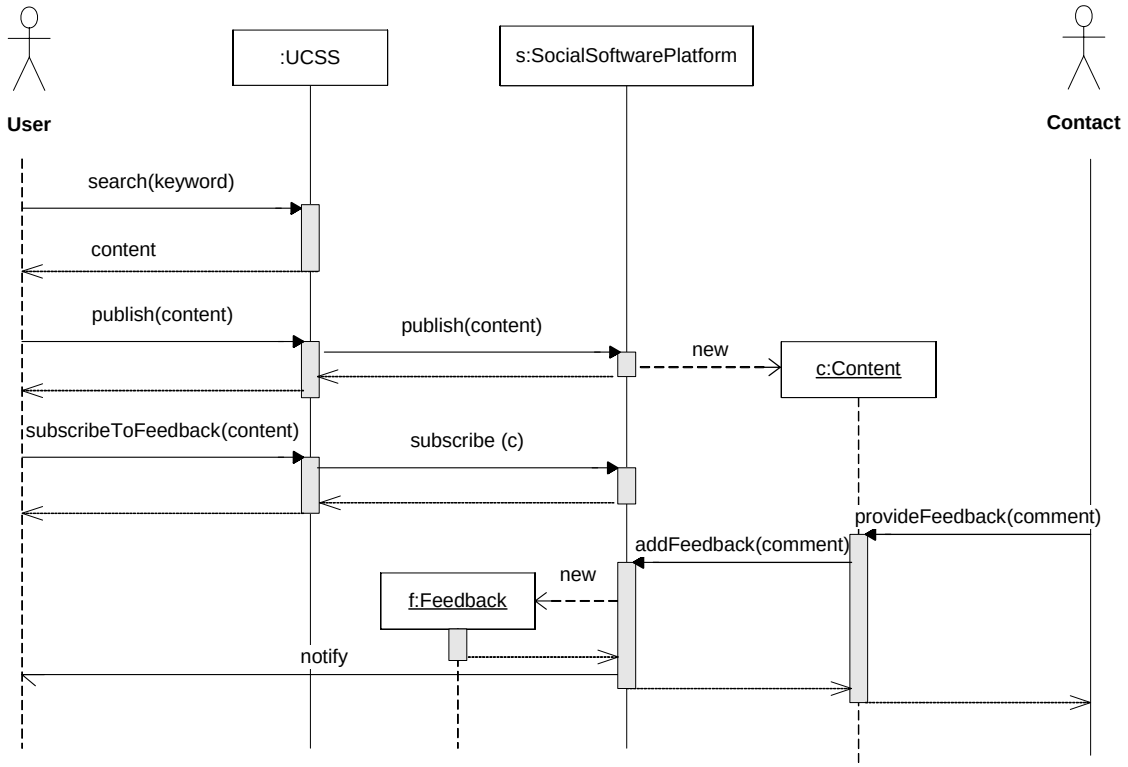


FIGURE 4.9. *Typical scenario of information dissemination*

Similar to information acquisition, the sub processes of information dissemination can also be executed manually or semi-automatically respectively according to the individual user's needs. The processes and use cases of information dissemination form two categories according to how content is handled, that is disseminated, at social software platforms.

Use cases that do not imply the involvement of contacts in accessing the individual user's information located at social software platforms constitute the category of information storage use cases. The second category denotes use cases that are directly related to the dissemination of content to contacts or platform users and is thus referred to as information publication.

Information Storage

The *post* process refers to storing a local content object – managed by a personal application – additionally on a social software platform. The *post as private* use case can either be applied to post a local content object to a social software platform. Further, a remote content object that was already posted to a social software platform can additionally be posted to another one. After this use case of the post process is accomplished, the content object posted at the respective social software platform(s) can later be published or shared with contacts as referred to by the use cases of the share process below.

This means, that multiple versions of semantically equivalent content objects are located at different content providers – at least in the repository of a personal application and additionally located at one social software platform. Hence, the requirement of tracing equivalent content objects and synchronizing them arises which is discussed further in Section 4.2.4.

The *move* process implies that a content object is posted to a social software platform whereas it is deleted at its original location in contrast to the post process. In this case, there is no need for synchronization of content objects. The move process may affect a local content object by the use case *move local* which is then moved to a social software platform. Otherwise, a remote content object is moved in the *move remote* use case to another social software platform and deleted from the original one. By contrast, moving a remote content object (again) to a personal application is represented by the copy process of information acquisition (see Section 4.2.1).

The *update* process refers to modifying the metadata of remote content objects by overwriting them by content attributes defined in the user-centered social software. This process is available in case the metadata representation of the content object at the social software platform is already applicable for that purpose. Otherwise, this process is enabled if the metadata representation of the content type the respective content object belongs to has been modified or extended at the social software platform in the appropriate way. The use case of this process is thus referred to as *update metadata*. By contrast, extensive modification of the content and metadata of a content object is addressed by the modify process (see Section 4.2.2) that is supported as a service of the respective content provider.

Information Publication

In the *share* process, a content object is posted to a social software platform in order to be made accessible to selected contacts. For this reason, the share process is supported by the definition of views for content objects and the composition of roles from the former (see Section 4.3). The share process first encompasses the *publish* use case which is defined as posting a local content object to a social software platform in order to make it publicly accessible which is exemplified in the scenario in Figure 4.9. Analogous to the *post as private* use case above, *publish* also refers to publishing an already remotely available content object at another social software platform.

The second use case of this process is *share with contacts* which refers to posting a content object to a social software platform for being accessed merely by contacts that are specified by the individual user. Depending on the selected contacts and the social software platform they are members of, this

use case can be applied several times for a single content object in analogy to the *publish* use case. In contrast to the latter, access to the shared content object can, though, be controlled on a fine-grained basis.

In order to consistently support the use cases of the share process, again synchronization is required (see Section 4.2.4 for details).

The *provide feedback* process refers to the social aspect of posting a comment as a textual note to a social software platform and attaching it to a content object of a contact (see Figure 4.9). The textual comment as feedback may be extended by a rating if this way of providing feedback is enabled by the social software platform. The *give feedback* use case refers to publishing a newly created comment (and rating) to the social software platform. By contrast, during the *post note as feedback* use case, a content object of a contact that was annotated by the individual user – performing the use case *add note accessible to contacts* (see Section 4.2.2) – can be commented using that previous note. The *post rating as feedback* is similar to the last use case in that a formerly created rating that is managed by the user-centered social software is posted to the social software platform the content object originates from. This use case is only available if the respective social software platform provides the rating of content as mentioned above.

By providing feedback for specific content objects of a contact, the individual user as the author of the feedback has no influence on the accessibility to his feedback any more; therefore, the feedback object is in this case published to the social software platform, that is, it needs to be considered as publicly available (see above).

A feedback object as a content attribute of the respective content object can itself be handled as a content object if its reference is included in the repository of the user-centered social software.

Information Acquisition	**Information Organization**	**Information Dissemination**
Manual Information Acq. *get by URI* *browse* Information Search *keyword search* *context search* *contact search* *full text search* *search related to any (C)* *search related to all (C)* *similarity search (C)* Information Subscription *subscribe to new content (p)* *get feedback (c)* *subscribe to feedback (c)* *copy content (c)*	Information Interpretation *classify by context (C)* *remove context (C)* *define link* *add unidirectional link (c)* *add bidirectional link (c)* *add inverse link(c)* *remove link (c)* *define property* *add property (c)* *remove property (c)* Information Evaluation *add private note (C)* *add note accessible to contacts (P)* *remove note (C)* *define mark* *add mark (C)* *remove mark (C)* *define rating* *add rating (C)* *remove rating (C)*	Information Storage *post as private (C)* *move local (C)* *move remote (C)* *update metadata (c)* Information Publication *publish (C)* *share with contacts (C, P)* *give feedback (c)* *post note as feedback (c)* *post rating as feedback (c)*

TABLE 4.1. *Use cases of social information management*

Table 4.1 gives a summary of all use cases introduced above, related to information acquisition, organization and dissemination managed by the user-centered social software. The use cases are structured according to the sections in which they have been described. Parameters of the use cases that are related to content objects or contacts are given in brackets. A single content object is denoted by a lowercase letter *c* whereas a set of content objects is represented by a capital *C*. The same holds for contacts depicted by *p* and *P* respectively.

4.2.4 The Synchronization Process

In order to achieve consistency when trying to support the aforementioned social information management processes by user-centered social software, synchronization of content objects plays a significant role. Synchronization is defined as the consistent update of semantically equivalent versions of a content object at different content providers.

With regard to social information management, the *copy content* use case of information acquisition and the uses cases of the post and share processes of information dissemination need to be extended by the synchronization process. The respective synchronization scenarios are illustrated in Figure 4.10 below. Content objects (depicted by squares) owned and managed by the individual user are on the one hand located on the personal device and on the other hand at the individual user's areas at social software platforms indicated by dashed ellipses. Remote content objects of contacts reside at the social software platforms outside the individual user's areas (see Section 3.3.2).

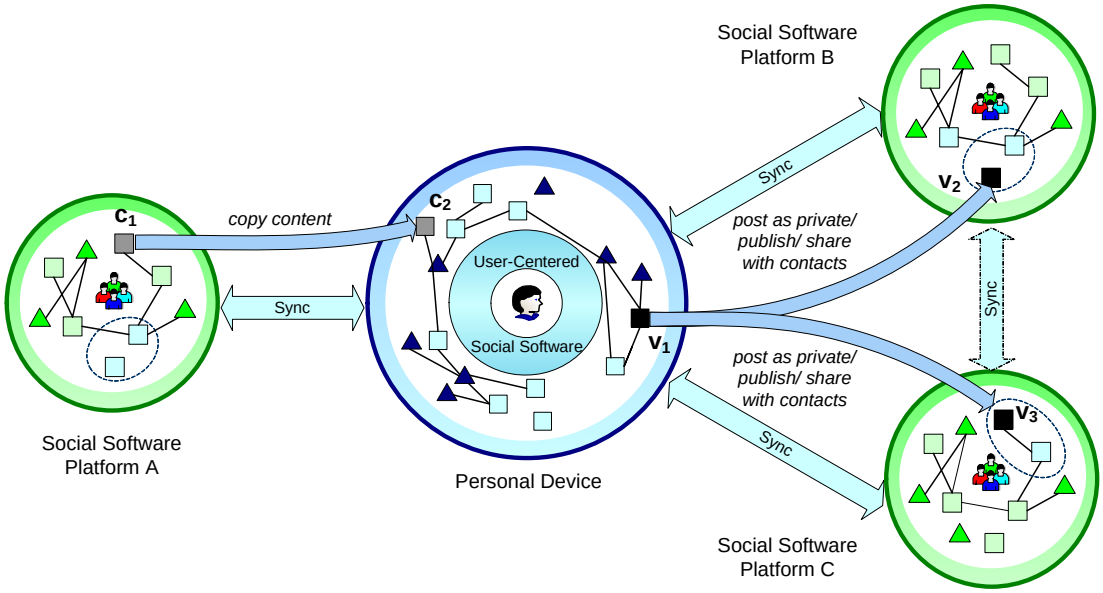


FIGURE 4.10. *Synchronization scenario for particular use cases of social information management*

The copy content use case is pictured by the content objects referred to as c_1 and c_2 respectively. In this use case, the remote content object c_1 is copied from social software platform A to the personal device as a local content object c_2. Since two equivalent versions of this content object exist, they need to be synchronized at later points in time if consistency is required. Nevertheless, the individual user may wish not to synchronize the local version of the content object but keep the version obtained in the *copy content* use case instead and even modify it locally. In general, the individual user cannot overwrite the remote content object with a locally modified version since the remote object is

managed by a contact who only has write access to it. The example in Figure 4.10 accordingly illustrates that the content object c_1 is not located in the individual user's area marked by the dashed ellipse and thus, during possible synchronization, c_2 is always updated.

The use cases of the post and share processes, that is *post as private*, *publish*, and s*hare with contacts* differ merely in the social aspect of access authorization to content objects but can be handled equivalently when applying the synchronization process. The example scenario for the involvement of the synchronization process in these use cases is pointed out by the content objects shaded in black in Figure 4.10. The local content object in the original version v_1 is – without loss of generality – shared with contacts on social software platform B as a semantically equivalent remote content object in version v_2. A need for synchronization arises from the application of all use cases according to the associated label between v_1 and v_2 in Figure 4.10.

Subsequent application of the use case *publish* – or *post as private* or *share with contacts* respectively – is depicted by the arrow between versions v_1 and v_3 of the content object. This in turn results in a need of the accomplishment of the synchronization process if the content objects are modified independently on the personal device and social software platform C. Since at that moment three versions of semantically equivalent content objects exist, synchronization is also required between social software platforms B and C indicated by a dashed bordered arrow in Figure 4.10.

Such a ternary synchronization scenario of the versions v_1, v_2 and v_3 of the content object can be realized by a reduction to two binary synchronization processes controlled by the user-centered social software on the personal device, however. This also holds true in the scenario of a content object that needs to be synchronized on an arbitrary number of content providers (see [Le02]).

Analogous to the previously described *copy content* use case, the individual user decides whether synchronization shall be performed and if this is the case, which content objects – and implicitly content providers – engage in synchronization. Especially in the *publish* and *share with contacts* use cases, the user might intend to share different versions of a content object with different contacts.

Synchronization of Multiple User-Centered Social Software Instances

Besides use case support, synchronization is further essential in the scenario where the individual user has access to multiple user-centered social software instances on several personal devices in addition to the social software platforms he is member of.

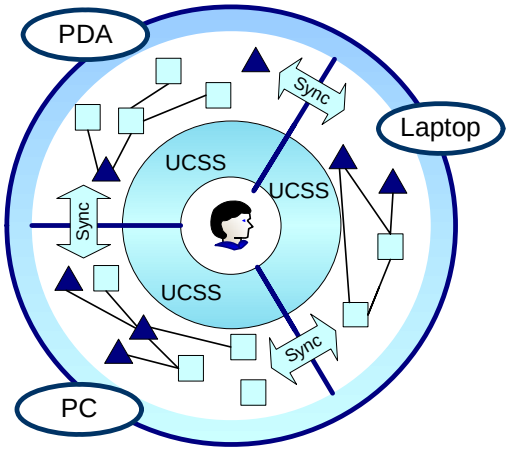


FIGURE 4.11. *Synchronization scenario for multiple user-centered social software instances*

The information repositories of the user-centered social software considering newly added content references and content attributes, task contexts and portfolios need to be held in a consistent state on each personal device. This requirement results in the need for synchronizing the repositories of multiple user-centered social software instances the individual user is working with. The synchronization scenario for this case is illustrated in Figure 4.11.

This scenario hence includes the execution of the synchronization process between stationary and mobile devices as content providers.

Synchronization Aspects

Two scenarios concerning user-centered social software exist that require synchronization of content objects as described above. First, when using a single instance of the user-centered social software and applying certain use cases, inconsistencies may occur between semantically equivalent versions of a content object located at different content providers. Second, the scenario of using multiple instances of the user-centered social software on several personal devices also requires synchronization.

The most significant aspects to be taken into consideration are the determination of synchronization control including the roles of the content providers, the synchronization strategy and the identification of semantically equivalent content objects at the content providers. According to [Le02], for each execution of the synchronization process, the active system – one of the user-centered social software instances – that controls the synchronization process needs to be determined since the roles of the systems may change.

Synchronization control also refers to specifying at which particular points in time (e.g. predefined intervals) the synchronization process is performed. The synchronization strategy refers on the one hand to the direction of the synchronization, that means it needs to be determined at which content provider the content objects are updated. On the other hand, it has to be specified whether synchronization is performed incrementally based on the modifications in the meantime between two executions of the synchronization process between two content providers or by full update [Le02].

The user-centered social software is capable of simply identifying semantically equivalent versions of a content object. This is due to the fact that as a pre-condition, content references managed by the user-centered social software are required to be uniquely identifiable by a URI according to the specification in Section 4.1. Further, since the application of all use cases is controlled by the user-centered social software, it can immediately keep track of equivalent versions of content objects that emerge.

When a set of content objects being also interconnected by several links is posted to content providers, transitive synchronization traversing links is required ([Le02] and [MaLe02a]). Therefore, to achieve a consistent state of content objects at the involved content providers, in the synchronization process, not merely single content objects need to be considered but a network of content objects instead.

Further issues address conflict solving strategies that occur under certain circumstances, for example if both versions of the content object to be synchronized have been modified independently since the last execution of the synchronization process. This scenario in general needs user interaction in order to determine how to achieve a consistent state of both versions of the content object, especially if a merging of the content should be applied. The steps of the synchronization process to be performed on information objects in an information system including user interaction for final decision control are discussed in detail in [Le02].

The specification of synchronization rules facilitates automatic update of equivalent versions of a content object. Rules then specify which content objects – or on a more fine-grained basis content attributes – located at a specific content provider are updated and thus overwritten by content originating from another content provider. By applying rules to the synchronization process, user interaction can be reduced. Synchronization rules are not further considered since they are beyond the scope of this thesis.

4.3 Role-Based Authorization Model

The role-based authorization model of user-centered social software is required to appropriately support the use cases of information dissemination. As described in the subsequent sections, the proposed model relies on the Role-Based Access Control (RBAC) model [FCK95] in order to adapt to the authorization models of social software platforms explained in Section 3.3.2. According to the RBAC model, roles allow a grouping of access rights represented by views on content elements. Contacts assigned to a role thus obtain the appropriate views on content elements which have been determined by the individual user.

4.3.1 Definition of Views

The basic concept for authorization control is established by views on content elements, i.e. content objects represented by content references and content attributes, as introduced in Section 4.1. In the scope of user-centered social software, authorization by read access to content elements is considered.

In specific cases, write access to content objects managed by the individual user may also be required for contacts like in the case of a teamlog (see Chapter 3.2.2). However, the consideration of write access is less significant since this software family focuses on the support of social information management from the individual user's perspective instead of cooperation and collaboration support of individuals or groups.

Hence, the authorization to be expressed by views refers to read access to content elements the individual user intends to maintain, publish and share at social software platforms. The support of information dissemination use cases by views in order to specify the accessibility of content elements is discussed in the following.

The view on a content element is determined by the particularly defined values *private*, *public* or *role-based* of *accessibility mark*s. Thus, the accessibility of content elements can be specified by applying the *add mark* use case.

A content attribute itself as a content element cannot be decomposed any further. Thus, the view on a content attribute is specified by the value of a single accessibility mark. Content attributes marked *private* may only be accessed by the individual user while those marked *public* are specified as accessible by any contact. The accessibility mark set to *role-based* implies that a content attribute is visible if the respective content object is included within a role (see Section 4.3.2).

The view on a content reference determines the accessibility of the original content and also includes the views on its associated content attributes.

The composition of the view on a single content object is illustrated in Figure 4.12. The example shows content attributes of different kinds that are associated to the content reference object by directed arrows. Further, the view on the content object and each content attribute is represented by the

assigned border color of the respective box. While the color red indicates that the accessibility mark is set to *private*, the color yellow denotes the value *role-based* of the accessibility mark and the color green is used to depict content attributes that may be publicly accessible.

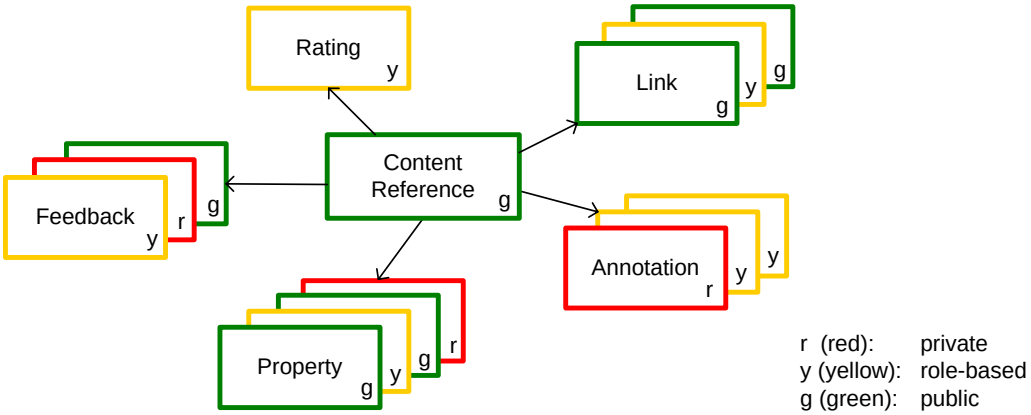


FIGURE 4.12. *Composition of the view on a content object*

4.3.2 Definition of Roles

In the authorization model of user-centered social software, a role encompasses a set of views where each view describes the accessibility to the content attributes of one content object in order to achieve fine-grained access control.

The *effective view ev*(c) on a content object c within a role is given by the set of content attributes as a subset of all associated content attributes *A*(c) that are either marked public or role-based according to (4.1).

$$ev(c) := \{d \mid d \in A(c) \wedge (d.accessibility = public \vee d.accessibility = role\text{-}based)\} \quad (4.1)$$

As illustrated in Figure 4.13, role r_1 for example includes the effective views on content objects c_1 and c_2 while role r_3 incorporates the effective views on all depicted content objects.

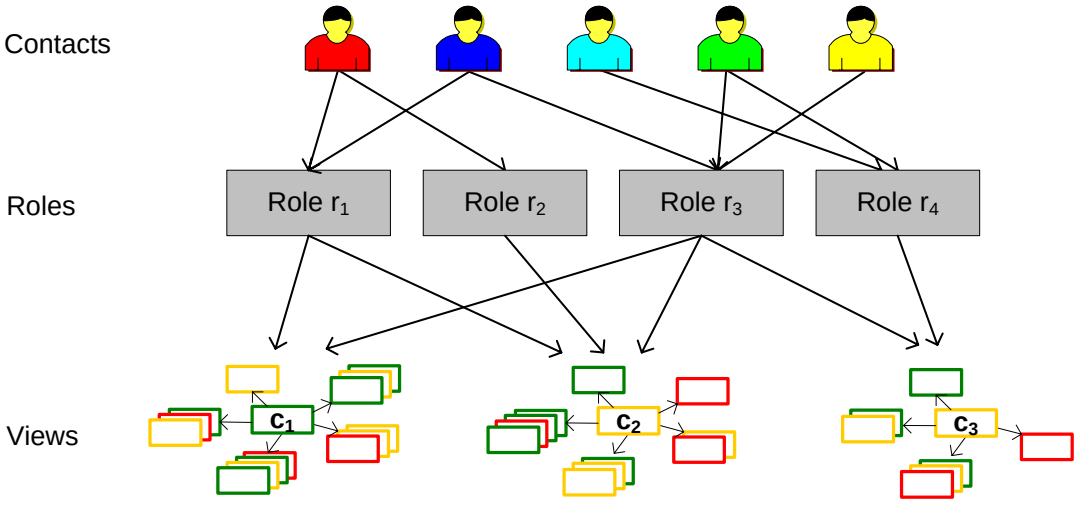


FIGURE 4.13. *Definition of roles*

The individual user's contacts are then assigned to the specified roles while a contact may obtain multiple different roles as further shown in Figure 4.13. The set of content attributes C contact p has read access to is therefore defined as follows in (4.2) expressed by the assigned roles r_i and the effective views on content objects c_j :

$$C_p := \bigcup_i r_i \quad \text{where} \quad r_i = \bigcup_j ev(c_j) \tag{4.2}$$

Figure 4.14 illustrates the mapping of user groups to roles in the individual user's area on a social software platform extending Figure 3.16. All contacts who are assigned to a user group should receive the same role for proper support of information dissemination use cases. The semantics of a role and a user group differ merely in their perspective such as a role refers to access rights for a single individual whereas a user group represents multiple individuals to whom access to content objects is assigned.

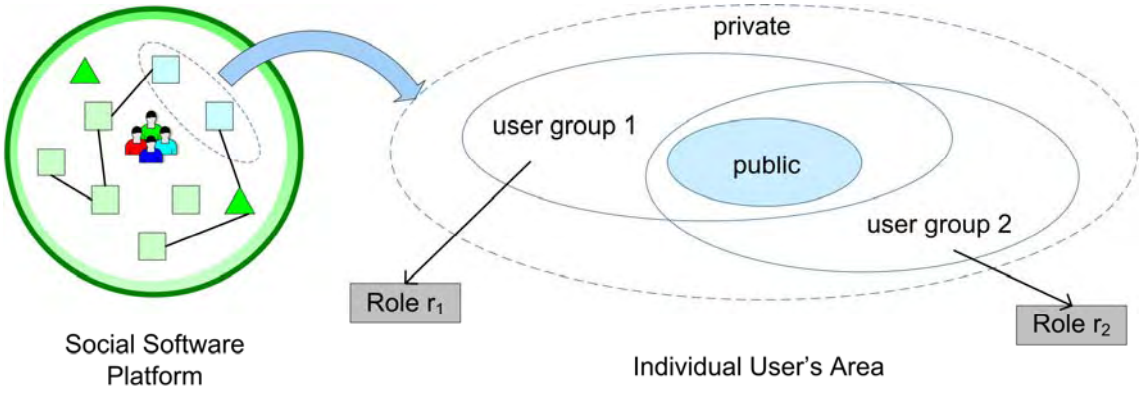


FIGURE 4.14. *Mapping of user groups to roles*

Significance of Roles for Information Dissemination Use Cases

The accessibility of a content object for contacts is significant when information dissemination use cases are applied.

A content element that is marked private cannot be posted to non-private scopes of the individual user's area on social software platforms and is therefore relevant for the information dissemination use cases *post as private* as well as for the use cases *move local* and *move remote*.

Role-based authorization facilitates the application of the use case *share with contacts*. When this use case is performed, contacts may access the content objects at the designated area of the respective social software platform. All these contacts need to possess roles according to which each of the content elements to be shared is accessible.

In order to make content elements publicly accessible by accomplishing the use case *publish* they are marked as *public*.

Role Hierarchy

The authorization model of user-centered social software facilitates the specification of a role hierarchy in order to let more specialized roles comprise more general ones.

In the scope of user-centered social software, role hierarchies for contacts strongly depend on the respective social context of the individual user who may define the roles family member, close friend, acquaintance and co-worker, for example, that are assigned to his contacts.

Concerning this example, it makes sense to define a simple hierarchy for the roles in the private context of the individual user as shown in Figure 4.15. The role family member is a specialization of the role friend in order that family members be allowed to access all content objects visible in the latter role. Further, the role acquaintance is more general than the role friend. This fact results in the implicit visibility of content objects that have been made accessible to acquaintances also by contacts assigned to the role friend.

Further, content objects to be disseminated to co-workers from a professional context of the individual user typically differ from those in the private context. Therefore, the role co-worker is omitted from the hierarchy of roles that was specified for the private context.

By applying a hierarchical definition of roles, the assignment of multiple roles to contacts can be avoided within social contexts of the individual user as shown by the above example.

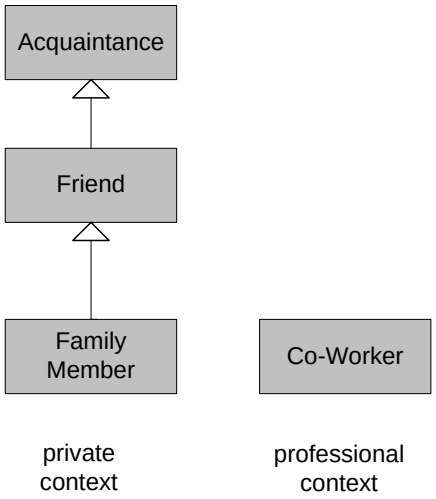


FIGURE 4.15. *Example of a role hierarchy in social contexts*

4.3.3 Identification of Contacts

Contacts have to be identified on social software platforms in order to correctly accomplish information dissemination according to the specified roles. Identifying a contact who is member of several social software platforms where he typically uses different pseudonyms is a complex task.

This issue can be addressed by record linkage methods that are designed to identify semantically identical data – which are content objects in the scope of user-centered social software – in either different repositories or within the same repository. Significant application areas include data integration, data cleansing and synchronization processes. The linkage flooding algorithm which is outlined in [LMR04a] and [LMR04b] and discussed in detail by Riedel [Ri03] is an example of a record linkage method that focuses on identifying semantically identical objects by analyzing their metadata and relations to other content objects.

Since information on a contact is considered as a content object representing the user profile, record linkage methods can be applied to identify identical contacts with slightly differing user

profiles on several social software platforms. In order to apply such methods to support the assignment of roles in the user-centered social software, the user profiles need to be accessible, however.

Otherwise, identification of contacts also needs to be taken into consideration when contacts are specified by the individual user whom he intends to share certain content objects with. In this case, social software platforms have to be determined which the designated contacts are members of. This scenario involves the examination of the roles of contacts who may implicitly access those content objects intended to share only with a specified contact. Therefore, it is unlikely the user-centered social software would be used for this purpose by the individual user. Peer-to-peer communication services instead can be applied more efficiently in this scenario.

4.4 Rule-Based Social Information Management Support

For appropriate support of social information management use cases elaborated in Section 4.2, the user-centered social software model incorporates the definition of different kinds of rules that specify the business logic of the aforementioned use cases. Parameterization of rules further facilitates individually customizable social information management support from the user's perspective.

By this method, acquired (incoming) and disseminated (outgoing) information is filtered. Concerning information organization, rules are required to facilitate automatic application of the respective use cases.

According to [Pa06], in rule-based systems different types of rules exist represented by *normative rules*, *deductive rules* and *reactive rules*. In addition, *constructive rules* are further distinguished by [BrMa05] which refer to data selection while normative rules describe integrity constraints on content objects. *Deductive rules* or *inference rules* represent a relation where *conclusions* can be deduced from the specified *premises*. Reactive rules – also termed *Event-Condition-Action (ECA) rules* – are defined in order to trigger an action in case the specified event occurs. By applying *active rules* that represent a special case of reactive rules, an action can be determined without reacting to an event [Pa06].

The specified rules can further be divided into two distinct categories with regard to social information management processes. Rules are either generally specified for all use cases of a process or need to be use case-dependent on a more fine-grained basis.

Parameterization of rules is accomplished according to the individual user's requirements where appropriate. The configuration of parameters is either performed manually by the user or accomplished automatically based on formerly observed user behavior.

The several kinds of rules for social information management support are discussed in the subsequent sections. For formal specification of rules, content objects are denoted by the letters c and d while contacts are referred to by the letter p and the individual user is denoted by u. Expressions to be parameterized are shown in italics. Further, the dot notation is used for attribute selectors.

4.4.1 Content-Related Rules

Rules that can be applied to content-related elements of content objects are referred to as *content-related rules* in the scope of user-centered social software.

Content-related rules can be completely parameterized by the individual user in a flexible manner. Two levels of parameterization abstraction exist for this purpose in the user-centered social software model. On the first level, a *rule schema* determines which content-related elements are restricted by

that content-related rule. On the second level, the *rule instantiation* specifies typed parameter values from set P for the restricted elements in the respective rule schema.

The elements encompass content-related information represented by original metadata from the content provider, content type, content attributes, context, views and – in the case of contact objects – also roles. The model of these interrelations with regard to content-related rules is illustrated in Figure 4.16.

In the following, rule schemata are denoted in italics while rule instances are marked by sequence numbers. Parameter values for rule instantiation are emphasized by single quotation marks if necessary.

Specific kinds of content-related rules applied in the user-centered social software model, that is, restriction rules, inference rules and similarity rules are discussed in the subsequent sections.

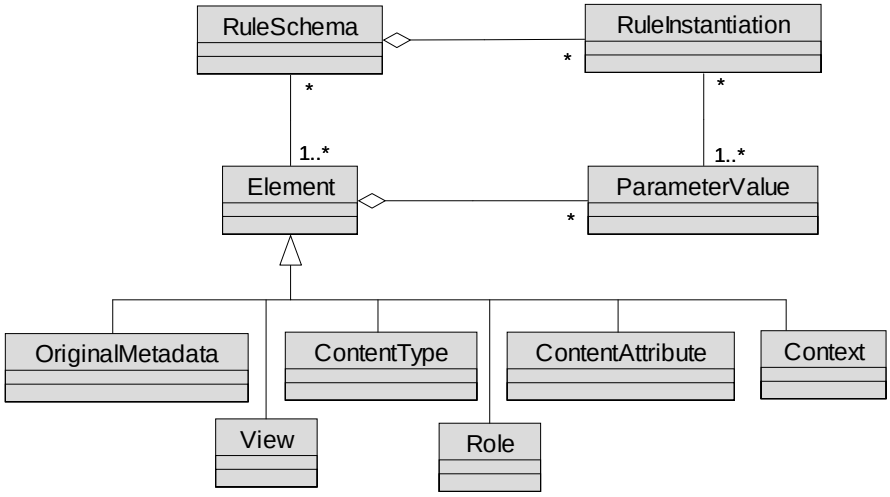


FIGURE 4.16. *Model of content-related rules*

Restriction Rules

A *restriction rule* is a constructive rule based on *content-related elements* as shown in Figure 4.16 and specifies a set of content objects that fulfill this restriction rule. Thus, a restriction rule can be efficiently realized using queries.

Restriction rules are denoted by R in the following. The boolean-valued function R(c) based on restriction rule R and parameterized by content objects c is given in (4.3).

$$R(c) = \begin{cases} \text{true}, c \in R \\ \text{false}, c \notin R \end{cases} \tag{4.3}$$

In the following example, a parameterized restriction rule is given as a textual expression according to a rule schema resulting in the rule instantiation R_1.

R_1 := content objects of content type ‘image’ with rating > ‘4’ and all annotations ‘marked private’

Inference Rules

Explicit links between two content objects can be deduced from *inference rules* which are based on content-related elements analogous to restriction rules. Inference rules especially take into account existing links including the characteristics of the link type. Inference rules are denoted by *N* and specified according to (4.4) where *N*(c,d) is a boolean-valued function.

$$N(c,d) := c\ \textit{relatedTo}\ d \tag{4.4}$$

The relation *relatedTo* in (4.4) is an arbitrary relation that is specified as a link type in the user-centered social software while *relatedToInv* denotes the corresponding inverse relation. The relation also determines the content types of c and d which need to correspond to the domain and range of the link type according to Section 4.1.1. Further, the following general rule schemata for inference rules are deduced from the algebraic properties of link types, i.e. transitivity, symmetry and asymmetry on the one hand and the rule schema for inverse links on the other hand given in (4.5) – (4.7) according to [NeMa98].

$$\text{Transitivity: } N_T(c,d) := c\ \textit{relatedTo}\ d \wedge d\ \textit{relatedTo}\ e \Rightarrow c\ \textit{relatedTo}\ e \tag{4.5}$$

$$\text{Symmetry: } N_S(c,d) := c\ \textit{relatedTo}\ d \Rightarrow d\ \textit{relatedTo}\ c \tag{4.6}$$

$$\text{Asymmetry: } N_A(c,d) := c\ \textit{relatedTo}\ d \Rightarrow \neg(d\ \textit{relatedTo}\ c) \tag{4.7}$$

$$\text{Inverse link: } N_V(c,d) := c\ \textit{relatedTo}\ d \Rightarrow d\ \textit{relatedToInv}\ c \tag{4.8}$$

An example for a simple inference rule according to (4.4) is given by N_1(c,d) that can be used to automatically establish an inverse link. This rule can be applied given that the domain of the link type *isAuthor* is specified as contact and the range is established as any content type.

$$N_1(c,d) := d\ \textit{isAuthor}\ c \Rightarrow c\ \textit{createdBy}\ d$$

Content objects can further be linked automatically by *implicit links* that are inferred from rules in order to support other use cases and do not necessarily take effect persistently in contrast to *explicit links*. This kind auf automatically created links is referred to by the general term *linkedTo* which does not imply any semantical meaning of the relation.

The following example of an instantiated rule schema N_2(c,d) clarifies the concept of links that are inferred on demand creating non-persistent implicit links that are neither based on link types. The content types of c and d are contact and e-mail respectively. In the set of parameter values P = {e, p, t} e represents a content object of type document and p is a contact while t defines a date literal.

$$N_2(c,d) := e\ \textit{isAttachedTo}\ d \wedge c\ \textit{isAuthor}\ e \wedge \neg\ p\ \textit{isAuthor}\ e \wedge e.\text{createdDate after } t \Rightarrow c\ \textit{linkedTo}\ d$$

Similarity Rules

Similarity rules are inference rules by which the relation *similarTo* can be deduced for content objects. Therefore, a *similarity rule* relies on content-related elements and operates on two content objects in order to determine whether they are similar.

Similarity rules in the scope of user-centered social software intend to not merely specify similarity of content objects based on their content but to include restrictions on further content-related elements that refer to the social context of the individual user.

For content objects c and d a similarity rule is denoted as S(c,d) and parameterized according to the similarity of content-related elements. S(c,d) is a boolean-valued function; if S(c,d) evaluates to true content objects c and d are similar and thus related by *similarTo* as given in (4.9).

$$S(c,d) \Rightarrow c \; similarTo \; d \tag{4.9}$$

Similarity defined by a similarity rule S(c,d) is a symmetric and transitive relation, that is formulas (4.10) and (4.11) hold for any valid similarity rule with regard to content objects c, d and e.

$$S(c,d) \Leftrightarrow S(d,c) \tag{4.10}$$

$$S(c,d) \wedge S(d,e) \Rightarrow S(c,e) \tag{4.11}$$

The set of content objects that are similar to content object c is denoted by $S(c)$ as defined in (4.12). Subsequent application of the similarity rule on elements $d \in D$ that are similar to c results in the transitive closure of c concerning similarity.

$$S(c) = \{d \mid S(c,d)\} \tag{4.12}$$

The following representative example of a similarity rule S(c,d) based on (4.9) illustrates the definition of similarity rules. In this case, two content objects are similar according to $S_1(c,d)$ if both have the same content type and either they have the same contact p as author or they share more than n contexts. Therefore, the set of parameter values results in P = {p, n} for rule instance $S_1(c,d)$. This similarity rule also relies on inference rule $N_2(c,d)$ as specified in the example above.

$S_1(c,d)$:= ((c.contentType = d.contentType) $\wedge$
(N_2(c,d) $\vee$ (p *isAuthor* c $\wedge$ p *isAuthor* d $\vee$ |{q | q *classifies* c $\wedge$ q *classifies* d}| > n)

Restriction rules and similarity rules are used as parameters in more specific rules for information acquisition, information organization and the rule-based definition of views and roles as explained in the subsequent sections.

4.4.2 Information Acquisition Rules

In order to determine the appropriate information sources – content providers and contacts – when working in a certain task context with the user-centered social software, rules are applied as shown on the left hand side in Figure 4.6. The rules that apply for information acquisition are denoted by A_i in the following with C specifying the set of content objects that can be acquired by the respective use case according to rule A_i.

(A_1) *Relevance for task context*: According to Section 4.1.2, content objects may only originate from content providers cp that are associated with the current task context X and also enabled in that task context for information exchange purposes. This rule is automatically parameterized according to the task context as given in the formal specification (4.13).

$$A_1 : C = \{c \mid c.cp \text{ associated with } X \wedge c.cp \text{ enabled in } X\} \tag{4.13}$$

(A_2) *Trustworthiness of information source*: Based on the considerations in Section 2.2.4, the individual user specifies contacts at content providers from whom he accepts acquired information in the given task context, that is *trusted contacts*. Therefore, according to this rule, only content objects

related to the individual user or trusted contacts are appropriate. This rule also implies that all content that is managed by personal applications is trustworthy.

Formally, the rule is applied to content objects and contacts as follows, where the set of trusted contacts is denoted by *T*.

$$A_2 : C = \{c \mid c \; relatedTo \; p \; \wedge (p \in T \vee p = u)\} \tag{4.14}$$

The individual user can parameterize this rule to define trustworthy information sources within a task context. The expression *relatedTo* in (4.14) may be parameterized by link types such as *managedBy*, *ownedBy* or *createdBy* for example. Particular contacts can be specified that fulfill certain requirements to be contained in set *T*.

The parameterization of *T* therefore results in the specification of trusted contacts similar to the following example using a restriction rule: Trusted contacts are those assigned to role r or being the author of more than n already referenced content objects with rating greater than m, that is $T := \{p \mid p \text{ assigned to } r \vee |(p \; isAuthor \; c \wedge c.rating > m)| > n\}$.

(A_3) *Content filtering*: Content-related restrictions for the use cases of the search process (see Section 4.2.1), i.e. selection of content objects, are parameterized by the individual user. In the search inside scope, restriction rules to all elements of content references managed by the user-centered social software are applicable as specified in Section 4.4.1.

The formal specification of a content filtering rule is given as a parameterized restriction rule that is to be fulfilled. Parameterization of this rule is ad-hoc in contrast to other information acquisition rules that are made persistent for later reuse.

$$A_3 : C = \{c \mid R(c)\} \tag{4.15}$$

An example for a parameterization according to (4.15) is the restriction of the individual rating of content objects to exceed a specified value m in order to be included in the search result in search inside scope, that is $R = \{c \mid c.rating > m\}$ in correspondence to Section 4.4.1.

(A_4) *Content subscription*: The individual user specifies content objects to be automatically acquired by the user-centered social software. This can either be new content of contacts or feedback to the user's own content from contacts according to the description in Section 4.2.1. Which content objects are subscribed to is either marked manually or automatically by the respective value of the predefined subscription mark (see Section 4.2.2).

Another possibility is to select content objects for subscription by restriction rules according to Section 4.4.1. Application of this rule overrides (A_2) when contradictory specified. The formal specification of the content subscription rule is given by (4.16) as follows.

$$A_4 : C = \{c \mid c \text{ marked } subscribed \vee R(c)\} \tag{4.16}$$

(A_5) *Relation of content*: This rule is applied to the use cases of relation search. It specifies on the one hand by which maximum path length *k* content objects may be linked indirectly in order to be considered as related. On the other hand, the relation of content rule is parameterized by link types in set *L* that need to connect content objects c and d along the path $P_n(c,d)$ in the scope of the relation search process. Formula (4.17) gives the general definition of content objects c and d being related in which case *R*(c,d) evaluates to true.

$$R(c,d) \Leftrightarrow \exists\, P_n\,(c,d),\ n \leq k : \forall\, c_i \in P_n(c,d),\ 0 \leq i \leq n : (c_i\ relatedTo\ c_{i+1} \wedge relatedTo \in L) \qquad (4.17)$$

Based on (4.17) the formal specification of the relation of content rule is given in (4.18) with regard to the *search related to any* and *search related to all* use cases respectively, C_I denoting the initial set of content objects.

$$A_5 : C = \{c \mid \forall\, d \in C_I : R(c,d)\} \vee C = \{c \mid \exists\, d \in C_I : R(c,d)\} \qquad (4.18)$$

(A_6) *Similarity of content*: Similarity rules need to be specified in advance in order to accomplish the *similarity search* use case as described in Section 4.2.1. These similarity rules $S(c,d)$ according to the specification in Section 4.4.1 determine for this use case which content objects are similar to an initial set of given ones C_I.

$$A_6 : C = \{c \mid \forall\, d \in C_I : S(c,d)\} \qquad (4.19)$$

An example for a similarity rule that can be applied in the *similarity search* use case is given in Section 4.4.1.

Table 4.2 summarizes the aforementioned information acquisition rules and classifies them according to use case dependency. The rules (A_1) and (A_2) need to be considered whenever an information acquisition use case is accomplished whereas rules (A_3) – (A_6) are use case-dependent each.

Rule	**Use Case Dependency**
(A_1) *Relevance for task context*	all
(A_2) *Trustworthiness of information source*	all
(A_3) *Content filtering*	*keyword, context, contact and full text search*
(A_4) *Content subscription*	*subscribe to new content subscribe to feedback*
(A_5) *Relation of content*	*relation search*
(A_6) *Similarity of content*	*similarity search*

TABLE 4.2. *Use case dependency of information acquisition rules*

4.4.3 Information Organization Rules

According to the remarks in Section 4.2.2, automatic accomplishment of information organization processes requires rule-based support by the user-centered social software. In the following subsections, reactive information organization rules denoted by O_i are discussed for the individual processes in compliance with Section 4.2.2 and based on considerations of the enhancement of personal information discussed in Section 2.2.4.

(O_1) *Automatic classification*: Content objects are automatically classified after information acquisition using the *classify by context* use case. Classification is based on contexts provided by the user-centered social software like categories or tags. On the one hand, a content object can be classified automatically by a context q that is contained in the original content as a keyword. On the other hand, a context q of similar content objects from $S(c)$ can be added to the content object as given in (4.20).

$$O_1 : \text{c contains q} \vee (\exists\, \text{d} \in S(\text{c}) \mid \text{q } \textit{classifies } \text{d}) \Rightarrow \text{q } \textit{classifies } \text{c} \tag{4.20}$$

(O_2) *Automatic linking of content objects*: Based on existing links between content objects and other content-related elements further relations applying to content objects can be inferred from rules according to (4.4). In this case, the use cases *add unidirectional link, add bidirectional link* or *add inverse link* are accomplished automatically based on a specified inference rule *N*(c,d) as stated in (4.21) with a particular link using predefined link types.

$$O_2 : N(\text{c,d}) \Rightarrow \textit{add unidirectional link } \text{c } \textit{relatedTo } \text{d} \vee \textit{add bidirectional link } \text{c } \textit{relatedTo } \text{d} \vee \textit{add inverse link } \text{d } \textit{relatedTo } \text{c} \tag{4.21}$$

Content objects are automatically related by equivalence (see Section 4.1.1) when semantically equivalent versions of a content object are managed at several content providers. According to Section 4.2, the application of certain use cases leads to this scenario which requires synchronization of content objects. In order to identify such objects, they are automatically related by the bidirectional link *equivalentTo*.

(O_3) *Automatic extension by properties*: Content objects can be automatically extended by properties based on similarity rules *S*(c,d) (see Section 4.4.1) which is analogous to automatic classification specified by rule (O_1). In this case, the use case *add property* is automatically performed. Further, the respective property pr must not be already defined by another value. Property values v are in this case added from similar content objects d based on (4.22).

$$O_3 : S(\text{c,d}) \wedge \text{d.pr} = \text{v} \wedge \text{c.pr} = \text{undefined} \Rightarrow \textit{add property}(\text{c.pr} = \text{v}) \tag{4.22}$$

(O_4) *Automatic marking*: Predefined marks can be used by the user-centered social software to automatically mark content objects according to particular characteristics. The marks can be set automatically for content elements that are selected according to a restriction rule *R*.

With regard to the accessibility mark, a set of content elements can be specified by the restriction rule to be marked as *public*, *private* or *role-based* respectively.

In case of the subscription mark, a set of content objects can be specified for subscription by a restriction rule *R* as referred to in the definition of rule (A_4) which need to be marked as subscribed by the user-centered social software.

When automatically applying a use case that adds a content-related element to a content object by rules (O_1) – (O_3), this element is marked by the value *automatic* of the *automatic organization* mark.

Therefore, the specification of the automatic marking rule is given in (4.23) with m referring to one of the values *public*, *private*, *role-based*, *subscribed* or *automatic* according to the aforementioned considerations.

$$O_4 : R(\text{c}) \Rightarrow \textit{add mark } (\text{c, m}) \tag{4.23}$$

(O_5) *Automatic removal*: Automatically added contexts, links and properties according to the information organization rules (O_1), (O_2) and (O_3) introduced above can be automatically removed applying this rule. Whenever the individual user removes an automatically added content-related element e, all other elements marked automatic and having the same value v are removed.

O_5 : e manually removed $\Rightarrow \forall$ c, c.e = v $\wedge$ e marked *automatic*:

remove context $\vee$ *remove link* $\vee$ *remove property* from c (4.24)

Table 4.3 gives an overview of the abovementioned information organization rules that support automatic application of the respective use cases.

Rule	Use Case Dependency
(O_1) *Automatic classification*	*classify by context*
(O_2) *Automatic linking*	*add unidirectional link, add bidirectional link, add inverse link*
(O_3) *Automatic extension by properties*	*add property*
(O_4) *Automatic marking*	*add mark*
(O_5) *Automatic removal*	*remove context, remove link, remove property*

TABLE 4.3. *Use case dependency of information organization rules*

The use cases of the annotate and rate process respectively have not been taken into consideration since they are only applied manually by the individual user and thus do not allow automatic application by means of rules.

4.4.4 Information Dissemination Rules

In order to determine whether information may be disseminated, inference rules are applied as shown on the right hand side of Figure 4.6. The deductive rules that apply for information dissemination are denoted by D_i in the following and determine the validity of the accomplishment of a use case.

The application of a use case *u* on local content object c to disseminate it to scope s in the individual user's area at a social software platform is denoted by *u*(c,s).

(D_1) *Authorization*: The validity of information dissemination use cases with regard to a content object depends on appropriate authorization of all contacts who may access that content object managed in the individual user's area at the respective social software platform.

Content objects marked as public can be disseminated to all scopes of the individual user's area at social software platforms. Therefore all information dissemination use cases are applicable. By contrast, content objects marked as private may only be posted to the private scope and merely the use cases of information storage may be accomplished. Intermediate cases are determined by the effective view on a content object within a particular role. The specification of the authorization rule is as follows in (4.25).

D_1 : c is visible in role corresponding to s $\Rightarrow$ *u*(c,s) is valid (4.25)

The rule of authorization in (4.23) implies as a special case the rule for dissemination of feedback. This means, the use cases of *give feedback*, *post note as feedback* and *post rating as feedback* may only be applied to content objects marked as public since the individual user has no influence on the scope in which the feedback will be managed in the contact's area at the respective social software platform.

(D_2) *Content representation*: Content elements created by the user-centered social software can only be posted to a social software platform if appropriate content representation is provided there. Original metadata from the social software platform referenced by the content object can in this case be updated by content attributes added in the user-centered social software (see Section 4.2.2). The appropriate representation refers to the type of the content elements on the one hand but also to its semantics on the other hand.

Appropriate content representation can be achieved in certain cases by transformation of content element representations used in the user-centered social software UCSS and specific social software platforms SSP, however. Therefore, a transformation function f_{SSP} per social software platform needs to be defined in advance of information dissemination of such a content element c. Transformation of the representation c_{UCSS} of content element c to the representation c_{SSP} at the respective content provider is given in (4.26).

$$c_{SSP} = f_{SSP}(c_{UCSS}) \tag{4.26}$$

The content representation rule of content element c is denoted in (4.27) with SSP representing a social software platform.

$$D_2 : c_{SSP} = c_{UCSS} \vee c_{SSP} = f_{SSP}(c_{UCSS}) \Rightarrow u(c,s) \text{ is valid} \tag{4.27}$$

Rating values exemplify the requirement for the transformation of content representation. If an appropriate content representation for ratings exists on the respective social software platform, it is required to map rating values r assigned by the individual user in the user-centered social software to adequate values r' on that platform by a transformation function based on (4.26) in order to disseminate them correctly. An example for the transformation of rating values is given in Section 5.4.3.

(D_3) *Single dissemination per platform*: The information dissemination use cases except *update metadata* to a social software platform SSP are only possible if that version of the local content object has not yet been posted there as formulated by (4.28).

$$D_3 : c \text{ is not yet disseminated to SSP} \Rightarrow u(c,s) \text{ is valid} \tag{4.28}$$

The abovementioned rules (D_1) – (D_3) hold for all use cases of information dissemination with minor exceptions in contrast to information acquisition rules summarized in Table 4.4.

Rule	Use Case Dependency
(D_1) *Authorization*	all
(D_2) *Content representation*	all
(D_3) *Single dissemination per platform*	all except *update metadata*

TABLE 4.4. *Use case dependency of information dissemination rules*

4.4.5 Rule-Based Definition of Views and Roles

It is a complex task for the individual user to explicitly specify views and roles due to the vast number of content-related elements even for a single content object. The definition of views and roles requires rule-based support by the user-centered social software instead, in order to provide an efficient means

of access control for information dissemination. The rules are applied on the level of views and roles respectively.

In order to explicitly determine a view for a single content object, access rights for all associated content-related elements need to be specified separately as explained in Section 4.3.1. Another possibility is an implicit definition of views that arises by using restriction rules to specify content elements whose accessibility marks should be set.

Application of Restriction Rules

A set of views for content objects can be defined by the application of rules compared to the explicit definition of the view per content object. A set of effective views represent a means to determine a role. In order to automatically determine a set of content objects, restriction rules are used that can be realized as queries as introduced in Section 4.4.1.

The following example shows the application of a set of restriction rules for supporting the definition of a role by the selection of accessible content objects.

Role r, labeled ‘co-worker’, is defined by the individual user by the specification of three rules where each determines a set of effective views on content objects. Only content-related elements of content objects that are not marked as *private* are accessible within the role. Parameter values within the textual expression of the rule instance are emphasized by single quotation marks. In the following example, the role r is specified as $r = (I_1 \cup I_2) \setminus E_1$ with

(I_1) Include content objects of content type ‘document’ associated to context ‘java technology’ where context is a ‘folder’ or ‘tag’

(I_2) Include content objects of type ‘email’ where ‘author’ is contact ‘John Miller’ or ‘Steve Garret’

(E_1) Exclude content objects with rating ‘< 3’.

The general specification of a role as a set of content elements is given in (4.29) where each restriction rule specifies a set of content objects to be included (denoted by I_i) or excluded (represented by E_j) respectively.

$$r = \bigcup_i I_i \setminus \bigcup_j E_j \tag{4.29}$$

Rule-based definition of a role may conflict in the case of particular content objects, however. Such a scenario occurs if according to the set of rules a content object is both included in and excluded from the role. In that case, exclusion rules obtain higher priority.

4.5 Summary

The family of user-centered social software is designated to support social information management from the individual user’s perspective. The model consists of the core concepts of content reference, contact and context that correspond to the abstractions of content, user and classification scheme respectively in the content provider model developed in the previous Chapter 3.

Social information management support facilitates information acquisition, organization and dissemination by specifying use cases that extend the scope of personal information management introduced in Chapter 2 to online social networks at social software platforms. The three primary processes of social information management are complemented by the synchronization of semantically equivalent content objects which is an auxiliary process to ensure consistency of the application of particular use cases. The role-based authorization model of user-centered social software further focuses on information dissemination with selected contacts on a fine-grained basis.

Efficiency of the individual user's social information management activities is achieved by customizable rule-based support. Several kinds of predefined and user-defined rules, i.e. constructive rules, deductive and reactive rules, appropriately specify the business logic of the respective use cases in the user-centered social software model.

For this purpose, rules are specified for each process of social information management. Information acquisition rules that specify task context-relevant and trustworthy information sources, content filtering for search, retrieval and subscription purposes and relation and similarity of content objects ensure adequate information supply when performing the respective use cases. Rules concerning information organization allow automatic classification and extension of content objects by additional metadata and linking together content objects.

The validity of the application of information dissemination use cases is primarily specified by the rule of authorization. Roles for supporting fine-grained access control to content objects in the case of information dissemination use cases are automatically established by means of rules.

Chapter 5 presents the design of the Social Organizer – an application that pertains to the family of user-centered social software and relies on the associated model presented in this chapter. The design of the Social Organizer is concerned with significant aspects of use case support from the individual user's perspective.

Chapter 5

Design of the Social Organizer

This chapter presents the design and considerations for the prototypical implementation of a user-centered social software, termed Social Organizer. The information model of the Social Organizer is designed to facilitate schema-free metadata management and cross-platform relation management. Social information management in the Social Organizer is enhanced by customization of predefined rule schemata and the application of user-defined rules for information acquisition, organization and dissemination. Use case support for social information management in the Social Organizer is illustrated based on a RIA user interface prototype.

5.1 Information Model

The Social Organizer is a personal application that relies on the user-centered social software model described in Chapter 4 in order to support social information management. The Social Organizer is therefore an instantiation of a user-centered social software.

The diagram in Figure 5.1 depicts the information model of the Social Organizer where the primary concepts of the user-centered social software model shown in Figure 4.1 are represented by corresponding entities.

Task contexts defining workspaces for the individual user (see Section 4.1.2) constitute the topmost abstraction within the model of the Social Organizer. Initially, one *default task context* exists that can be modified by the individual user (see attribute *isDefault*). Content providers are assigned to and enabled in task contexts. The abstract concept of the content provider establishes a super class for both personal applications and social software platforms with regard to the user-centered social software model (see Figure 5.1). Social software platforms encapsulate attributes for representing the URL and the individual user's authentication information in the form of login and password.

The attribute *enabled* of the entity representing associated platforms denotes whether a social software platform is enabled for information exchange in a specific task context. In each task context of the Social Organizer, all accessible personal applications are enabled as content providers for

information organization purposes, though. Task contexts aggregate multiple portfolios that contain collections of content reference objects for reuse whose original content is managed at content providers. Each content provider is able to manage content objects of one or more dedicated content types (see Section 3.2). A content type may represent a collection (e.g. feed) which is indicated by the attribute *isCollection*.

Content references encapsulate the URI of the content object they refer to in order to be uniquely identifiable as required by the user-centered social software model. There is a distinction between *external* and *internal URI*s in the Social Organizer: The URI that specifies a remote content object at a social software platform is a URL. An example of an external URI is www.20six.de/weblogEntry/12345. By contrast, the URI of a local content object is an artificial identifier assigned by the Social Organizer that is composed of a prefix denoting the respective personal application and the identifier of the content object used in that application in the form of <personal application>/<id>. The internal URI of a local content object managed by the personal information manager Microsoft Outlook introduced in Section 2.3.1 could thus be given as follows: outlook/12345.

Contact is a distinct entity defined as a specialization of content reference to ensure that user profile information on an individual at specific content providers – exemplified in the model by a *username* at the platform and *last name* and *first name* as the respective attributes – is equally treated compared with other content objects represented by content references. It is nevertheless required to make a distinction between content and contact within the information model since contacts are real-word subjects that need to obtain authorization on content objects (see Section 5.1.6).

The Social Organizer provides context information in the form of flat-structured tags that facilitate additional classification of content reference objects and can be considered as best practice on social software platforms as discussed in Section 3.2. This is achieved in the model by introducing the predefined content type *tag* with an associated String property type for textual representation of the keyword. A content reference object can be classified by multiple tags to which it is linked by the *classifies* relation (see Section 5.1.3). An alternative to model tag information is to use a separate entity representing a tag which does not scale for further kinds of context information, however.

Whether *published copies* of a local content object have been disseminated to (further) social software platforms besides the *master copy* can be deduced from the entity Equivalence (see Figure 5.1). Published copies can merely be located at social software platforms but not at personal applications according to the semantics of the information dissemination process (see Section 4.2.3). The master copy of a content object is located at a content provider or if none is specified, the repository of the Social Organizer is the content provider. Which content reference object is the master copy at the respective content provider is an immutable characteristic.

The entity Equivalence is responsible for synchronizing semantically equivalent versions of a content object – by the operation *sync()* – and thus determines whether the affected content objects are synchronized manually or automatically at preconfigured intervals. Moreover, the type of the synchronization determines which version of the content object is used to update other versions. Values of the corresponding attribute *syncType* can be *noSync*, *most recent* or *master*. In the latter case, all equivalent versions of the content object are then updated based on the master copy during the synchronization process.

In the following sections, further significant concepts comprised in the information model of the Social Organizer are discussed in detail.

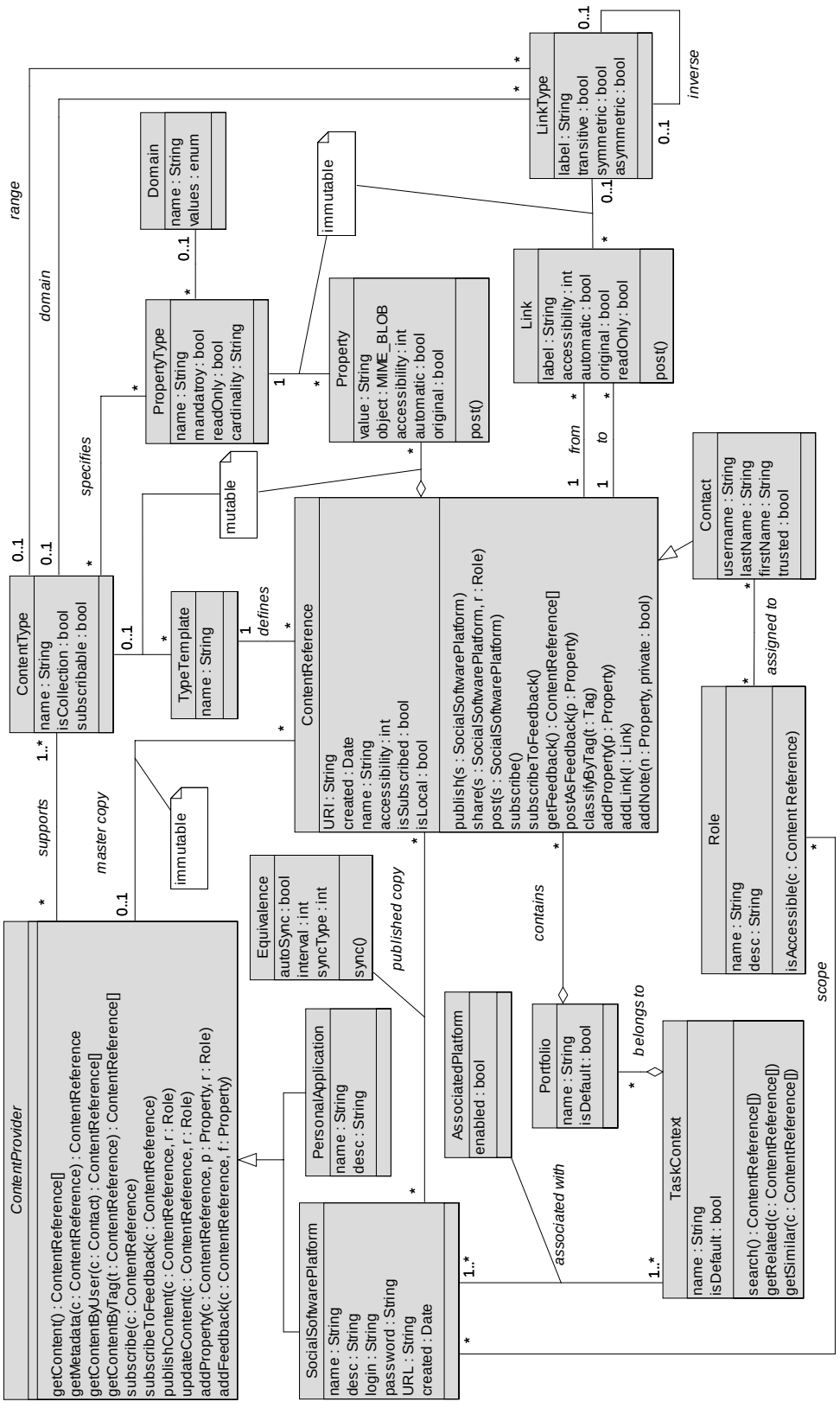


FIGURE 5.1. *Information model of the Social Organizer*

5.1.1 Content Attributes

Within the Social Organizer, all four kinds of content attributes exist that have been defined in the user-centered social software model in Section 4.1.

The model of links and properties is identical to the entities of the user-centered social software model presented in Figure 4.2. By means of this model, a flexible schema-free metadata and semantic relation management of content objects is facilitated similar to the workplace environment Haystack proposed by Karger et al. [Ka05].

The information model of the Social Organizer extends that model by explicitly defining domains in order to describe value ranges of property types. A domain is specified here by the enumeration of valid values. As shown in Figure 5.1 by corresponding attributes, it may further be specified whether the specification of a value is mandatory with regard to a property of a particular property type and if the value should be read-only. Further, the cardinality of the property type indicates the occurrence of properties of that property type associated to a content reference object.

The value of a property may either comprise a String representation or a BLOB representation for media-typed properties such as images, video or audio content. The association of a property to a content reference is mutable.

Concerning links, Section 5.1.3 summarizes predefined link types of the Social Organizer. The label of the link type is represented by the corresponding attribute *label* in the information model in Figure 5.1. Characteristics of link types comprised in the information model are transitivity, symmetry and asymmetry that depend on the represented relation and are indicated by appropriate attributes of the entity link type. Further, the specification of inverse link types is included in the information model of the Social Organizer. The characteristic of reflexivity is disregarded concerning link management in the user-centered social software model since links are merely established between two different content reference instances.

The entity link also comprises an attribute *label* in order to enable the representation of ad-hoc links in the Social Organizer that do not conform to a specific link type. A link can also be specified as read-only (e.g. if it should not be removed).

Predefined marks used in the Social Organizer in compliance with the user-centered social software model are represented by specific attributes of the respective entities in the information model and are described in Section 5.1.4. User-defined marks in the Social Organizer are represented by property types and domains. In this case, the property type defines the label of the respective mark while its values are defined by the associated domain.

Content type-independent metadata reflecting the individual user’s opinion are also contained in the information model exemplified by feedback, annotation and rating. In order to keep the model of content attributes simple, properties and links are used to represent these content attributes.

Feedbacks are represented implicitly in the information model by the predefined link type *feedbackOf* as explained in Section 5.1.3 since feedbacks themselves are content reference objects. Multiple content objects can be linked to a content reference object as feedback. By contrast, annotation and rating are represented as predefined property types as described in Section 5.1.2.

5.1.2 Predefined Property Types

Predefined property types are applied in the Social Organizer to represent the content attributes *annotation* and *rating*.

As shown in Table 5.1, annotations as textual notes can be represented by a String property type. Annotations attached to content objects are in general neither mandatory nor read-only. Multiple annotations can be attached to a content reference by the individual user which is denoted by the cardinality of n.

A rating can also be added to content reference objects in the form of a property of type integer according to Table 5.1. The individual user may configure the interval for the rating. Dissemination of the rating to social software platforms – if rating of content is enabled – requires that the individual user specify a mapping of his own rating values used in the Social Organizer to those at the respective social software platforms according to Section 4.4.4.

Further, the property type *ruleExpression* is specified in order to represent rule expressions within the Social Organizer (see section 5.2 for details).

Property Type	Domain	Mandatory	ReadOnly	Cardinality
annotation	String	false	false	n
rating	int	false	false	0-1
ruleExpression	String	false	false	n

TABLE 5.1. *Predefined property types in the Social Organizer*

5.1.3 Predefined Link Types

The Social Organizer provides predefined link types in order to fulfill the requirements of the user-centered social software model. These predefined link types are described in the following.

Equivalent objects are represented by the entity equivalence that incorporates semantically equivalent versions of a content object located at different content providers. For this purpose, a link of link type *equivalentTo* is automatically created between two content objects after applying certain information dissemination use cases described in Section 4.2.4. In fact, the transitive closure of the relation *equivalentTo* defines the set of content objects the entity equivalence refers to. Besides the characteristics of this link type resulting from the equivalence relation, that is symmetry and transitivity, it is required that the related objects be of the same content type as shown in Table 5.2. The inverse link type of *equivalentTo* is represented by itself.

Feedback objects are represented by the entity of content references with content type feedback. The feedback of a content object is thus indicated by an appropriate predefined link *feedbackOf* between two content objects. Therefore, the domain of this link type includes all content types while the range is restricted to feedback objects and represents an asymmetric relation. This is a rather implicit representation of feedbacks compared to the original user-centered social software model.

Further, the link type *classifies* is provided in order to associate content objects with additional context information in the form of tags. Thus, the domain of this link type is given by the predefined content type tag whereas the range encompasses all content types. Two tags can be linked together by this means to denote that the concepts they represent are related.

In order to indicate the contact – or the individual user himself respectively – who manages a certain content object at a social software platform the predefined link type *isOwner* is provided and automatically applied by the Social Organizer. The domain of this link type is thus contact and the range is determined by arbitrary content types. The corresponding inverse link type *ownedBy* is also given below in Table 5.2, both relations being asymmetric as a consequence.

A common case to be applied either manually by the individual user or automatically by the Social Organizer is to denote the author of a content object of any type. The predefined link type *isAuthor* is provided for this purpose. The corresponding inverse link type is termed *createdBy* as shown in Table 5.2. All characteristics correspond to those of the previously mentioned link types *isOwner* and *ownedBy* respectively.

Similar content objects are automatically linked by the link type *similarTo* (see Table 5.2) in order to make the similarity relation persistent for later reuse by the appropriate use cases. Otherwise, similar content objects are linked transiently by the link type *linkedTo* (see below) on demand, that is, whenever a use case is applied that requires the identification of similar content objects. The characteristics of the *similarTo* link type are analogous to those of *equivalentTo* described above – the only difference lies in the fact that similar content objects need not necessarily be of the same content type.

The link type *linkedTo* is applied for transient linking of content objects of arbitrary content types by implicit links. According to Section 4.4.1, automatic link creation is used in order to support certain social information management use cases.

Table 5.2 summarizes the predefined link types in the Social Organizer. In the table, the domain, range and characteristics of the link types including algebraic properties are given according to Section 4.1.1. Further, the inverse link type is given if existent.

Link Type	Domain	Range	Characteristics	Inverse Link Type
equivalentTo	all	all	symmetric, transitive domain.contentType = range.contentType	*equivalentTo*
feedbackOf	all	feedback	asymmetric	-
classifies	tag	all	asymmetric	-
isOwner	contact	all	asymmetric	*ownedBy*
ownedBy	all	contact	asymmetric	*isOwner*
isAuthor	contact	all	asymmetric	*createdBy*
createdBy	all	contact	asymmetric	*isAuthor*
similarTo	all	all	symmetric, transitive	*similarTo*
linkedTo	all	all	-	-

TABLE 5.2. *Predefined link types in the Social Organizer*

5.1.4 Predefined Marks

Predefined marks as required by the user-centered social software are the accessibility mark, the subscription mark and the automatic organization mark. The semantics of marks are additionally used to specify further characteristics of entities in the information model of the Social Organizer as described in the following.

The *accessibility mark* facilitates content objects and content attributes to be marked as *public*, *private* or *role-based*. This mark is represented by integer values of the attribute *accessibility* of content reference, property and link in the information model in Figure 5.1. The attribute set to *public* denotes the public mark and the attribute value *private* represents the private mark as given in Table 5.3. The value *role-based* indicates that the content object can be made accessible to contacts in roles.

Thus, the accessibility mark is part of the authorization model described in Section 5.1.6. The default value for newly added content objects is *role-based*.

The *subscription mark* denotes whether a (subscribable) content object – represented by its content reference – has been subscribed to by the individual user. This boolean mark is realized by the boolean attribute *isSubscribed* according to the information model of the Social Organizer in Figure 5.1.

In the user-centered social software model, the automatic organization mark is specified that indicates whether a link or property has been automatically added to a content object within the Social Organizer according to the rules of automatic information organization. This mark is represented by the boolean values of the attribute *automatic* of the respective entities property and link of the information model shown in Figure 5.1.

The information model of the Social Organizer comprises the additional *location mark* denoted by the attribute *isLocal*. This boolean-valued mark is a derived attribute of the content reference entity and describes whether a content reference refers to a local or remote content object, i.e. managed by a personal application or at a social software platform.

An additional mark in the Social Organizer is the *original content* mark also applied for properties, and links to indicate whether these content attributes have been added to a content object within the Social Organizer and are managed there. The value of the mark is set to *added* in this case. Otherwise, if original metadata of content providers is represented according to the information model, the mark is set to the value *original*. The original content mark is represented by the boolean-valued attribute *original* for each of the abovementioned entities according to Section 5.1.7.

Trusted contacts can be marked accordingly in order to specify the rule of trustworthiness of the information source (A_2) introduced in Section 4.4.2 by the *trust mark*. Thus, the boolean values *trusted* and *non_trusted* are defined for the *trusted* attribute of contacts.

Table 5.3 gives an overview of the predefined marks and their possible values in the Social Organizer. Additionally, the affected entities of the information model are shown along with the attribute representing the respective mark.

Mark	**Values**	**Entities**	**Attribute**
accessibility	*private*, *public*, *role-based*	Content reference, Property, Link	accessibility
subscription	*subscribed* (`true`), *not subscribed* (`false`)	Content reference	isSubscribed
automatic organization	*automatic* (`true`), *manual* (`false`)	Property, Link	automatic
location	*local* (`true`), *remote* (`false`)	Content reference	isLocal
original content	*original* (`true`) *added*	Property, Link	original
trust	*trusted* (`true`) *non_trusted* (`false`)	Contact	trusted

TABLE 5.3. *Predefined marks in the Social Organizer*

5.1.5 Schema-Free Metadata Management

The Social Organizer facilitates flexible schema-free metadata management by supporting the specification of untyped content objects in contrast to objects of rigidly defined content types managed at content providers.

For this purpose, type templates are introduced. Untyped content objects are associated to a predefined type template referred to as *DefaultType* and can be extended by arbitrary properties representing user-defined metadata. Each type template is associated to a content type except the DefaultType. From the associated content type, property types are inherited. The type template a content object is associated with may also be altered.

Figure 5.2 depicts a scenario of schema-free metadata management in the Social Organizer. An untyped content object is newly created within the Social Organizer and is extended by two properties. The property p1 is of the predefined type rating and assigned the value 4 which corresponds to one of the values in the specified domain. Further, a p2 of user-defined type is added to the content object that indicates a color value from the associated domain.

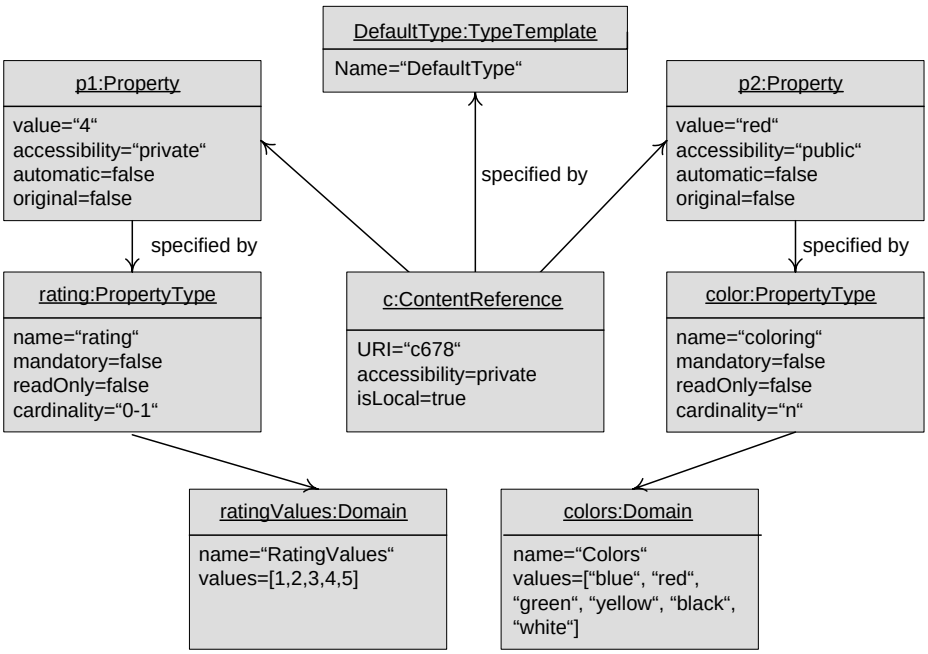


FIGURE 5.2. *Example of schema-free metadata management*

5.1.6 Role-Based Authorization

The Social Organizer incorporates a role-based authorization model to sketch the idea of the user-centered social software model. The accessibility mark plays a significant role in the design of the Social Organizer for this purpose. Further, contacts are assigned to roles as shown in the information model in Figure 5.1.

The views on content attributes represented by properties and links are determined by the respective value of the accessibility mark (see Table 5.3). The accessibility mark of a content reference object further denotes the default value for the view of associated content attributes and is thus inherited by newly added content attributes to that content object until explicitly specified otherwise.

Hence, a role in the Social Organizer encompasses read access to a set of content objects that are not marked as *private* along with associated properties and links which are marked either *public* or *role-based*. If the content reference is marked as *private*, it is not visible in any role.

A role is thus implicitly composed of the effective views on content objects according to the user-centered social software model. All content attributes of content objects are accessible within a role unless they are marked as *private*.

In order to specify which content objects are accessible within a role, restriction rules according to Section 5.2.2 are applied. By the operation *isAccessible()* (see Figure 5.1) it can be determined whether a particular content object is accessible within a role conforming to the specified rule.

Roles are associated to logical scopes at content providers represented by user groups in compliance with the user-centered social software model. A role can be associated to user groups at multiple content providers according to the information model of the Social Organizer in Figure 5.1.

Concerning role management, contacts that represent the same individual at different social software platforms need to be identified to ensure consistent application of information dissemination use cases in the Social Organizer. Since contact is a specialization of the content reference entity, user profiles of contacts can be considered as representing equivalent content objects on different social software platforms without synchronizing them, i.e. specifying the synchronization type *noSync* for the respective equivalence object.

Automatic identification of contacts is beyond the scope of the Social Organizer design though, which means that the individual user has to explicitly specify equivalent user profiles instead.

However, the user profiles belonging to the individual user himself can be identified as being equivalent since account information of integrated social software platforms is known in the Social Organizer. In this case, the individual's own user profile information can be updated and synchronized with social software platforms.

Predefined Roles

The Social Organizer specifies four general predefined roles that are required in the default specification of trusted contacts according to common predefined user-groups at social software platforms (see Section 3.2). These roles are labeled *family member*, *friend*, *acquaintance* and *co-worker* and are analogous to predefined user groups on social software platforms.

The initial set of roles in the Social Organizer hence has four elements. These predefined roles become applicable when contact objects are associated to them based on rules or they are associated with at least one contact.

5.1.7 Representation of Original Content Objects

In order to support use cases of information organization on content objects located at content providers, besides the content reference the original metadata of a content object also needs to be represented appropriately in the Social Organizer. For this purpose, the model of property types and properties is used.

The example in Figure 5.3 illustrates that the attributes title, text and creation date of a weblog entry are represented by property types 'title', 'text' and 'creationDate' which are already specified or added as property types to the content type weblog entry. The specific values for that weblog entry are then represented by property objects within the Social Organizer (see Figure 5.3).

Original relations between two content objects are represented by link types and links according to the defined associations of the domain-specific service model of a particular content provider (see Section 3.3.1). For example, if a specific weblog at a content provider has a weblog entry, an ad-hoc link can be established between the two objects. An alternative is to define a link type *entryOf* accordingly and relate the weblog entry and the weblog by a link based on that newly defined link type (see Figure 5.3).

The content itself (e.g. document text) is not represented within the Social Organizer but merely a preview on the content can be given for information organization purposes.

Context information based on classification mechanisms of content providers can be represented by content references of corresponding content types similar to the predefined content type tag. Content objects that are classified by a given context such as a category or folder are then related to that context information by links of an appropriate newly specified link type.

For example, an e-mail at the content provider could be classified by a folder which is established as a distinct content type. The e-mail can then be linked to the representation of the folder object by establishing a link of type *isInFolder*.

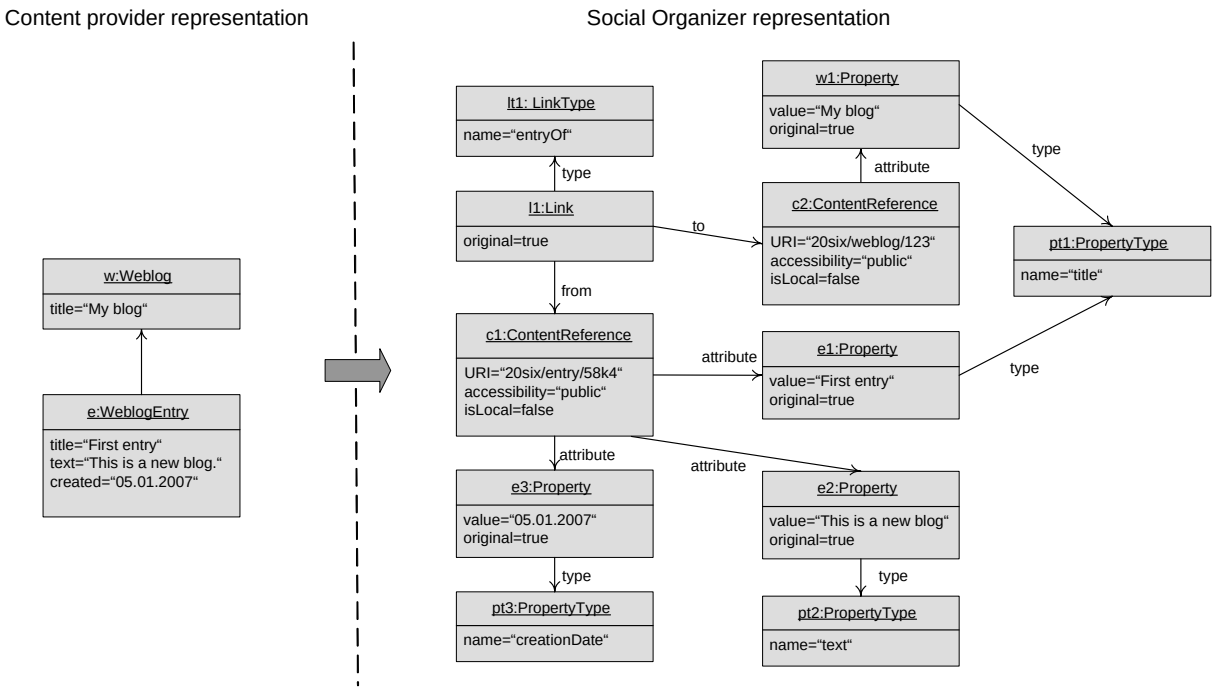


FIGURE 5.3. *Representation of original content objects*

Original metadata, associations and context represented by content providers are marked by the original content mark as explained in Section 5.1.4 and shown in the example of Figure 5.3.

Metadata that is updated in the Social Organizer by the corresponding use case has to be converted accordingly from the information model of the Social Organizer to the domain-specific service model of the content provider in order to be disseminated. Technical issues of transformation of content representations are beyond the scope of this thesis, however.

5.1.8 Integration of Content Provider Services

The operations of the entity content provider in the information model in Figure 5.1 denote the services according to Section 3.3.3 that need to be integrated from content providers in order to be able to support the use cases of social information management. Social software services – including the special case of personal application services – depicted in Figure 3.17 are associated to operations of the information model of the Social Organizer and indicated in brackets in the following.

Considering information acquisition and retrieval, selected use cases of information search and information subscription are supported within the Social Organizer. Therefore, retrieval services and subscription services of content providers have to be integrated that are indicated by the operations *getContent()* (‘Get content by criteria’), *getContentByUser()* (‘Get content by user’), *getContentByTag()* (‘Get content by tag/category’), *subscribe()* (‘Content feed’) and *subscribeToFeedback()* (‘Feedback feed’) respectively.

For obtaining a particular content object for information organization purposes the operation *getMetadata()* (‘Get metadata of content’) needs to be supported by each content provider that obtains the original metadata of the content object identified by its reference. If modifications to original metadata in the Social Organizer are propagated to a content provider, the operation *updateContent()* (‘Update content’) is applied.

With regard to information dissemination, the operation *publishContent()* (‘Publish content’) describes posting a content object to be accessible in a specific scope at the content provider which is determined by the given role. The role determines which content attributes of a content object are shared with contacts that are assigned to that role and have access to the corresponding scope. Concerning the *publish* use case, the role can be omitted.

Use cases for extending the content objects by content attributes within the Social Organizer do not require the integration of content provider services but are initiated by appropriate operations of the content reference entity represented by *classifyByTag()*, *addProperty()*, *addLink()* and *addNote()* respectively (see Figure 5.1).

Added properties of a content reference object representing annotations or a rating can be added as feedback at social software platforms by applying the operation *addFeedback()* (‘Add local feedback/rating’).

Further content attributes represented by properties and links that have been added to a content object by the Social Organizer during information organization may be posted to the social software platform if the operation *addProperty()* is supported and appropriate content representation is available there (see rule (D_2) in Section 4.4.4). This operation relies on the respective *post()* operations of the aforementioned entities (see Figure 5.1).

Based on the operations of content providers mentioned above, more specific use case support is facilitated within a task context. Information acquisition from content providers within the search outside scope is accomplished by *search()* that relies on the abovementioned retrieval services of content providers according to the specified search criteria (see use cases *keyword search*, *contact search*, *context search* in Section 4.2.1).

Particular retrieval-based use cases of information acquisition within the search inside scope are initiated by the corresponding operations of the task context, that is, *search()*, *getRelated()* and *getSimilar()* which operate on the repository of the Social Organizer.

The use cases mentioned above are initiated by the appropriate operation on a specific content reference object and are represented by the operations *post()*, *publish()*, *share()*, *subscribe()*,

subscribeToFeedback(), getFeedback() and *postAsFeedback()* shown in the information model of the Social Organizer in Figure 5.1.

5.2 Specification of Rules

Support of social information management use cases according to the user-centered social software model requires that the Social Organizer provide predefined rule schemata and default instantiations for them. Information acquisition, organization and dissemination rules on the one hand rely on the specification of predefined content-related rules. On the other hand, the specification of trusted contacts is essential in order to appropriately perform information acquisition while the definition of roles is required for information dissemination on a fine-grained basis.

In addition to predefined rule schemata the individual user may specify further rule schemata in the above-mentioned categories. Nevertheless, user-defined rules are optional in contrast to predefined ones. Both predefined and user-defined rule schemata are instantiated by particular parameter values that are either assigned immediately when specifying the rule like in case of content filtering or originate from configuration settings by the individual user. Default instantiation of predefined rules enables working with the Social Organizer without further user configuration.

In the model of the Social Organizer, valid rules rely on boolean and relational operators as indicated by the examples in the following subsections since the expressiveness of such rules is sufficient to illustrate enhanced social information management support.

The specified rules can usually be realized using queries (e.g. selection of content objects for subscription or the definition of roles) and program code-based algorithms (e.g. validity checking of information dissemination use cases). However, deductive and reactive rules (e.g. similarity of content objects, automatic creation of properties and links) can be optimized by rule engine support.

Rules are represented within the Social Organizer by a predefined content type *rule*. The property type *ruleExpression* shown in Table 5.1 incorporates the rule expression in textual form according to query syntax or the syntax of the respective rule engine (see Section 5.4.2 for an example). Rule expressions can only be added to content objects of content type rule.

In the following subsections, rules are expressed in compliance with the formal notation used in Section 4.4 for the user-centered social software model along with the conventions for variable names that have been introduced there. The expressions in the formulas refer to general content reference objects by the letters c, d and e respectively whereas the letter p usually indicates a contact. The letter t denotes a content object of content type tag within the Social Organizer.

5.2.1 Predefined Rule Schemata

The application of content-related rule schemata is required for several rules according to the user-centered social software model. The next subsections describe predefined rule schemata for content-related restriction, inference and similarity rules along with the relevant use cases.

Content Filtering

Content filtering rules are required in order to conform to rule (A_3) defined in Section 4.4.2. For this scenario a predefined restriction rule schema exists.

Content filtering rules are instantiated on demand by parameter values by the individual user in order to perform the *keyword search* use case and its specializations *context search* and *contact search* respectively according to Section 4.2.1. The three aforementioned use cases are associated within the two scopes search inside and search outside of the search process as explained in Section 5.3.1 below.

The restricted content-related elements are in this case the original metadata and properties of the content object and context information (e.g. associated tags in the Social Organizer). At least one of these elements is enforced to contain the specified keyword(s) in order that the content object appears in the search result. The rule schema further includes restriction on the content type and the interval of the creation date of content objects.

Trusted Contacts

The definition of trusted contacts required for the application of rule (A_2) is also based on the specification of a restriction rule. The rule schema defining trusted contacts p in the Social Organizer is given by (5.1) according to Section 4.4.2 and based on an inference rule. Roles defined in the Social Organizer are denoted by r_i establishing the set of roles R.

On the one hand, the rule schema considers contacts to whom specific roles are assigned. On the other hand, further restrictions on contacts are included in the rule schema (5.1) in order to determine that they pertain to the set of trusted contacts, that is, contacts who own content objects that were rated higher than a specified value by the individual user. Therefore, the rule schema matches trustworthiness of contacts based on the individual user's own opinion. This is represented by the appropriate threshold value n of the metrics in (5.1) indicating trustworthiness based on content objects owned by the respective contact.

$$T(p) := p \text{ assigned to } r_i \,,\, r_i \in R \vee (\exists\ c \mid p\ \textit{isOwner}\ c\ \wedge c.\text{rating} > n)$$
$$\Rightarrow \textit{add mark 'trusted'} \text{ to } p \qquad (5.1)$$

Instantiation of the abovementioned rule schema requires the specification of a subset of roles and a parameter value n. For this rule schema, merely one instantiation per task context is possible. The individual user can define different rule schemata in order to specify the set of trusted contacts though and instantiate them appropriately in existing task contexts.

Inference Rules

According to rule (O_2) content objects can automatically be linked within the Social Organizer either by explicit links of link types determined by the particular inference rule or implicit links of type *linkedTo* (see Section 5.1.3) respectively.

With regard to explicit links, several kinds of predefined inference rule schemata are specified in the Social Organizer for these purposes.

Inference rules exist that are based on algebraic properties of predefined link types comprised in the information model of the Social Organizer according to Section 5.1.3. These six general inference rules are represented by N_1 through N_6 and given in formulas (5.2) – (5.7) below.

Inference rules N_1 and N_2 refer to the characteristics symmetry and transitivity of the *equivalentTo* relation between content objects as described in Section 5.1.3.

$$N_1(c,d) := c\ \textit{equivalentTo}\ d \Leftrightarrow d\ \textit{equivalentTo}\ c \qquad (5.2)$$

$$N_2(c,d) := c\ \textit{equivalentTo}\ e\ \wedge e\ \textit{equivalentTo}\ d\ \Rightarrow\ c\ \textit{equivalentTo}\ d \tag{5.3}$$

Analogous to the previous case, inference rules N_3 and N_4 hold for the similarity relation represented by the *similarTo* link type.

$$N_3(c,d) := c\ \textit{similarTo}\ d\ \Leftrightarrow d\ \textit{similarTo}\ c \tag{5.4}$$

$$N_4(c,d) := c\ \textit{similarTo}\ e\ \wedge e\ \textit{similarTo}\ d\ \Rightarrow\ c\ \textit{similarTo}\ d \tag{5.5}$$

Inference rules N_5 and N_6 are applied in order to deduce inverse relations in both directions from appropriate predefined link types in the Social Organizer according to Table 5.2.

$$N_5(c,d) := c\ \textit{isAuthor}\ d\ \Leftrightarrow d\ \textit{createdBy}\ c \tag{5.6}$$

$$N_6(c,d) := c\ \textit{isOwner}\ d\ \Leftrightarrow d\ \textit{ownedBy}\ c \tag{5.7}$$

Automatically establishing implicit links is primarily applied in order to support the determination of similarity between content objects in the Social Organizer. For this purpose, further predefined rules are specified as shown in formulas (5.8) – (5.10) deduced from predefined link types and associations in the information model.

Rule N_7 specifies that two content objects of arbitrary content types are related if they have the same author. From N_5 this inference rule also holds true for the inverse link type *createdBy*.

$$N_7(c,d) := e\ \textit{isAuthor}\ c\ \wedge e\ \textit{isAuthor}\ d\ \Rightarrow\ c\ \textit{linkedTo}\ d \tag{5.8}$$

Similarly, from rule N_8 it can be deduced that two content objects c and d are related if they are classified by the same tag t given in formula (5.9).

$$N_8(c,d) := t\ \textit{classifies}\ c \wedge t\ \textit{classifies}\ d\ \Rightarrow\ c\ \textit{linkedTo}\ d \tag{5.9}$$

Inference rule N_9 indicates that an arbitrary content object c and a feedback d will be automatically related by implicit links if the feedback is already associated with an object e which is equivalent to c.

$$N_9(c,d) := c\ \textit{equivalentTo}\ e\ \wedge d\ \textit{feedBackOf}\ e\ \Rightarrow\ c\ \textit{linkedTo}\ d \tag{5.10}$$

Predefined inference rules N_7 – N_9 in the Social Organizer that are not based on the required characteristics of link types may be disabled by the individual user.

Similarity Rules

Rule schemata for similarity rules need to be specified in the Social Organizer in order to perform the *similarity search* use case on the one hand and automatic classification and automatic extension of content objects by properties on the other hand. It is possible to define several similarity rules in compliance with the user-centered social software model.

Similarity rules are applied within the user-centered social software model in support of three use case-related rules of information acquisition and organization.

The rule established for the *similarity search* use case in (A_6) requires a schema for a similarity rule and an adequate instantiation. Further, automatic classification of content objects as described by rule (O_1) and automatic extension by properties referring to rule (O_3) that can be applied optionally also involve similarity rules.

For the aforementioned purposes, two similarity rule schemata are specified given in (5.11) and (5.12).

Rule S_1 defines similarity of two content objects in terms of similarity in content, context and contact relations in the scope of user-centered social software. It states that two content objects are similar if both have the same content type and either they have the same contact p in common as author or they share more than n tags t according to formula (5.11).

$$S_1(c,d) := (c.contentType = d.contentType) \wedge ((p\ \textit{isAuthor}\ c \wedge p\ \textit{isAuthor}\ d) \vee |\{t \mid t\ \textit{classifies}\ c \wedge t\ \textit{classifies}\ d\}| > n) \Rightarrow c\ \textit{similarTo}\ d \quad (5.11)$$

For default instantiation of S_1 the parameter value of n needs to be configured by an integer value greater than 0.

Similarity rule S_1 is extended by S_2 given in (5.12) in order that objects which are merely related by direct links of any link type or ad-hoc links respectively are explicitly declared as being similar. S_2 is significant in order to combine the use case *similarity search* with the *search related to any* use case (see Section 5.3.2).

$$S_2(c,d) := c\ \textit{relatedTo}\ d \Rightarrow c\ \textit{similarTo}\ d \quad (5.12)$$

Predefined similarity rules can be disabled if the individual user merely intends to apply user-defined similarity rules to content objects.

5.2.2 User-Defined Rule Schemata and Rule Instantiation

Two scenarios exist in the Social Organizer for the application of user-defined rule schemata. Restriction rules and inference rules are applied in order to support particular use cases as explained in detail below. Further, user-defined restriction rule schemata are required for the specification of roles. Such user-defined rules are optionally specified by the individual user.

In the following, examples for user-defined rule schemata and instantiation are explained. Parameter values in rule schemata are emphasized by single quotation marks.

Restriction Rules for Use Case Support

Concerning content subscription (A_4), only user-defined rule schemata and instantiations exist in order to subscribe to a set of content objects.

Considering the use case *subscribe to new content*, only content objects located at social software platforms and not yet referenced by the Social Organizer are determined by the instantiated rule and subscribed to continuously.

An example for a restriction rule applied for content subscription in the Social Organizer is as follows in (5.13) subscribing to content objects c created by a favored contact.

$$c\ \textit{isAuthor}\ \text{‘Miller’} \Rightarrow \textit{subscribe to new content}(c) \quad (5.13)$$

Another example for a rule considering the use case *subscribe to feedback* can be established as given by (5.14) with u in turn denoting the individual user.

$$u\ \textit{isOwner}\ c \wedge \text{’poems’}\ \textit{classifies}\ c \Rightarrow \textit{subscribe to feedback}(c) \quad (5.14)$$

Whenever a set of content objects is subscribed to, all content objects are automatically marked *subscribed* applying the *add mark* use case according to (O_5).

By contrast, the application of (O_5) is not required in order to subscribe to a single content object that is already managed within the Social Organizer. Such content objects can be manually marked *subscribed* by the individual user.

User-defined restriction rules for automatic marking (O_5) are merely established and instantiated in an ad-hoc manner in two cases. The first case is to mark a set of content objects *subscribed* in compliance with a restriction rule for subscription as mentioned above. The second case is that the individual user requests to add a certain value of the accessibility mark to a set of content objects that meet the rule's requirements.

An example is to mark continuously all content objects c as *private* that are classified by the tag 'finance':

$$\text{'finance'}\ \textit{classifies}\ c \Rightarrow \textit{add mark}\ (c, \textit{'private'}) \tag{5.15}$$

Other marks can only be applied automatically according to predefined rules (see Section 4.2.1). The application of the automatic organization mark in order to automatically add content-related elements is only performed in combination with the rules (O_1) – (O_3) as explained in Section 4.4.3 and the location mark is implicitly deduced from the content provider of content objects.

Inference Rules for Use Case Support

In addition to rules N_7 through N_9 introduced in Section 5.2.1, the individual user can define further inference rules for automatic linking of content objects. The two scenarios below illustrate content type-specific application of user-defined inference rules.

The first scenario exemplifies user-defined inference rules merely based on predefined link types of the Social Organizer. Content objects to be linked to a weblog entry c according to user-defined inference rules are determined in this scenario. Thus, the restriction of the content type of c to weblog entry is further omitted from the specification of inference rules for this scenario depicted in Table 5.4.

$N_{U1}(c,d) := t_1\ \textit{classifies}\ c \wedge \ldots \wedge t_k\ \textit{classifies}\ c \wedge t_i\ \textit{classifies}\ d,\ 1 \leq i \leq k \wedge c \neq d \wedge$
$(d.contentType = weblog \vee d.contentType = weblogEntry \vee d.contentType = email)$
$\Rightarrow c\ \textit{linkedTo}\ d$

$N_{U2}(c,d) := p\ \textit{isAuthor}\ c \wedge p\ \textit{isAuthor}\ d \wedge c \neq d \wedge$
$(d.contentType = weblogEntry \vee d.contentType = feedback) \Rightarrow c\ \textit{linkedTo}\ d$

$N_{U3}(c,d) := f_1\ \textit{feedbackOf}\ c \wedge f_k\ \textit{feedbackOf}\ c \wedge d\ \textit{isAuthor}\ f_i,\ 1 \leq i \leq k$
$\Rightarrow c\ \textit{linkedTo}\ d$

TABLE 5.4. *Scenario of user-defined inference rules using predefined link types*

The first rule N_{U1} in Table 5.4 determines that an object of type weblog, weblog entry or e-mail is related to the weblog entry by an implicit link if it shares at least one of the tags t_i the weblog entry is classified by. Similarly, N_{U2} implies that content objects of type weblog entry or feedback are related to weblog entry c if they have the same author p. Rule N_{U3} as the third example in the set of rules for

the abovementioned scenario states that the author d of at least one of the feedback objects associated with that weblog entry should be related to that weblog entry itself.

The second scenario demonstrates the application of user-defined inference rules involving user-defined link types that are not included in Table 5.2. This scenario focuses on which content objects can be linked explicitly or implicitly to an object c of content type e-mail according to inference rules specified by the individual user.

With regard to content objects of type e-mail link types *isSender* and *isRecipient* in this example indicate relations to contact objects that represent the sender and recipient of the e-mail respectively. Further, the link type *isAttachedTo* is applied in order to establish links between an e-mail and a document object that is attached to that e-mail. The set of rules for the second scenario is given in Table 5.5.

Inference rule N_{U4} exemplifies a user-defined rule establishing an explicit link between a contact and an e-mail object. The sender p of the e-mail implies in this case that it represents the author of the e-mail object.

$N_{U4}(c,p) := p\ \textit{isSender}\ c \Rightarrow p\ \textit{isAuthor}\ c$
$N_{U5}(c,d) := p\ \textit{isSender}\ c \wedge (p\ \textit{isSender}\ d \vee p\ \textit{isRecipient}\ d) \Rightarrow c\ \textit{linkedTo}\ d$
$N_{U6}(c,p) := d_1\ \textit{isAttachedTo}\ c \wedge \ldots \wedge d_k\ \textit{isAttachedTo}\ c \wedge$ $p\ \textit{isAuthor}\ d_i \wedge d_i.\text{createdDate} > D,\ 1 \leq i \leq k \Rightarrow p\ \textit{linkedTo}\ c$ (D parameter value representing a date literal)

TABLE 5.5. *Scenario of user-defined inference rules involving user-defined link types*

By contrast, N_{U5} and N_{U6} demonstrate reasonable user-defined inference rules in order to enhance automatic linking of content objects by implicit links in addition to predefined inference rules. Rule N_{U5} implies that e-mails composed by the same contact as sender and having the same recipient are also related to the original e-mail c. The user-defined link types *isSender* and *isRecipient* implicitly restrict the domain and range to contact and e-mail objects respectively so that these restrictions need not be specified within the rule expression.

Rule N_{U6} determines that a contact p is related to the e-mail c if he is the author of one of the documents d_i attached to the e-mail and the document has been created later than the specified date literal D – see attribute createdDate in Table 5.5. This example therefore demonstrates an inference rule that also refers to the metadata of the involved content objects in addition to relations between content objects – the latter being the most significant kind of parameter in a rule schema for the application of inference rules.

Inference rule N_{U6} is additionally depicted in the diagram of Figure 5.4 in order to clarify the establishing of a new link by a complex user-defined inference rule. The initial situation of existing unidirectional links is depicted by solid arrows while the deduced implicit link is illustrated by a dashed arrow in the diagram. Content objects found in rule N_{U6} are distinguished by their content types which are shown by appropriate icons in Figure 5.4. In addition, the required attribute createdDate of documents is illustrated next to the respective icons.

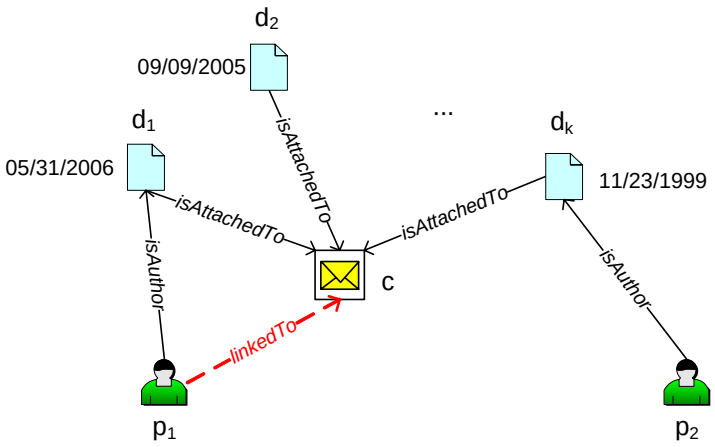


FIGURE 5.4. *Application of the inference rule* N_{U6}

Besides inference rules illustrated by the aforementioned examples, the individual user can further specify similarity rules that also rely on user-defined inference rules.

Rule Schemata for Roles

Roles are in general defined by appropriate restriction rules as described in Section 4.4.5. For this purpose, rule schemata are instantiated and composed in order to result in a set of content elements that are accessible within that role.

The Social Organizer supports inclusion and exclusion of content objects based on restriction rules in compliance with the user-centered social software model. These restriction rules need to be defined in advance in order that the respective content objects can be selected for inclusion or exclusion within the role.

Table 5.6 exemplifies the rule-based composition of accessible content objects for the predefined role *Co-Worker* as described in Section 5.1.6. Sets of included content objects are denoted by I_i whereas sets of content objects to be excluded from accessibility are indicated by E_j. The three example rules presented in Table 5.6 hence specify that on the one hand content objects of type document or e-mail are accessible within the role *Co-Worker* if they are classified by the tag ‘project’ but not by the tag ‘draft’. On the other hand, content objects associated with an author who is assigned to the role *Co-Worker* are also visible in the role unless they are classified by the tag ‘draft’.

Co-Worker = $(I_1 \cup I_2) \setminus E_1$
I_1 := {c | (c.contentType = document $\vee$ c.contentType = email) $\wedge$ ‘project’ *classifies* c}
I_2 := {c | p *isAuthor* c $\wedge$ p assigned to ‘Co-Worker’}
E_1:= {c | ‘draft’ *classifies* c}

TABLE 5.6. *Rule-based definition of the predefined role ‘Co-Worker’*

The definition of a role in this way could have been expressed by a single rule expression using boolean operators. However, this composition of rules to specify a role facilitates the reuse of restriction rule schemata and also their instantiations.

5.3 Use Case Support

The design of social information management support in the Social Organizer is presented in this section. The information model in Figure 5.1 depicts from a static view which use cases of the user-centered social software model are facilitated by the Social Organizer. By contrast, this section is devoted to the explanation of support for social information management processes in the Social Organizer.

The user interface of the Social Organizer is the most significant part from the individual user's point of view in order to perform the use cases of information acquisition, organization and dissemination. Therefore, use case support is illustrated based on the user interface of the Social Organizer in the following subsections. Essential concepts of the user-centered social software model like different content types, ratings or the accessibility mark are illustrated by means of appropriate user interface metaphors.

5.3.1 Interaction Design

The interaction design of the Social Organizer reflects support for the processes of social information management. It is related to selected social information management use cases in compliance with the information flow in the user-centered social software model illustrated in Figure 4.2.

Figure 5.5 depicts the main view of the Social Organizer user interface. Two menu items exist that are located in the menu bar and refer to entities which need to be configured in advance of working with the Social Organizer.

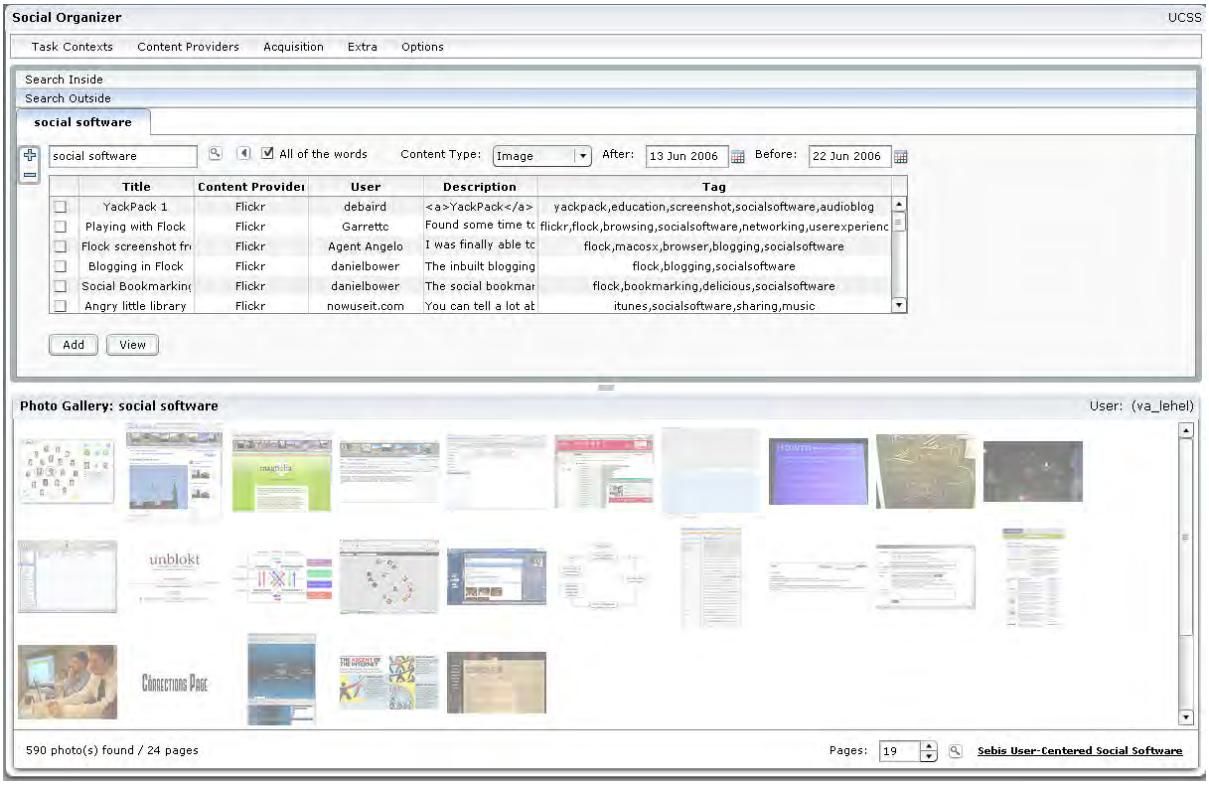


FIGURE 5.5. *The collection-based main view of the Social Organizer user interface*

The menu item *Content Providers* encapsulates the management of social software platforms and personal applications as content providers that are added to the Social Organizer for information exchange purposes. Since the prototypical realization of the Social Organizer focuses on the integration of several social contexts the individual user acts in, social software platforms are more important for interaction design considerations of the prototype.

The specification of task contexts – enabled by a menu item of the same identifier – is optional due to the availability of a default task context in the Social Organizer (see Section 5.1). A further characteristic lies in the fact that all integrated content providers are automatically associated with the default task context.

In compliance with the workspace concept of the user-centered social software model, collections of content reference objects can be managed in portfolios of the current task context. For each task context a *default portfolio* exists. For portfolios, the folder metaphor is adopted in compliance with equivalent application in information portals such as the infoAsset Broker described in [We02].

Collection-Based and Single-Content Views

The main view of the Social Organizer user interface in Figure 5.5 illustrates the emphasis on collection-based social information management support. The top panel of the main view consists of elements supporting information search in the two different scopes and constitutes the *collection-based view* on the information available to the individual user. The bottom panel contains further details on the respective search result by providing previews of items as exemplified in Figure 5.5. This main view implicitly refers to the current task context the individual user is working in. Figure 5.5 presents the search outside scope visualized as the open view on the top panel exemplifying content filtering based on the rule schema introduced in Section 5.2.1.

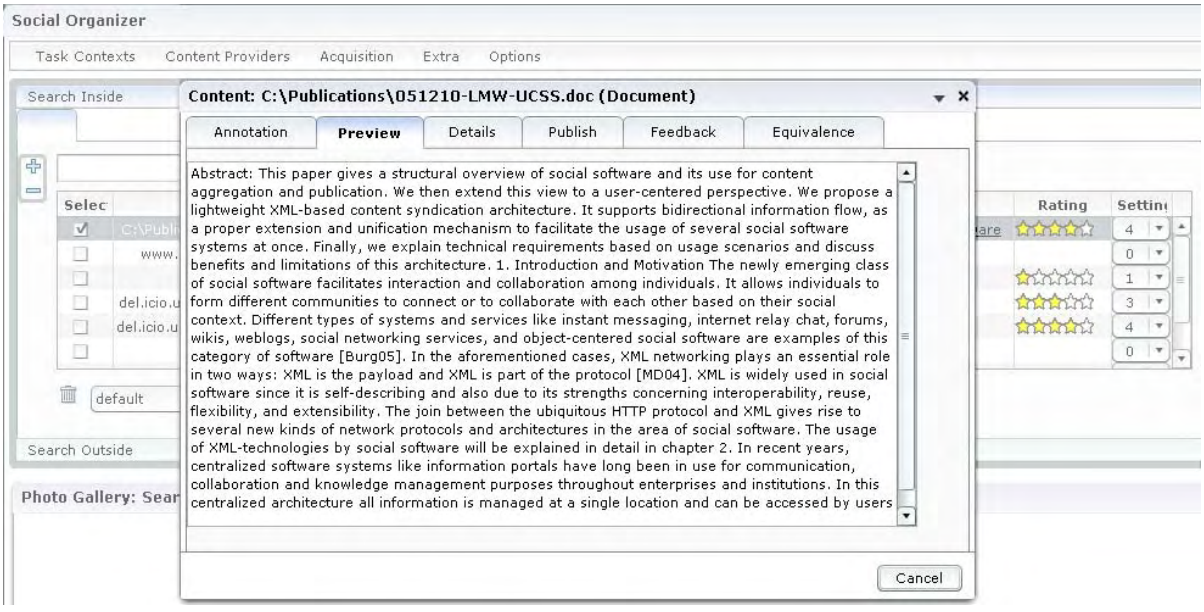


FIGURE 5.6. *The single-content view of the Social Organizer user interface*

Further, the *single-content view* is an essential part of the Social Organizer user interface in order to visualize content-related elements that pertain to a specific content object. The single-content view is placed as a popup window over the collection-based view according to Figure 5.6 and is composed of

a set of tabs that realize specific social information management use cases and are discussed below accordingly.

Figure 5.6 shows the preview of the selected content object of type document in the corresponding tab *Preview* which is a content type-dependent user interface element.

Information Flow

The first step in the information flow supported by the Social Organizer is information acquisition from content providers that are assigned to the current task context of the individual user involving both local and remote content objects. Information is either acquired manually by the *get by URI* or *browse* use cases respectively that are located in the *Acquisition* menu since these are less significant for exemplifying social information management support.

With regard to information search, content objects are searched for in the search outside scope of the collection-based view. Content objects that have been acquired from content providers performing the use cases of the search process can then be selected to be managed in the Social Organizer. Information search is discussed in detail in Section 5.3.2.

Subscription to updates of content objects at social software platforms is an additional way to acquire content objects. Its use cases are thus applied to distinct content objects that are of content types being subscribable (see Figure 5.1). Section 5.3.3 is devoted to the illustration of information subscription support in the Social Organizer.

Information organization as the second step of the information flow is supported collection-based by obtaining content objects in the search inside scope in order to perform information interpretation and evaluation use cases on them. Accordingly, information organization use cases can only be accomplished on content objects already referenced or managed by the Social Organizer.

Succeeding information acquisition, the Social Organizer provides support for manual application of the use cases of the classify, link and extend processes during information interpretation. The individual user applies these use cases by selecting particular content objects from the search result and displaying them in the single-content view. Information organization use cases referring to information interpretation are also automatically applicable to content objects; however, these aspects are beyond the scope of interaction design that describes the individual user interacting with the Social Organizer in order to perform social information management use cases. Information interpretation support in the Social Organizer is presented in Section 5.3.4.

Use cases concerning information evaluation, that is the use cases of the annotate, mark and rate process are also supported in the single-content view. Support of significant use cases of information evaluation in the Social Organizer is explained in Section 5.3.5.

The third step in the information flow relies on the use cases of information dissemination that can be performed in the single-content view.

The Social Organizer supports role-based authorization for the use cases of the post and share processes by facilitating to post content to adequate logical scopes in the individual user's areas on social software platforms according to Section 4.3.2. The content objects can merely be privately accessible, shared with selected contacts or publicly visible by any registered or anonymous platform user in compliance with the respective use cases *post as private*, *share with contacts* and *publish*. Specific use cases of information storage concerning moving content objects to social software platforms and updating metadata are omitted from the interaction design of the Social Organizer since they do not involve social aspects of information dissemination.

Further, feedback to content objects of contacts is disseminated and thus posted to social software platforms by the use cases of the provide feedback process. Section 5.3.6 is concerned with the support of selected use cases of information publication.

The following subsections discuss in detail the way significant social information management use cases are supported by the Social Organizer.

5.3.2 Information Search

In this section, the realization of selected use cases of information search within the Social Organizer is explained.

The collection-based view of the Social Organizer is responsible for the support of information search. The search outside scope depicted in Figure 5.5 is applied for information acquisition while the search inside scope shown in Figure 5.7 is used for information retrieval as a prerequisite for information organization.

Both information search scopes are organized in search tabs in order to retain search results in the current session of the task context.

The several use cases of information search – *keyword search*, *contact search* and *context search* respectively – are integrated in both scopes in order to obtain content objects that meet specific requirements according to the specified content filtering rule schema discussed in Section 5.2.1. The displayed information on content objects in the search result depends on the content type if the latter is filtered, that is, not set to *All* (see Figure 5.7.).

In Figure 5.5, the *context search* use case is illustrated in the search outside scope. The search result is established by a list of content objects located at content providers and not yet referenced by the Social Organizer. In the example, objects of content type image are searched for at the associated content providers of the task context – in this case the photo sharing platform Flickr [Flic07] – that match the tag ‘social software’ and have been posted to the platform in the specified time interval.

Information provided on the content objects in the search result includes the content provider, the user who has published the image at the platform and the associated tags. Further, content type-dependent attributes such as title and description of the respective image are shown.

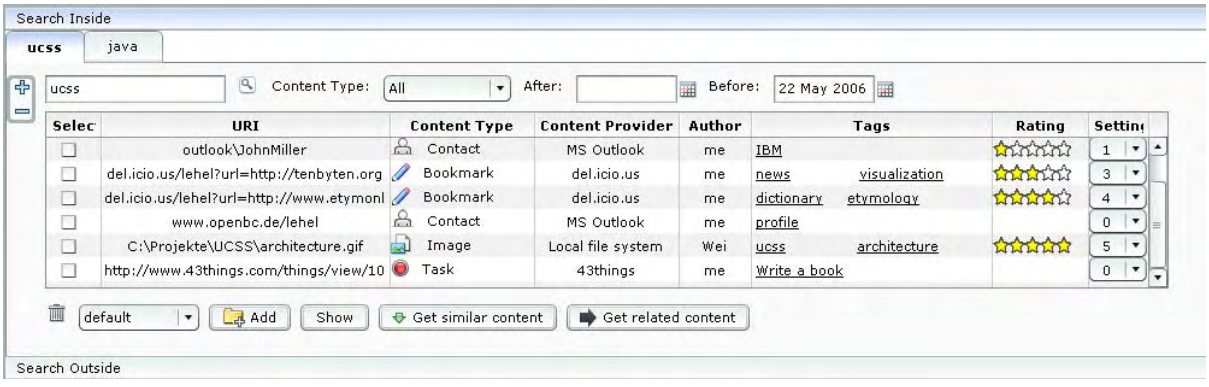


FIGURE 5.7. *Information search in the search inside scope*

Figure 5.7 illustrates the application of *keyword search* which is exemplified by the keyword ‘ucss’ in the search inside scope, that is, over content objects already added as references to or managed by the Social Organizer. Further content filtering is not used in the above example, so that objects of any

content type may appear in the search result which is visualized as a list of content objects in the user interface.

Information on a content object in the list of the collection-based view encompasses its URI, its content type and the content provider the referenced object is located at. Further, information on the author is provided if available – specified by the *isAuthor* link – and the tags the content object is associated with. The rating of the content object is also shown in the collection-based view and can be modified there directly by the individual user without applying the single-content view.

Context search is additionally enhanced by the suggestion of tags on user input similar to Google Suggest [Goog07b] or the service provided by the social bookmaking service del.icio.us [Deli07].

Content objects can be selected from the list in order to show details on them in the single-content view. Two significant use cases of information search exist that can only be applied in the search inside scope, i.e. the use cases *search related to all* and *similarity search*, discussed in the following subsections.

Use Case *Search Related to All*

The use case *search related to all* is applied within the search inside scope in order to obtain content objects that are related to all selected ones on the respective search tab by arbitrary links. The application of this use case is indicated by the button labeled ‘Get related content’ in the view in Figure 5.7. The result of this use case is displayed in a further search tab of the search inside scope.

Whether two content objects are related is specified by the relation of content rule (A_5) of the user-centered social software model. Configuration of the parameter values – the path length and link type of the relations – of the according rule schema is discussed in Section 5.4.3.

When merely one content object is selected in the list, this use case facilitates displaying all related content objects to that object either by direct links, or links up to the specified path length which is determined in the configuration settings of the Social Organizer (see Section 5.4.3).

Use Case *Similarity Search*

Application of the use case *similarity search* results in a set of content objects that are similar to the set of selected ones in the previous search result initially based on similarity rule S_1 according to (5.11).

The s*imilarity search* use case is performed in the search inside scope and thus performed on content objects already referenced by the Social Organizer due to application of the results of the corresponding similarity rule(s). The search result therefore appears in a new tab in the search inside scope.

By applying this use case, all content objects are obtained that are linked to at least one of the content objects in the initial set by the relation *similarTo*. In Figure 5.7, the application of this use case is illustrated by the button labeled ‘Get similar content’ in the search inside scope.

5.3.3 Information Subscription

Two significant use cases of the subscribe process exist whose support in the Social Organizer is discussed in the following subsections.

Use Case *Subscribe to New Content*

Subscription to subscribable content objects – determined in the first place by the content type – takes place in the single-content view by means of the use case *subscribe to new content*. Typically, the individual user subscribes to a content object of a type feed whose items are composed of content objects of content types such as weblog entry, feedback or image with further objects enclosed optionally per item (see Section 3.2.2). Therefore, subscribable feeds are selected for subscription in the menu *Acquisition* or marked as subscribed in the single-content view (see also Section 5.3.5).

Newly acquired content objects associated with a feed in a specific task context can be shown initially in the search inside scope of the collection-based view in order to notify the individual user of recent content when he starts working with the Social Organizer.

Use Case *Subscribe to Feedback*

A further significant use case of information subscription is *subscribe to feedback*. In order to apply this use case, a content object needs to be located at least at one social software platform. Otherwise, the application of the *publish* or *share with contact*s use case is required in advance in order to enable obtaining feedback from contacts for a content objects that has been posted to a social software platform. Afterwards, the individual user subscribes to feedback for that content object at the respective social software platform.

FIGURE 5.8. *The 'subscribe to feedback' use case*

Figure 5.8 depicts the result of the application of the *subscribe to feedback* use case on the *Feedback* tab of the single-content view that contains a list of the associated feedback objects of a content object. Figure 5.8 exemplifies a content object of the individual user which is of content type weblog entry and is located at a specific social software platform – the weblog community 20six [Twen07] in this example. Two feedback objects are presented in the list in Figure 5.8 indicating a preview of the comment text of the feedback and the associated contacts who are users at the social software platform.

5.3.4 Information Interpretation

Support of information organization use cases related to information interpretation within the Social Organizer is presented in this section. Figure 5.9 depicts information organization in the *Details* tab

which is a content type-independent user interface element. The details consist of the original metadata from the content provider on the one hand exemplified above the separator line. On the other hand, the content object can be rated by the individual user and enriched by tags within the Social Organizer. In this part of the single-content view, new content attributes in the form of properties and links can further be added to the respective content object – in this case the user profile of a contact.

Selected use cases of the related processes classify, extend and link are discussed in the following subsections.

FIGURE 5.9. *Information organization in the single-content view*

Use Case *Define Link*

The individual user is supported in the definition of further link types within the Social Organizer in addition to predefined link types presented in Table 5.2 by the use case *define link*. As shown in Figure 5.10, the label and an optional description of the semantics of the link type are specified. Further, the domain and range are determined and either set to a specific content type or *any* that indicates an arbitrary content type. The inverse link type and further characteristics according to the information model in Figure 5.1 are optionally specified for the link type.

In the example of Figure 5.10, the link type *isAuthor* is established, the domain of content types being contact and the range set to any content type. The inverse link type is specified as being *createdBy* which has been defined in advance. Additionally, the algebraic property of asymmetry has been specified as a characteristic for this link type since domain and range of the link are different.

Use Case *Add Link*

The *add link* use case is significant in the Social Organizer since it realizes cross-platform relation management of content objects according to the user-centered social software model.

This use case is manually applied in the *Details* tab of the single-content-view as shown in Figure 5.9. The *add link* use case is indicated by the button labeled ‘New Link’ which enables selecting a link type and a particular content object in order that the content object displayed in the single-content view is manually linked to the chosen one.

In the Social Organizer, automatic linking of content objects is additionally provided depending on the inference rules described in Section 5.2, that is, the use case *add link* can be automatically applied.

Automatic accomplishment of this use case does not involve user interaction and is therefore not further considered on the user interface.

FIGURE 5.10. *The 'define link' use case*

Use Case *Add Property*

The use case *add property* is manually applied in the *Details* tab in the single-content view as presented in Figure 5.9. In compliance with the information model of the Social Organizer presented in Figure 5.1, property types are specified by the individual user in advance for specific content types. In succeeding steps of information organization, properties according to these property types can be added to a particular content object.

Automatic extension of content objects by user-defined properties is also supported by the Social Organizer based on similarity rule S_1 in an analogous manner to the *classify by tag* use case (see below).

Use Case *Classify by Tag*

Manual classification of content objects according to tags is supported by the Social Organizer as pointed out in the information model in Figure 5.1. The individual user can classify content objects referenced or managed by the Social Organizer by tags in the *Details* tab of the single-content view (see Figure 5.9). Analogous to *context search* (see Section 5.3.2), suggestion of tags is provided for the *classify by tag* use case.

According to the information model of the Social Organizer, classification of a content object by tags is performed by application of the *add link* use case using the link type *classifies*.

Automatic classification of content object is also supported by the Social Organizer – initially based on similarity rule S_1 – in compliance with the user-centered social software model.

5.3.5 Information Evaluation

Information evaluation encompasses the use cases of the processes annotate, mark and rate. In the following subsections support of these use cases within the Social Organizer is described.

Use Cases of the Annotate Process

Support of the annotate process in the Social Organizer is illustrated in Figure 5.15 that depicts the *Annotation* tab of the single-content view. The two use cases *add private note* and *add note accessible to contacts* are controlled by setting the accessibility mark for an annotation property as *private* or specifying the default value *role-based* respectively. Hence, the individual user composes an annotation to be attached to a content object and marks it as *private* – indicated by the checkbox labeled 'private' in Figure 5.15 in order to perform the first use case. The *add note accessible to contacts* use case of the annotate process is applied by default.

Use Cases of the Mark Process

Manual application of the *add mark* use case with predefined marks is supported in the Social Organizer. According to Section 5.1.4, the accessibility mark and the subscription mark can be set manually for a particular content object. An example is presented by the abovementioned use cases of the annotate process where the accessibility mark needs to be specified appropriately by the individual user.

Automatic marking of content objects is provided for the location mark, automatic organization mark and the subscription mark in compliance with the user-centered social software model (see Section 4.4.3).

Use Cases of the Rate Process

Content objects can be rated in the single-content view or the collection-based view within the Social Organizer by applying the use case *add rating*. In compliance with related social software platforms like YouTube [Yout07] shown in Figure 3.11, the metaphor of stars is applied for ratings in order to emphasize the specified value as the individual user's preference on the respective content object.

Application of the *add rating* and *remove rating* use cases is visualized in the *Details* tab of the single-content view in Figure 5.9. These use cases are supported in the collection-based view – namely in the search inside scope according to Figure 5.7 – in addition.

The use case *define rating* which is used for the definition of rating values in the Social Organizer is applied in the configuration settings (see Section 5.4.3).

5.3.6 Information Publication

Support of the use cases of information dissemination in the Social Organizer that require the specification of roles is described in the following subsections.

Specification of Roles

It is essential for information dissemination purposes to support the specification of roles in advance. The rule-based specification of a role is shown in Figure 5.11 with the prerequisite that restriction rule

schemata have already been defined and instantiated. In this case, already existing restriction rules for inclusion and exclusion of content objects can be applied for the specification of the role. Otherwise, it is possible to use newly established restriction rules as indicated by the button labeled 'New Restriction Rule' in Figure 5.11.

Primarily, Figure 5.11 depicts the selection of a set of rules for inclusion and exclusion of accessible content objects that meet the requirements of the selected restriction rules from all available restriction rule instantiations. In compliance with the user-centered social software model, the set of rules for inclusion and exclusion of content objects is composed by disjunction while exclusion rules are considered having higher priority over inclusion rules.

A further essential function is the assignment of contacts to the specified role as shown at the bottom part of the user interface in Figure 5.11. Contacts to be assigned to that role are selected from all contacts referenced by the Social Organizer. Equivalent contacts – having been specified manually as representing the same individual with regard to user profile information at different social software platforms – are shown as a single contact in the list of all contacts.

Rules specified on the user interface need to be transformed into appropriate query syntax – like in the case of restriction rules exemplified in Figure 5.11 – or a rule engine syntax which is not further considered within the scope of this thesis.

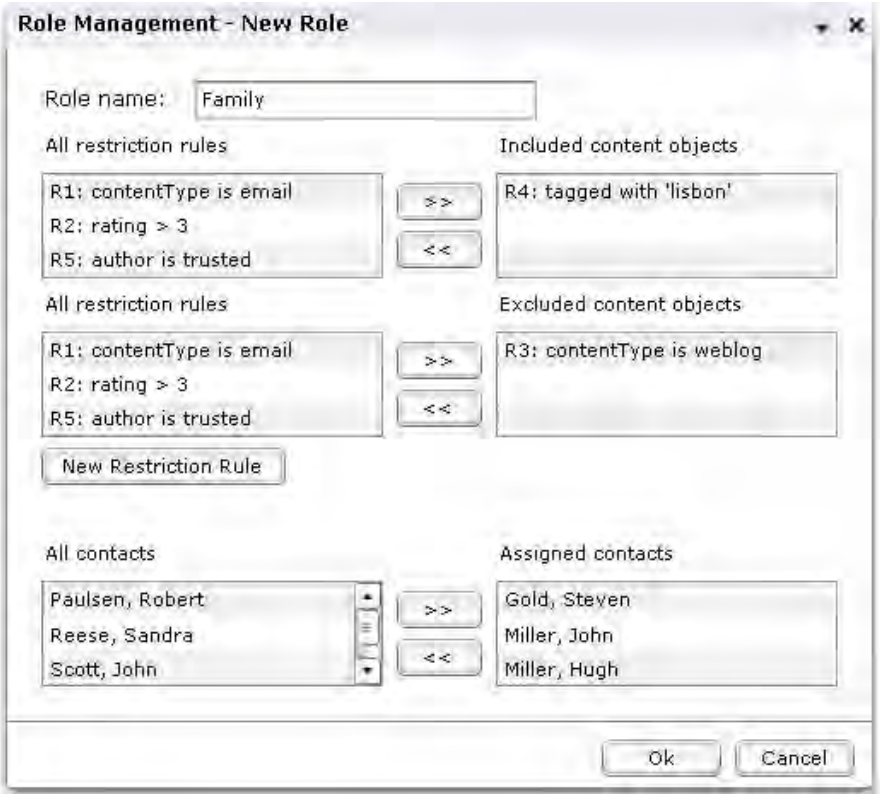


FIGURE 5.11. *Rule-based specification of a role*

By means of the Social Organizer, the individual user is able to visualize the effective view on a content object for a specified role as shown in Figure 5.12. If role-based access is enabled for a particular content object, the effective view indicates which content attributes of the object are visible in the selected role.

Although tags are represented by content objects of a special content type and linked to content objects (link type *classifies*) according to Section 5.1, from the individual user's perspective classification by tags is distinguished from links added to a content object (see Figure 5.12).

As further depicted in Figure 5.12, accessibility of properties and links of a content object is indicated by the traffic light metaphor in the Social Organizer which relies on the values of the respective accessibility mark. The color red (left) signifies that a content attribute is marked as *private* (e.g. property *rating*). The color yellow (middle) denotes that a property or link (e.g. link *enclosedIn*) is marked *role-based* and thus accessible within the current role which is *Friends* in the example of Figure 5.12. In an analogous manner, the color green (right) is appointed if the value of the accessibility mark for a content attribute is set to *public* and visible for any contact (e.g. tag *monaco*).

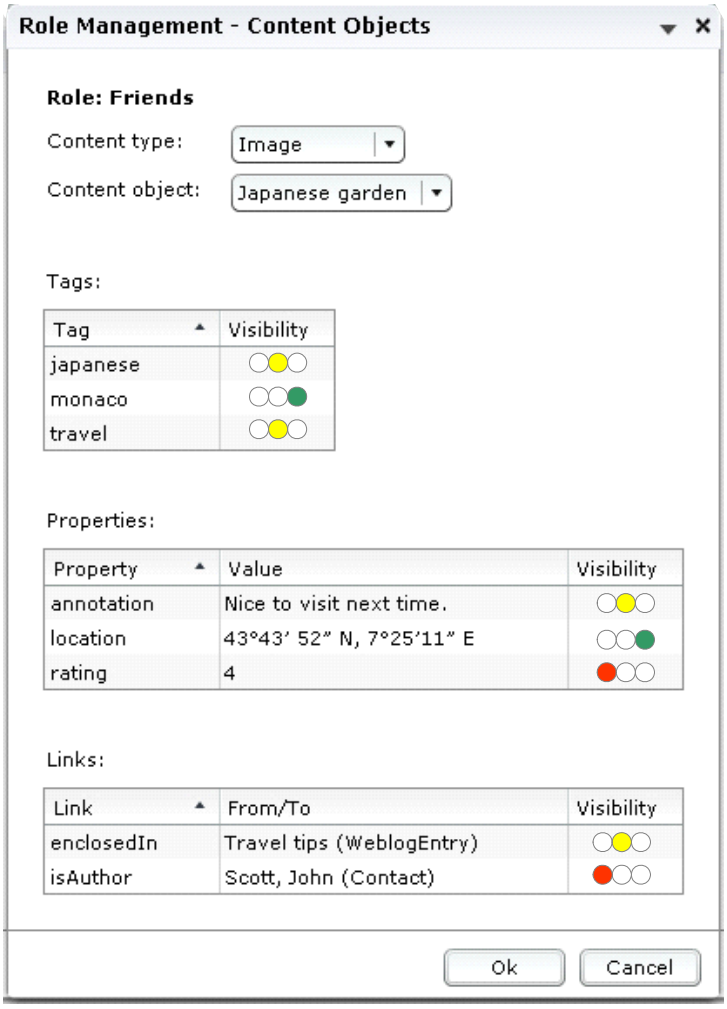


FIGURE 5.12. *Effective view on a content object by a role*

Use Case *Publish*

Publication of content objects to social software platforms in order to share them with designated contacts is shown in Figure 5.13 by the corresponding tab *Publish* of the single-content view. On this tab, a list of social software platforms as content providers is shown that are assigned to the current task context and support the content type of the content object which is displayed in the single-content view.

In the example of Figure 5.13, the bookmark object located at the social bookmarking service del.icio.us [Deli07] is additionally posted to the platform Linkroll [Link07a]. For that platform automatic synchronization and the subscription to feedback from contacts was specified by the individual user. The platform Simpy [Simp07] depicted at the top of the list also provides the management of content objects of type bookmark but the current object has not (yet) been posted there.

The individual user can monitor content objects that have been identified as being equivalent by the Social Organizer based on the application of the use cases of the post process. As indicated by the rightmost tab *Equivalence* of the single-content view in Figure 5.13, content objects equivalent to the displayed one can be optionally visualized.

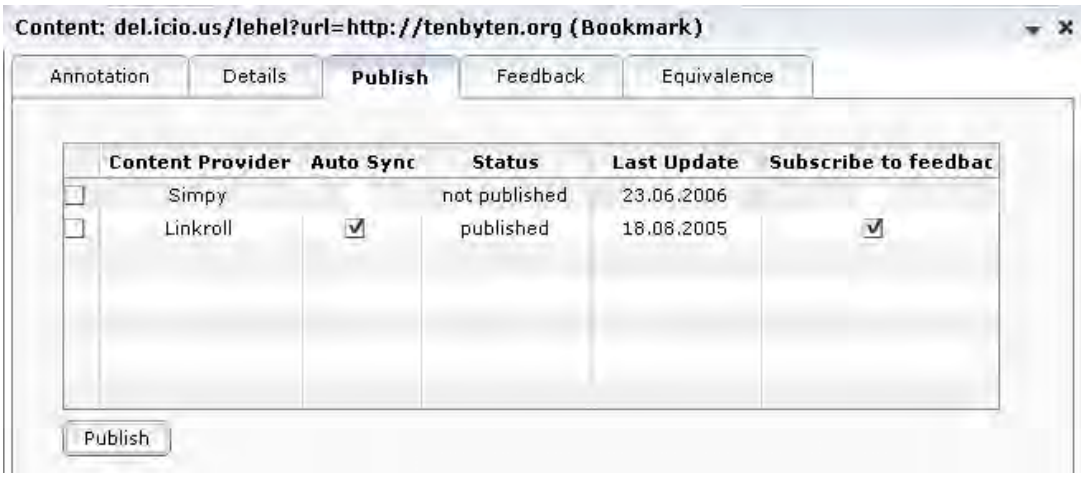


FIGURE 5.13. *The 'publish' use case*

In the case of contacts, however, the individual user manually determines which user profile information from different social software platforms represent the same individual. He is especially able to manage his own distributed user profiles at those platforms identified by pseudonyms. Hence, the Social Organizer enables the manual identification of contacts for authorization purposes. This scenario is exemplified in Figure 5.14 for the contact 'Miller, John'.

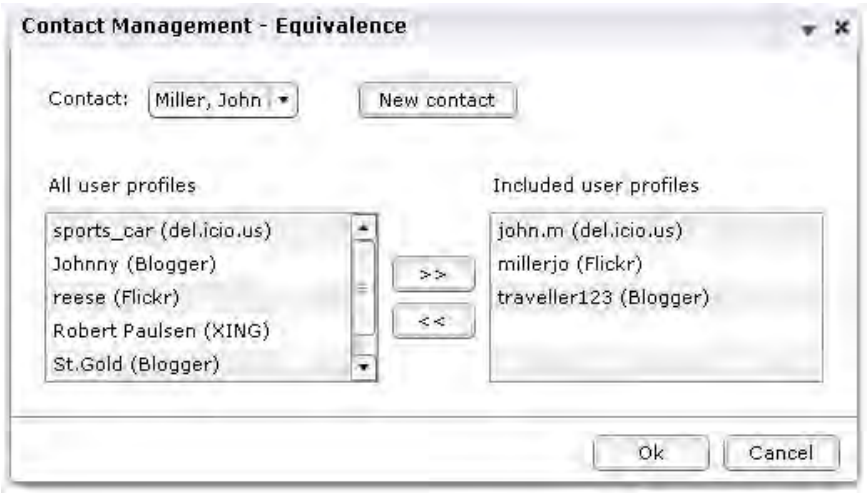


FIGURE 5.14. *Equivalence of user profiles*

Use Case *Post Note as Feedback*

The realization of supporting the *post note as feedback* use case within the Social Organizer is illustrated in Figure 5.15 in the *Annotation* tab of the single-content view. The *add note accessible to contacts* use case explained in Section 5.3.5 is a prerequisite for the use case *post note as feedback* since the individual user needs to attach an annotation to a content object in advance. That annotation may then be posted to the social software platform and attached to the respective content object of a contact.

Figure 5.15 exemplifies attaching a note to a bookmark object with the button labeled 'Post as Feedback' enabled since the annotation is not marked as *private* in this case.

FIGURE 5.15. *The 'post note as feedback' use case*

5.4 Implementation Aspects

Based on the design considerations for appropriate social information management support introduced in the previous sections, implementation aspects of the Social Organizer are explained in the following. The architecture of the Social Organizer is discussed within a centralized deployment scenario along with implementation details on each component. Further, configuration and extensibility issues of the Social Organizer are described.

5.4.1 Architectural Overview

Figure 5.16 presents an overview of the logical three-tier architecture of the Social Organizer. With regard to social information management support, an intuitive user interface is an essential part of the user-centered social software which invokes services provided by the business tier. The user interface for support of significant use cases is discussed in detail in Section 5.3.

The business tier incorporates two main components. On the one hand, the business logic component is responsible for support of information acquisition, organization and dissemination use cases relying on the considerations for the specification of rules discussed in Section 5.2.

On the other hand, social software services and services of personal applications are embedded by the service integration component for use case support according to the operations of the content provider entity presented in Section 5.1.8. A further issue is content representation of content providers within the information model which is discussed in Section 5.1.7. A separate adapter is required for each content provider – personal application or social software platform – in order to achieve scalability considering the integration of new content providers within the Social Organizer.

The persistence tier is established by the information repository that persistently stores content objects referenced and managed by the Social Organizer and is thus based on the information model presented in Section 5.1.

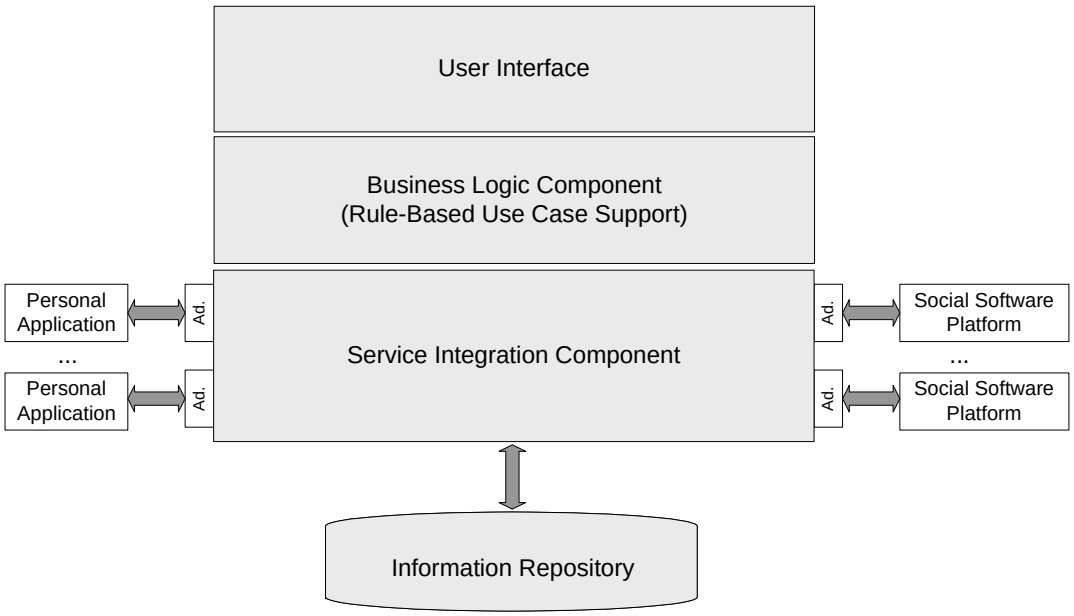


FIGURE 5.16. *Logical architecture of the Social Organizer*

Deployment Scenario

The architecture of the Social Organizer is realized based on a centralized deployment scenario shown in Figure 5.17. Thus, interface-based dependencies between the components conform to the layered architecture in Figure 5.16. Implementation-dependent execution environments are omitted from the illustration of the deployment scenario, though.

Implementation of the Social Organizer as a centralized application determines that the software components on each tier are located at a single node represented by the personal device. Personal application services are located at the personal device and represented by a multi-component in order to indicate that several personal applications can be integrated in the Social Organizer as content providers. Merely social software services are located as components at multiple nodes since they represent independent social software platforms.

This scenario is reasonable in the case of a stationary personal device or a mobile end user device that has sufficient resource capacity.

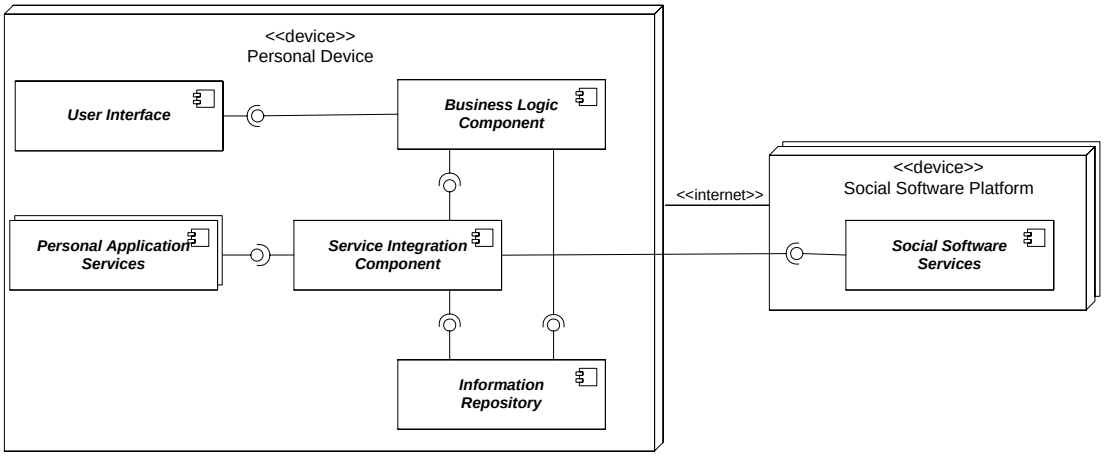


FIGURE 5.17. *Deployment scenario of the Social Organizer components*

5.4.2 Social Organizer Components

The following subsections describe implementation aspects about the software components the Social Organizer is composed of.

User Interface

In order to illustrate use case support of social information management by the Social Organizer as an innovative personal application, a web-based user interface relying on Rich Internet Application (RIA) technologies is reasonable.

Several RIA technologies exist for the development of web-based rich client applications, Ajax [Ga05] – an acronym for Asynchronous JavaScript and XML – and Adobe Flex [Flex06] representing two widely-adopted RIA technologies for Web 2.0 applications. Ajax incorporates several technologies of DHTML and XML [Ga05] whereas Flex is based on MXML and ActionScript representing proprietary technologies that compile into Flash applications [Flex06].

The RIA technology Adobe Flex [Flex06] has been applied for rapid prototyping of the Social Organizer user interface. The execution environment for Flex applications is an application server (e.g. Apache Tomcat [Tomc07]) and the Adobe Flash player which itself is executed within a web browser as shown in Figure 5.18.

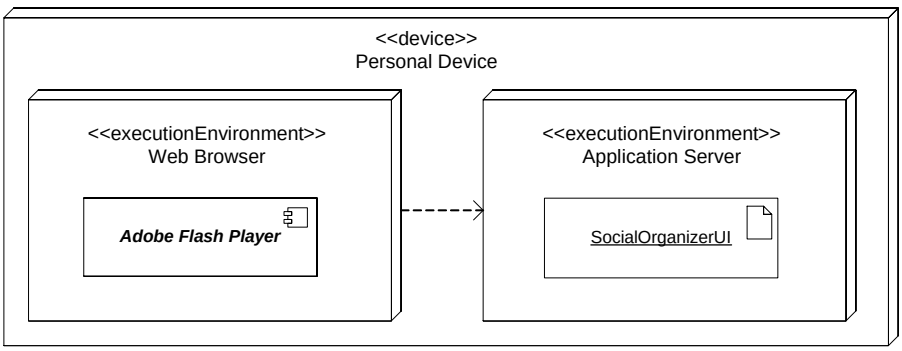


FIGURE 5.18. *Execution environment of the Social Organizer user interface*

Parts of the prototypical implementation of the user interface are the result of a system development project described in [Zh06]. The respective user interface views are shown throughout Section 5.3 to demonstrate support of significant social information management use cases within the Social Organizer.

Business Logic Component

The business logic component provides rule-based support for selected social information management use cases according to the specified business rules which provide an abstract representation of inherent and customizable business logic (see Section 5.2). While the specified business rules are usually realized by simple database queries and program code-based algorithms deductive and reactive rules can further be optimized by rule engine support [Ro03].

Thus, concerning constructive rules such as content filtering and content subscription rules for information acquisition purposes and business rules applied for the presentation of content–related elements on the user interface, an approach based on selection queries and program-code based algorithms is followed within the Social Organizer prototype.

```
(defrule S1
   (ContentReference ?c (contentType ?type))
   (ContentReference ?d (contentType ?type))
   ( OR  ((ContentReference ?p (contentType “Contact”))
          (Link ?l1 (linkType “isAuthor”) (from ?p) (to ?c))
          (Link ?l2 (linkType “isAuthor”) (from ?p) (to ?d))
         )
         ((ContentReference ?t (contentType “Tag”))
           (?count <- (accumulate (bind ?x 0)
                      (bind ?x (+ ?x 1))
                      ?x
                      ((Link ?l1 (linkType “classifies”) (from ?t) (to ?c))
                       (Link ?l2 (linkType “classifies”) (from ?t) (to ?d)))))
           (> ?count ?n))
         )
    )
   => (assert (Link (linkType “similarTo”) (from ?c) (to?d)))
)
```

TABLE 5.7. *Example of a similarity rule in Jess syntax [Fr06]*

For reasoning over the information model of the Social Organizer applying deductive and reactive rules, a rule engine such as Jess [Fr06] can be applied that provides an efficient mechanism for if-then conclusions over the underlying facts based on pattern matching (see Rete algorithm [Fo82]). The rule set of predefined rules is specified in advance according to Section 5.2 and constitutes the production memory of the rule engine. Facts are established from the information model and held in the working memory of the rule engine. A rule fires if its premise is fulfilled as a reaction to changes on the working memory [Jess06].

Table 5.7 illustrates the rule syntax of Jess [Fr06] applied on the example of the similarity rule given in (5.11).

Service Integration Component

The service integration component is responsible for integrating the services of content providers in order to facilitate use case support and provide higher-level services for the business logic component according to Section 5.1.8. Therefore, the service integration model operates on the domain-specific models of the content providers (see Section 3.3.1).

Adapters are implemented in Java for selected social software platforms (e.g. Flickr [Flic07], del.ici.ous [Deli07], Blogger [Blog07], XING [Xing07]) within the scope of the Social Organizer prototype described in the system development project report by [St07]. However, personal applications are not yet integrated in the prototype.

Information Repository

The information repository of the Social Organizer relies on the information model introduced in Section 5.1. The information repository can be realized by any adequate persistence mechanism including RDBMS, OODBMS and native XML databases.

Concerning the Social Organizer prototype, the appropriate database schema is realized using an RDBMS (MySQL [MySQ07]) which additionally requires the implementation of an object-relational mapping component that complements the business tier in the architecture of the Social Organizer in Figure 5.16. This component can be realized by an open source Java persistence framework such as Hibernate (see [BaKi05]) in the Social Organizer prototype.

5.4.3 Configuration and Extensibility

The user interface can be configured by individual users in order to adjust it to their specific needs. In the Social Organizer prototype, elements on the user interface such as the different tabs in the single-content view can be disabled by the individual user if he does not intend to visualize them.

The Social Organizer can further be adjusted to the individual user's needs by configuring and extending the appropriate content-related elements and rules.

A psychologically and sociologically founded initial configuration of the required content-related elements and rules – including rule schemata and instantiation – is beyond the scope of this thesis and is therefore subject to further research. Initial configuration values can then be specifically customized for each individual user, for example based on user modeling theories or exhaustive user studies.

The next sections thus propose an initial configuration adapted from conventions used at social software platforms according to Section 3.2. Parameter values are initially set to predefined configuration values that can be (re)adjusted by the individual user while working with the Social Organizer.

Initial Configuration of Content-Related Elements

Content Types – Predefined content types that exist independently of integrated content providers are *contact* to represent persons, *tag* to represent context information, *rule* and *feedback*. Initially, the Social Organizer is able to handle the following content types in order that significant social software platforms be integrated for social information management support: *document*, *e-mail*, *image*, *weblog*, *weblog entry*, *bookmark* and *feed*. In order to be capable of representing context information of

integrated social software platforms, the additional content types of *categories* and *folders* need to be provided.

Rating – The default interval for the value range of the rating is set to integer values from 0 to 5 in the prototype of the Social Organizer. The intended semantics of this initialization is that the rating 0 indicates an unspecified rating value for a content object referenced or managed by the Social Organizer, i.e. it has not yet been rated by the individual user. The value 1 thus signifies the lowest possible rating while value 5 represents the highest rating following the conventions of social software platforms like YouTube [Yout07] for example.

The individual user may define an arbitrary interval for the value range of the rating, however. User-defined configuration of rating values in the Social Organizer is illustrated in Figure 5.19 (left). For rating values, transformation functions for appropriate social software platforms can be specified.

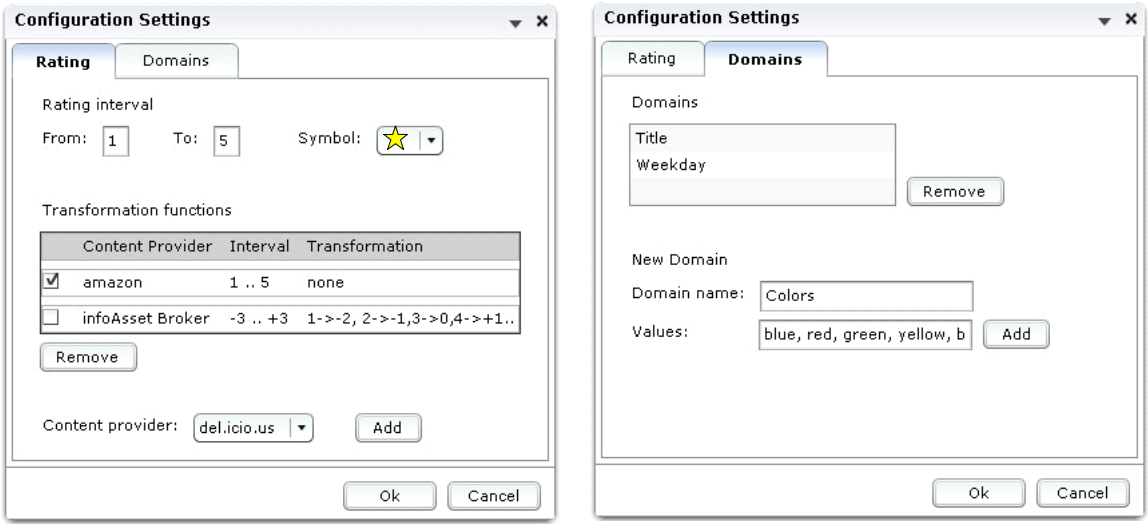


FIGURE 5.19. *Configuration settings in the Social Organizer*

Initial Configuration of Rules

Trusted Contacts – In order to initially specify trusted contacts, the four predefined roles (see Section 5.1.4) are used as parameter values and n as the minimal value for the rating is set to an average value of 3 – given that the default interval for the rating is 1-5 as stated above. However, contacts need to be specified, related to content and assigned to roles by the individual user in order that the rule of trusted contacts becomes applicable.

The initial instantiation for trusted contacts represents the default configuration for each newly created task context in the Social Organizer.

Relation of Content – The rule schema in (A_5) that specifies whether two content objects are related is adopted from the user-centered social software model but requires a default instantiation. The path length k is initially set to the value 1, that is, merely direct links are followed by default in order to determine related elements. The set of link types is set to *any*, also taking ad-hoc links into consideration.

The user may specify the path length from values between 1 and 5 due to a dense network structure of content objects that results from the application of inference rules on the information model. The set of link types may be configured manually by selecting a subset of all defined link types – predefined and user-defined – in the Social Organizer.

Similarity Rules – For initial instantiation of S_1 the parameter value of n that specifies the minimum number of tags two content objects need to share in order to be considered similar is set to the default value 3 in the Social Organizer according to observations of significant tag-based classification mechanisms at social software platforms such as del.icio.us [Deli07].

For user-defined instantiation of the predefined similarity rule S_1, this value can be configured by an arbitrary integer value by the individual user.

Extensibility of Entities

Besides user configuration of predefined elements, the Social Organizer facilitates user-defined extension of property types and link types on the one hand and content-related elements and rule schemata on the other hand.

User-defined extension of property types and link types is addressed by the respective social information management use cases *define property* and *define link* which are supported by the Social Organizer. The former use case includes the user-defined specification of domains which are not initially provided.

User-defined configuration of domain values in the Social Organizer is depicted in Figure 5.19 (right).

The extensibility of content types is required for newly integrated social software platforms and personal applications whose supported content type(s) cannot yet be handled by the Social Organizer. This extensibility cannot be achieved merely by user configuration but requires modifications on the implementation level.

5.5 Summary

The Social Organizer represents an innovative personal application that is designed to support social information management and relies on the user-centered social software model presented in Chapter 4. Thus, the information model of the Social Organizer is based on the core concepts of content reference, context and contact and also incorporates the model of task contexts for the definition of workspaces.

The model of properties and links is adopted in order to represent user-defined content attributes and original metadata from content providers. Content type-independent metadata such as annotation and rating that are added by the individual user during information evaluation obtain an appropriate representation by means of properties within the Social Organizer. Establishing links between arbitrary content objects based on user-defined link types and ad-hoc links enables cross-platform semantic relation management of content objects.

The Social Organizer further facilitates flexible schema-free metadata management by supporting the specification of untyped content objects in contrast to objects of rigidly defined content types managed at content providers.

In order to fulfill the requirements of the user-centered social software model, predefined property types (e.g. *annotation* and *rating*), links (e.g. *classifies*, *isAuthor*, *equivalentTo*) and marks (e.g. *accessibility*, *trust*) are introduced. Concerning context information, the Social Organizer facilitates tagging of content objects by linking them to objects of the predefined content type *tag*.

Social information management is further supported by information acquisition, organization and dissemination rules in compliance with the user-centered social software model. Within the Social Organizer, constructive, deductive and reactive rules can be specified. Predefined rule schemata exist for content filtering for information search, trusted contacts and inference rules based on algebraic characteristics of predefined link types. Parameterization of predefined rule schemata and the specification of user-defined rules in the social Organizer illustrates enhanced social information management support customized to the individual user's needs.

Support of information acquisition, organization and dissemination use cases is illustrated on a web-based user interface prototype applying the RIA technology Adobe Flex [Flex06]. The Social Organizer user interface consists of two main views. The collection-based view enables the integration of information acquisition and retrieval services and the accomplishment of information organization on collections of content objects. The single-content view provides further details on a particular content object and presents subscription, organization and dissemination services that can be applied to that object.

Role-based access control of content objects for information dissemination purposes is exemplified on the user interface by rule-based definition of roles and illustration of accessibility of content attributes of a content object by contacts who are assigned to that role.

Discussion of technological and architectural issues of further components of the Social Organizer prototype such as the service integration component to integrate retrieval, publication, subscription and feedback services of content providers and the integration of a rule engine in order to realize the business logic component is beyond the scope of this thesis and subject to further research.

Chapter 6

Conclusions and Prospects

This chapter summarizes the contributions of this thesis relying on the presented user-centered social software model and the design of the Social Organizer and points out possible directions for future research.

6.1 Summary of Contributions

The aim of this thesis is to characterize the concept of *user-centered social software* as an innovative approach for personal information management support by emphasizing information exchange in the individual user's social contexts. The focus lies on extending the support of PIM activities from personal applications to social software platforms since these innovative and widely-adopted Web 2.0 communities facilitate the formation of online social networks and thus enable the representation of the individual user's several social contexts.

Efficient PIM support requires unification across several personal applications that merely support selected content types and PIM activities – like e-mail clients, media tools or mind mapping tools – according to observations in relevant literature (see [Ki00], [JoBr05], and [KaJo06]). Over the last years, individual users have increasingly become members of newly emerging social software platforms that results in distributed management of their personal information. Therefore, a need arises for unification of PIM activities across those web community platforms. Since social software platforms also facilitate information acquisition from and publication and sharing with contacts it is reasonable to extend the scope of personal information management to the aforementioned activities in online social networks referred to as *social information management* within this thesis.

Social software platforms focus on particular content types (e.g. bookmark, contact, image, weblog, video) and support metadata management, information search, publication and sharing, subscription, commenting and collaborative classification in social contexts based on various underlying social networks. Hence, individual users can link together and exchange information with selected contacts by means of social software in contrast to conventional personal applications. However, functional features and information models of social software platforms reveal limitations regarding the

individual user's perspective, most significantly lack of service integration, flexible metadata management and cross-platform relation management of content objects.

Within the scope of this thesis, a classification of several kinds of social software is proposed. Further, a generic content provider model is developed that incorporates common characteristics of domain-specific content provider models of social software platforms, i.e. management of content, classification schemes and user profiles, and can also be applied to describe personal applications as content providers. The content provider model is complemented by a categorization of retrieval, publication, feedback and subscription services of social software platforms.

Based on these considerations, the family of user-centered social software is proposed – as opposed to the term object-centered social software coined by Rheingold [Rh05] which denotes a specific kind of social software platforms – in order to refer to an alteration of the view on social software platforms to the individual user's perspective instead of emphasizing shared content objects.

The user-centered social software model is characterized by the three primary concepts of content reference, context and contact that correspond to the abovementioned abstractions of the content provider model.

User-centered social software is designed to support individual users in social information management activities referred to as use cases of information acquisition, organization and dissemination. The essential functional requirements for adequate unified PIM support are fulfilled by the proposed model of user-centered social software by integrated services for searching, classifying and linking together arbitrary types of content objects and extending them by flexible user-defined metadata across personal applications and social software platforms.

Further requirements of social information management that result from information exchange in social networks are also addressed by the user-centered social software model. Hence, integrated support of publication, controlled sharing of content objects with selected contacts and subscription to content of contacts on social software platforms are provided by means of bidirectional integration of personal information managed by personal applications and social software platforms. Therefore, the integration of social software services is included in the model in order to present an integrated view on the available content to the individual user.

Efficiency of the individual user's social information management activities is achieved by customizable rule-based support. Several kinds of predefined and user-defined rules, i.e. constructive rules, deductive and reactive rules, appropriately specify the business logic of the respective use cases in the user-centered social software model.

For this purpose, rules are specified for each process of social information management. Information acquisition rules that specify task context-relevant and trustworthy information sources, content filtering for search, retrieval and subscription purposes and relation and similarity of content objects ensure adequate information supply when performing the respective use cases. Rules concerning information organization allow automatic classification and extension of content objects by additional metadata and linking together content objects.

The validity of the application of information dissemination use cases is primarily specified by the rule of authorization. Roles for supporting fine-grained access control to content objects in the case of information dissemination use cases are automatically established by means of rules.

Within the scope of this thesis, the Social Organizer is designed representing a personal application that conforms to the user-centered social software model and provides rule-based support of social information management. Parameterization of predefined rule schemata and the specification of user-

defined rules demonstrate social information management support customized to the individual user's needs.

User-centered support of information acquisition, organization and dissemination use cases is illustrated on a web-based user interface prototype applying innovative RIA technologies. The Social Organizer user interface consists of two main views. The collection-based view enables the integration of information acquisition and retrieval services and the accomplishment of information organization on collections of content objects. The single-content view provides further details on a particular content object and presents subscription, organization and dissemination services that can be applied to that object.

Role-based access control of content objects for information dissemination purposes is exemplified on the user interface by rule-based definition of roles and illustration of accessibility of content attributes of a content object by contacts who are assigned to that role.

The Social Organizer provides a unified view on private and shared content by integrating services and content from personal applications and social software platforms and thus enhances information acquisition and dissemination. Concerning information organization, the Social Organizer illustrates the benefits of schema-free metadata and cross-platform semantic relation management.

6.2 Future Work and Research Directions

The results presented in this thesis lead to future work with regard to the development of a full prototype that relies on the user-centered social software model and also establish the basis for further research directions.

Wei [We07] is concerned with technological and architectural issues regarding the realization of a user-centered social software exemplified by the Social Organizer. Thus, the service integration component of the Social Organizer prototype to integrate retrieval, publication, subscription and feedback services of content providers considering technological aspects (e.g. web APIs, protocols and formats) and the realization of the business logic component according to the specified rules are essential issues. Further, the Social Organizer as a personal application also needs to be evaluated based on a user study.

Since the Social Organizer is designed as a prototype for use on stationary devices, an appropriate version for mobile devices is still an issue for future research with an adequate user interface and supporting those use cases of social information management that are reasonable in the mobile scenario.

In the following, possible extensions to the user-centered social software model are discussed and the peer-to-peer scenario for user-centered social software instances is outlined.

6.2.1 Extensions to the User-Centered Social Software Model

So far, the user-centered social software model does not consider automatic information acquisition, organization and dissemination based on observations of the individual user's actions.

On the one hand, results of social network analysis performed on the respective social software platforms can make a significant contribution to automatically determine trustworthy contacts the individual user tends to engage in information exchange with. In this case, the user would not need to explicitly parameterize or specify the corresponding rule.

On the other hand, collecting activity-based metadata by user behavior monitoring (e.g. frequency of access to a content object, classification behavior) may lead to suggestions of appropriate content objects and services applicable to them within a task context. Thus, implicit ratings of content objects can be used in order to rank content objects by relevance in the collection-based view to enhance efficiency of information acquisition and retrieval.

Further, collaborative filtering (see [BaSh97]) could be applied in order to recommend new content of interest for individual users based on other users' preferences at social software platforms in addition to content-based filtering.

The special case of social information management support in the mobile scenario requires further extensions to the model of user-centered social software. Since personal tasks in the mobile scenario depend on the user context (e.g. acquiring information on a sight in a foreign city or searching for the nearest gas station) it is necessary to include appropriate rules in order to specify constraints in case of changing availability of information sources and the individual user's varying spatio-temporal context (see [MiKo04]).

6.2.2 Peer-to-Peer Scenario of User-Centered Social Software

Web content is primarily managed at designated web platforms such as social software platforms today. However, evolution of service-oriented architectures (SOA) (see [PaGe03]) on the web may result in a social network of individual users who do not access centralized services in order to obtain web content but directly access selected personal information of contacts that have been made available on personal devices. Thus, a peer-to-peer scenario for social information management support becomes increasingly significant and has to be addressed by the user-centered social software model.

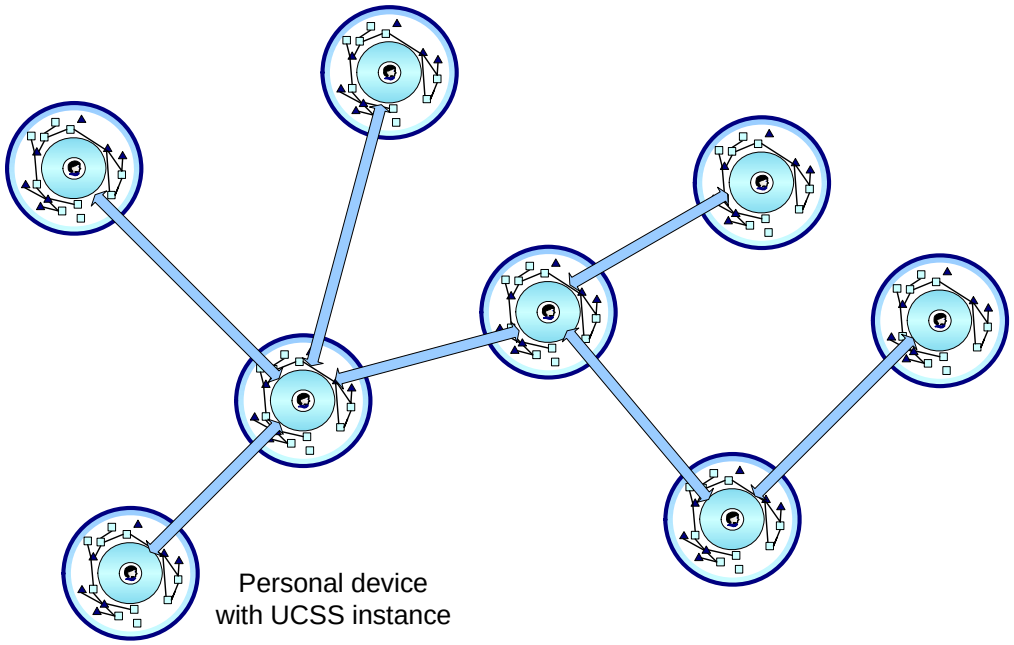


FIGURE 6.1. *Peer-to-peer scenario of user-centered social software*

Figure 6.1 illustrates the peer-to-peer application scenario of user-centered social software. In such a scenario, individual users apply their user-centered social software (UCSS) instances in order to

directly exchange personal information with selected contacts without mediation by social software platforms.

The user-centered social software also needs to provide the necessary social software services presented in Section 3.3.3 instead of merely providing integration of existing services. Further, synchronization scenarios between peers (UCSS instances) are essential as described in Section 4.2.4.

A benefit of the peer-to-peer architecture lies in centralized management of personally owned information on the individual user's personal device(s). Further, distributed identity management can be omitted by using a central repository of peers to facilitate identification of contacts by their personal device instead of maintaining multiple user profiles on social software platforms that otherwise need to be recognized as being equivalent, i.e. representing the same individual.

The peer-to-peer scenario is especially significant with regard to personal devices within a mobile ad-hoc network (see [CCL03]) since it may lead to ad-hoc information exchange opportunities with contacts (e.g. encountering individuals with similar interests).

Bibliography

[Av06] AVRAM, G.: *At the Crossroads of Knowledge Management and Social Software*. *Electronic Journal of Knowledge Management*, vol. 4, no. 1, pp. 1-10, 2006. Source: http://www.ejkm.com/volume-4/v4-i1/Avram.pdf. Retrieved January 2007.

[Ba95] BARREAU, D.K.: *Context as a Factor in Personal Information Management Systems*. *Journal of the American Society for Information Science*, vol. 46, no. 5, pp. 327-339, John Wiley & Sons, New York, NY, USA, 1995.

[Ba99] BARGER, J.: *Weblog resources FAQ*, 1999.
Source: http://www.robotwisdom.com/weblogs/index.html. Retrieved January 2007.

[Ba03] BARABÁSI, A.–L.: *Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume, New York, 2003.

[Ba06] BÄCHLE, M.: *Social Software*. *Informatik Spektrum*, vol. 29, no. 2, pp. 121-124, Springer, 2006.

[BaAl99] BARABÁSI, A.–L. and ALBERT, R.: *Emergence of scaling in random networks*. Science, 286, pp. 509–512, 1999.

[BaKi05] BAUER, C. and KING, G.: *Hibernate in Action*. Manning Publications, Greenwich, CT, 2005.

[BaSh97] BALABANOVIĆ, M. and SHOHAM, Y.: *Combining Content-Based and Collaborative Recommendation*. *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[BBN03] BERGMAN, O., BEYTH-MAROM, R. and NACHMIAS, R.: *The User-Subjective Approach to Personal Information Management Systems*. *Journal of the American Society for Information Science and Technology*, vol. 54, no. 9, pp. 872-878, 2003.

[BHL01] BERNERS-LEE, T., HAENDLER, J. and LASSILA, O.: *The Semantic Web*. *Scientific American*, 284 (5), pp. 34–43, 2001.

[Bi79] BIDDLE, B.J. *Role Theory. Expectations, Identities and Behaviors*. Academic Press, New York, 1979.

[BiZa05] BIELENBERG, K. and ZACHER, M.: *Groups in Social Software: Utilizing Tagging to*

Integrate Individual Contexts for Social Navigation. Master Thesis, Universität Bremen, 2005.

[BlAt03] BLUMBERG, R. and ATRE, S.: *The Problem with Unstructured Data*. *DM Review magazine*, February, 2003.

[Blog07a] BLOGGER DATA API *documentation*. Source: http://code.google.com/apis/gdata/blogger.html. Retrieved January 2007.

[Bo01] BOARDMAN, R. *Category Overlap between Hierarchies in User Workspace*. In *Proceedings of IFIP INTERACT'01: Human-Computer Interaction*, pp.759-760, 2001.

[Bo02] BOARDMAN, R.: *Workspaces That Work: Towards Unified Personal Information Management*. In *Proceedings of 16th British HCI Conference, People and Computers XVI*, vol. 2, pp. 216-217, 2002.

[Bo04] BOARDMAN, R.: *Improving Tool Support for Personal Information Management*. Ph.D. Thesis, Department of Electrical and Electronic Engineering, Imperial College London, UK, 2004.

[BrMa05] BRY, F., MARCHIORI, M.: *Ten Theses on Logic Languages for the Semantic Web*. In *Proceedings of the International Workshop on Principles and Practice of Semantic Web Reasoning*, LNCS vol. 3703, pp. 42-49, Springer, 2005.

[BrMi05] BRICKLEY, D. and MILLER, L.: *FOAF Vocabulary Specification*, July 2005. Source: http://xmlns.com/foaf/0.1/. Retrieved January 2007.

[Bu45] BUSH, V.: *As We May Think. The Atlantic Monthly*, July 1945, pp. 101–108, 1945.

[Bu92] BURT, R.S.: *Structural holes*. Harvard University Press, Boston, MA, 1992.

[Bu97] BUCKLAND, M.K.: *What is a "Document"? Journal of the American Society for Information Science and Technology*, vol. 48, no. 9, pp. 804-809, 1997.

[Ca06] CALLAHAN, S.: *Data, Information, Knowledge: a sensemaking perspective*, March 2006. Source: http://www.anecdote.com.au/archives/2006/03/data_informatio.html. Retrieved January 2007.

[CCL03] CHLAMTAC, I., CONTI, M., LIU, J. J.-N.: *Mobile ad hoc networking: imperatives and challenges*. *Ad Hoc Networks*, vol. 1, no. 1, pp. 13-64, 2003.

[CDT06] CUTRELL, E., DUMAIS, S.T. and TEEVAN, J.: *Searching to Eliminate Personal Information Management*. *Communications of the ACM*, vol. 49, no. 1, pp. 58-64, 2006.

[ChHa05] CHURCHILL, E.F. and HALVERSON, C.A.: *Social Networks and Social Networking*. IEEE Internet Computing, vol. 9, no. 5, pp. 14–19, 2005.

[CMN83] CARD, S.K., MORAN, T.P. and NEWELL, A.: *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc. Publishers, Hillsdale, New Jersey, 1983.

[Co03] COATES, T.: *On Permalinks and Paradigms…*, 2003. Source: http://www.plasticbag.org/archives/2003/06/on_permalinks_and_paradigms/. Retrieved

January 2007.

[CoBe02] COOL, C. and BELKIN, N.J.: *A Classification of Interactions with Information*. In *Proceedings of the Fourth International Conference on Conceptions of Library and Information Science*, pp. 1-15, Libraries Unlimited, 2002.

[Cu07] CUMMINGS, N.J.: *NetVis Module – Dynamic Visualization of Social Networks*. Source: http://www.netvis.org. Retrieved January 2007.

[CuDu06] CUTRELL, E. and DUMAIS, S.T.: *Exploring Personal Information*. *Communications of the ACM*, vol. 49, no. 4, pp. 50-51, 2006.

[Da94] DAVENPORT, T.H.: *Saving IT's Soul: Human-Centered Information Management*. Harvard Business Review, vol. 72, no. 2, pp. 119-131, 1994.

[Da05] DAVENPORT, T.H.: *Personal Knowledge Management*. Source: http://www.babsonknowledge.org/2005/11/personal_knowledge_management.htm. Retrieved January 2007.

[Da06] DAHM, M.: *Grundlagen der Mensch-Computer-Interaktion.* Pearson Education, 2006.

[De03] DESSLOCH, A., MAIER, A., MATTOS, N. and WOLFSON, D.: *Information Integration – Goals and Challenges*. *Datenbank-Spektrum*, vol.6, p. 7-13, 2003.

[DET80] DANIEL, S., ELLIS, J. and TRUSCOTT, T.: *USENET – A General Access UNIX Network*. Duke University, 1980.

[Do01] DORSEY, P.A.: *Personal Knowledge Management: Educational Framework for Global Business*. Millikin University, 2001.
Source: http://www.millikin.edu/pkm/pkm_istanbul.html. Retrieved January 2007.

[Do03] DÖRING, N.: *Sozialpsychologie des Internet. Die Bedeutung des Internet für Kommunikationsprozesse, Identitäten, soziale Beziehungen und Gruppen.* Hogrefe, Verlag für Psychologie, Göttingen, 2003.

[Dr93] DRUCKER, P.F.: *Post-Capitalist Society*. Harper Business, 1993.

[Du03] DUMAIS, S.T., CUTRELL, E., Cadiz, J.J., JANCKE, G., SARIN, R. and ROBBINS, D.C.: *Stuff I've Seen: A System for Personal Information Retrieval*. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 72-79, ACM Press, New York, NY, USA, 2003.

[Du07] DUSSEAULT, L.: *Vision of Chandler 1.0*. Source: http://www.chandler.osafoundation.org/1.0_vision.php. Retrieved January 2007.

[Ef03] EFIMOVA, L.: *What's in it for me? Using weblogs to understand knowledge workers*. Paper abstract, Telematica Instituut, Enschede, Netherlands, 2003.

[ErRe59] ERDŐS, P. and RÉNYI, A.: *On random graphs*. *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[Eu02] EUZENAT, J.: *Personal Information Management and the Semantic Web*. W3C Semantic Web Activity: Advanced Development, 2002.

Source: http://www.w3.org/2001/sw/Europe/200210/calendar/SyncLink.html. Retrieved January 2007.

[FCK95] FERRAIOLO, D.F., CUGINI, J.A. and KUHN, D.R.: *Role-Based Access Control (RBAC): Features and Motivations. 11th Annual Computer Security Applications Proceedings*, 1995.

[FFG96] FERTIG, S., FREEMAN, E. and GELERNTER, D.: *Lifestreams: An Alternative to the Desktop Metaphor*. In *ACM SIGCHI Conference on Human Factors in Computing Systems Conference Companion (CHI '96)*, pp. 410 - 411, ACM Press, New York, NY, USA,1996.

[Fl05] FLORIDI, L.: *Semantic Conceptions of Information.* In *Stanford Encyclopedia of Philosophy*, 2005. Source: http://plato.stanford.edu/entries/information-semantic/. Retrieved January 2007.

[Flex06] ADOBE FLEX 2, technical white paper, 2006. Source: http://www.adobe.com/products/flex/whitepapers/pdfs/flex2wp_technicaloverview.pdf. Retrieved January 2007.

[Flic07a] FLICKR API *documentation*, 2006. Source: http://www.flickr.com/services/api/. Retrieved January 2007.

[Fo82] FORGY, C.L.: *Rete: A Fast Algorithm for the Many Pattern / Many Object Pattern Match Problem. Artificial Intelligence*, vol. 19, 1982.

[Fr00] FREEMAN, L.C.: *Visualizing Social Networks. Journal of Social Structure*, vol. 1, no. 1, 2000.

[Fr04] FREEMAN, L.C.: *The Development of Social Network Analysis.* BookSurge, LLC, North Charleston, SC, 2004.

[Fr06] FRIEDMAN-HILL, E.J.: *Jess – The Rule Engine for the Java Platform*, December 2006. Source: http://herzberg.ca.sandia.gov/jess/docs/Jess70p1.pdf. Retrieved January 2007.

[FrGe96] FREEMAN, E. and GELERNTER, D.: *Lifestreams: A Storage Model for Personal Data*. In *ACM SIGMOD Record (ACM Special Interest Group on Management of Data)*, vol. 25, no. 1, pp. 80-86, 1996.

[Ga04] GALLA, M.: *Social Relationship Management in Internet-based Communication and Shared Information Spaces*. Dissertation, Department of Informatics, Technische Universität München, Germany, 2004.

[Ga05] GARRETT, J.J.: *Ajax: A New Approach to Web Applications*, February 2005. Source: http://www.adaptivepath.com/publications/essays/archives/000385.php. Retrieved January 2007.

[GeoT07] GEOTAGS, *geographical identification metadata format*. Source: http://geotags.com/geo/geotags2.html. Retrieved January 2007.

[GeoU07] GEOURL, *geographical identification metadata format.* Source: http://www.geourl.org. Retrieved January 2007.

[GHW97] GARTON, L., HAYTHORNTHWAITE, C. and WELLMAN, B.: *Studying online social networks*. *Journal of Computer Mediated Communicatio*n, vol. 3, no. 1, 1997.

[Go99] GODBOUT, A.J.: *Filtering Knowledge: Changing Information into Knowledge Assets*. *Journal of Systemic Knowledge Management*, January, 1999.

[Gr73] GRANOVETTER, M.: *The Strength of Weak Ties*. American Journal of Sociology, vol. 78, no. 6, pp. 1360–1380, 1973.

[Gr04] GRAHAM, A.: *Arranging Ideas: Knowledge Management in Human Terms*. 2004. Source: http://blog.contentious.com/archives/2004/07/16/arranging-ideas-knowledge-management-in-human-terms. Retrieved January 2007.

[GrHo06] GREGORIO, J. and DE HÓRA, B. (eds.): *The Atom Publishing Protocol*. December 2006. Source: http://ietfreport.isoc.org/all-ids/draft-ietf-atompub-protocol-12.txt. Retrieved January 2007.

[Ha69] HARARY, F.: *Graph Theory*. Addison-Wesley, Reading, MA, 1969.

[Ha96] HAYTHORNTHWAITE, C.: *Social Network Analysis: An Approach and Technique for the Study of Information Exchange*. *Library and Information Science Research*, vol. 18, no. 4, pp. 323–342, 1996.

[Ha05] HAMMOND, T., HANNAY, T., LUND, B. and SCOTT, J.: *Social Bookmarking Tools (I): A General Review*. *D-Lib Magazine*, vol. 11, no. 4, 2005.
Source: http://www.dlib.org/dlib/april05/hammond/04hammond.html. Retrieved January 2007.

[He05] HERRING, S.C., KOUPER, I., PAOLILLO, J.C., SCHEIDT, L.A., TYWORTH, M., WELSCH, P., WRIGHT, E. and YU, N.: *Conversations in the Blogosphere: An Analysis "From the Bottom Up"*. In *Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS'05)*, IEEE Press, 2005.

[He06] HEARST, M.A.: *Clustering versus Faceted Categories for Information Exploration*. *Communications of the ACM*, vol. 49, no. 4, pp. 59-61, 2006.

[HiDu03] HILL, R. and DUNBAR, R.: *Social Network Size in Humans*. *Human Nature*, vol. 14, no. 1, pp. 53–72, 2003.

[Hu01] HUBERMAN, B.A.: *The Laws of the Web – Patterns in the Ecology of Information*. MIT Press, 2001.

[Hu05] HUBERMAN, B.A.: *Information flow in social networks*. Complexity workshop, Hewlett Packard, Grenoble, November 2005. Source:
http://www.psych.lse.ac.uk/complexity/Workshops/Grenoble/
HubermanGren28Nov.pdf. Retrieved January 2007.

[Jo04] JONES, W.: *Finders, keepers? The present and future perfect in support of personal information management*. *First Monday*, vol. 9, no. 3, 2004. Source: http://www.firstmonday.org/issues/issue9_3/jones/index.html. Retrieved January 2007.

[JoBr05] JONES, W. and BRUCE, H.: *A Report on the NSF-Sponsored Workshop on Personal*

Information Management, Seattle, WA, 2005. Source: http://pim.ischool.washington.edu/final%20PIM%20report.pdf. Retrieved January 2007.

[Ka05] KARGER, D.R., BAKSHI, K., HUYNH, D., QUAN, D. and SINHA, V.: *Haystack: A Customizable General-Purpose Information Management Tool for End Users of Semistructured Data*. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR 2005)*, pp. 13-27, 2005.

[KaJo06] KARGER, D.R. and JONES, W.: *Data Unification in Personal Information Management*. *Communications of the ACM*, vol. 49, no. 1, pp. 77-82, 2006.

[Ki00] KIRSH, D.: *A Few Thoughts on Cognitive Overload*, *Intellectica*, vol. 30, pp. 19–51, 2000.

[KlCa04] KLYNE, G. and CARROLL, J.J. (eds.): *Resource Description Framework (RDF): Concepts and Abstract Syntax*, 2004. Source: http://www.w3.org/TR/rdf-concepts/. Retrieved January 2007.

[Kr04] KRCMAR, H.: *Informationsmanagement*. Springer Verlag, Berlin, 2004.

[La88] LANSDALE, M.W.: *The psychology of personal information management*. *Applied Ergonomics*, vol. 19, no. 1, pp. 55-66, 1988.

[Le01] LEHEL, V.: *Entwurf und Realisierung eines Basisdienstes zur Unterstützung arbeitsteiliger Geschäftsvorgänge in einem Unternehmensportal*. Software Systems Group, Technische Universität Hamburg-Harburg, Germany, 2001.

[Le02] LEHEL, V.: *Synchronisation von Informationsobjekten zwischen Portalen*. Diploma Thesis, Software Systems Group, Technische Universität Hamburg-Harburg, Germany, 2002.

[LeCu01] LEUF, B. and CUNNINGHAM, W.: *The Wiki way: quick collaboration on the Web.* Addison-Wesley, London, 2001.

[LeMa04] LEHEL, V. and MATTHES, F.: *Integration von Weblog-Funktionen in eine betriebliche Standardsoftware zum Wissensmanagement*. In proceedings *Wissensmanagement - Wandel, Wertschöpfung, Wachstum (KnowTech 2004)*, pp. 217–224, Munich, Germany, 2004.

[LeMa05] LEHEL, V. and MATTHES, F.: *User-Centered Social Software - Der Wissensarbeitsplatz der Zukunft?* In proceedings *Wissensmanagement - Motivation, Organisation, Integration (KnowTech 2005)*, pp. 369-376, Munich, Germany, 2005.

[LMR04a] LEHEL, V., MATTHES, F. and RIEDEL, S.: *Linkage Flooding*. Technical Report 0402, Chair Software Engineering for Business Information Systems, Department of Informatics, Technische Universität München, Germany, 2004.

[LMR04b] LEHEL, V., MATTHES, F. and RIEDEL, S.: *Linkage Flooding: Ein Algorithmus zur dateninhaltsorientierten Fusion in vernetzten Informationsbeständen.* In proceedings *Informatik verbindet (Informatik 2004)*, vol. 1, pp. 346–355, Gesellschaft für Informatik, Ulm, Germany, 2004.

[LMS03] LEHEL, V., MATTHES, F. and STEINFATT, K.: *Weblogs als ein innovatives Instrument des betrieblichen Wissensmanagements*. In proceedings *Interaktion in Bewegung (Mensch & Computer 2003)*, pp. 229–238, Stuttgart, Germany, 2003.

[LMW05] LEHEL, V., MATTHES, F. and WEI, S.: *Towards a Lightweight User-Centered Content Syndication Architecture*. In proceedings *Berliner XML-Tage 2005*, pp. 129–134, XML-Clearinghouse, Berlin, Germany, 2005.

[LyVa03] LYMAN, P. and VARIAN, H.R.: *How Much Information? 2003*. Source: http://www.sims.berkeley.edu/how-much-info-2003. Retrieved January 2007.

[Ma03] MAYFIELD, R.: *Blog Tribe Social Network Mapping*, 2003. Source: http://radio.weblogs.com/0114726/2003/01/02.html. Retrieved January 2007.

[Ma05] MATSUMURA, N., ASAKO, M., SHIBANAI, Y., OHSAWA, Y. and NISHIDA, T: *The dynamism of 2channel*. *AI & Society*, vol. 19, no. 1, pp. 84–92, 2005.

[Ma06] MARCHIONINI, G.: *Exploratory Search: From Finding to Understanding*. *Communications of the ACM*, vol. 49, no. 4, pp. 41-46, 2006.

[MaJo06] MARSHALL, C.C. and JONES, W.: *Keeping Encountered Information*. *Communications of the ACM*, vol. 49, no. 1, pp. 66-67, 2006.

[MaLe02a] MATTHES, F. and LEHEL, V.: *Dokument- und Kontaktsynchronisation mit mobilen Datenbanken: Anforderungen und Lösungsansätze aus Sicht von Unternehmensportalen*. In *Tagungsband Workshop Mobile Datenbanken und Informationssysteme – Datenbanktechnologie überall und jederzeit, März 2002*, pp. 3–9, Otto-von-Guericke Universität-Magdeburg, Germany, 2002.

[MaLe02b] MATTHES, F. and LEHEL, V.: *Persönliche Informations- und Wissensportale als dualer Ansatz zu Unternehmensportalen*. In proceedings *Informatik bewegt (Informatik 2002)*, pp. 551–557, Gesellschaft für Informatik, Dortmund, Germany, 2002.

[Mi67] MILGRAM, S.: *The Small World Problem*. Psychology Today 1, pp. 60–67, 1967.

[Mi69] MITCHELL, J.C.: *The Concept and Use of Social Networks.* In MITCHELL, J.C. (ed.): *Social Networks in Urban Situations*, pp. 1–50, Manchester University Press, Manchester, 1969.

[MiGa04] MIKA, P. and GANGEMI, A.: *Descriptions of Social Relations*. *First Workshop on Friend of a Friend, Social Networking and the Semantic Web*, Galway, Ireland, 2004.
Source: http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/descriptions_of_social_relations/. Retrieved January 2007.

[MiKo04] MIKALSEN, M. and KOFOD-PETERSEN, A.: *Representing and Reasoning about Context in a Mobile Environment*. In *Modeling and Retrieval of Context*, CEUR Workshop Proceedings, vol. 114, pp. 25-35, 2004. Source: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-114/paper3.pdf. Retrieved January 2007.

[Mo37] MORENO, J.L.: *Sociometry in Relation to Other Social Sciences*. Sociometry, vol. 1, no. 2, pp. 206–219, 1937.

[Mo53] MORENO, J.L.: *Who shall survive?* Beacon House, Beacon, New York, 1953.

[Mo04] MOOSBRUGGER, B.: *Interaktive selbstorganisierende Karten in einem Informationsportal*. Diploma Thesis, Department of Informatics, Technische Universität München, Germany, 2004.

[Mu99] MÜLLER, C.: *Networks of 'personal communities' and 'group communities' in different online communication services*. In ARMITAGE, J. and ROBERTS, J. (eds.): *Exploring Cyber Society. Social, Political, Economic and Cultural Issues*. University of Northumbria at Newcastle, UK, vol. 2, 1999.

[Na02] NARDI, B.A., WHITTAKER, S., ISAACS, E., CREECH, M., JOHNSON, J. and HAINSWORTH, J.: *Integrating Communication and Information through ContactMap. Communications of the ACM*, vol. 45, no. 4, pp. 89-95, 2002.

[NeMa98] NEŠETŘIL, J. and MATOUŠEK, J.: *Invitation to Discrete Mathematics*. Oxford University Press, 1998.

[No88] NORMAN, D.A.: *The Psychology of Everyday Things*. Basic Books, 1988.

[NoSa05] NOTTINGHAM, M. and SAYRE, R. (eds.): *The Atom Syndication Format*. IETF RFC 4287, December 2005. Source: http://www.ietf.org/rfc/rfc4287.txt. Retrieved January 2007.

[NoTa95] NONAKA, I. and TAKEUCHI, H.: *The Knowledge-Creating Company*. Oxford University Press, Oxford, 1995.

[OiRe93] OIKARINEN, J. and REED, D.: *Internet Relay Chat Protocol*. IETF RFC 1459, May 1993. Source: http://tools.ietf.org/html/rfc1459. Retrieved January 2007.

[OPML00] OPML 1.0 *Specification*, September 2000. Source: http://www.opml.org/spec. Retrieved January 2007.

[Or05] O'REILLY, T.: *What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software*, 2005. Source: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. Retrieved January 2007.

[Pa88] PATASHNIK, O.: *BibTex bibliographic reference format*, 1988. Source: ftp://sunsite.unc.edu/pub/packages/TeX/biblio/bibtex/distribs/doc/btxdoc.tex. Retrieved January 2007.

[Pa03] PAQUET, S: *Socio-technological approaches to facilitating knowledge sharing across disciplines*, Ph.D. Thesis, Université de Montréal, Canada, 2003.

[Pa06] PĂTRÂNJAN, P.-L. (ed.): *Use Cases for a Rule Interchange Format*. REWERSE project document, 2006. Source: http://rewerse.net/deliverables/m24/std-d1.pdf. Retrieved January 2007.

[PaGe03] PAPAZOGLOU, M.P. and GEORGAKOPOULOS, D.: *Service-Oriented Computing. Communications of the ACM*, vol. 46, no. 10, pp. 25-28, 2003.

[Po58] POLANYI, M.: *Personal Knowledge: Towards a Post-Critical Philosophy*. University of Chicago Press, Chicago, 1958.

[Po66] POLANYI, M.: *The Tacit Dimension*. Doubleday, New York, 1966.

[Qu03] QUAN, D. BAKSHI, K., HUYNH, D. and KARGER, D.R.: *User Interfaces for Supporting Multiple Categorization*. In *Proceedings of INTERACT 2003*, pp. 228-235, IOS Press, 2003.

[QuKa04] QUAN, D. and KARGER, D.R.: *How to Make a Semantic Web Browser*. In *Proceedings of the 13th International WWW Conference (WWW '04)*, pp. 255-265, ACM Press, New York, NY, USA, 2004.

[Ra97] RAUTENSTRAUCH, C.: *Effiziente Gestaltung von Arbeitsplatzsysteme: Konzepte und Methoden des Persönlichen Informationsmanagements*. Addison-Wesley-Longman, Bonn, 1997.

[Ra00] RASKIN, J.: *The Humane Interface. New Directions for Designing Interactive Systems*. Addison-Wesley, 2000.

[Ra04] RAVASIO, P.: *Personal Information Organisation: Studies on User-Appropriate Classification and Retrieval Strategies and their Implications for Information Management Systems Design*. Dissertation, Swiss Federal Institute of Technology Zurich, 2004.

[RaRa04] RAVID, G. and RAFAELI, S.: *Asynchronous discussion groups as Scale Free and Small World Networks*. *First Monday*, vol. 9, no. 9, 2004.
Source: http://firstmonday.org/issues/issue9_9/ravid/index.html. Retrieved January 2007.

[Rh05] RHEINGOLD, H.: *Why some social network services work and others don't*, May 2005. Source: http://www.smartmobs.com/archive/2005/05/07/why_some_social.html. Retrieved January 2007.

[Ri03] RIEDEL, S.: *Entwicklung eines Modells, Verfahrens und softwaretechnischen Rahmenwerks für Record Linkage in semantischen Netzen*. Diploma Thesis, Software Systems Group, Technische Universität Hamburg-Harburg, Hamburg, Germany, 2003.

[RiGa98] RIVA, G. and GALIMBERTI, C.: *Computer-mediated communication: identity and social interaction in an electronic environment*. *Genetic, Social and General Psychology Monographs* 124, pp. 434–464, 1998.

[Ro03] ROSS, R.G.: *Principles of the Business Rule Approach*. Addison-Wesley, 2003.

[Ro04] RÖLL, M.: *Distributed KM - Improving Knowledge Workers' Productivity and Organisational Knowledge Sharing with Weblog-based Personal Publishing*. In *Tagungsband zur BlogTalk 2.0*, pp. 139–164, Donau-Universität Krems, Kulturwissenschaften, Vienna, 2004.

[SBD06] SHNEIDERMAN, B., BEDERSON, B.B. and DRUCKER, S.M.: *Find that Photo! Interface Strategies to Annotate, Browse, and Share*. *Communications of the ACM*, vol. 49, no. 4,

pp. 69-71, 2006.

[Sc00] SCOTT, J.: *Social Network Analysis. A Handbook*. 2. ed., Sage, London, 2000.

[Sh48] SHANNON, C.E.: *A Mathematical Theory of Communication*. *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[Si06] SIFRY, D.: *State of the Blogosphere, April 2006 Part 1: On Blogosphere Growth*. Source: http://www.sifry.com/alerts/archives/000432.html. Retrieved January 2007.

[Sp06] SPURGIN, K. M.: *The Sense-Making Approach and the Study of Personal Information Management.* Personal Information Management Workshop at SIGIR 2006, Seattle, 2006.

[St04] STEFFENS, U.: *Metadaten-Management für kooperative Anwendungen.* Dissertation, Software Systems Group, Technische Universität Hamburg-Harburg, Hamburg, 2004.

[St07] STAGE, A.: *Erstellung einer Komponente zur Unterstützung der Integration von ausgewählten Social Software Plattformen*. System development project, Department of Informatics, Technische Universität München, Germany, 2007.

[Te03] TEPPER, M.: *The rise of social software*. *netWorker*, vol. 7, no. 3, pp.18–23, ACM Press, New York, NY, USA, 2003.

[Te07] TEEVAN, J.: *Supporting Finding and Re-Finding through Personalization*. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, USA, 2007.

[TJB06] TEEVAN, J., JONES, W. and BEDERSON, B.B.: *Personal Information Management*. *Communications of the ACM*, vol. 49, no. 1, pp. 40-43, 2006.

[Trac04] TRACKBACK *Technical Specification 1.2*, August 2004.
Source: http://www.sixapart.com/pronet/docs/trackback_spec. Retrieved January 2007.

[Vo06] VÖLKEL, M., KRÖTZSCH, M., VRANDECIC, D., HALLER, H. and STUDER, R. *Semantic Wikipedia*. In P*roceedings of the 15th international conference on World Wide Web*, pp. 585-594, ACM Press, New York, NY, USA, 2006.

[VoIP07] VOICE OVER IP protocol. Source: http://www.fcc.gov/voip/. Retrieved January 2007.

[Wa99] WATTS, D.J.: *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press, 1999.

[Wa02] WATSON, I.: *Applying Knowledge Management: Techniques for Building Corporate Memories*. Morgan Kaufmann, 2002.

[WaFa94] WASSERMAN, S. and FAUST, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, 1994.

[WBG06] WHITTAKER, S., BELLOTTI, V. and GWIZDKA, J.: *Email in Personal Information Management*. *Communications of the ACM*, vol. 49, no. 1, pp. 68-73, 2006.

[We99] WELLMAN, B. (ed.): *Networks in the global village: life in contemporary communities*.

Westview Press, Boulder, Colorado, 1999.

[We02] WEGNER, H.: *Analyse und objektorientierter Entwurf eines integrierten Portalsystems für das Wissensmanagement*. Dissertation, Software Systems Group, Technische Universität Hamburg-Harburg, Hamburg, 2002.

[We07] WEI, S.: *User-Centered Social Software – A Software Architecture Reconciling Personal Information Management and Object-Centered Social Software Platforms* (working title). Dissertation, Department of Informatics, Technische Universität München, Germany, 2007 (in preparation).

[WeBe97] WELLMAN, B. and BERKOWITZ, S. D. (eds.): *Social Structures: A Network Approach*. Jai Press, Greenwich, Connecticut, 1997.

[Wh06] WHITE, R.W., KULES, B., DRUCKER, S.M. and SCHRAEFEL, M.C.: *Supporting Exploratory Search*. *Communications of the ACM*, vol. 49, no. 4, pp. 36-39, 2006.

[Wi88] WILSON, T.D.: *On user studies and information needs*. *Journal of Librarianship*, vol. 37, no. 1**,** pp. 3-15, 1988.

[Wi94] WILSON, T.D.: *Information needs and uses: fifty years of progress*. In VICKERY, B. C. (ed.): *Fifty years of information progress: a Journal of Documentation Review*, pp. 15-51, Aslib, London, 1994.

[Wi05] WINER, D.: *RSS 2.0 Specification*. Source: http://blogs.law.harvard.edu/tech/rss. Retrieved January 2007.

[WKR02] WOLBER, D., KEPE, M. and RANITOVIC, I.: *Exposing Document Context in the Personal Web*. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI)*, pp. 151-158, ACM Press, New York, NY, USA,2001.

[XFN07] XFN – XHTML FRIENDS NETWORK *Specification*. Source: http://www.gmpg.org/xfn/. Retrieved January 2007.

[XiDo99] XIONG, R. and DONATH, J.: *PeopleGarden: Creating Data Portraits for Users*. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, pp.37-44, ACM Press, New York, NY, USA, 1999.

[Zh06] ZHENG, Y.: *Evaluation of the RIA Technology Macromedia Flex for User-Centered Social Software*. System development project, Department of Informatics, Technische Universität München, Germany, 2006.

Web Sites

[Amaz07] AMAZON, online store. Web site: http://www.amazon.com. Retrieved January 2007.

[Blog07] BLOGGER, weblog community. Web site: http://www.blogger.com. Retrieved January 2007.

[Boxn07] BOX.NET, file storage and sharing service. Web site: http://www.box.net. Retrieved January 2007.

[Cite07] CITEULIKE, scientific reference sharing service. Web site: http://www.citeulike.org. Retrieved January 2007.

[Conn07] CONNOTEA, scientific reference sharing service. Web site: http://www.connotea.org. Retrieved January 2007.

[Cope07] COPERNIC DESKTOP SEARCH. Web site: http://www.copernic.com/. Retrieved January 2007.

[Deli07] DEL.ICIO.US, social bookmarking service. Web site: http://del.icio.us. Retrieved January 2007.

[Desk07] DESKTOPTWO, online personal desktop. Web site: http://desktoptwo.com. Retrieved January 2007.

[Digg07] DIGG, news sharing platform. Web site: http://www.digg.com. Retrieved January 2007.

[Ebay07] EBAY, online auction platform. Web site: http://www.ebay.com. Retrieved January 2007.

[Ecli07] ECLIPSE RICH CLIENT PLATFORM, development platform for rich client applications. Web site: http://www.eclipse.org/home/categories/rcp.php. Retrieved January 2007.

[Flic07] FLICKR, photo sharing service. Web site: http://www.flickr.com. Retrieved January 2007.

[Floc07] FLOCK, social web browser. Web site: http://www.flock.com/. Retrieved January 2007.

[Fort07] 43THINGS, task sharing service. Web site: http://www.43things.com. Retrieved January 2007.

[Frie07] FRIENDSTER, social networking service. Web site: http://www.friendster.com. Retrieved January 2007.

[Furl07] FURL, social bookmarking service. Web site: http://www.furl.net. Retrieved January 2007.

[Goog07a] GOOGLE, web search engine. Web site: http://www.google.com. Retrieved January 2007.

[Goog07b] GOOGLE SUGGEST, web search engine. Web site: http://labs.google.com/suggest/. Retrieved January 2007.

[Goog07c] GOOGLE DESKTOP SEARCH. Web site: http://desktop.google.com/. Retrieved January 2007.

[Goog07d] GOOGLE READER, news aggregator service. Web site: www.google.com/reader. Retrieved January 2007.

[Goow07] GOOWY, online personal desktop. Web site: http://www.goowy.com. Retrieved January 2007.

[ICQ07] ICQ, instant messaging service. Web site: http://www.icq.com/. Retrieved January 2007.

[IE07] MICROSOFT INTERNET EXPLORER, web browser. Web site: http://www.microsoft.com/windows/ie/ie7/technology/default.mspx. Retrieved January 2007.

[Ito07] JOI ITO'S WEB, personal weblog. Web site: http://joi.ito.com/. Retrieved January 2007.

[Itun07] APPLE ITUNES, multimedia player. Web site: http://www.apple.com/itunes/overview/. Retrieved January 2007.

[Iwiw07] IWIW, social networking service. Web site: http://www.iwiw.org. Retrieved January 2007.

[LEO07] LEO, online dictionary. Web site: http://dict.leo.org. Retrieved January 2007.

[Link07] LINKEDIN, social networking service. Web site: http://www.linkedin.com. Retrieved January 2007.

[Link07a] LINKROLL, social bookmarking service. Web site: http://www.linkroll.com/. Retrieved January 2007.

[Live07] LIVEJOURNAL, weblog community. Web site: http://www.livejournal.com. Retrieved January 2007.

[Lotu07] IBM LOTUS ORGANIZER, personal information manager. Web site: http://www-142.ibm.com/software/sw-lotus/products/product2.nsf/wdocs/orgfact. Retrieved January 2007.

[Medi07] MEDIAWIKI, wiki software. Web site: http://www.mediawiki.org. Retrieved January 2007.

[Micr07] MICROFORMATS WIKI. Web site: http://www.microformats.org/wiki/Main_Page. Retrieved January 2007.

[Mova07] MOVABLE TYPE, weblog publishing software. Web site: http://www.movabletype.org/. Retrieved January 2007.

[Mozi07] MOZILLA FIREFOX, web browser. Web site: http://www.mozilla.com. Retrieved January 2007.

[MySp07] MYSPACE, social networking service. Web site: http://www.myspace.com. Retrieved January 2007.

[MySQ07] MYSQL, open source RDBMS. Web site: http://www.mysql.com. Retrieved January 2007.

[Netv07] NETVIBES, Web 2.0 desktop. Web site: http://www.netvibes.de. Retrieved January 2007.

[Omea07] JETBRAINS OMEA PRO, unified PIM tool. Web site: http://www.jetbrains.com/omea. Retrieved January 2007.

[OTN07] ORACLE TECHNOLOGY NETWORK discussion forums. Web site: http://forums.oracle.com/forums/categoryHome.jspa. Retrieved January 2007.

[Outl07] MICROSOFT OUTLOOK, personal information manager. Web site: http://office.microsoft.com/en-us/outlook/HA101656351033.aspx. Retrieved January 2007.

[Page07] PAGEFLAKES, Web 2.0 desktop. Web site: http://www.pageflakes.com/. Retrieved January 2007.

[Pers07] PERSONALBRAIN, mind mapping software. Web site: http://www.thebrain.com/. Retrieved January 2007.

[Phot07] PHOTOTAKERS, discussion forum. Web site: http://www.phototakers.com/forum/. Retrieved January 2007.

[Pica07] PICASA, image organizer. Web page: http://google.picasa.com/. Retrieved January 2007.

[Pics07] PICSEARCH, image search engine. Web site: http://www.picsearch.com/. Retrieved January 2007.

[Pubs07] PUBSUB, weblog search engine. Web site: http://www.pubsub.com. Retrieved January 2007.

[RSSB07] RSSBANDIT, feed reader. Web site: http://www.rssbandit.org/. Retrieved January 2007.

[RSSO07] RSSOWL, feed reader. Web site: http://www.rssowl.org/. Retrieved January 2007.

[Ryze07] RYZE, social networking service. Web site: http://www.ryze.com. Retrieved January 2007.

[Shop07] SHOPPING.COM, product comparison platform. http://www.shopping.com/. Retrieved January 2007.

[Simp07] SIMPY, social bookmarking service. Web site: http://www.simpy.com/. Retrieved January 2007.

[Skyp07] SKYPE, Internet telephony service. Web site: http://www.skype.com. Retrieved January 2007.

[Slid07] SLIDESHARE, slide sharing service. Web site: http://slideshare.net. Retrieved January 2007.

[Soci07] SOCIALIZER, unifying social bookmarking service (by Pierre Far). Web site: http://ekstreme.com/socializer/. Retrieved January 2007.

[Tech07] TECHNORATI, weblog search platform. Web site: http://www.technorati.com/. Retrieved January 2007.

[Tomc07] APACHE TOMCAT, open source application server. Web site: http://tomcat.apache.org/. Retrieved January 2007.

[Tril07] TRILLIAN, unifying instant messaging service. Web site: http://www.ceruleanstudios.com/. Retrieved January 2007.

[Twen07] 20SIX, weblog community platform. Web site: http://www.20six.co.uk. Retrieved January 2007.

[Visu07] VISUALMIND, mind mapping software. Web site: http://www.visual-mind.com/. Retrieved January 2007.

[Wiki07] WIKIPEDIA, wiki encyclopedia. Web site: http://www.wikipedia.org. Retrieved January 2007.

[Wiki07a] WIKIWIKIWEB, first wiki system.
Web site: http://www.c2.com/cgi/wiki?WikiWikiWeb. Retrieved January 2007.

[WLM07] WINDOWS LIVE MESSENGER, instant messaging service. Web site: http://get.live.com/messenger/overview. Retrieved January 2007.

[Word07] WORDPRESS, weblog software. Web site: http://wordpress.org/. Retrieved January 2007.

[Xang07] XANGA, weblog community. Web site: http://www.xanga.com. Retrieved January 2007.

[Xing07] XING (formerly OPENBC), social networking service. Web site: http://www.xing.com. Retrieved January 2007.

[Yaho07] YAHOO DESKTOP SEARCH. Web site: http://desktop.yahoo.com/. Retrieved January 2007.

[Yout07] YOUTUBE, video sharing platform. Web site: http://www.youtube.com. Retrieved January 2007.