TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Sicherheit
in der Informatik

Ubiquitous Personal Information Management

Jens Heider

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. Johann Schlichter
Prüfer der Dissertation:	1. Univ.-Prof. Dr. Claudia Eckert
	2. Univ.-Prof. Dr. Uwe Baumgarten

Die Dissertation wurde am 07.06.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 19.09.2011 angenommen.

Zusammenfassung

In vielen Arbeitsbereichen nimmt die Menge der persönlich verwalteten Informationen immer weiter zu. Dabei steigt nicht nur der benötigte Speicherplatz. Durch die ebenfalls zunehmende Vielfalt von Anwendungen, Speicherorten und die Nutzung von unterschiedlichen Zugangsgeräten in stationären und mobilen Arbeitsumgebungen wächst auch der Aufwand für einen nutzbringenden Einsatz der abgelegten Informationen. Dies gilt nicht nur für das Auffinden von Informationen, sondern auch für entferntes Abrufen, Verteilen, Bearbeiten und Speichern über Anwendungsgrenzen hinweg. Zudem besteht durch die isolierte Betrachtung von Daten unterschiedlicher Quellen ein ungenutztes Potential für die Informationsverwaltung, da ergänzende Daten anderer Quellen nur aufwändig manuell berücksichtigt werden können. Um dem resultierenden Produktivitätsverlust entgegenzuwirken und das bestehende Potential abgelegter Informationen effektiver und sicherer nutzen zu können, wird in dieser Dissertation die Vision der ubiquitären persönlichen Informationsverwaltung entwickelt, die allgemeine Lösungsarchitektur entworfen und als Proof-of-Concept für eine Anwendungsdomäne implementiert.

Das Ziel der vorgeschlagenen Lösungsarchitektur ist dabei, für alle verwalteten Daten eine globale Zugriffs- und Ortstransparenz zu erzeugen und dem Nutzer eine universelle persönliche Informationsplattform zur Verfügung zu stellen, die ergänzend zu den originären Anwendungen für quellenübergreifende Nutzungsszenarien zur Verfügung steht. Der grundlegende Ansatz beruht auf der Verschränkung neuer Konzepte der Informationsverwaltung, der Mobilität und der IT-Sicherheit: Mit dem Konzept der homogenen Darstellung gespeicherter Informationen, verknüpft aus beliebigen Datenquellen, wird dabei zunächst die Basis für die anwendungsunabhängige Nutzung geschaffen. Auf dieser deskriptiven Darstellung wird die Interaktion mittels mobiler und stationärer Endgeräte entworfen, die an die Anforderungen unterschiedlicher Nutzungsumgebungen angepasst sind. Die IT-Sicherheit als Schutz vor kompromittierenden Angriffen auf Systemkomponenten ist jedoch für einen globalen Informationszugriff nicht nur Voraussetzung; durch die Einbeziehung persönlicher Endgeräte und der globalen Informationsverknüpfung wird auch die Handhabung der Schutzmechanismen beim vertraulichen Umgang mit Informationen vereinfacht.

Als erster Beitrag zur Umsetzung der Vision wird das Konzept des persönlichen digitalen Wissens vorgestellt, mit dem eine Repräsentation der vom Nutzer gespeicherten und strukturierten Informationen erzeugt wird. Mit Hilfe generischer Regeln und Entwurfsmuster können heterogene Inhalte und Strukturen aus beliebigen Datenquellen erfasst und in eine universelle einheitliche Darstellung überführt werden. Die Vereinheitlichung ermöglicht dabei die autonome Verknüpfung von Informationen aus unterschiedlichen Quellen, um vormals getrennte Aspekte von Informationen zusammenzufügen. So entsteht eine digitale Wissensrepräsentation über die abgelegten

Informationen, deren Schnittmenge mit dem präsenten Wissen des Nutzers auch für den Zugriff auf Teile außerhalb der Schnittmenge eingesetzt werden kann.

Ein generisches Interaktionsmodell für Wissensrepräsentationen ist der zweite Beitrag dieser Dissertation. Der pfad-basierte Ansatz dient in dem vorgestellten Lösungskonzept als Grundlage für alle Nutzerinteraktionen. Über einen geschlossenen Interaktionszyklus wird sowohl die Navigation innerhalb der Repräsentanz ermöglicht als auch die Interaktion mit den repräsentierten Inhalten. Es entsteht somit nur geringer Lernaufwand, und die Umsetzung benötigt nur wenige Interaktionselemente. Zudem ermöglicht das Modell die dynamische Erstellung und Nutzung relevanter Einstiegspunkte zur effizienten Interaktion mit den Inhalten. Das Interaktionsmodell ist dabei unabhängig von der Nutzerschnittstelle und kann daher durch unterschiedliche visuelle Umsetzungen an die Erfordernisse von mobilen und stationären Umgebungen angepasst werden.

Mit der graphen-basierten Suche in Wissensrepäsentationen, als dritter Beitrag, wird ihre assoziative Struktur für das Auffinden von abgelegten Informationen eingesetzt. Der vorgestellte bidirektionale Breitensuche-Algorithmus liefert korrelierte Ergebnisse zu zwei und mehr Eingabeelementen aus der Repräsentation. Über ein Gewichteschema und definierte Operatoren kann der Nutzer für die Suche zudem den bevorzugten Relationstyp vorgeben und Ähnlichkeiten, Unklarheiten und Unterschiede zwischen den Eingabeelementen ausdrücken. Für Situationen, in denen eine Schlüsselwortsuche unzureichende Ergebnisse liefert, kann so assoziatives Wissen des Nutzers für die Suche nach Informationen eingesetzt werden.

Der vierte Beitrag widmet sich einem zusätzlichen Schutz gegen kompromittierende Angriffe auf server-basierte Informationsdienste. Ziel des Ansatzes ist es, zur Laufzeit die Bedrohungen durch sicherheitsrelevante Implementierungsfehler zu reduzieren. Der Schutz basiert auf der erweiterten Nutzbarmachung von Betriebssystemrestriktionen. Dazu stellt der Ansatz einen strukturierten Designprozess für sichere Serverdienste vor, der bei der Aufteilung monolithischer Architekturen in getrennte Komponenten unterstützt. Der beabsichtigte zusätzliche Schutz basiert dabei mit der Funktionsaufteilung auf dem Prinzip der geringst möglichen Privilegien und dem Einsatz von unabhängigen Komponenten, die die Operationen mit Assets schutzzielspezifisch kontrollieren können. Die Aufteilung von Dienstfunktionen in isolierte Komponenten dient der Reduktion von zu erteilenden Privilegien durch das Betriebssystem, was die Reichweite einer kompromittierten Komponente einschränkt. Den verbleibenden Manipulationsmöglichkeiten einer kompromittierten Komponente wird durch eine paarweise Kontrolle entgegengewirkt. So wird ein zusätzlicher Schutz ermöglicht, durch den Angreifer mindestens zwei der isolierten Komponenten kompromittieren müssen, um erfolgreich die definierten Schutzziele verletzen zu können. Die geschützte Serverarchitektur für das persönliche Informationsmanagement wurde nach diesem Ansatz umgesetzt.

*»What man knows not, is needed most by man,
And what man knows, for that no use has he.«*

Faust I, Outside the town gate
Johann Wolfgang von Goethe
translated by George Madison Priest

Contents

Chapter 1

Introduction

1.1 Motivation

Finding desired information in the multitude of fast growing sources of personally stored data is considered a key problem in many working environments. Although digital document management already drastically improves the information management by providing full-text search and retrieval via direct network access, the time spent for searching information still increases (e.g., more than nine hours per week [FDMC05], including about three hours per week of search in vain [FV06]). This waste of time is not only frustrating, it significantly reduces the efficiency of the available working power. An improvement of these tasks therefore can save money by supporting the users in information search and management tasks.

However, searching is commonly only the beginning in a process of working with information. Having found what was searched for, there is a method needed to retrieve, process, store and share it (cp. [TJB06]). Of course there are already established ways to accomplish these tasks, but they are bound to data-specific interfaces and require a user to switch between applications, resulting in a great loss of efficiency. In case of modern working environments there is also the need for frequent switches between working devices and locations. Thus, the more different data sources, devices and environments are involved, the less current technology provides sufficient support for an efficient working: Each data source access with a device has its own interface that has to be used to bridge between storage locations and tasks the user wants to perform.

Providing the required remote accessibility to users, information security aspects become also more important for enterprises, due to the increased interaction with data outside the protected corporate's network perimeter. Increasing the possibilities and efficiency for personal information management therefore also requires solutions on how to preserve the protection against the increased attack surface.

The following key observations therefore are being taken into account for this thesis:

- Various formats and locations of information contained in communication media, documents, notes and organizer entries poses a challenge to the users in their daily work with information.
- Content of data sources is kept and managed separately with specialized interfaces. Therefore, users can use the potential of information spread across multiple data sources only manually.
- For a given search task, users need to think of fitting keywords and are often required to switch between different user interfaces. In mobile scenarios this is rendered an even more complex task because of the diversity of remote interfaces.
- The more security measures with user interaction are introduced that are perceived as an obstacle for efficient working, the more users tend to lower their effort in properly dealing with them. This decreases the measures' effective strength.

It will be shown, that the potential of autonomously processing user-managed data together with today's network connectivity has the potential for superior solutions. Their aim is to address these observations by increasing the possibilities for users to instantly and securely work with their stored data [HB06, HS08, Hei04].

1.2 Vision

The vision that is carried with this thesis is the paradigm of so-called *Ubiquitous Personal Information Management*. It servers as a remote control for the universe of information organized by individuals, bridging the differences in content handling, storage location and access technology. With a single ubiquitous interface, users can efficiently solve tasks instantly at the time they occur — employing current travel time; accessing information unexpectedly needed at customer's premise and the like — rather having to postpone it until a fitting environment is available. The location of the actual data source, its interfaces and the way the user has created the data is rendered irrelevant for the interaction with this interface.

> Ubiquitous Personal Information Management Systems allow an individual to securely search, access, organize and distribute his personally structured and scattered digital content anywhere, anytime and with any device.

Such systems are intended to focus on supporting individuals by increasing the flexibility of managing their *personally* organized content, rather than providing access for a whole group to data that was structured by other individuals. Especially in focusing on that personal aspect, the envisaged system can use the context that the collective data of one user has been created in and its given structures, to support the user's recognition and reminiscence. Presenting the user's content in an network of interlinked descriptive content elements should enable a user to map parts of *his* recollection with the digital representation to reassemble missing parts. Via associative connections presented between these parts, the user can find content by thinking of the surrounding situation. So the system is intended to use the already existing data to reconstruct a digital image of the things that caused the storage and structuring of the data. Even if this image would be pitted, blurred and sometimes even incorrect due to small data sets and misinterpretations, it would still offer a beneficial starting point for exploring the data to extract desired information.

Also part of the vision is the consideration of security aspects during all information managing processes. However, security not only is a precondition for establishing such a far-reaching access technology. The envisaged character of the systems is also predestined to extend and simplify the capabilities a user experiences during distributing information to other individuals in a secure way. Using the instant availability of the system via personal devices increases the possibilities to seamlessly exchange cryptographic information for securely sharing content with individuals in local proximity. Thus, digital content can be exchanged in analogy with content handed-over personally: The system can guarantee that the digital content is only made available to the person with whom the exchange was initiated.

The vision combines therefore elements of information management with ubiquitous access and the consideration of applied security. These three technology pillars are intended to be seamlessly combined to complement and support each other. This is covered in the thesis with the topics: creating a representation of user-structured data, providing a generic applicable interaction concept with them, describing ways to securely solve information management tasks with the new capabilities and protecting the system against compromising attacks.

1.3 Objectives

This thesis presents research on the improvement of personal information management to cope with the increasing amount of data to be organized by individuals and the need to create efficient and secure accessibility in any situation. Since information management is already consuming efforts, the advocated solution is intended to improve the benefit of information management performed already without burdening the user to invest more effort or change his way of organizing things. The objective is to design and develop a framework for such a management platform to improve the support for users in their daily tasks with information by increasing the possibilities for interaction with stored content.

The overall objective of this thesis can be summarized as follows:

> This thesis develops the design of an architecture together with a prototype serving as a proof of concept for a generic information access platform that provides access and location transparency. It focuses on corporate users, their usage scenarios and their personally organized data sources by providing a unified and ubiquitously available interface. The design incorporates today's hardware and is completely independent of the content to manage. The platform adapts to user-given structures, provides secure access even outside the corporate network perimeter and strengthens the security for personal information exchange.

The particular objective to include already available hardware is driven by the idea to remove or at least lower obstacles that might would prevent a fast adoption of results. As the design is intended to provide an additional aid to users, the accomplished benefit should not be abolished by the requirement of expensive new or additional hardware. It should improve the possibilities for working with already managed information by utilizing unused capacities. Besides the demonstration of the overall interaction, the objective for the prototype is therefore also to elaborate the feasibility even on cheap low-end devices.

Removing the borders for information access is considered an objective as a direct consequence from considered key observations. The possibility for an instant access regardless of the personal location and the location of the information's storage is a common request: It enable users to settle a task just when it occurs, rather than having to wait for reaching a fitting environment while the necessary efforts have aggregated and other tasks have to be delayed. Considered this objective on its own, of course available tools addresses these situations already by providing transparency on network layer. However, the objective in this thesis is considered for a generic information interface, independent of application's data sources types, locations and structures. Thus, in this thesis an additional transparency on application layer is postulated to seamless work with information, rather than having to working with multiple applications to solve one information management task.

Personally organized data is the target of the platform. Compared to public available data or data organized by other individuals, personally organized data has the advantage of being linked to an incident, which caused its storage, being related to the individual. This relation between the data and the user should be used without requiring additional effort for the user. Thus, a supplemental possibility and methodology for accessing the data should be established without interfering with the current way the data is accessed or managed.

Finally, protecting the managed information against threats introduced with new access and distribution methods is a central objective for the envisaged design. Protection should be provided both: (i) enforced as non-optional measure to prevent users from circumventing it accidentally and (ii) comprehensible and easy, to induce an understanding of the gain in security. Thus, many different protection schemes have to work properly together for securely accessing scattered data sources from arbitrary devices and locations. This prerequisite of usable security in practice is demanded not only for communication protection, but also for the overall usage scenario. Besides the usability aspects this includes the consideration of possible attacks on all involved components. In consequence, these attacks may effect all layers of interfaces and also include those are targeted on compromising components by exploiting implementation flaws. Sophisticated protection measures against these attacks are considered necessary for an approach that creates a single point of access to the complete asset of personally stored information.

1.4 Contributions

The key contribution of this thesis is the design and the prototypical implementation of a novel information management platform that supports secure ubiquitous information access with a unified interaction model. It is based on the seamless combination of technologies for information management, mobility and security, with the goal to complement and support each other. The solution concept comprises four main contributions in the technological fields of knowledge representation, interaction, search and security, described in the following sections.

The developed prototype stimulates the envisaged information management platform as a proof of concept of the underlying ideas, the knowledge representation and the interaction model. It forms the basis to conduct user trials and to evaluate both, the paradigm of ubiquitous personal information management and its architectural realization to show the feasibility of an universal interface for information access and management.

1.4.1 Personal Knowledge Representation from arbitrary Data Sources

Considered for enterprises, data is stored and managed by an individual as the result of a working task. The first contribution of this thesis is the concept to create a homogeneous representation from this heterogeneous, user-structured data. The generic concept to autonomously unify and create interconnections between data is considered in this work the foundation for supporting new efficient ways for ubiquitous information access and management, such as the unified interaction model for personal information and search strategies via associative approaches, which are also presented in this thesis.

The representation is intended to rebuild a part of the knowledge that the user may recall about an incidence connected with the data. The more information can be interconnected in this representation, the more aspects that a user may be still aware of can be also found in the representation as reference to information the user is looking for. Therefore the concept interconnects related data from multiple data sources to enrich the representation with elements that represent different but related aspects, which are caused by the same working task.

Furthermore, the concept is independent from data sources structures. It unifies existing hierarchical, property and type relations and preserves user-given structures in the representation. Many of the extracted relations are introduced by the individual way the user has organized his data. It is the extraction and preserving of these individual structures that should help the user to navigate inside his data. Preserving relations in a user's individual organization during the process of unifying arbitrary data sources is why the concept is considered a novel help to users. But it is also the reason why the concept is only useful to that particular user for the most part. This is what is considered the *personal* aspect of the concept.

Unused potential of data that is kept separately in multiple data sources so far should unfold when the data is presented to the user in an interconnected view. However, the data remain untouched and is still available via its data source's original interfaces. The user is therefore presented an additional way to use his data via a new layer on top of his managed content. This layer provides a navigation across data source boundaries via the autonomously extracted structures.

The proposed concept of a personal knowledge representation does not need user interaction for its creation. Thus, the user does not have to invest effort other than is already spent for organizing his data sources. Only computing resources have to be invested for the additional access possibilities created from the already managed data sources. Besides the general concept, important parts of the contribution are therefore the proposed patterns and generic rules for the Topic Maps technology (initially presented by Heider et al. in [HB06]), which create the personal knowledge representation from arbitrary data sources with unified and preserved user-given structures.

1.4.2 Generic Interaction Model for Knowledge Representations

Regardless of the way a personal knowledge representation is created, it has to be presented the user in a fitting way to be of any use to him. The second contribution therefore describes a generic interaction model for knowledge representations. It uses a path-centric approach together with a closed interaction cycle to present an interface completely independent of data source properties and structures. It is intended to provide both: a variety of dynamically generated entry points to the representation and navigation between its elements. Fitting entry points are vital to improve the efficiency by reducing the distance to desired elements during manually browsing the representation. After entering the representation, comprehensible navigation options then help users to access the right information.

The generic interaction model is designed to be feasible for limited mobile devices as well as for resource rich desktop systems. Different visual implementations of the model were designed for mobile and stationary systems to evaluate the applicability.

With the proposed interaction process, users can interact with the representation regardless of the content's original interface and interaction capabilities. Users browse through the representation by navigating via relations to desired elements. Other possible interactions with these elements — such as viewing, editing or distributing — can be defined also inside the representation's data structures. Thus, the representation also contains the information about available operations on presented information types and how to process and interact with the original data sources for that operation. Therefore, new operations for interaction added to the representation can be used directly without adjusting the user interface. These properties make the interface universal for any operation on contained information elements.

1.4.3 Graph-based Search on Knowledge Representations

With the ability to navigate and browse inside the representation, the user already is enabled to manually search for information by (i) following associative paths between entry points and connected information pieces or (ii) by just performing a classic keyword search. With the third contribution of this thesis, this is extended by also utilizing the structure of the representation for automated associative searches.

Representations consists of hierarchies as well as property and type relations. These interconnections between information elements are interpreted for the introduced search method as a graph. The proposed *bidirectional Breadth-first Search algorithm* (published initially by Heider et al. in [HS08]) provides correlated results based on the user-given structure of his data. First fitting shortest paths are calculated between multiple selected information elements, which are then used with defined operations to identify intersecting result elements. Users only have to select related informations and are presented associative results, which can be further used for interaction. This search method extends the possibilities of users to find desired information contained in their managed data.

Input for the search is performed by using the same interaction model as described for navigation: By just marking two or more elements to which the user recalls a relation with the searched information, the graph-based search calculates result elements without having to burden the user with precise but complex query languages. Even in cases an exact hit is not contained, the result can be used as starting point for a manual search in the local proximity of the result elements.

Advanced search possibilities are offered by also taking the relations' type into account. As an option in the proposed approach, the search results can be influenced by applying weights to the relation types. With the introduced weight schema for hierarchies, property and type relations, the user can choose between predefined search modes. With these modes, the user decides which type of relation to the desired information is preferred. Furthermore, the approach introduces binary operations on relation vectors of result sets. These operations can be used by advanced search interfaces to let the user express more detailed searches. With these interfaces, users specify similarities, uncertainties and differences between selected elements and the searched information.

1.4.4 Generic Approach for Protection against compromising Attacks

Any system providing a far-reaching information access — such as the envisaged platform — is threatened by attacks trying to compromise components. Therefore, the security design proposes an additional line of defense to make attacks more difficult to succeed.

This work motivates a tightened usage of operating systems' restriction capabilities to prevent exploit code from violating protection goals. But, as long as services are designed as monolithic server architectures, one implementation flaw may endanger all involved data. This is due to the fact that monolithic services can not make use of fine grained operation system restrictions to prevent an exploit from manipulating the complete service. Therefore, an attacker can abuse any of the services' privileges to access arbitrary data. As a consequence, an important objective proposed in this thesis is the separation of critical functions into components isolated by the operating system.

The approach for creating separated components from a monolithic service — called *intra-service privilege borders* in this work — is intended to help create a protection that require an attacker to compromise and exploit at least *two* isolated service components to accomplish the mission of the attack successfully.

With this contribution, a novel approach for creating an architecture of isolated service components is proposed. It requires an attacker to defeat at least 2 of n independent components to succeed. Thus, the approach describes a process for splitting up the monolithic service into multiple components. A component in this approach is an isolated executable with its own protected memory, preventing unauthorized interfering with the component's functionality. The goal of the approach is to strengthen the protection against attacks that compromise and manipulate service functionality by exploiting implementation flaws. This is realized within the concept by (i) the separation of service functionality into components to reduce required privileges per component and (ii) by guarding relevant operations for protection goals with disjoint components to prevent exploiting a single point of attack.

This approach is then used to design the actual isolated service components for ubiquitous personal information management systems. The resulting service components, created by applying the approach, illustrate the feasibility for the application scenario to arrange the functionality in components that are isolated by the operating system in the intended way. Together with the performed security considerations, the proposed architecture can be used for the protection of actual ubiquitous personal information management system implementations.

1.5 Organization of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 discusses the related work and the involved technologies out of the thesis's application context. Since the solution concept is considered to consist of the three pillars *information management*, *mobility* and *security*, research on each of them is addressed to identify available and missing building blocks, but also to motivate finding solutions to oppositional requirements demanded by each of them. The organization of the chapter therefore reflects the conflicting areas that have to be solved for ubiquitous personal information management.

In Chapter 3 the generic approach for creating personal digital knowledge is introduced as part of the proposed solution concept. With this approach, information about personally organized data from arbitrary data sources is used to create a homogeneous digital representation. The chapter describes the design patterns for systems that use these representation to provide an associative information access across data sources. Additionally, the chapter introduces the interaction model for such representations and the novel way to use it for searches and information management.

The demand for security is addressed in Chapter 4. It describes the specific protection required for the envisaged systems to counteract risks that may be originated by the new way of interacting with personal information. Classic security measures of server-based services are extended with additional lines of defense against attacks aiming at compromising and manipulating service components. First the novel approach to create this sort of protection is described as a general approach and then it is applied to create the additional protection for personal information management systems.

Having described the conceptual aspects, Chapter 5 presents a proof of concept implementation called *MIDMAY* (Mobile Information Distribution, Management and Access for You!) to verify the concepts applicability. The chapter discusses the implementation specific aspects derived from the solution design presented in this work.

The proof of concept implementation was used to analyze the characteristics of the solution concepts applied to the information worker domain. In Chapter 6 these results are described and interpreted regarding expectations and assumptions made for the underlying concept. Properties of the implementation are presented and evaluated for productive environments, to decide in which fields further research is beneficial.

A conclusion of the achieved overall results and an outlook to future work is given in Chapter 7.

Chapter 2

Related Technologies

This chapter discusses related work and technologies out of the application context described in the introduction. Since the solution concept is considered to consist of the three technologies *information management* (Section 2.1), *mobility* (Section 2.2) and *security* (Section 2.3), research on each of them is addressed to identify available and missing building blocks, but also to motivate finding solutions to oppositional requirements demanded by each of them. The organization of this chapter therefore reflects the conflicting areas that have to be solved for ubiquitous personal information management. Section 2.4 then combines the mutual determining topics. It discusses the aspects of related work used for this thesis and the contribution of this work to the described problem space of *ubiquitous personal information management*.

2.1 Information Management

The management of information is a well recognized requirement, which was already pointed out in 1982 by John Naisbitt who reasoned that we are drowning in information but are starving for knowledge [Nai82]. Simply being able to access information or even owning them seems not to be the crucial point, rather than being equipped with a technology

a. to efficiently find parts of information out of the mass; and

b. to connect them to valuable knowledge.

Both emerging challenges necessary to be solved have caused research on technologies to manage information. Addressing aspect (a), strategies have evolved to provide an interface for the user to enter keywords contained in the searched information. Those approaches are either based on manually attaching keywords to content or utilize automated indexing algorithms, which retrieve the keywords directly from the content. Related technologies, aiming at this way of information management, predominantly consider the problem space from the perspective of classic search engines, which are further described in Section 2.1.1.

The second contribution to information management, visible as aspect (b), is the development of languages and structures to store, process, leverage and exchange information. By looking from that perspective, the *relations* between information are more emphasized to provide a cognitive style of querying. This is especially applicable for information one already got in contact with, hence all kinds of personal information collections.

> »An important feature of personal collections is that people are familiar with many details and characteristics about their information, as well as the contexts surrounding their use of it. When looking for personal information, you may remember the general topic of the item, who it was from, where you filed it, or roughly when you saw it.«
>
> Edward Cutrell et al. [CD06]

Related technologies that are aiming on leveraging these relations therefore are based on a knowledge representation, built from information the user wants to manage. The concept of knowledge representations and available standards and implementations is therefore described in Section 2.1.2.

2.1.1 Information Search

Electronic systems that offer personal storage of information and support an easy access to the stored information were envisaged already in 1945 when Vannevar Bush had proposed *Memex* (a portmanteau of "memory extender") [Bus45]. Bush described the device to be able to display books and films from a remote library by automatically following cross-references from one entry to another. Nowadays, systems providing search functionality have to retrieve information stored on personal computers.

> »Retrieval thus depends on indexing, i.e. on some means of indicating what documents are about. Indexing in turn requires an indexing language with a term vocabulary and a method for constructing request and document descriptions. Indexing is the base for retrieving documents that are relevant to the users' need. It has to be supported by a search apparatus that specifies conditions for a match between request and document descriptions, and modulation methods to alter these descriptions if no match is initially forthcoming.«
>
> David D. Lewis and Karen Sparck Jones [LJ96]

Looking at the emerging field of personal search tools for personal computers, one can see the approaches currently followed to assist the user in finding his stored information. In her PhD thesis, Na'ama Aharony [Aha07] shows that the typical user interface of available desktop search tools are described by the following characteristics.

1. defining a query as a small set of keywords,

2. retrieving a set of documents that are related in some way to the query keywords,
3. presenting the user with these documents ordered according to some (default) primary ranking scheme, and
4. allowing the user to re-order the document list according to secondary ranking schemes, which are generally attributes such as last-access date, filename, purported relevance of the documents' content to the query, etc.

Therefore these tools are based on letting the user choose keywords that are used together with ranking mechanisms to produce valuable search results. This ranking is considered as the most important factor for retrieving accurate results, so very much research is invested in developing better suggestion and learning algorithms. Unlike the proposed ranking mechanism by Aharony and its evaluation [CDZ07, Aha07], only little insights are published about the question of how to rank desktop search results without further assistance by the user. On the other hand, the technology of manually providing metadata for stored content to ease the later retrieval is discussed more widely. Such an approach, based on the Memex vision, is proposed by the *MyLifeBits* project. Its aim is to create a "lifetime store of everything" [GBL+02]. The main concept is based on manually adding content and annotations to a database. This way the interface can provide access to entries via different dimensions, such as keywords, time and thumbnails.

Regardless if the user manually adds keywords to his content, or if the content is indexed autonomously through software components, the approaches in this field are mainly based on indexes that do not consult the contents context or semantic. The main benefits of these search techniques are the proofed possibility for efficient and easy to use implementations. Its foundation is the concept of digital libraries, visible in the way content is added and handled like in the *UpLib* approach [JP03]. Any kind of data presentable as images are addressed in this system, making it versatile and applicable in many areas. Although its aim is to organize explicitly added content by providing efficient full-text search and visual presentation techniques, the inherent drawback for daily work content is the limitation of the search problem to the occurrence of keywords in the content without a consideration of their meaning, the relation to other content, and the underlying working process e.g. it was received in. So it is up to the user to remember the right relevant keywords that produces not to many results, hiding an important entry in the mass, but nevertheless do not exclude the searched information due to its required tight description.

A likewise new approach therefore is the usage of ontologies and agents to provide improved results, in case the user did not hit the exact keyword contained in the content to find. In the project called *haystack*[1] a personal portal, providing user

[1] See haystack project homepage at `http://groups.csail.mit.edu/haystack/`

interfaces and views, is constructed by agents with the help of user and system defined ontologies [AKS99, HKQ02]. This is a step towards databases of knowledge for personal information, however, the approach still treats the semantics of data sources separated, resulting in isolated representations for each source.

To overcome this shortcoming of utilizing desktop specific information, [NP05] proposes to include the email and browsing context into the search and ranking functionality. It combines context ontologies with the classic full text search method to use *activity-based metadata* and *authority transfer annotations*. Activity-based metadata describe context information relevant for finding and connecting the resources stored on the desktop, whereas authority transfer annotations help to rank retrieved resources in a personalized way. The prototype uses the open source project *Beagle*[2] as underlying desktop search infrastructure and extends its regular full-text indexing capabilities with contextual metadata and ranking.

Using the semantic in the information space of a local desktop has brought up the term *Semantic Desktop* for the concept of interpreting these information as *Semantic Web* [BLHL01] resources. Each resource is then identified by a Uniform Resource Identifier (URI) to access them and to express its relations to other resources [SBD05]. A Java-based implementation of this concept is the open source application framework *IRIS*[3], which also serves as the knowledge store and user interface for the CALO[4] research project. IRIS provides views and contextual navigation across an extensible suite of integrated office applications, based on an ontology knowledge store with the ability to allow the user to model relations between resources. A semantic desktop for mobile devices is proposed in the SeMoDesk project [WW08], which focuses on correlating personal information available on mobile devices. It adapts the framework for Personal Information Models (PIMO) [SvED07] to represent relations between information organized and created on smartphones.

Another example for the Semantic Desktop concept is based on local desktop web servers, creating the peers of the access to stored information. This open source framework is called *Gnowsis Semantic Desktop*[5]. It provides a unified interface to browse and search the local data arranged in an RDF representation of the original sources [Sau05]. The framework was funded by the research project EPOS[6] and is continued as reference implementation of parts of the *NEPOMUK*[7] project. Aperture[8] is an open source library for crawling and indexing information sources such as

[2] See http://beagle-project.org/

[3] IRIS: Integrate. Relate. Infer. Share.; see http://www.openiris.org/

[4] CALO: Cognitive Assistant that Learns and Organizes; see http://www.ai.sri.com/project/CALO/

[5] See http://gnowsis.opendfki.de/

[6] EPOS: Evolving Personal to Organizational Knowledge Spaces; see http://www3.dfki.uni-kl.de/epos/

[7] NEPOMUK: Networked Environment for Personalized, Ontology-based Management of Unified Knowledge; see http://nepomuk.semanticdesktop.org

[8] http://sourceforge.net/apps/trac/aperture/wiki/Overview

file systems, websites and mail boxes. All metadata is mapped to properties of the NEPOMUK Information Element Ontology[9] to allow uniform processing of the crawled and extracted information.

So all these approaches converge to *knowledge representations* due to the unified access to typical desktop information sources. However, the aim of these approaches is to use the relations between data to navigate between resources and to rank them, rather than building a user-centric homogeneous digital representation by merging arbitrary personal information sources — still preserving user-given structures and context — for ubiquitous browsing and searching through the knowledge, as it is one goal of this work.

2.1.2 Knowledge Representation

Knowledge Representations are described for various purposes with different approaches. Before considering actual technologies usable for the intended scenario of this thesis, first the roles of a knowledge representation should be distinguished. In [DSS93] Randall Davis et al. have identified five distinct roles a knowledge representation (KR) can play in general:

- A KR is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.
- It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends.
- It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences.
- It is a medium of human expression, i.e., a language in which we say things about the world.

Examples of artificial languages and models used primarily for this kind of knowledge representation include the *Knowledge Interchange Format* (KIF), the *Web Ontology*

[9] NIE: NEPOMUK Information Element Ontology;
see http://www.semanticdesktop.org/ontologies/nie/

Language (OWL) and the *Knowledge Machine* (KM), to name only some famous ones. Common to all of theses languages are their basic purposes. Besides the specification of facts and their relation to each other, a further major goal of these languages is to specify a syntax that can be interpreted by software to process the contained information, instead of just presenting them to humans. The ability of this *reasoning* should help the system better understanding the content but also the search request. However, a software following this approach requires a well designed foundation of interpretable facts for every domain it should be used in. The creation of such *ontologies* is still a major effort even for a single small specialized domain, caused by the complexity and ambiguity of human language. Additionally, although there are surely common processes and context relations applicable for all users, each individual user organizes his information — logical as well as physical — in a more or less different way, as shown by William Jones et al. in their publication about organizing personal information to get things done [JPGB05]. For this reason, individual ontologies have to be applied for the actual user, if the resulting system should adapt to the user's way of thinking, rather than requiring the user to adapt to the system's organizing method. This preparing effort for ontologies is a major drawback for a system that should be usable by average users in any domain. For this thesis the knowledge representation is therefore only considered as data storage to prevent the imperative of fitting ontologies. Even so the approach should not require an ontology, the approach should not exclude adding ontologies at any time to enhance the way the user can interact with his knowledge. So technologies have to be considered that allow for a flexible way to store and process relations between facts and which also provide the possibility for reasoning. Contemplable technologies were found with *RDF* and *Topic Maps* with their considerably researched concepts and the available tooling.

RDF

The *Resource Description Framework* (RDF) was developed to aid the vision of the *Semantic Web* [BLHL01] by providing a structuring and describing of resources available in the World Wide Web.

> »The Semantic Web is a Web of actionable information — information derived from data through a semantic theory for interpreting the symbols. The semantic theory provides an account of "meaning" in which the logical connection of terms establishes interoperability between systems.«
>
> Tim Berners-Lee et al. [SBLH06]

A central concept of this vision is therefore the process of reasoning. This process should provide answers to users questions based on content that contain interpretable semantic. This semantic is stored in the RDF metadata model. It is based upon the idea of making statements about resources in the form of subject-predicate-object expressions. In RDF terminology these statements are called triples, which represent a directed relation to the object. The subject denotes the resource, and the predicate denotes aspects of the resource and expresses a relationship between the subject and the object.

In Figure 2.1 an overview of the related RDF standards is given, arranged by their corresponding layer. It starts with the *syntax layer*, containing the RDF/XML syntax specification [W3C04c]. This W3C recommendation defines an XML syntax for RDF in terms of the W3C recommendations *Namespaces in XML*, *XML Information Set* and *XML Base*. The language N3, residing also in the syntax layer, is an alternative to RDF's XML syntax. Its design goal was to be compact and readable, but also is extended to allow greater expressiveness [BL06].

The *data layer* on top of the syntax layer defines the concepts of RDF, such as its graph data model, an URI-based vocabulary and its data types. Together this forms a formal semantic which provides a dependable basis for reasoning about the meaning of an RDF expression [W3C04d]. However, RDF itself provides no means for defining application-specific classes and properties. Instead, such classes and properties are described as an RDF vocabulary, using extensions to RDF provided by the *RDF Vocabulary Description Language 1.0: RDF Schema*. It describes properties in terms of the classes of resource to which they apply and consists of a collection of RDF resources that can be used to describe properties of other RDF resources in application-specific RDF vocabularies [W3C04b].

These basic capabilities for describing RDF vocabularies can be extended when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. Residing also on the *constraint layer*, the *Web Ontology Language (OWL)* is used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. The W3C recommendation OWL is a revision of the DAML+OIL web ontology language [W3C04a]. DAML+OIL was built from the original DAML ontology language DAML-ONT (October 2000) in an effort to combine many of the language components of OIL [W3C01].

Since RDF was developed to support the semantic web vision, its goal is to provide a data model for the description of resources delivered by the World Wide Web. In this data model, resources can be described by structured metadata. It serves the groundwork for automated mechanisms to reason about the importance of resources regarding search queries issued by humans.

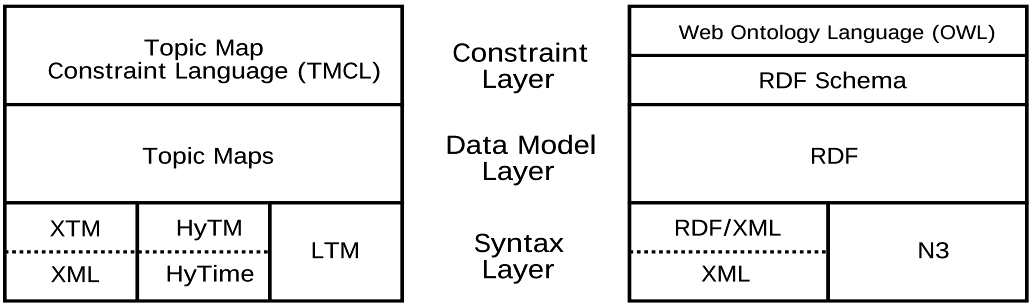


Figure 2.1: *Classification of Topic Maps and RDF Standards [Gar03]*

Topic Maps

In parallel to the development of RDF, a second initiative developed a technology for indexing information resources called *Topic Maps*. Topic Maps[10] facilitate the description of knowledge structures and their interconnection with existing information resources. Residing on top of these resources, a topic map builds a structured information layer which can be enriched with further details about them [Pep00]. A topic map therefore represents information using *topics*, which instances are the basic constructive elements. Relationships between them are established via *associations*, expressed with roles each member can play in a relationship. This network is connected to physical information using *occurrences* between topics and relevant information resources. All of these conceptual elements are represented by topic instances, forming the structure of a topic map.

Topic Maps as a format to describe and to exchange knowledge structures has become the ISO Standard 13250, which is based on SGML[11]. It uses the ISO 10744 HyTime standard for linking and addressing, and so the syntax is known as *HyTime Topic Maps* (HyTM). In October 2001 the *XML Topic Map Syntax* (XTM) has been added to the ISO Standard, to overcome some shortcomings and to adapt Topic Maps to the web [Int02]. Therefore, the current ISO Standard 13250 defines the interchange syntaxes XTM and HyTM, but does not explain how they relate to one another. Because of the well supported properties of XML and a rare use of HyTM in the field, this thesis focuses on XTM as syntax for the interchange of topic maps, whenever

[10] The phrase "topic maps" is used in two ways in ISO/IEC13250: as a capitalized proper noun, "Topic Maps", denoting the name of the standard; and as the plural of a common noun "topic map", which describes a specific topic map. Similarly, the non-capitalized phrase is plural, meaning more than one topic map.

[11] The Standard Generalized Markup Language is a metalanguage, designed to define markup languages for documents. See `http://www.w3.org/MarkUp/SGML/`

details of this layer have to be considered. At the time of this writing, XTM is specified in the current version 2.0 [Into6b].

The *Linear Topic Map notation* (LTM)[12] is a third alternative on the syntax layer. It is a simple textual format for topic maps and provides a convenient way to maintain topic maps. Additional to this syntax outside the ISO standard, there is also the AsTMa* language family[13], which offers languages for Topic Map authoring (AsTMa=), constraining (AsTMa!), manipulation (AsTMa+) and querying (AsTMa?). Its design goal was to provide an optimized authoring for humans by reducing the amount to type compared to XTM or LTM, and a non-orthogonal design that attempts to follow a *do-what-I-mean* instead of a *do-what-I-say* approach. As this thesis focuses on a software-based creation of knowledge representations, LTM and AsTMa= can be considered in this context as an alternative to convert the representation into a more human readable form for developing and research.

Moving further up to the constraint layer of the stack in Figure 2.1, the *Topic Map Constraint Language (TCML)* extends the standard with the possibility to constrain any aspect of the topic map data model, defined in [Into6a]. TCML consists of the parts *TMCL-Schema* and *TMCL-Rule*. The former can be used to describe constraints for topics and associations, whereas the latter is used to define rules for the validation of topic maps. Non standardized languages on the constraint layer are the constraint language AsTMa! and the *Ontopia Schema Language (OSL)*.

Compared to RDF, a further aspect to note about the constraint layer is that topic maps on their own are intended to be able to represent and to manage any kind of subjects and relationships between them and would be able to represent any kind of ontological framework without additional concepts. For example, the Topic Maps standard does not require to declare topics as types before using them as such. This builds the foundation of a single representation that claims to be able to express "anything about anything whatsoever". However, it should be noted that this point of view is also criticized.

> »[..] this very claim has attracted criticism from experts in semantics, formal logic and ontologies, and among the Semantic Web community, sometimes interpreting it as a puff of smoke hiding poor formal foundation ... although this capacity to 'say anything about anything' is also claimed by RDF. But it's true that the formal foundation and conceptual model of Topic Maps remain a subject of debate — not to say a bone of contention — inside the Topic Maps community itself.«
>
> Bernard Vatant [Vato4]

Following the criticism above, there should be a separation between the instances stored and their given semantics. On the other hand, exactly the flexibility and simplicity of an integrative approach are used by real-world application in recurrent patterns, linking association types to specific role types, occurrence types to topic

[12] See http://www.ontopia.net/download/ltm.html

[13] See http://astma.it.bond.edu.au/

types, role types in association types to topic classes, showing all the symptoms of the presence of some underlying, implicit if not explicit, ontology inside the topic map data model [Vat04]. This is the result of rules, applied indirect during creation of the topic maps. Especially if topic maps are created mainly automatic, the underling ontology is given indirectly with the code that produces and processes the map.

One way to populate topic maps in such an automated way is shown by the *TMHarvester*[14] project. It is a subproject of the topic map engine for Java (TM4J) which generates topic maps from various data sources, like a collection of documents, mp3 repositories and databases.

In contrast to the centralized approach of TMHarvester, in the *TMweb* vision, the concept of *Autonomous Topic Maps* (ATM) should provide a decentralized creation of implicit and explicit linked topic maps [Mai07]. This approach allows topic maps to carry their own construction manual which can be executed by generic interpreters in various application contexts. The execution of these construction manual produces new topic maps according to the modeling method defined by the manual. All of these created topic maps will be instances of the same modeling method, which goal is to address the problems of merging and querying maps of different authors.

2.2 Mobility

When defining mobility from a communications perspective, it commonly means being able to move freely while staying connected. Ubiquity on the other hand, resting in the title of this thesis, means in this context universal connectivity. Hence, considered in the context of information management, ubiquity means universal access to information: anywhere, anytime and with any device. So this again includes also mobile situations in which a user would really move while accessing his information, requiring mobility features. As these situations by far pose the most challenges for the applied approaches, related mobility technologies will be described in the next sections. Because of the need for network access technologies, first the aspects of *mobile communication* will be addressed in Section 2.2.1. Then Section 2.2.2 focuses on research related to *mobile information access* and its required server backends on application level.

2.2.1 Mobile Communication

Current mobile devices already provide an interesting range of technologies for mobile communication. Various bearers cover scenarios in *Wireless Personal Area Networks* (WPAN) (e.g. IrDA, Bluetooth, Ultra Wideband Broadcast (UWB)), *Wireless Local Area Networks* (WLAN), and *Wireless Wide Area Networks* (WWAN) that uses cellular

[14] See http://tm4j.org/tmnav/development/userdoc/en/tmharvest.html

networks (e.g. GSM, GPRS, UMTS, WiMAX, Mobitex, iDEN) to provide data services for different communication tasks. Especially WPAN and WLAN technologies are often used to communicate cost efficiently and are proposed to leverage the location context, offering location based services by pushing individualized content to mobile devices [KH06]. Because of the large coverage, WWAN are considered in general as "always available", providing convenient connectivity for mobile devices, even though unpredictable lost of connections have to be considered as well.

Combining coverage and cost efficiencies is addressed by the 3GPP standardisation effort called *Generic Access Network* (GAN)[15] which is an access layer technology allowing for seamless roaming and handover between WLAN and UMTS. Mobile devices can use these services with their application software to seamless communicate with servers on the Internet, regardless of source's and destination's location, current used bearer or changes between them caused by moving through different coverage situations. An IETF standard called *Mobile IP* also aims at providing a seamless handover by retaining IP addresses, providing network layer mobility.

Although the availability of seamless handover and roaming technology will make mobile communication more reliable and cost efficient in case of public available WLANs, the constraints for mobile computing can be still summarized in general like described in [MMBB02]:

- Frequent disconnections while fast moving
- High bandwidth variability
- Limited screens, resources and battery power
- Susceptible to damaging data due to theft and accidents.

Smartphones and PDAs address these constraints on protocol level with lightweight mobile standards like WAP [WAP99], and implementations of other common Internet protocols such as HTTP and IMAP, improved by compression and caching. Smartphones can combine the Internet communication with features based on cellular network technologies like SMS, MMS or WAP-PUSH [WAP01]. Besides its user-centric utilization, all of these can be also used as a communication channel to installed client software, which makes them interesting for configuration updates depending on location context (see e.g. [LE01]) and also for direct notifications (see e.g. [ATAdL06]), since these services provide asynchronous communication, including storage of incoming data in case the devices is not reachable currently. In contrast to the classic *pull* technologies initiated by the mobile client, especially the latter *push* standard provides an efficient information flow towards the mobile device regarding network consumption and reliability. In particular, the device does not have to leave communication channels open to receive notifications, which lowers the energy consumption,

[15] Also known as Unlicensed Mobile Access (UMA), the former name for the system.

hence improves standby time, and removes traffic overhead for periodically polling for changes. In [OKAD04] these properties of WAP-PUSH are used to efficiently deliver personalized information from XML sources to mobile clients, showing the opportunities for this approach.

2.2.2 Mobile Information Access

Access to information in *nomadic scenarios* — which can be considered a compromise between totally fixed and totally mobile systems [MCE04] — and *mobile scenarios*, forms the mobility aspects of ubiquitous information access in this thesis. So in this section, examples of technologies are described that address parts of functionalities relating to concepts to be applied for the unbound information access proposed in the introduction. First the general technologies for *single-server-based* approaches that focuses on information exchange — as opposed to *mobile middlewares* for managing services in distributed systems such as Jini, CORBA, et al. classified in [MCE04] — is discussed before special aspects of mobile access to Topic Maps and other applied technologies are described.

Information Exchange Technologies

Concepts that allow for bidirectional interaction have to be considered when a mobile device should be able to control personal information stored in remote locations. Actions can be initiated on both sides, either because of information access from the client or because updates occur on the remote side that require a notification on the mobile device. Therefore common technologies in wired systems like SOAP[16] would require SOAP server components for both communication partners. Especially for small mobile devices these larger components have to be tailored to memory and platform constraints. Additionally it is beneficially to improve the resource consuming HTTP communication overhead for mobile scenarios by using a WAP binding for SOAP. In [GB04] the improved SOAP latency and performance have been demonstrated for this binding, even though some tests were performed on simulators because of device restrictions. Other tests showed further possible improvements of the message size by utilizing compression and leveraging knowledge of the underlying WSDL[17] description [ADJ05]. Therefore currently only the footprint for a combined client and server makes SOAP somewhat resource consuming [PTB06], although there are clearly benefits for the mobile usage when considering the transaction concept and the fine-grained information flow into remote methods and back.

[16] SOAP originally stood for Simple Object Access Protocol. The acronym was dropped with version 1.2 of the standard as it was considered to be misleading. http://www.w3.org/2000/xp/Group/1/06/f2f-pminutes

[17] Web Services Description Language, see http://www.w3.org/TR/wsdl

Message Oriented Middlewares (MOM) [RMB01] already supports this essential bidirectionality with the publish/subscribe (pub/sub) concept, which is a well-established solution for asynchronous interaction. In this concept the senders characterize messages into classes to which subscribers can express interest in. The processes of publishing and subscribing is therefore decoupling the communication, which allows for a dynamic information exchange between frequently unavailable communication partners. In [PHJ02] the concept of pub/sub has been extended for the constrains of mobile environments. A proof of concept implementation showed the feasibility of the concept and provided insights for further research to improve content presentation and adaptation. In addition, the general advantages of pub/sub in mobile scenarios over the concepts of *Remote Procedure Calls* (RPC) were researched in [OPK+05] by analyzing performance and effectiveness of different approaches during failure situations of client and server nodes including disconnection of communication links. The results of this analysis indicate that pub/sub systems are more durable than client/server models in error prone mobile and ubiquitous environments. To leverage this durability, the proof-of-concept implementation called *HandHeld Message Service* (HHMS) addresses the information exchange with standard pub/sub APIs like provided by the *Java Message Service* (JMS). The HHMS architecture focuses with this approach on reliable message services for handheld devices. These services should seamlessly integrate mobile devices with wired domains and show options to extend the pub/sub paradigm to wireless environments [OF04].

The *Open Mobile Alliance Data Synchronization Specification* (formerly known as SyncML[18]) is a lightweight XML-based data synchronization standard. Originally designed to provide a vendor independent way to synchronize Personal Information Manager (PIM) data, such as notes, dates, and contacts, between components, the standard has become well supported and has been extended for other data as well [HTMP02]. In addition the specification is independent from the transport layer and therefore usable with a wide range of protocols such as HTTP, WSP[19] or OBEX[20]. With the *Java Specification Requests 230* (JSR 230: Data Sync API) there will be also a highlevel API for J2ME enabled mobile devices, if vendors will start to implement the optional package for their devices [Mah04].

Mobile Access to Topic Maps

In the particular case of information stored in Topic Maps, recent research has also addressed the demand of a remote access to Topic Maps. The *Topic Map Remote Access Protocol* (TMRAP) describes an abstract web service interface for remote access to Topic Maps. It can be used to access a Topic Maps repository to query fragments

[18] See `http://www.openmobilealliance.org/syncml/`

[19] Wireless Session Protocol; part of the Wireless Application Protocol (WAP)

[20] OBject EXchange; protocol commonly used for serial communication via Bluetooth and IrDA

or update a topic map, or to listen for updates to parts of a topic map [Gar05]. The concept of transport bindings allows the usage of the same backend with different technologies, such as HTTP and SOAP.

A similar approach was presented with the *RESTful Topic Maps Interaction Protocol* (TMIP). The *Representation State Transfer* (REST) architectural style generally provides a near-identity transformation for creating, reading, updating, and destroying online information resources [Fie00]. Accordingly, TMIP uses classic HTTP mechanisms, such as its methods and additional HTTP headers, to control the modalities of the data exchange between the parties. Additionally, an URL regime was defined to address maps (and other objects) on the server-side [Bar05].

Also distributed approaches for Topic Maps have been designed, tailored for propagating knowledge in adhoc or P2P networks. A system called *Shark* allows peers to share knowledge in Topic Maps form whereby the system controls which information is extracted from one knowledge base [SG02]. The system serializes the topic map fragment into XTM and uses KQML [FWWe93] for the interchange.

Applied Information Mobility

On application level, proof of concept systems already have been designed to provide mobile access to information in a generalized form. Focused on providing mobile services such as print, fax and view, the system called *Satchel* is designed especially to support distinctive features of mobile document work [LEF+00]. It is based on a web server that transforms the content to a proprietary format through a gateway [FPJea00]. Because of the used web technologies, the interaction primitive is based on hyperlinks.

Other projects focuses on providing user-centric personal digital libraries, which are also accessible via mobile devices. An example is *PBLIB* which provides each user with a general purpose personal digital library [ACGSLJ05]. Based on this approach, a mobile digital library is proposed, consisting of client-side applications, a data server and a mobile communication middleware [ACGSGS+05].

Enterprise services are addressed also by projects for mobile workers. *iMobile* is an enterprise mobile service platform that allows resource-limited mobile devices to communicate with each other and to securely access corporate contents and services. The architecture uses infolets for creating an abstract view of an "information space" using appropriate protocols (e.g., HTTP for the Web, JDBC for database access, X10 for home network control, and LDAP for directory services) to connect to a backend server [CHJ+03]. Other proprietary technologies to exchange information between mobile devices and intranet servers, like the BlackBerry Enterprise Solution developed by the Canadian-based company Research in Motion, offer also key features for mobile information access [Whe03]. The established generic data channel from mobile devices into the corporate network and its inherent push functionality can be also used by client applications to synchronize and access personal information in mobile scenarios.

2.3 Security Technologies

Besides the aspects described for information management and mobility, the scenario demands also for an integration of security technologies. An efficient usage is important to produce a balanced security level against known attack threats, without preventing user convenience. The following sections are structured by the related technologies for *Communication Security* (Section 2.3.1) and *Device Security* (Section 2.3.2).

2.3.1 Communication Security

In ubiquitous personal information management systems, the security of communication has to be addressed for the following two use cases:

a. the user's device has to communicate with the information source

b. dedicated individuals should be able to participate indirectly from the user's managed information though controlled distribution by the user

A classic and well researched security aspect for both use cases is therefore the requirement of a secure transmission of data. Besides the task to choose from the various approaches in cryptography to protect authenticity, integrity and confidentiality of data while in transit, also the gap between cryptographic theory and real-world cryptographic applications have to be bridged, like shown by Niels Ferguson and Bruce Schneider in their book "Practical Cryptography" [FS03], to prevent typical implementation flaws (e.g, missing countermeasures against side channel attacks, flaws in the key negotiation and insecure storage management).

The second general aspect for communication security is to complete the usage of publicly evaluated algorithms with rules for the design they are integrated in. On a very abstract level this is done in the engineering principles for information technology security [SHF01]. These 33 principles build a first foundation that should be considered in every security design (e.g., principle 7: "Implement layered security (Ensure no single point of vulnerability).", principle 11: "Minimize the system elements to be trusted." and principle 24: "Implement least privilege.").

From a more fundamental point of view, the main question in communication security always is: "How to share and manage secrets with each communication partner?". Especially in the research community, the main question is about the distribution of secret keys and its secure management, instead on trusting on the secrecy of algorithms. On the other hand, there are also fitting arguments that secrecy of security relevant algorithms or aspects also add protection, mostly argued by experts in the military and intelligence areas. In [Swi04] these two paradigms "there is no security through obscurity" and "loose lips sink ships"[21] are economically analyzed by comparing the evaluated cost and benefits for defenders and attackers. Following this model, the targeted solution of this work will more likely benefit from building the security on disclosure than it would harm the security through the knowledge an attacker learns from it.

In use case (a) the user's device and the communication counterpart has to share a secret to establish communication authenticity and confidentiality. This is commonly done by pairing the communication endpoints with a mutual exchange of certificates, like provided with the *Transport Layer Security* (TLS) standard. Since this has to be perform only between one server and a limited number of devices per user, and both communicating parties are well known to each other, no third party is required to act as a certificate authority in the application context of this work.

However, for the design of authenticity and confidentiality in *mobile ad-hoc networks* (MANETs) — the foundation of use case (b) — the key management schemes are essential. By definition, MANETs differentiate themselves from existing networks by the fact that they rely on no fixed infrastructure [ZH99]. Solutions for this type of networks would therefore suit well the idea of ad-hoc communication between users who want to share information. Existing approaches differ in the way how authenticity is provided. The survey on peer-to-peer key management for mobile ad-hoc network shows the following two classes of key management schemes [MDM07]. *Authority-based MANETs* use pre-established relationships created offline by an authority prior to network formation. The advantage for use case (b) would be, that no online access is required to authenticate a communication partner. In contrast the *fully self-organized MANETs* do not have any form of online or offline authority [CBH03]. A fundamental assumption therefore is that each communication node is its own authority domain, which makes these approaches typical for securing personal communication on application level, whereas the *authority-based* approach is used to secure networking mechanisms such as routing [CHB06].

In-between the authority-based and self-organized approaches, the key management can be also performed with *partially* or *fully distributed* authorities. These approaches split up the authorities secret key, using a *threshold cryptography scheme* [Sha79], so that some or all nodes share a part of the overall secret. If a new node can obtain some minimum (threshold) number of partial certificates, the node can combine them to produce one complete certificate [Tse07]. Other approaches combine the threshold scheme with *ID-based public-key management* [DA04]. This is a public-key cryptography in which the public key of all participants are generated from arbitrary strings that are connected to their identity, such as IP addresses for network nodes or

[21] A famous World War II slogan emphasizing the importance of keeping information secret. See `http://www.nh.gov/nhsl/ww2/loose.html` for posters telling vivid stories.

email-addresses for users. An important advantage of the ID-based cryptography is the possibility to send encrypted data without the need to ask the recipient for his public key, as shown in a complete scheme with *Weil pairing* on elliptic curves [BF03]. This could lead to a more efficient solution for end-to-end confidentiality for systems without immediate connectivity to a supporting server, especially if the cellular security infrastructure would be used to perform the still required bootstrapping of security associations as proposed in [AKG$^+$07]. However, since the corresponding private key is generated by a centralized (or hierarchical distributed) private key generator (PKG), *key escrow* is inherent in identity-based cryptography [BF03].

Without the above summarized approaches of a mutually trusted authority, the user needs an additional trustful communication channel to the recipient, to perform a device authentication[22]. Especially in the mobile scenario, common possibilities for such trustful channels are provided if the user shares the same physical context with the recipient. A very early approach leverages a physical contact between devices to prevent *man-in-the-middle attacks* during authentication [SA99]. Other contexts are used with radio link properties, such as the distance measuring proposed in [CCH06], to make sure that the communication partner is the authentic node inside a measured and displayed range. The presence inside the same physical context can be also proven by measuring environmental conditions. In [MG07] accelerator sensors inside both communication devices are used to detect similar movements. Shaking both devices in one hand then establishes an authenticated connection, based on the unique movement patterns. However, the components required by all these techniques are not commonly available in current consumer products.

Devices with Bluetooth interface could profit from a similar authentication technique for ad-hoc communication. The Bluetooth Special Interest Group provides in *Bluetooth Core Specification v2.1* [Blu07] an *Out-of-Band* man-in-the-middle protection either via numeric comparison and by employing the *Near Field Communication* (NFC) technology. Since at the time of this writing NFC-enabled devices are still rare, no evaluation has proven the proposed security characteristics in the field so far.

Other approaches bridge this gap by using existing technologies to provide a user-friendly way for comparing previously communicated data between sender and receiver. By recognizing manipulation during the unprotected transmission on application level, man-in-the-middle attacks can be ruled out during the essential establishment of a shared secret. With the help of modern camera-phones, [MPR05] proposes the use of two-dimensional barcodes to visualize a hash value of the received data. In this approach, the sending party uses the integrated camera to record the barcode from the display of the receiving phone. Then it compares the recognized barcode value with the hash of the transmitted data. With even less interaction, the

[22] Note that in this scenario the mutual authentication of the users' physical identities has to be performed in a classic way, such as with ID cards, in any case. A device authentication can only prove the identity of the communicating devices.

approach presented in [HS07] uses *IFS-fractals* to visualize hash values of the initial key negotiation. The receiving party then only has to select one of four displayed fractals that equals the one displayed on the sender's device. It is created from the received data and the other three images are created from random values. By comparing the images on sender's and receiver's device, manipulations during the transmission can be ruled out and additionally the users are forced to really compare the images, which prevents the user from always confirming the equality of images just by habit.

These and other examples are used in [May08] to create a taxonomy of spontaneous device authentication methods from a user's point of view. The first dimension distinguishes between direct and indirect *user sensibility*, the second between the two interaction scenarios *input* and *verify*. The combination of the dimensions creates four classes that define the information flow between sources and sinks for the considered out-of-band mediums *key pad input*, *display*, *camera*, *laser*, *infrared*, *audible*, *ultra-sound*, *motion* and *radio frequency*.

2.3.2 Device Security

With the ability to access any personally stored information, the security of the used device becomes vital for the overall security, because the device extends the access beyond the network security perimeter in which the data source is located. Therefore, the device has to be protected physically and logically to prevent loss of confidentiality in case of lost/stolen[23] devices and active or passive attacks aiming at abusing the trust relation between device and server. Besides the usage of personal firewalls, virus protection and secure communication, important protection mechanisms are therefore the encryption of stored content, automatically locking the device after a short time of inactivity and the possibility of the device to wipe all data, if the wrong password is entered more than the preset limit [Hal04].

Especially on resource-constrained devices like smartphones, the provided security of these additional protection tools on application level is often reduced because of computing power constrains, usability aspects and sometimes even work against each other[24].

Most users of resource-constrained devices therefore will have to trust the already integrated protections. In this segment the *Java 2 Micro-Edition Connected Limited Device Configuration* (J2ME CLDC) in combination with the *Mobile Information Device Profile*

[23] According to CSI/FBI Computer Crime and Security Survey 2006 [GLLR06], losses from laptop or mobile hardware theft increased from $19,562 in 2005 to $30,057 per respondent of the survey in 2006. 47% of the 616 respondents reported laptop or mobile hardware theft, which is the second highest reported attack type compared to 65% virus attacks, 42% insider abuse of net access and 32% unauthorized access to information.

[24] E.g, if a scanner tool cannot recognize a virus anymore because the scanner uses the raw access method that bypasses the decryption filter of the storage encryption tool. In this case the scanner only analyses encrypted data and cannot recognize fitting signatures of existing malware.

(MIDP) is the platform of choice when it comes to developing platform independent mobile applications. However, the study [DSTZ05] of its security aspects with the purpose of providing a security evaluation for this Java platform has confirmed that serious vulnerabilities already exist in the reference implementation of MIDP 2.0, which causes vulnerabilities on some phones, while other phones followed a restrictive approach in implementing the J2ME CLDC platform. Besides these vulnerabilities caused by implementation flaws, when considering the single point of access to be protected in the application context of this work, there are also security related problems caused by the open design of current smartphone operating systems such as Symbian, Windows Mobile, Android and those specific to device manufactures, such as Apple's iPhone and RIM's BlackBerry. Many of them have been designed with classic protections against threats in scenarios driven by mobile phone usage, but the devices now evolve to an access point of the corporate's intranet with all its values (see e.g., [HB11]). In case of physical attacks to the device, users cannot prevent their mobile devices from being manipulated through malicious software installations or the exchange of removable media. Extensions applied on application level cannot fully remedy a missing protection of system resources, because they can be removed by attackers with physical access to the device. One example for this threat can be shown with an application that uses password protected SSL client certificates. The application will either have to ask the user for the secret to unlock the certificates private key or will retrieve it from a persistent storage to ease operation. In the latter case an attacker with physical access to the device can obtain certificate and private key by directly pulling the secret from the persistent storage with no effort, because the storage in general and specifically the Java record store is not protected against attacks from outside (see [DSTZ06]). In the other case, the attacker can replace the original application with a crafted one that steals the unlock credentials from the user while it is entered. With this knowledge, an attacker can impersonate the victims identity in the application context. Even SIM-based (Subscriber Identity Module) installation- and configuration-locking and storage encryption cannot prevent an attacker who possesses a stolen device from exchanging the SIM to unlock the device for the manipulation. Even if the storage copied from the device is not readable in the first place, because of a SIM-based encryption, any required secret can be questioned from the user by returning a manipulated device, which transmits this secret to the attacker (know as "evil maid" attack).

This shows the twofold problem: How to protect the device from being manipulated and how can a user verify that he is really operating his own device instead of a replacement of it, before the user enters a credential that can be abused by an attacker.

Integrated trustworthy hardware components could offer a solution. This concept known from the *Trusted Platform Module* (TPM), proposed by *Trusted Computing*

Group[25], can provide security anchors by helping a third party to verify that the software has not been changed. With the functionality called *binding*, data can be encrypted using the TPM endorsement key, a unique RSA key burned into the chip during its production, or another trusted key descended from it. *Sealing* encrypts data similar to binding, but in addition specifies a state in which the TPM must be in order for the data to be decrypted [Tru07]. Such TPMs have even shown to be usable and useful in next-generation operating systems that utilize virtualization technologies. In [SE08] an efficient approach for using TC-technology in virtual environments was proposed by extending the TPM specification. These trusted computing approaches can increase the security compared to software-only mechanisms. Therefore also mobile devices should benefit from trusted computing concepts to provide a stronger protection in scenarios the attacker already possesses the user's mobile device.

A design for a *Mobile Trusted Module* (MTM), also proposed by *Trusted Computing Group*, could offer such a mobile trusted computing base. In [Die07] an integrated architecture for trusted computing for Java-enabled embedded devices was presented that also gives an answer to the question of how a mobile trusted-computing-enhanced system could be implemented with currently available technology. Furthermore, in [SKK08] an approach for the practical design and implementation of the MTM concept was proposed with a solution for the take-ownership of the device by the user and the migration of user credentials between devices. All these aspects indicate the possibilities for a secure foundation for mobile applications, but even if available, these security anchors have to be included into the design of the application, its implementation and the usage processes.

2.4 Conclusion

First of all, the described approaches have shown that no existing technology on their own addresses the needs and requirements for the given scenario. However, this does not imply the need for a complete new approach, since the presented concepts can be used as well engineered underlying technology. This section focuses on the question how these technologies can be combined for their usage in ubiquitous personal information management and what needs to be extended additionally.

2.4.1 Involving User Knowledge

The technologies that use and create context in personal data sources provide the functionality to enhance the search by building and leveraging a personal knowledge base. The Topic Maps standard do already provide a sound basis for this. But why using Topic Maps for a design that is supposed to autonomously generate knowledge

[25] An industry organization formed to define, develop and promote open standards for trusted computing and security technologies (see https://www.trustedcomputinggroup.org)

if RDF is designed for the semantic web? Its vision is leveraging the semantic of content to provide a machine-interpretable knowledge base. So RDF seems to fit quite well for an approach with machine-generated interpretable knowledge. The Topic Maps technology, on the other hand, better suits the human aspects regarding readability and it is completely self-defining, which eases extensibility and supports general applicability during runtime. Therefore, the main argument can be described with the difference in the design goal of this approach. If the system should be designed to autonomously reason about the stored personal knowledge, then RDF would be a good choice because its design is focused on representing interpretable information and much research is done in the area of reasoning. However, the approach presented in this thesis is based on the idea that it would be already a great help for users, if they are shown the relations between their stored information to ease the management and to improve the retrieval process. This thesis will therefore show how this management can be achieved without the need for an artificial intelligence preprocessing of the results. Such a processing would be helpful of course, but is deemed not realizable in a generic way in the foreseeable future because of a missing global ontology and the absence of a fitting solution to overcome the related *semantic gap*[26]. Therefore the main design goal in this work is to use realistic assumptions about the practicability without presuming or limiting the approach to a highly specialized application field, like it would be the case if requiring a specialized ontology for every application domain and user.

Because of the *personal* dimension, an approach seems feasible, if the user with his physical knowledge can be integrated together with the stored digital equivalent to mutually complete each another. This human component is the reason why Topic Maps technology is proposed in this thesis for building the knowledge representation. Building a networked layer between the user and his personal data, the Topic Maps identity concept is ideally suited to distinguish between electronic and other resources, making it possible to describe also relations to both in parallel. In addition, its merging concept defines an automatic process for combining multiple representation, which makes them modular and provides the possibility for a distributed and iterative generation of knowledge representations.

However, existing approaches in this research field currently do not address concepts for connecting information of heterogeneous data sources. But this is a key aspect for building a knowledge representation that should aid the user in finding content in these data sources. Connecting data would allow to leverage the mapping of user-known facts to portions of the data sources by providing navigational structures across any given source. A fitting user interface have to be supported by any device to provide a ubiquitous access. Currently no approach provides such an access to information for application-independent usage. Moreover, the way users

[26] The difference between two descriptions of the same object by different linguistic representations

interact with knowledge representations for information retrieval are characterized by browsing, query languages and full text search. This thesis will add a new way of navigation and searching by leveraging the user-given structures of data sources. The goal is to provide the possibility for search operations that are easy to use by everyone, yet powerful enough to be usable in any application field.

Therefore the Topic Maps technology is proposed as an underlying technology that will be extended in this work for an ubiquitous personal information management with proposals for

- the generic concept for building a homogeneous knowledge representation from heterogeneous data sources by still preserving existing structures
- the operational framework for autonomous topic map creation and merging
- the generic navigation interface concept for the created topic maps
- the usage of graph theory to provide a powerful search functionality on topic maps
- the distribution concept to dedicated recipients for personal context-enriched information

All of these extensions are user-centric; they involve the user's knowledge about facts which is applied to provide a personalized aid for the daily work with personal information.

2.4.2 From Pull- and Push- to a Remote Control-Technology

The second key aspect for this thesis is the possibility for the user to access stored information anytime anywhere. Many commercial products claim that this is already possible and even more research is going on to improve the networking technologies and its application to seamless deliver information to mobile users, just if they were using their stationary office- or home-PCs. And the mobile networks technologies have shown that the availability of various communication channels from and to mobile devices have increased the information flow for mobile workers, like with mobile email push approaches.

However, all shown approaches consider the mobile device just as a tiny PC that uses multiple clients for different communication tasks, just like the PC on the work desk. The constrains of limited resources are of course addressed, but the different ways users need to work with information in mobile scenarios is not properly addressed with the concept of a miniaturized PC with separated applications to be used for a single information management task. For example, the task of securely sending a remotely stored information to a contact involves the remote search in (at least) one data source, the transfer of this information to the currently operated device, the search for the recipient address and the composition of a transfer container to submit the information to the recipient. Any information in this scenario is either pulled by different client applications or pushed to a communication-specific application on the device through server initiative. The only possibility for an information transfer between these applications is the file storage of the device or an existing clipboard, which makes it a time consuming task that involves using multiple clients and a manual control of the final information transfer.

For the ubiquitous personal information management therefore the paradigm is changed from a PC usage concept to the concept of a universal information interface introduced by the remote control technology that emerges from the combination of the pull- and push-technology. The control technology uses existing network channels to keep the user informed about his personal information, independent of the application the user has used to create, store or communicate it. It removes the need to use various mobile application and services to retrieve stored personal information or to trigger an action on them. This is possible because information on its own is universal. This paradigm splits the processing intelligence between server — responsible for collecting and preprocessing of information into a global context — and the device, which uses its capabilities to implement the provided abstract navigation interface. In this paradigm, each node of the provided navigation interface is linked to a piece of information. Its default action is the transition to other linked nodes, but a node can also define other actions for the server. Actions can retrieve connected content, distribute content to dedicated users or even control the status of the heating at home by accessing its current status and its settings via the representing node.

This proposed change of the paradigm for mobile information access can resort to the full range of mobile communication channels. On application level the existing approaches are extended for the concept of the remote control technology by

- combining the interaction steps of navigation with the control concept
- proposing a concept for generic remote interaction with personal information

Additionally, like shown also in other approaches, the user needs to be provided access to their digital knowledge even in stationary scenarios with public devices. This gives the user the control over his personal knowledge also in nomadic scenarios. However, the mobility aspects also influence the concept for information processing and create security relevant questions for the security design. To address these connections the constraints caused by mobility have been split-up and were integrated into the two concept chapters: *Personal Digital Knowledge* and *Security Concept*.

2.4.3 Applicable Security

IT security is the third crucial aspect of ubiquitous personal information management because of the inherent concentration of desirable data from multiple sources in one access concept. From an attackers point of view this access concept is another target to evaluate if it offers a more economic access to protected data than it would be possible through conventional attack techniques.

Currently, the technology to protect mobile devices is mainly focused on transferring security concepts from the PC-world such as virus protection and encryption. The threats of lost/stolen devices can be addressed with these additional countermeasures applied on application level to supplement the protection provided by current operating systems. However, attacks using direct software manipulation with physical access to the devices are not covered. The required secure privilege management currently does not have a trustworthy anchor. Especially smartphones suffer therefore from missing protection of storage, which enables arbitrary applications to read and modify data. And the available access management for J2ME applications, which regulate access to phonebook and other components, also assumes that only the authorized user operates the device. Without a hardware-based trust anchor inside mobile devices, these shortcomings can only be counteracted by raising the effort for attackers to get physical access to the device.

Considering the access concept for ubiquitous personal information management in this way creates two claims:

a. The security concept should protect targets of additional functionality caused by the remote control technology at least as strong as the information would be protected in conventional scenarios.

b. The protection has to be strong enough compared to the economical equivalent for the attacker aiming at the concentrated data in a single attack spot.

The claim (a) is addressed in many approaches with classic security concepts for communication protocols and cryptography. If considered during the design phase, the risks for these targets can be evaluated according to the used cryptographic principles. This makes the protection comparable to the state-of-the-art protection for a conventional access. The consequence for this thesis is, that no new security technology has to be created. The important impact of this claim is the integration of the security in a way the user will accept and which he cannot bypass - neither by mistake nor willful. Especially in mobile use cases and for encrypted communication with other users this is always a trade-off between functionality and security, because of the limited device resources and the problem of heterogeneous applications, since it is in general not realistic to force other involved users to use the same technology.

Claim (b) on the other hand requires to think of a concept that is capable of protecting the access to stored information with multiple lines of defense. This again can be achieved by combining conventional security concepts with independent secrets, which they rely on, for every layer. So no single breach of a protection mechanism should provide direct access to the complete stored knowledge.

This thesis will address both claims with a proposal for a layered security design, which combines the protection of the conventional server-side with an applied approach for key management driven by mobile devices. Since these devices play the role of a remote control, they are used in the same fashion to organize the key exchange between users.

Chapter 3

Personal Digital Knowledge

Todays challenges with data arise from using it efficiently. Various locations and methods are used in parallel to store the huge amount of data, growing constantly with every new application of computers. Although there is much information contained in, personally stored data is currently only used separately with a specialized interface for each and every source, to perform tasks like searching and accessing of desired information. Therefore, the user is forced to use multiple applications and strategies to search for already known content. Even approaches providing search interfaces based on full-text search for multiple applications do not solve this sufficiently. They only shift the problem from finding the content to providing the right keyword contained in the content the user hopes to retrieve. Additionally these approaches do not utilize existing structures and semantics of stored data.

Tagging, the second major trend to deal with this challenge, is very time consuming. Manually adding metadata to every piece of data to rise the chance of finding it with given keywords in the future seems to work only when performed by larger communities. The effort is spread over many people and the outcome is beneficial for all of them, resulting in a motivating ratio between effort and benefit. However used for personal data, tagging has to be performed individually, and the effort additionally invested for those organizing tasks rises soon beyond the benefits for a single person, especially in cases the user does not recall the allocated keywords (cp. [BPH09, BVKKS08]).

Therefore, this chapter proposes the concept of aiding the user in his daily work with information beyond manual tagging and keyword search. But before introducing the solution design, Section 3.1 first explains necessary terms in the context of this thesis.

As a first result of this thesis, the new concept of leveraging redundancies in personal data sources is proposed in the following Section 3.2. A major advantage of this concept initially presented by Heider et al. in [HB06] is the intention that no user interaction is needed to create new connections between the already managed data of disjunct data sources. This ability to autonomously create interconnections between data is considered in this work a key concept for supporting new efficient ways for ubiquitous information access and management, like the unified access concept for personal information and search strategies via associative approaches also presented in this thesis.

For applying the redundancy concept, patterns and generic rules for the Topic Maps technology are proposed in the subsequent Section 3.3 to create an efficient knowledge representation from arbitrary data sources. These topic maps are build with a homogeneous representation concept that unifies hierarchical, type and property relations but preserves arbitrary user-given data structures. With this second contribution, the created topic maps offer a unified interaction with the stored information to the user in an intuitive way, yet leaving the user interface independent of the integrated data sources or the actual capabilities of the used client platform.

After having created a representation of the stored data, the proposed model of a closed interaction cycle then builds a generic information access on top of the homogeneous topic maps, now using their interlinked graph structure (Section 3.4). With this contribution, the interaction possibilities with these knowledge graphs are described for the intended simplification on working with information across different data sources. Vital entry points to the graphs, navigational operations and new strategies of finding items are inspected as use cases for the interaction model.

Abilities of these graphs are extended in the next contribution with a novel approach (published initially by Heider et al. in [HS08]) by applying graph theory for searching in topic maps (Section 3.5). By marking nodes representing associated information of the searched item, the proposed bidirectional *Breadth-first Search* provides correlated results based on the user-given structure of his data. This search method further increases the possibilities of users to find desired information contained in their managed data.

In addition to the inherent access to personal information and its integral retrieval approach, a further aspect of the aspired ubiquity is the proposal of a direct and seamless distribution concept of personal information to arbitrary recipients (Section 3.6), closing this chapter.

3.1 Term Explanation and Differentiation

Although used in computer and information science very commonly, the terms *data*, *information* and *knowledge* do have slightly different meanings in the involved research fields. Furthermore, these terms are needed in this thesis to differentiate between different interpretation layers in the description of the approach. The explanations given in the following therefore specify the terms for the presented work to avoid ambiguities.

3.1.1 Data and Metadata

A starting point for specifying the terms data, information and knowledge is bringing them into an hierarchical order regarding the level of interpretation by the user. From this point of view the term *data* is the most basic element in the used hierarchy, since no interpretation by the users is involved.

> **Data**: For this thesis, the term data is considered for the application in information management. In this context, data is defined by R. Hayes as »recorded symbols« [Hay92].

Furthermore, the term data should denote an element of arbitrary data sources. An example for such a data source is the email repository associated to a given email address.

> **Data source**: The term denotes a multiset *ds* of bit strings. *ds* is the repository from which data can be accessed.

In addition to the term data, the term *metadata* is important in this thesis for leveraging the characteristics of stored data. Metadata is commonly understood as data about data and characterizes for example the who, what, when, where and how related to a particular set of data.

> **Metadata**: »Metadata is structured data which describes the characteristics of a resource. [..] A metadata record consists of a number of pre-defined elements representing specific attributes of a resource, and each element can have one or more values.« Chris Taylor [Tay03]

The Dublin Core [WKLW98] is one well fitting example of a metadata standard[1] in in terms of Taylor. It is used to describe properties of data for common applications. In the same way, this thesis will focus on the use of metadata as a descriptor of data. This descriptor will provide finding aid for the user and the possibility to interconnect data by its properties. This is not limited to objective input such as the "properties" metadata generated when creating a file in a word processor or spreadsheet application. It also refers to subjective metadata created in the personal context such as assignment of keywords or summarization of content in an abstract (cp. [DHSW02]).

One common property defined by Dublin Core is the term *subject*. In this thesis it is *not* used as synonym for the term *topic*, since the latter is also used by the Topic Maps technology to describe the basic constructive elements of a topic map.

[1] For other common metadata standard it may be referred to [Mas09]

Subject: The metadata property *subject* describes with keywords or in a short phrase what the content of a resource is about. (cp. [WKLW98])

3.1.2 Information

From a technical perspective, an interpretation of the stored data can be characterized by the term *information*, like described by Roger Clark. This is the next level in the term hierarchy of interpretation.

> »*Information* is data that has value. Informational value depends upon context. Until it is placed in an appropriate context, data is not information, and once it ceases to be in that context it ceases to be information.«
>
> Roger Clark [Cla99]

This definition emphasizes the context that data is interpreted in. In the focus of this thesis the context would be defined by the creator and user of his personal data. In a further formalization, this view of an interpretation of data towards information is given by the *General Definition of Information (GDI)*:

> »Over the last three decades, several analyses in Information Science, in Information Systems Theory, Methodology, Analysis and Design, in Information (Systems) Management, in Database Design and in Decision Theory have adopted a General Definition of Information (GDI) in terms of *data* + *meaning*.«
>
> Blackwell Guide to the Philosophy of Computing and Information [Flo03]

The mentioned *meaning* clearly depends on the interpretation of a human being. This is applicable, since the condition of meaningfulness, as postulated in the following definition GDI.3, may only be judged by intellect:

> »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.«
>
> Blackwell Guide to the Philosophy of Computing and Information [Flo03]

GDI is a common definition of information applicable for many areas. However, like the definition by Clark, it describes a quite general view that does not involve the roles and interaction between software and their human users during the interpretation of data. As this thesis involves processes between software, human users and their data, the definition needs to be adapted to distinguish between the work of interpretation beneficially provided by the proposed software and the interpretation work the user has still to provide with his intellect. Therefore in this thesis, the mentioned context in GDI is always set by human users, aided by software.

> **Information**: The term *information* will be used in this thesis as the interpretation of data by humans together with the help of software.

An example for the help of software can be given by looking at data stored in a digital organizer. The raw data stored on the data carrier does not contain any information for a human in most cases. But the user can interpret the entry in the digital organizer with the help of the software to "I have an appointment at ten o'clock with Steve".

> **Stored information**: When using the phrase *stored information*, the interpretation and its envisioning of the underlying stored data by the user is meant. All stored information of a user is sometimes referred to as *personal space of information* [JT07].

The stored information can be classified by one or more types. Therefore the term *information type* is introduced.

> **Information type**: The term *information type* refers to the classification of the corresponding stored information. The type characterizes the stored information and builds a subordinate group to which all information of that type belong.

Note that the data "Steve" can be already interpreted as information if the user recognizes the string as a name, since this represents at least the information "There is a person named Steve". The *information type* in this example can be therefore expressed by defining a type "person", "name" or both of them.

Depending on the tasks a user is involved in, the stored information taken from his data sources form a domain. This *information domain* concept is used to build an information management that is independent from a specific digital content the user is working with, because no assumptions about the content and its interpretation have to be made. However, to be able to describe detailed benefits for tasks that a user has to perform with information, examples can be given by considering a specific information domain. A common example is the office worker domain[2], settled around the subject of organizing tasks and cooperating with other office workers.

> **Information domain**: The term *information domain* describes an application field of information types around a central subject, defined by the tasks of the user and his environment.

Outside of the interpretation of stored plain data, also the surrounding has to be considered, which commonly can be expressed by the term *context*. It is defined by Oxford Dictionary [Oxf98] as "the set of facts or circumstances that surround a situation or event". As proposed by Moschgath in her research about context-dependent

[2] This example is further described in Section 3.2.3

access control in ubiquitous computing [Mos02], the term *context information* is more specifically understood for ubiquitous computing as interpreted environmental data. For that reason her translated[3] definition will be also used in this work.

> **Context information**: The term *context information* will be understood in this work as any kind of environmental data, which are measurable, ascertainable or computable, and which may could have influence on the application. Environmental data in this case are object attributes as well as object relations.

3.1.3 Knowledge

So far the terms *data* and *information* have been explained for this thesis. The next interpretation level in the mentioned hierarchy is then given by the term *knowledge*. In the following a general definition is given that supports the idea of information accumulation by humans.

> »The term *knowledge* is often used to refer to a body of facts and principles accumulated by mankind in the course of time.«
>
> Roger Clark [Cla99]

An important foundation for this work is to extend this basic understanding of knowledge with the two dimensions *personal* and *digital*. In this case, the personal dimension specifies who is owning the knowledge. *Personal* knowledge is therefore an accumulation of facts and principles by a given individual. On the other hand, the term *digital* knowledge should expresses this accumulation in the binary world. It describes what is accumulated and how it is made persistent. Following the argumentation above, digital knowledge is considered the interconnection of information contained in stored data. Bringing together both dimensions personal and digital is the foundation of the following sections.

> **Personal digital knowledge**: The term *personal digital knowledge* is used to represent a body of data, its interconnections and the interpretation of these by its owner.

Considered by humans, the *personal digital knowledge* in its whole is more than the amount of all his stored data, because of the visibility of the relations between its data. "Steve is the author of a document stored on my notebook, which I received by email during a meeting in Berlin last year." is therefore individually interpreted data, hence information, retrieved from stored files, emails and organizers, which is accumulated

[3] The original German definition was given by Moschgath with: "Unter Kontextinformationen werden in dieser Arbeit die verschiedensten Umgebungswerte verstanden, welche messbar, ermittelbar oder berechenbar sind, und Einfluss auf die Anwendungen haben könnten. Umgebungswerte sind sowohl Objekt-Attribute wie auch Objekt-Beziehungen." [Mos02]

to personal digital knowledge. Adding new information to digital knowledge, like adding an upcoming appointment with Steve, increases the possibilities for further interpretation of it, since new interconnections to other information can be created.

But of course one has to distinguish between the knowledge of a user and its digital representation built through accumulation of stored information. The bigger the intersection between them, the more the user can intuitively use the digital knowledge, because he already knows interesting portions of it. But also parts of the digital knowledge not known to the user, and therefore outside the intersection, can be very interesting for him, as it reveals relations otherwise unknown to the user. This process is often referred to as *information mining*, which usually analyzes large databases in order to identify valid, useful, meaningful, unknown and unexpected relationships [KB03, Bor00]. However for this thesis, providing new information, automatically generated by interconnecting data, is considered as positive side effect. The main focus is lead on using the digital knowledge to find, to manage and to distribute stored data in an intuitive and efficient way.

3.2 Leveraging Redundancies in Data Sources

This chapter proposes the basic concept of leveraging redundancies in personal data sources. These redundancies will be used to create interconnections between data to provide further ways of access via connected data. Based on the subsequent assumptions for current information processing, the concept is explained and illustrated with examples of redundancies in different information domains.

3.2.1 Assumptions

When looking at content of data sources, there is one thing they have in common. They all specify types for their stored data to give it a meaning. Sometimes this is done explicitly, sometimes it is implicit inside the application using the data. Regardless of the actual way the data is typed, storing data without preserving this classification would be useless. Thus, when considering multiple data sources of one information domain, it seems to be inevitable that the inherent information types do at least overlap to a certain degree. In the domain described by common office work, examples for these types are subject, location, time, person or object. So the first basic assumption of this section is:

Assumption 3.1. *If two or more data sources belong to the same information domain they contain information types that share the same semantic.*

If Assumption 3.1 is fulfilled for an application of the concept, then there is also a high probability that also *equal data* is contained in multiple data sources. This equal data is used for different purposes in each data source, but this redundancy is the result of the data source's self-reliance. This is the reason for the second assumption:

Assumption 3.2. *Data sources of the same information domain contain recurrent data that describe the same subject.*

An example for the effect described in Assumption 3.2 is the occurrence of names of persons. The names are stored independently in multiple data sources to describe an authorship, to indicate interactions with the person or simply as a part of an address.

Applying the view created by Assumption 3.1 and 3.2 on multiple sources, one will observe much *redundancy* of data. The next section will present the basic concept for leveraging this effect to create personal digital knowledge.

3.2.2 Basic Concept

The goal of the basic concept is to interlink stored information to leverage the potential of redundancies for aiding the user in his daily work with information. By combining the body of data with its interlinking, a representation is created that can be interpreted by the owner of the data. This personal digital knowledge then contains the relations between the stored information across all included data sources. On this level, stored information is represented by a descriptive subject. On the one hand, the subjects are connected to other subjects and on the other hand the subjects are linked to the data they represent. This way the representation forms a network of interlinked subjects above the represented data.

Sharing equal subjects with multiple instances of data then lead to the desired effect of interconnecting various stored information across the borders of data sources. With these bridges, originally unrelated information of different application now can be used to connect the inherent data source structures.

The naming of the subjects is an important aspect for the representation, since the user should understand for which data they stand. Subjects in the knowledge representation are therefore created from metadata that describes the data. Depending on the information domain, this metadata is already included in sources or have to be extracted first (see Section 3.3.2). In any case no additional effort for the user should be involved, to help the user focusing on his primary tasks. Additionally, the content should remain untouched in its original location to remain maintainable for the user also with his familiar applications. Reasons for the importance to keep the original hierarchies are described by the study presented in [JPGB05]. It shows that folders often are used for more than just returning stored information in a structured way. A folder structure frequently resembles a "divide and conquer" problem decomposition with subfolders corresponding to major components of a project. Therefore folders are also representing a user's evolving understanding of a project and its components. Letting the user work with his familiar structures, no existing access to data or the managing possibilities of interfaces are changed or inhibited.

Linking together the metadata of different data sources offers the possibility to connect information maintained by multiple applications. This way, the benefit for searching is created by providing the user the opportunity to recall an information maintained by one application, to find an information stored by another application. The more sources are connected, the more the user can choose from starting points for his search to follow the paths in his autonomously created digital knowledge to the desired information.

Additional metadata can be also retrieved from other sources, leveraging context information automatically generated by preexisting knowledge or ambient sensor devices (see [BHH06]). However, this thesis will focus on information directly extracted from data sources more widely-used.

3.2.3 Examples for Information Domains

The interlinking of stored information can be visualized more precisely when considering a specific information domain. The first example is taken from the office worker domain, which builds a typical and well known information domain. Common data sources in office environments are email clients, digital organizers, text and spreadsheet documents and many forms of web-based applications. One example to show the general concept of merging different types of data sources should be the usage of a web-based application for managing travel expenses. Among others, this data source contains information about the user's location at certain dates, which are useful to enrich the digital knowledge. It even contains information about certain actions performed during a business trip, if these have caused a bill handed in for refunding. If the user recalls one of these actions, this information can be also used very efficiently for searching information beyond the aspects of the business trip, if connected with other sources of the information domain.

Table 3.1: *Office Worker Domain: Examples of reoccurring types in data sources, which are usable to interlink data*

data source	information types			
	person	subject	date	location
email	sender, recipient	of email	of submission	-
organizer	attendee	of entry	of event	of meeting place
document	author	title	of creation	-
travel exp.	employee	trip description	of event	destination

Table 3.1 shows the relation between common information types, their occurrence in data sources, and the interpretation for the using application in the table entries. Now consider the situation if data with a type contained in the same column of the

table are deemed equal in terms of the knowledge representation. The information about a person's *authorship* of a document can then be linked in the first table column automatically via the *attendee* entry of the organizer to the person's attendance at a meeting. This example creates a connection between a person and items related to this person. In the same way, *attendees* of meetings can be linked via the *sender* entry of an email to written or received emails. And senders of emails can be connected to information about their journeys, leveraging the travel expense data.

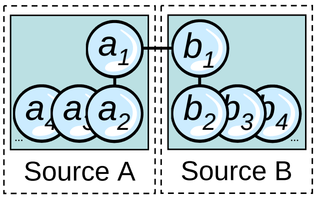


Figure 3.1: *Relations between Entries of Data Sources*

These examples are connections built only in the column *person*, but of course also the connections already contained inside each data source are interesting. The connections are represented in Table 3.1 as a set of entries aligned in one row. By interlinking entry a_1 stored in data source A with entry b_1 contained in a different source B, also new relations between other data of the sources are created (see Figure 3.1). If entry a_2 in A is related in any way to a_1, and B contains an entry b_2 related to b_1, then the user can follow the connection from a_2 to b_2 via the newly created relation between a_1 and b_1. Therefore, the transitive relations $a_2 \leftrightarrow a_1 \leftrightarrow b_1 \leftrightarrow b_2$ can be used for the new connections $a_2 \leftrightarrow b_2$, $a_2 \leftrightarrow b_1$ and $a_1 \leftrightarrow b_2$. An example for this effect is the implicit connection between a subject of an event in the organizer to attachments in emails sent by attendees of this event.

The Finance Information Domain

Similar to the example for the office worker domain, Table 3.2 presents reoccurring information types of data sources residing in the finance domain. The example considers *electronic bank statements* accessible online, *scanned paper bills* and *offerings* stored in a web-based back office system. In addition, *standing orders* of reoccurring payments and the overall *finance status*, calculated in data sources, provide valuable input data for personal digital knowledge. Likewise to the office worker example, the reoccurring types can be used to interlink various data sources. An offering then gets related to a bill, which in turn is related to the resulting entry of a bank statement after accomplishing the payment, to name only one example. When comparing Table 3.2 and Table 3.1, one can also imagine the effect that occurs if both information domains are combined into the same digital knowledge. In this case, email addresses residing in the digital organizer will have a relation to vendors, and appointments with customers are connected to offerings. The shown examples should not be used to limit the concept to just these types. Other information domains do have very different requirements, so Section 3.3 will present an universal concept that can be used for arbitrary data sources.

Table 3.2: *Examples of reoccurring types in data sources found in the Finance Domain*

data source	information types				
	account	value	person	subject	date
statement	payer, payee	amount	payer, payee	of transaction	of transaction
bill	payee	amount	payee	of purchase	of purchase
offering	acceptor	prize	vendor	goods	of delivery
stand. order	payee	amount	payee	purpose	next exec.
finance stat.	owner	status	owner	account name	of status

3.3 Homogeneous Knowledge Base using Topic Maps

So far, the possibilities of interlinking stored information by leveraging redundancies have been described in the previous section. Now this section considers the conceptional mechanisms for the creation of a homogeneous representation using Topic Maps. In Chapter 2 the technologies *Topic Maps* and *RDF* have been introduced and evaluated for their capabilities to build a homogeneous knowledge base. Topic Maps accordingly should help the user to deal with information whereas RDF was designed to enable software to interpret it. Furthermore the statements created with associations and roles in Topic Maps are easy to understand by humans. Adding new topics and association types is also a simple task performed directly inside a topic map and does not require modification of an external schema like in RDF. Combining structure, data and types in one uniform concept is therefore the major advantage of the Topic Maps paradigm for the intended use. Therefore the software autonomously creating these maps does not need to distinguish between them — and even more important — it does not need to understand the difference between data and its types. So types are represented just as another interlinked information. This way, on the representation level, their is no need for a differentiation since any subject can be stored as a *topic* instance.

> »Topic maps are organized around topics, which represent subjects. That is, in a topic map you find topics. Every topic you find represents a subject out in the real world that it is a symbol or stand-in for in the topic map. The definition of subject is essentially "anything whatsoever". What this means is that from the point of view of a topic map, objects are just a special kind of subject.«
>
> Lars Marius Garshol [Gar04]

Using this statement as foundation, the following subsections will create a homogeneous knowledge base with Topic Maps. First the representation of different data source structures that are acting as input for the knowledge base will be unified with patterns (see Section 3.3.1). Then in Section 3.3.2, universal rules will be designed to aid the creation of digital knowledge independent from certain information domains. Based on these rules the extractor concept is described in Section 3.3.3, whose task is to perform the physical instantiation of digital knowledge for the user.

3.3.1 Unifying the Structure with Patterns

Since the user should be aided in finding and working with his stored information, the created representation should provide a unified navigation and access, regardless which data sources have been included. Therefore a generic creation model is required which offers typing and structuring of digital knowledge contained even in multidimensionally structured data sources. The dimensions of the data sources in this case refer to the structure of the contained data.

The model has to be generic to be applicable for any available data source and should create a homogeneous structure. This is the precondition for a unified representation that does not require specific visualization functions for every information type extracted from arbitrary data sources.

The homogeneous structured representation also should offer the possibility to find related pieces of information, so generic patterns have to be defined for an interconnection between them, too. On the top level, this is achieved by applying the *Hierarchical Classification Pattern* and the *Faceted Classification Pattern* — proposed by Kal Ahmed (see [Ahm03]) — for constructing association, type and role topics. The next paragraphs show the application of these patterns for the creation of the unified representation. Using the patterns with the *Published Subject Indicators* (PSI) of the introduced topics across the represented data from different data sources includes the hierarchical structures in a unified way into the representation.

The Pattern for Multi-Hierarchies

Each data source to be used as input for digital knowledge may contain its own hierarchical structures. Considered on its own, these structures are obvious and have a definite semantic. Combining multiple sources and their structures in a homogeneous way, however, demands to unify the underlying semantic. This has to be done without changing the structures, as this would require artificial intelligence or the supplying aid of the user. Instead, the semantic of associations is represented via unified types, added during the inclusion process. On that layer one specialized component is needed, called extractors in the following, which is aware of the structures and its semantic for this data source. All extractors therefore can use the same unified types correctly because of their specialized understanding of the processed data source. Using equal types for associations across multiple data sources then unifies the semantic for the processing through other layers, but still preserves their differences of associations and therefore the recognizability for the user.

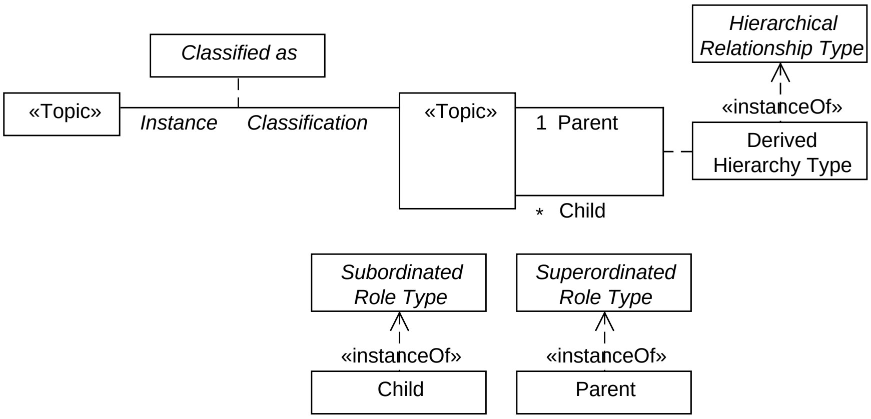


***Figure* 3.2:** *Hierarchical Classification Pattern in UML [HB06]*

Following this concept, the *Hierarchical Classification Pattern* (see Figure 3.2) is used to define multiple hierarchies with different associations. For example the two relations *parent-child* and *container-containee* are mapped to different associations but represent the same hierarchical semantic. This semantic is expressed by giving both associations the *Hierarchical Relationship Type*. The *Superordinate Role Type* and *Subordinate Role Type* are used for instances playing a hierarchical role in these associations to unify the semantic for hierarchical relations. A topic is then made an instance of a class inside the hierarchy by using the *Classified As* association. This way all hierarchical associations can be explicitly marked and therefore treated homogeneous as their semantic is preserved by the used types.

This is the first step for the unified representation because now multiple hierarchical relations are presented uniformly by the digital knowledge. The same is applicable for unifying semantics of property relations and type relations.

The Pattern for a Global Root

A further step towards a homogeneous knowledge representation is the creation of a designated entrance point to all networked information. This is done with the help of the *Faceted Classification Pattern* (see Figure 3.3) that provides a concept for an efficient finding of all hierarchies and their root elements. Every hierarchy is represented in this pattern through a topic of the type *Facet*. This topic is associated with the top level root element via the *Facet Has Root* association. The used association is of the type *Facet Has Hierarchy*. The result is a single root element for all represented hierarchies, building a hierarchy of multiple hierarchies. This way all included hierarchies can be visualized equally without any changes in the browsing component even if they are expanded dynamically.

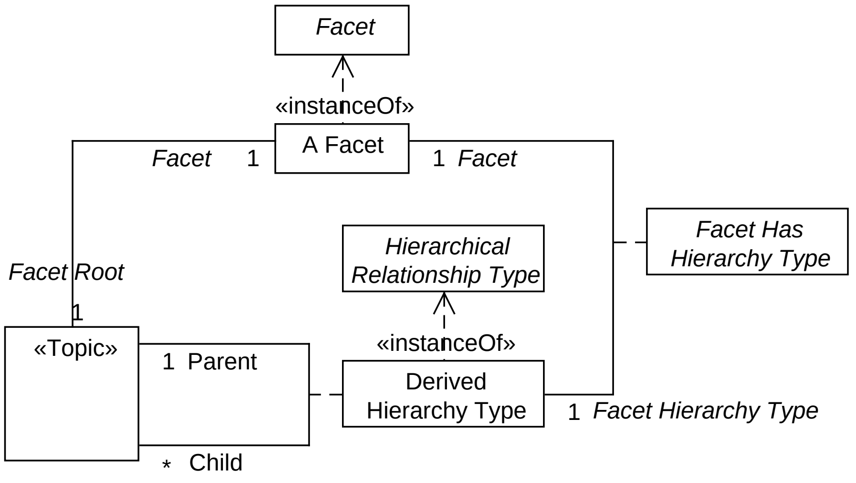


***Figure* 3.3:** *Faceted Classification Pattern in UML [HB06]*

3.3.2 Universal Rules for Digital Knowledge Creation

With the approach described in the previous section, all kinds of hierarchical relations can be represented in a generic model. The next step is to define rules for the representation of the information to be accessible. The primary goal is to build a representation that offers an easy way to access all stored information via appropriate topics. The representing topics should contain the information the user most likely will recall to access the linked data.

The key issue here is the re-usability of types across different data sources such as *person*, *subject*, *date*, *location* and any other information type occurring in at least two data sources, as shown in Section 3.2. These types are represented as dedicated topics, unifying the classification of other topics without limiting the represented information in the topic map. The more of these general topics are identified in the metadata of data source content and used as topic types, the more interconnections by associations can be established across the sources, and the easier the users can follow their own paths in mind in a natural way to find the desired piece of information they are looking for.

In the next sections universal rules for the creation of digital knowledge will be described by identifying relevant information and transferring it in a unified Topic Maps structure. The patterns described in the previous section are independent of the kind of source and their dimensionality and are applied to all data sources[4]. On top of these patterns further rules are defined to unify the data source representation.

[4] Thus, each indexed data source will be considered as a facet and all relations will be mapped with the described hierarchical classification pattern.

By combining these rules, the methods developed in this thesis are used when implementing instances of specific extractors for data sources (see Section 3.3.3).

First simple structures will be addressed before the proposed rules are expanded by advancing the number of dimensions contained in the considered data sources. This approach is used to show the applicability for the full range of possible sources independent of an information domain.

One-dimensional Sources

A common visualization for one-dimensional sources are lists. They can be transformed to the Topic Maps model, independent of their actual format in the data source, by only considering the one-dimensional character of the structure. The basic concept is to build a hierarchy that contains all leaves directly below its root element, as shown in Figure 3.4. The figure shows the mapping of a one-dimensional data source to a topic map.

Proposition 3.1. *One-dimensional sources are uniformly mapped with Algorithm 3.1 to the representation by*

- *creating a topic for every source's entry and associating it hierarchically with the facet root topic of the source (line 3 and 6–8).*
- *preserving the order of the entries using the Topic Map* sort name[5] *construct for every entry's topic. The item is created from an increasing counter value. This way the construct acts via it's PSI[6] as key for the sorting (line 4–5).*
- *creating topics for context properties collected from outside data sources and connecting the property topics with their main entries of the source via associations with a type that marks them as a property relation (line 9).*

In the case of one-dimensional sources, the properties connected to the main entries with Algorithm 3.2 are taken from sources providing context information. Such context sources contain much more information than could be integrated completely — by means of integrating it with an extractor and merging it into the representation, as described for other sources in Section 3.3.3 — so the context information is only added to the representation if fitting connections to other data exist to reduce the scattering of unconnected data.

If these one-dimensional sources are chosen by the user to be contained in his digital knowledge, it is presumed that all entries contained should be directly accessible inside the representation, such as in the case of a list of team members. This offers the interconnection with data retrieved from other sources. In case the user only wants

[5] Sort names are a particular form of variant name that will be sorted on the value property.

[6] Variant items whose scope property contains a topic item whose subject identifiers property contains the string "http://psi.topicmaps.org/iso13250/model/sort"

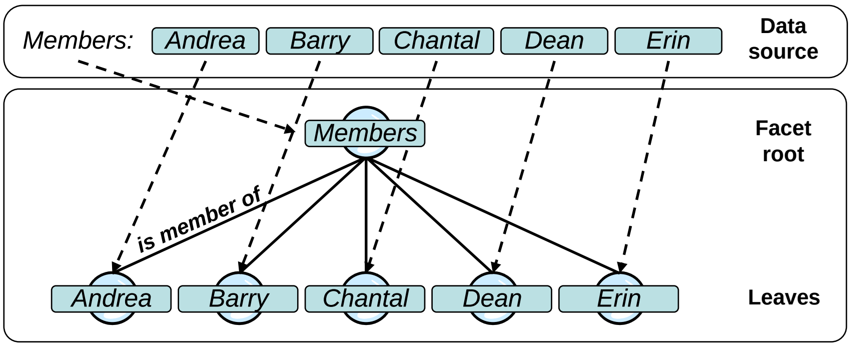


Figure 3.4: *Example for Mapping a one-dimensionally structured Source to a Topic Map*

Algorithm 3.1 Mapping one-dimensionally structured Sources to a Topic Map

```
Require: GetDimension($entry\_list$) = 1
Ensure: $entry\_list$ elements mapped to topics in $topicmap$
function Map_1d($entry\_list$, $t_{root}$, $topicmap$, $context$)
    for each $entry$ in $entry\_list$ do
        $t_{entry} \leftarrow$ create topic from $entry$
        $t_{variant} \leftarrow$ create variant name from Inc($counter$)
        $t_{entry} \leftarrow$ link $t_{variant}$ to $t_{entry}$ with $sort\_scope\_PSI$
        $a_{hierarchy} \leftarrow$ create association between $t_{entry}$ and $t_{root}$
        $topicmap \leftarrow$ add $t_{entry}$ and $a_{hierarchy}$ to $topicmap$
        $topicmap \leftarrow$ create hierarchical types and roles for association
        $topicmap \leftarrow$ AddProperties($context(entry)$), $t_{entry}$, $topicmap$)
    return $topicmap$
```

Algorithm 3.2 Associating Properties to a Topic

```
Require: IsInTopicmap($t_{entry}$, $topicmap$)
Ensure: topics associated as property to $t_{entry}$
function AddProperties($property\_list$, $t_{entry}$, $topicmap$)
    for each $property$ in $property\_list$ do
        if InformationTypeIsUseful($property$) then
            $t_{prop} \leftarrow$ create topic from $property$
            $a_{prop} \leftarrow$ create association between $t_{prop}$ and $t_{entry}$
            $topicmap \leftarrow$ add $t_{prop}$ and $a_{prop}$ to $topicmap$
            $topicmap \leftarrow$ create property types and roles for association
    return $topicmap$
```

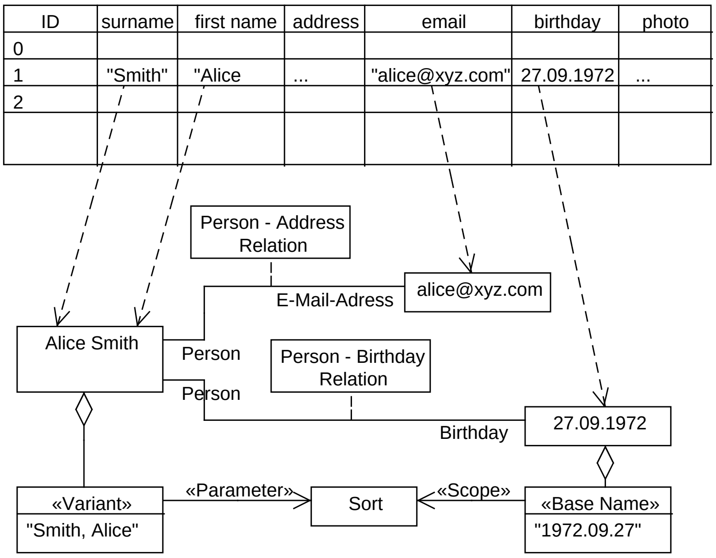


Figure 3.5: UML Mapping of a two-dimensionally structured Example Source to a Topic Map [Ber06]

to access the source as a single linked item, without adding topics of contained data into the representation, the system can be told so by indicating data source types that should be processed without connecting the content. This is useful if the represented list contains e.g. numeric measuring data that is not beneficial to be associated to other sources in terms of an information search.

Two-dimensional Sources

Two-dimensional sources are commonly visualized as a table and therefore will be also mapped to the knowledge representation as a hierarchical structure. For processing such tables, the proposed Algorithm 3.3 first creates a main subject for each row. The subject is taken from entries of one or more keyed columns, depending on how useful this is for the user to distinguish multiple topics. This combination of table entries during the mapping to the representation is shown as an example in Figure 3.5 by using first name and surname together as a single subject. The subject list for all rows is then processed in the same way described for one-dimensional sources with Algorithm 3.1 to create a hierarchically structured connection between the rows.

For each row the hierarchically connected entries may be additionally associated to entries of other relevant information types contained in the same row. In the same way as shown for one-dimensional sources, the entries of each row are therefore passed as properties to the *AddProperty()* function shown in Algorithm 3.2. It associates

Algorithm 3.3 Mapping two-dimensionally structured Sources to a Topic Map

```
Require: GetDimension($table$) = 2
Ensure: $table$ elements mapped to topics in $topicmap$
function Map_2d($table$, $t_{root}$, $topicmap$, $context$)
    $entry\_list$ ← collect all entries from the keyed column in $table$
    $topicmap$ ← Map_1d(entry_list, $t_{root}$, $topicmap$, $context$)
    for each $row$ in $table$ do
        $t_{key}$ ← use keyed topic from $topicmap$                ▷ created in Map_1d
        $row$ ← remove keyed element from $row$
        $topicmap$ ← AddProperties($row$, $t_{key}$, $topicmap$)
    return $topicmap$
```

properties to the row's main topic if the property is judged useful. This in turn depends on the definition and creation of information types most important in the information domain the knowledge should be used in. General rules for this decision are discussed subsequently.

Proposition 3.2. *As a general rule for the creation of topics, any information type should be taken from a two-dimensionally structured source that*

a. is found in more than one data source used,

b. will offer memorable attributes,

c. or provides additional context information.

Any other information does not have to be included in the topic map because in this case the data is already accessible via the reference contained in the metadata of the considered source [HB06].

Principle (a) in Proposition 3.2 targets at the created potential when leveraging interconnections between data sources. Merging extracted equal topics from different sources automatically collects all associations of those topics to a single topic instance. This topic then holds the relations to others, now referring to data contained in multiple data sources. This way relations across borders of data sources are created.

The second principle (b) aims at facts the user might recall when looking for desired information. Thus, facts about the referenced information are taken from its metadata or the content itself, to create topics with a property association to the topic that represents the stored content. Using these included topics, the user can access the referenced data in an efficient way because of the increased connectivity for potential search alternatives. A special application of principle (b) is the consideration of metadata that is commonly used to divide instances into classes. Such metadata should be used in the representation to visualize the membership of referenced data to a certain class. These collections, which are created by defining topics representing their classes' subjects, are hierarchically associated to the instances' topics. A collection therefore can then be used as a navigational aid to list all instances. At the same time collections are useful as general placeholders in search queries where they represent a collective property in contrast to a selected instance that owns this property. Using the generic concept of collections instead of defining new individual topic types for every property has the advantage of unifying access and its representation. It underlines the character of building groups of equal instances, whereas using individual named type topics focuses on distinguishing referenced information by their property. Implementors of specific extractors (see Section 3.3.3) are familiar with the metadata contained in the addressed data sources and therefore can decide which representation form to choose, based on the intention of metadata fields.

In addition, principle (c) proposes to classify information to enrich the representation. Including additional context information to the personal digital knowledge, derived from other data sources than the actual considered one, allows the system to facilitate the user in finding related information by mapping context that the user might recall. Beside stored context, this also refers to context information generated by components involved in user actions, such as context provided by the mobile client and its observing capabilities. This aspect will be discussed in depth in Section 3.6.4.

A special consideration is required if tables do not contain only *one-to-one* relations. In the case of *one-to-many* relations, also the related tables have to be inspected for information, jugged against the same rules of Proposition 3.2, and associated to the keyed topic if appropriate. However, associations made in this process sometimes can not be given a type for the representation, if their semantic is not specified in the data source and their meanings are only known to the application that is using the source. In this case, a generic *is related to* association should be used to preserve the relation and to prevent innumerable useless different associations. The same is true for *many-to-many* relations, which may also be contained in some data sources.

Multi-dimensional Sources

Of course one will encounter also multi-dimensional sources such as in the field of data warehousing. These sources can be addressed by first projecting them to a two-dimensional view, like it is performed for storing such data in relational databases. An example for a corresponding model is given in [AGS97] which illustrates the mapping of multi-dimensional structures on top of relational database systems. Those mapped two-dimensional structures than can be handled as described in the previous section.

A second approach is the projection of the multi-dimensional source to a hierarchy, which is the precondition for the proposed Algorithm 3.4. This projected hierarchy is mapped to the representation with the *Map_1d()* function. Then the top level

Algorithm 3.4 Mapping n-dimensionally structured Sources to a Topic Map

```
Require: GetDimension($table$) $> 2$
Ensure: $table$ elements mapped to topics in $topicmap$
function Map_nd($table$, $t_{root}$, $topicmap$, $context$)
    $entry\_list$ ← collect all entries from the keyed column in $table$
    $topicmap$ ← Map_1d(entry_list, $t_{root}$, $topicmap$, $context$)
    for each $row$ in $table$ do
        $t_{key}$ ← use keyed topic from $topicmap$                ▷ created in Map_1d
        $row$ ← remove keyed element from $row$
        $topicmap$ ← AddProperties($row$, $t_{key}$, $topicmap$)
                                   ▷ now all entries of $row$ have topics in $topicmap$
        for each $entry$ in $row$ do
            $t_{entry}$ ← use entry's topic from $topicmap$
            if GetDimension($entry$) $> 2$ then
                $topicmap$ ← Map_nd($entry$, $t_{entry}$, $topicmap$, $context$)
            else if GetDimension($entry$) $= 2$ then
                $topicmap$ ← Map_2d($entry$, $t_{entry}$, $topicmap$, $context$)
            else
                $topicmap$ ← Map_1d($entry$, $t_{entry}$, $topicmap$, $context$)
    return $topicmap$
```

table created during the projection is processed row by row. For each row, column entries are associated to the hierarchical entry via the *AddProperties()* function to preserve useful property relations. Depending on the dimensionality of each entry, the *Map_nd()* function is called recursively. For entry dimensions of two and one, the functions *Map_2d()* and *Map_1d()* are called, respectively.

For the described purposes of finding connecting information, the challenge in projecting multi-dimensional data to a single hierarchy is choosing the dimension that offers the most valuable information for the user, qualified by Proposition 3.2. Whereas simple structured data sources often benefits from the modeling of properties and metadata through additional views onto the information objects, this is not the case for multi-dimensional structures. By choosing one dimension for the projection, loss of possibilities for queries to this data have to be taken, providing the advantage of interconnection with other sources and a fast access. Otherwise the complete data source has to be represented inside the topic map. This would contrast the idea of a linked access to stored data and would result in a disproportionally decrease of overall performance, caused by the huge amount of data to manage. Thus, the complete representation of all information stored in multi-dimensional sources is only reasonable for data that provides important connectivity to other data sources, or whose data is important in its whole for the considered information domain, making a direct access with all its relations indispensable for an improved work with information.

3.3.3 Extractor Concept

The homogeneous representation is intended to be built without user interaction by using components that apply the rules proposed in the preceding section. In this section the need for the flexibility and extensibility of such components is taken into account by proposing a framework that manages the separated components and aggregates their topic maps into a global representation. These components are called *extractors* in this thesis, as they are actively *extracting* information to build the knowledge representation. They are used via the framework's interfaces for collecting information from configurable data sources.

The architecture of extractors and the interaction with the framework will be described in the next sections on a conceptual level. The architecture supports a convenient integration of arbitrary extractors that are specialized on their data sources intended to access. Common functionalities, like the traversing of tree structures, the work with Topic Maps objects, and the configuration and handling of extractors are addressed in the concept by a basic extractor, which can be efficiently extended for other sources.

In a further step, the application of elements and mechanisms provided by the Topic Maps standard are proposed for supporting the extractor concept. A suitable naming schema for the application context is described this way to unify the presentation of stored information to the user and to provide unified identification of topics for the automated processing. This is a foundation for the application of the Topic Maps merging paradigm. It replaces multiple instances of equal topics with a single one that aggregates all properties and associations of the former topics. A staged process of this merging concept is proposed to efficiently combine the independent topic maps of each extractor into a single consistent representation.

Extractor Architecture

Every extractor needs work to independent from other system components to perform his task: the collection of relevant information for the global knowledge representation. Any data deemed useful is stored as a topic inside an individual topic map maintained by the extractor instance. Besides the topics directly referring to the physical storage locations, also topics are created that describe properties of the indexed data. Because of the specialized character of each extractor, the semantic of the data source is known and can be reproduced by choosing appropriate *Published Subject Indicator* (PSI) and *Published Subject Identifier* (PSID) for the topic types. Examples for this approach are presented in the implementation Section 5.2.

Furthermore each extractor component is also capable of retrieving an entry of its inspected data source, if presented the associated topic. The retrieval component uses the *Subject Identifier* of the topic to pinpoint the corresponding piece of information inside a certain data source instance and returns its content for a further processing inside the framework. This provides the functionality to directly use referenced data as described in Section 3.6 for distributing content to external recipients.

The possibility to also include knowledge from other software components is addressed by a second design element tightly integrated into the architecture. With the creation of this concept, which is called *input maps* in this thesis, the inclusion of information independent of stored data is added. The data contained in input maps is also maintained with the Topic Maps engine, therefore the name should be understood as an identifier for the source it represents, like in the terms *extractor map* and *merged map*, instead of a new technology. This concept is useful in case of input generated by the user on the fly, for example when providing additional relations of topics during navigation. Additionally all kind of externally generated information and predefined knowledge maps, which provides an ontology functionality for a certain information domain, can be included this way.

Naming Schema of Topics

For the extractors the following mechanisms in the Topic Maps standard are proposed for usage in the presented application field. All topics are created uniquely in this schema by the extractor instance responsible for the data source type.

Proposition 3.3. *These principles regarding topic identification have to be applied by all extractors:*

a. *No special* scopes *are defined, so by default all topic properties are inside the* Unconstrained Scope *[PG02].*

b. *Every topic representing a stored entry inside a data source is given a* Subject Identifier*. It is uniquely created out of the unique data source identifier and properties of the entry.*

c. *All other topics are given unique* Subject Indicators *created using a global PSI prefix for the corresponding type together with its instance name.*

d. *The* Base Name *of a topic is created from the information to represent with the intention to provide a name for recognition by the user.*

e. *The* Topic Naming Constraint *is not used, so* Base Names *are not considered as unique.*

Principle (a) in Proposition 3.3 is due to the subject of an automated processing that should be even possible for data whose context cannot be identified by software.

As a consequence, the user has to dissolve potentially incorrect associations between data during a search. This is deemed acceptable in the field of personal knowledge management, because the user can rebuild the context on his own and an incorrect association does not cause damage.

The referencing of stored data entries is performed with principle (b). This way all data is directly addressable via a unique identifier. Other non-addressable data, like names and dates, are given subject indicators in principle (c). By using the proposed PSI prefix for the corresponding type, meaningful indicators can be created to indicate equal topics for a later merging.

Principle (d) and (e) are required to provide the user a simple, meaningful but still computable information about the represented data. In many information domains unique names are rather an exception, especially if considering data across multiple sources. Additionally there is a controversial discussion if the *Topic Naming Constraint* is useful [Fre02], since it anyway forces scope statements to keep names unique. Imagine a phone book to be represented in a topic map. The registered persons could be connected with associations to their phone numbers. However, the names are not unique, which requires the phone numbers to be included in the scope of the person topics to fulfill the naming constraint. This in turn implies that the resulting topic map contains duplicated information, which have to be kept consistent over time.

The principles of Proposition 3.3 are considered the conceptional minimum of Topic Maps elements that have to be used in the context of this thesis to receive unified topic maps. The application of these principles and corresponding examples are shown in the proof of concept chapter in Section 5.2.1.

The Basic Process of Merging

The framework is intended to merge the separated topic maps to a global one. This is a resource consuming process that is required once to generate a merged map, which is used then as the unified knowledge representation. The Topic Maps merging process ensures that whenever two topics inside different maps are known to represent the same subject they are merged in the final representation. As described in Proposition 3.3, topics representing different subjects can have the same base name. Thus, the merging of topics is based on subject identification by *Published Subject Indicator* (PSI) and *Published Subject Identifier* (PSID), rather than on their not unique base names.

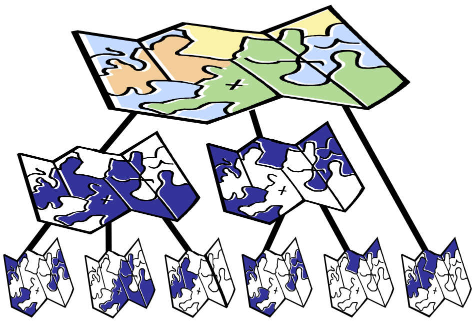


Figure 3.6: *Staged Merging of Topics Maps*

To reduce the effort for a repeated merging in case one of the extractor maps or input maps have changed, a staged approach is proposed (see Figure 3.6). This approach merges groups of maps to intermediate maps, which then are merged to the global one. In consequence only a subset of topics has to be merged after an update. Tests with the proof of concept implementation have shown a significant speedup compared to a serial merging of all maps (see Chapter 6). The grouping of maps in stages depends on the frequency they are used for merging. Maps that do not need to be merged often should be grouped, where as those with frequent changes should be merged directly into the global map.

However, some extractor maps may change quite often like in case of those representing email inboxes. Thus, it is necessary to distinguish between the kind of changes in data sources to decide about possible methods to handle these updates. Either data is added, updated or removed.

External adding of information to a data source can be handle efficiently: The event can be recognized by the corresponding extractor through protocol features or, if the data source type lacks this functionality, by periodically comparing the source with the previously generated topic map of the extractor. The new data then can be merged efficiently into the corresponding maps without having to process any unchanged data.

In case of a removal of information things are getting more complex. Even if the data source supports a sufficient notification for this event, it would be very resource consuming to identify all topics in the merged map that have been created as a result of the existence of the now removed information. Additionally, many data sources lacks the capability of unambiguously identify objects beyond their removal, because newly added objects can obtain the same identity and takeover the place of removed objects.

Fortunately, there is a reasonable solution for this problem by dropping the requirement that the representation should reflect removed data immediately. This is justifiable because the existence of information that is based on removed data inside the representation, does at most have an effect on provided search results, which in that case could contain entries the user can not access anymore. To reduce these void items, the extractors can be used to reinspect the data source from time to time, which automatically builds an up to data representation, after merging the updated extractor map with other intermediate maps to a new global merged map.

Proposition 3.4. *Keeping the representation up to date with reasonable effort requires therefore three principles:*

- *Add newly incoming data directly to each individual extractor map and simultaneously to the current global merged map.*
- *Use a staged merging process for extractor maps and input maps.*
- *Address removed data by reinspecting all data sources from time to time depending on the amount of data removed in a selectable interval.*

Besides this direct approach of physically merging topic maps, there is also the possibility of virtually putting separated maps together inside the Topic Maps engine[7]. The maps then reside separated, but the Topic Maps engine handles requests on the fly as if the maps were merged. However, this has the disadvantage of spending additional computing effort for every request, which also even increases with the size of each included map. For the described application, this is a major drawback, because of the decreasing access performance. An immediate access is deemed far more important in this use case than up-to-dateness. Furthermore the effort necessary to physically merge maps can be performed in the background when resources are available. Besides other restrictions when virtually merging maps, described in the implementation Section 5.2, this consideration is the reason for choosing the physical merging approach for the design of the extractor concept.

3.4 Navigation in Knowledge Graphs

After having described the creation of the knowledge representation, now the navigation inside the resulting graph should be presented. The following sections therefore show the basic interaction concept for navigation inside the knowledge graph. Its primary design goal is a generic access method that is independent of data source specific properties. The postulated universality thus creates the flexibility to use the same interface for arbitrary information domains without adaption efforts.

The second design goal is directly connected to the limited interaction capabilities of smart phones and other devices with reduced key pads. Especially on smaller mobile devices, the input of many characters is a disadvantage that should be avoided to speed up the search procedure. This requires to consider the character of mobility. It comprises basic conditions for bandwidth and latency of mobile communication, as well as the limited display and interaction capabilities of mobile devices.

To address the two aspects of generic access and mobile device constraints already in the underlying design, this thesis proposes a path-centric concept with a closed interaction cycle in Section 3.4.1. This concept is adaptable to different platform properties, as it describes the principles for navigation inside the generated knowledge graphs without specifying or requiring a particular graphical interface. The concept is then extended in Section 3.4.2 with further navigation elements to aid the user to keep track of their digital knowledge. Combining the concept to an interface, Section 3.4.3 provides examples, to give a first visual impression of the intended interaction process, which illustrates the foundation used in Section 3.5 to create the search interface.

[7] In TM4J the corresponding interface is called `UnifiedTopicMap` (see http://www.tm4j.org/)

3.4.1 Using a Closed Interaction Cycle

As topic maps represent a network of topics interconnected by associations, it is obvious to use the spawning paths between topics for navigating through the graph. But how should the user interface look like for a network with more than tens of thousands nodes and its dense connections with associations? Presenting the whole graph would be far beyond the capabilities of mobile devices, but presenting only single nodes (topics) at a time disposes the user's survey of the relations to the rest of his digital knowledge. Therefore, a trade-off between a graph-centric and a node-centric navigation approach (see [DKF+03]) is proposed in this section. As a result, the approach follows a path-centric paradigm, which is an integral part of the concept design by providing orientation guidance, a convenient navigation method, and a possibility for the search result visualization. Because the structure of paths reoccurs in all of the former aspects, the user can recognizes this concept easily during his interaction with the interface, which decreases the required time to get familiar with a new kind of interaction process.

Thus, this thesis proposes an interaction model called *Click and Cycle* — referring to its closed structure — which is based on three states forming a cycle (see Figure 3.7(a)). In this model, initially presented by Heider et al. in [HB06], each state consist of a list of elements of which one can be selected by the user. With these selected element, each state offers at least one transition to another state. The primary transition is the navigation process along the paths inside the graph. Additionally, each state provides transitions invoking generic actions on the states' current element. In the first state, these elements are terms which represent a subset of the digital knowledge vocabulary. Elements of the second state are topics and in the third state the elements are associations.

The starting point of the cycle is the term selection state. Therefore, the next section first introduces the creation of terms before the interaction concept is described.

Splitting up Topics

Although the graph consists of topics as their underlying node elements it is built of, the descriptive character of a topic provided by the base name often can be further itemized.

The creation of topic base names from natural languages provides the possibility to decompose them into their parts, called *terms* here. These are the pieces of titles, names — or more generally considered: pieces of all subjects — the user will probably recall and try to search for. Each unique term then is linked in the concept to all topics it is found in. As every topic is treated this way, regardless of its type and usage, a complete personal vocabulary of terms is created by the concept. Thus all terms contained in the representation are known to the system in advance and the user can be offered a list of fitting terms after having entered only a few characters. Therefore void search queries can be prevented, which also reduces the produced network traffic. Additionally, the survey of search alternatives is improved and thereby the input of the search criteria is sped up in general. This advantage can be also used for the desktop interface where the personal vocabulary even increases its potential if leveraging the display and interaction capabilities with highlighting methods, drag and drop functionality, and graphically visualizing term relations.

The second advantage of splitting up topics into terms, is the implicit generation of linguistic relations between topics through its equal terms, which can aid the user in finding related information that is not explicitly linked by the topic map. Of course sometimes, these relations are not correct in the sense of the actual search, like it is obvious in the case of homonyms or acronyms. However, as the framework is designed for humans, it can be assumed that they are used to deal with those multiple meanings.

Interaction Concept

In the Click and Cycle concept, a generic interaction with the knowledge graph starts by choosing a term that the user recalls in relation to a searched topic (see Algorithm 3.5). If too many terms fit the initial selection in this term selection state, further details can be specified in the *redo transition* (see Figure 3.7(a)) to narrow down the available terms until the best fitting term can be selected. Selecting a fitting term uses the *navigate transition* to advance to the next state, which presents a list of topics containing the selected term. In this state, three transitions are available: *navigate transition*, *mark transition* and *redo transition*. The common case is the navigate transition, which selects a topic to see all of its associations by advancing to the next state. In contrast, the *mark transition* marks the current selected topic for a topic operation. Marking a topic puts it in a virtual bag of accentuated topics. After this bag is filled with the collected topics, interactions can be performed with the bag's items, e.g. for the search functionality (see Section 3.5) or for distributing connected data to other recipients (see Section 3.6). Like for the term state, the *redo transition* is used to narrow down the amount of available topics until the best fitting topic can be selected.

Moving on in the cycle by selecting the topic, in the following association selection state the user can select the desired association. Again two transitions are available from here: *redo transition* and *navigate transition*. Like for the other states, the redo transition can narrow down the presented elements until the fitting element is found. Whereas using the navigation transition closes the cycle by advancing to the term selection state, which lists all related terms contained in associated topics. If the desired topic still is not found, a further cycle can be used to follow other paths.

Navigating this way inside a topic map is independent from the contained data types and content. If new sources or data types are merged into the representation,

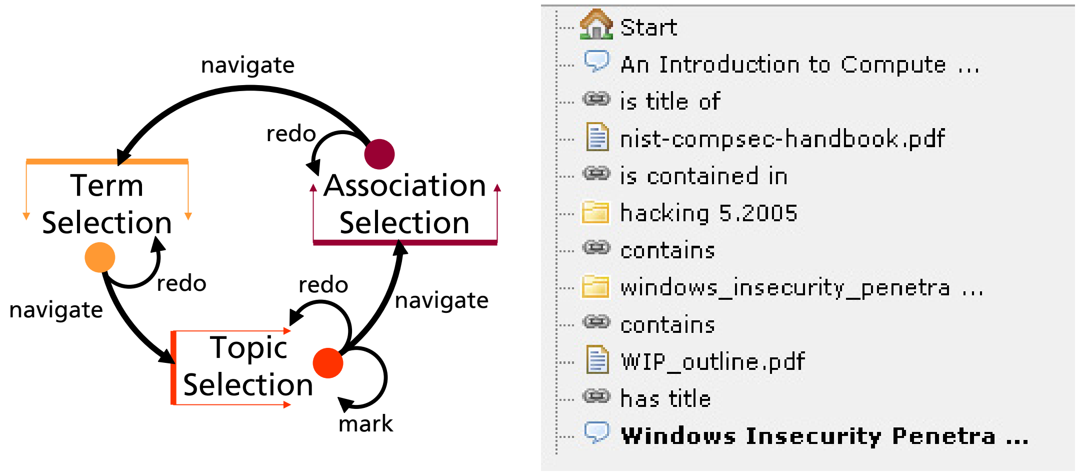
Algorithm 3.5 Click and Cycle: State Transitions for Graph Navigation

```
function Cycle
    info ← queryUserInput( )
    list ← getTerms(info)
    state ← term
    while true do
        showItems(list)
        item ← getSelectedItem( )
        if isLeafItem(item) = false then
            list ← getSubList(item)                        ▷ redo transition
        else
            if state = term then
                list ← getTopics(item)
                state ← topic                              ▷ navigate transition
            else if state = topic then
                if isMarked(item) then
                    addToCollection(item)                  ▷ mark transition
                else
                    list ← getAssociations(item)
                    state ← association                    ▷ navigate transition
            else if state = association then
                list ← getTerms(item)
                state ← term                               ▷ navigate transition
```

the code and the interface design do not have to be changed. This universal interface uses the structure and typing defined in the topic map. From this perspective, the presented interface is described by the data itself. Thus, the collecting extractors are responsible in an indirect way for creating the appearance of the user interface. This is reasonable since it is their constructed capability of understanding the source's structures. Extending the personal digital knowledge is therefore a task of adding new extractors to the framework, but the visualizing components stay completely untouched, pulling its interactive mechanism from the actual universal knowledge representation. Only a few navigational elements are necessary to enable users to keep track of their journeys through their personal knowledge space. These elements are all based on the representation's basis, namely the universal concept of topics.

3.4.2 Navigation Elements

Simplifying the navigation, users are allowed to navigate backwards and forth through the cycle, to visit views they have passed through. This is offered directly by conceptual navigation elements available from each node. So the movement on paths is both easy to understand and simple to implement. But the primary question is



Start
An Introduction to Compute ...
is title of
nist-compsec-handbook.pdf
is contained in
hacking 5.2005
contains
windows_insecurity_penetra ...
contains
WIP_outline.pdf
has title
Windows Insecurity Penetra ...

(a) State Graph for Navigation

(b) History of Navigation

Figure 3.7: *Navigating through the Topic Map by cycling the Interface [HS08]*

how to find an entrance to the representation? Transfered to the path interface, this is a question about which topic should be chosen as the starting node. Although at this point no information about the search interest is specified, the user has to be offered an entrance point to the topic map. In best case, the displayed entrance topic is already the one representing the information the user is looking for, however, also a closely related topic will be helpful. Therefore, first the general concepts for entering the graph and navigating inside are described, before Section 3.5 goes into detail about the advanced search techniques and their background in graph theory.

The following methods were evaluated to enter the graph navigation at useful topics. Each method serves a slightly different purpose depending on the kind of topic that users are looking for and the strategy they want to follow.

Keywords as Point of Entrance

In most cases users might remember a certain part of text contained in metadata of the information they are looking for. Therefore, it is useful to provide an interface to enter keywords contained in the topic that represents the searched resource. To make entering of keywords easier, a personal vocabulary can be automatically generated from all topics by splitting the *Base Name Strings* into terms (see Section 3.4.1). This way, users can browse through available keywords and the search interface can selectively provide suggestions for keywords, as users have to type only some characters of them.

With this approach, entry points are presented like in common keyword search interfaces. The difference here is the ability to also use the search result's relation to other topics. Users therefore do not have to hit the desired information directly nor even have to try to remember parts of its name, as long as a related term is known inside the representation. Because of the personal vocabulary the presence of such a term is visualized immediately when entering a few characters of it. If it is not displayed as a keyword, the users can rethink their search options, using the direct feedback during the search process. So using this strategy is indicated, if any term of the desired topic or one in close relation to it is known, because then the access to the referenced information is quite easy.

Hierarchical Root Topic

The most classic method to present an entry point is based on the hierarchy information contained in the topic map. Because of the used *Hierarchical Classification Pattern* and *Faceted Classification Pattern* (initially applied for this purpose by Heider et al. in [HB06]), all indexed data sources are represented in the topic map with their hierarchical structure. They share a common designated root topic, which can be presented to the user (see Figure 3.8). Beneath this topic, the next hierarchical level contains the root topics of each single source, connecting their trees to a global one. This way, the user is enabled to follow the paths to the leaves of the tree that is preserving the hierarchical structure of all sources. Of course, all other non-hierarchical associations existing in parallel — not displayed in Figure 3.8 for the sake of clarity — can also be used to navigate within the topic map.

This method is useful if the user remembers the hierarchical location in the original data source, which now can be accessed uniformly by using a single interface. It is much like the common way operating systems organize data in hierarchical systems. The separated roots of the trees — in that case the hard disks and network resources — are presented in a single view, which can be used to start the navigation. In case of the created topic map, this concept is used equally to provide the user a comparable interface, even though it is only virtually composed and therefore provides a complete overview across every and all indexed sources, regardless of their physical location and independent to which hardware device it belongs.

The root topic offers a method to find resources in case users are aware of the original structure they have given a particular source and now want to access the contained data in a familiar way when working with their sources. So using the root topic represents the strategy of a classic access that would be possible similar if working locally with the resource. However, now it can be used also remotely and independent of the protocol that is used to establish the connection. This simplifies the usages in a way, that makes it also desirable to use this method even in cases a direct access would be also possible, because with only one interface any information can be retrieved across all data source borders.

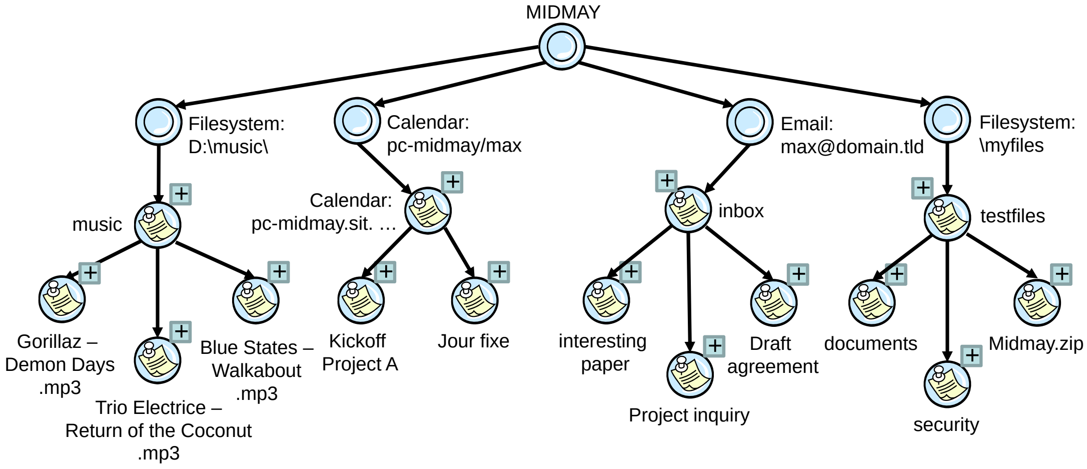


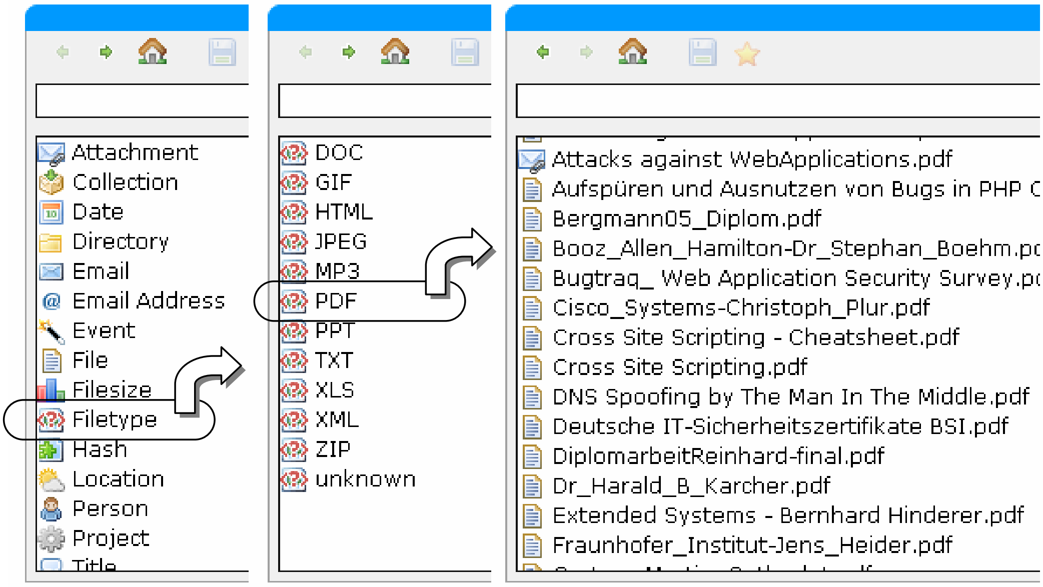
Figure 3.8: *Screenshot of a generated topic map with its root element. It offers a point of entrance using the hierarchy facets described in Section 3.3.1.*

Clustering of Properties

This point of entrance makes use of the closed nature of an information domain by clustering recurrent properties contained in the topic map. A common approach for this is the usage of the type information, stored for all topics for the clustering. The interface then presents a list of generated clusters, which entries lets the user choose one of the topic types contained in the topic map. By selecting one type entry, a list of all topics of that type can be shown.

In Figure 3.9 this approach is shown for the office domain and the clustering of file types. First the topic type *filetype* is selected that represents common file types found in the indexed sources. These topics are connected via a "*is filetype of*" association to the topics that are representing data with the selected property. Choosing this association, the *click and cycle* interface automatically presents all files with the selected filetype. Thus, no special functionality is required other than initially listing the topic types contained in the topic map, which is a cheap operation for the topic engine regarding the required computing effort.

In the example the usage of this point of entrance is useful to find documents of a certain file type, regardless of its physical location in a data source. Even the kind of data source and its special instance is not relevant for the user to know beforehand, as shown in the example where one entry is contained in an attachment indicated by the attachment icon. This also applies for searches for other topic types such as *location*, *date* or even the size of a file. The latter is a good example for another useful application of clustering. As the user very likely will not search for the exact size of a file up to the last digit, this property of a file is instead stored as an association to a topic representing a range of reasonable sizes. A single topic then acts as property



- Attachment
- Collection
- Date
- Directory
- Email
- Email Address
- Event
- File
- Filesize
- Filetype
- Hash
- Location
- Person
- Project
- Title

- DOC
- GIF
- HTML
- JPEG
- MP3
- PDF
- PPT
- TXT
- XLS
- XML
- ZIP
- unknown

- Attacks against WebApplications.pdf
- Aufspüren und Ausnutzen von Bugs in PHP C
- Bergmann05_Diplom.pdf
- Booz_Allen_Hamilton-Dr_Stephan_Boehm.pc
- Bugtraq_ Web Application Security Survey.pc
- Cisco_Systems-Christoph_Plur.pdf
- Cross Site Scripting - Cheatsheet.pdf
- Cross Site Scripting.pdf
- DNS Spoofing by The Man In The Middle.pdf
- Deutsche IT-Sicherheitszertifikate BSI.pdf
- DiplomarbeitReinhard-final.pdf
- Dr_Harald_B_Karcher.pdf
- Extended Systems - Bernhard Hinderer.pdf
- Fraunhofer_Institut-Jens_Heider.pdf

Figure 3.9: *Point of Entrance using Type Lists [HS08]*

for files of the size smaller than 10kb, another one for files between 10kb and 100kb and so on, covering all file sizes in selectable ranges. These property topics are given the type *filesize*, which then provides a similar access as described for files types. So again, the navigational possibilities for the user are defined inside the topic map, constructed during collecting information by the extractor components.

Additionally, displaying a list of all topics representing the types contained in the topic maps does also have benefits for the search process described in Section 3.5. The list can be also used to mark the complete topic type to be relevant for a search query instead of a single topic instance of the type. Considered this way, displaying all type information is also an elemental functionality deemed useful for convenient navigation and search inside a topic map.

History Path

When navigating through personal knowledge it is often necessary to have the possibility to go back and forth between previously visited topics. The user is offered this navigation element inside all views. To aid him beyond a single move back or ahead, a history can present his path from the point of entrance to the current position in the graph (see Figure 3.7(b)). This provides an additional aid to the user, locating his position in the representation and offers contextual meanings for the actual node (see [PK00]).

Drawing a path visualizes the relation of the intermediate topics and serves as navigation shortcut to topics and associations. Choosing a topic from the path lets the user enter the related association view of this topic, displaying the available associations. Whereas selecting an association entry inside the path brings up the topic view, containing all related topics connected to the previous node in the path via the selected association. So the user instantly can revisit topics and redo navigational decisions, to follow another path in the graph. At the same time, paths can be displayed even on small displays quite easily. And if more space on the display is available, path visualization can be even enhanced with more sophisticated display techniques by showing also unvisited adjacent topics in the surrounding of each contained node in the path.

Favorites

Topics that are used often are proposed to be marked as favorites. This is useful for example if the topic represents a document used frequently, like in reoccurring business processes the submitting of trade terms to partners. Choosing a topic from the list of favorites provides an instant access to the referenced document from any client. The user then can select the referenced document to be transmitted easily to any recipient, without performing a new search for it.

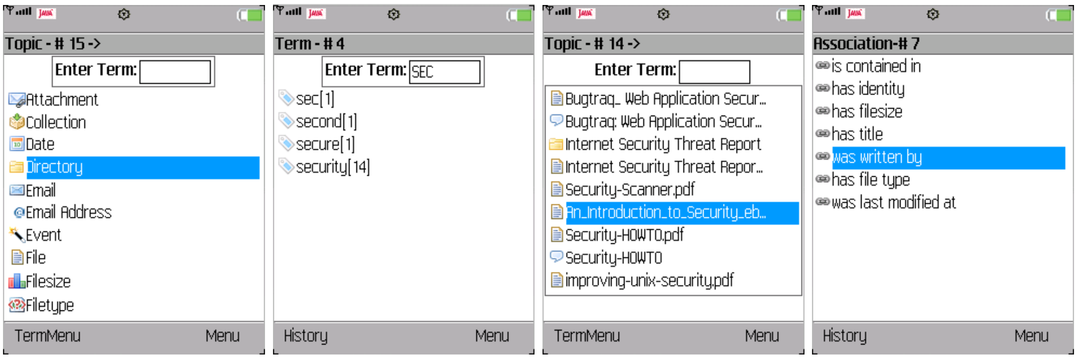
Depending on the data and its personal structuring, it often also makes sense to use favorites for topics that are frequently used in search queries. This way, selecting the required topics as input for the query can be simplified to only a few selections. Optionally, it is also possible this way to store the actual selection of topics for a query to be easy reproducible in future requests.

3.4.3 Examples for Navigational Strategies

After having described the general concept of navigation inside the knowledge representation, the following sections should show some applications and strategies emerging from the concept. Examples in the office domain are described to visualize the general strategy, usable also in other domains. The presented strategies have been tested for feasibility, utility and performance aspects with the mobile MIDMAY client. Although the proof of concept implementation will be presented primary in chapter 5, the presented screenshots should support the clarity of the strategies with the visual experience a user could grasp.

Navigation from Topic to Topic

Every navigation initially starts with the decision about the entrance point. For this example the user wants to find a document with a name that the user recalls at least partly. Therefore entering the first letters of the contained term "security" brings up



(a) Main View (b) Term View (c) Topic View (d) Association View

Figure 3.10: *The interaction starts by either selecting a type in the main view (a) or by entering characters to display the available terms from the personal vocabulary (b). After selecting the term "Security" all related topics are presented (c). Selecting a topic view entry brings up the association view (d), which closes the cycle after selecting an association. (Screenshots of MIDMAY's J2ME-Client)*

the *term selection view* of the interface (see Figure 3.10(a)). It contains all terms starting with the entered characters and presents the amount of topics each term is contained in. This information is added to the term entry in squared brackets. Choosing the term "security" then brings up the *topic selection view* (compare Figure 3.7(a)) comprising all fourteen topics containing this term. As the icons of the *topic view* indicate (see Figure 3.10(c)), the displayed topics are of different types. This is a very important thing to envision when trying to think of a fitting keyword for the entering of the graph, because picking a topic provide users with the possibility to specify what subject they are looking for, instead of specifying where to search for it. Searching for the subject "security" thereby is independent of the storage of related information. By choosing an entry of the topic list, the users thus select entry points and together with this decision they also use the inherent information type. The users then enter the *association selection view* (see Figure 3.10(d)), which presents the available relations to other pieces of information. In the case of the selected file, users can choose to see properties of it, they can choose to retrieve this information or they choose to use the property topics to navigate to another desired information. Using the "was written by" association can be used therefore to find out other data related to the author of the actual document, simply by traversing the topic representing the author and then using the mirrored association "is author of". This way, other files, emails or even appointments with the author can be found by navigating along the paths from topic to topic.

Leveraging known Structure of Data Source

Sometimes no information about the stored data to find is known. But even without knowing keywords of a document, a user may recall structural parts of the presumed data source. For example this user usually stores documents related to security in a directory named "Security-Docs", in which subfolders have been created to sort documents by subjects. With this clue about his personal structured digital knowledge, he first fetches the term "security", which in turn visualizes also the topic that represents the "Security-Docs" directory. Choosing the "contains" association then lists the directory's representation of the content including also its subfolders. Because he presumes his document inside the subfolder "USB", he uses the associations to browse this folder's content and finally finds the desired document. Of course he could have also used the terms "USB", "docs" or any other term contained in topics in the surrounding of the searched topic. Thus, leveraging the personal knowledge with the structure of a data source increases the amount of the representation's easy accessible entry points.

However, structure does not only stand for the one found in filesystems. Another good example is the interconnection of properties extracted from metadata. Imagine music files with metadata stored in the common ID3[8] format inside the content. The MIDMAY metadata extractor (see Section 5.2.5) recognizes this metadata and generates additional topics representing it. The content of the "Album" and "Genre" field for example is used to create a topic with the type *collection* and a base name built of the field's content. The *collection* topic is associated with the "contains" association to the topic holding the reference to the physical file. Thus, searching for a particular music file can be easily performed also via the name of the album it is published on, or even with the name of the genre it probably belongs to. Having retrieved a term contained in the album name, one can access the corresponding collection representing the content of the album. It offers access to all music files belonging to this collection, independent of their physical storage location and the data source type. This also includes attachments, databases, and in general any other source that contains data recognizable as files. The user can move from topic to topic to explore other related metadata and can also move back to the collection content by using the history function to start a new path to another music title. Therefore, in this example the structure consists of the clustering provided by metadata. Because the user knows about this clustering, he can leverage the structure as it is preserved inside the topic map.

Another example of the same type is the concept of projects. Topics representing a projects are associated with all resources that are involved in it. This way topics rep-

[8] ID3 is a metadata container most often used in conjunction with the MP3 audio file format. It allows information such as the title, artist, album, track number, or other information about the file to be stored in the file itself.

resenting persons, documents and appointments are rebuilt inside a usable structure by autonomously bringing together information from LDAP[9] and calendar servers. These structure provide interesting connections also for searching in other directions because this structure reside beside all other relations and provides therefore new paths for the user to leverage his digital knowledge.

Using Paths in Mind

Especially in cases very little is known about the stored piece of information to retrieve, the personal digital knowledge can help the user by leveraging its data source connecting capabilities. Let's say user Tom has stored the topic representing the person *Steve* as a favorite, because of many good documents authored by him that Tom often requires. Access to all those documents is made an easy task then and even authored documents received in the future will be accessible via the stored favorite topic "Steve". Now Tom uses this topic as the entry point to his digital knowledge, because he knows that Steve attended a meeting of which Tom currently recalls neither date nor name. However, this meeting is known to have an attendee who is the author of a desired document that Tom tries to find. Thinking of this circumstance, Tom uses the "is attendee of" association, showing all meetings Steve attended together with Tom. This way the relevant meeting topic can be found and Tom follows its "has attendee" association to all other invited persons. Seeing the list of persons, Tom now remembers the name of the author, chooses the related topic, and follows the "is author of" association, to display all topics referencing to information this person has authored. So this strategy is based on using relations to known information to retrieve information that the user currently can not think of.

However, following this paths in mind sometimes will be difficult in cases intermediate topics of the path to follow are connected with lots of other topics via the same associations. This will be addressed in the next chapter by letting the system calculate reasonable paths after selecting its endpoints, dropping the need to manually following the path by trying out multiple possibilities to reach the desired topic.

3.5 Finding Information using Graph Theory

So far, only navigational aspects on personal digital knowledge have been described. They already provide the possibility of searching topics and browsing the representation, however, they miss functionality for more sophisticated search tasks. Especially in the case the user can describe the searched information with other facts than a keyword, there should be a way of specifying the search interest. So this section is about the approach for providing input about information the user is interested in,

[9] Lightweight Directory Access Protocol, an application protocol for querying directory services

to let the system find possible matching candidates for that request. This should go beyond the specification of terms that have to be contained in the information's metadata. Although specifying multiple terms reduces the number of interaction cycles as more information is provided at the same time, it lacks the possibility to leverage the inherent structure of the topic maps. The result then depends on the description capabilities of the used terms and the suggestion algorithm to assist the user with an automatic context-based topic search like proposed in [MLRM04]. However, even using such an approach does not change the problem for users to find the fitting terms contained in the content that they hope to retrieve.

In contrast, this thesis presents an approach that generates paths of relations between topics. The paths' aim is to show pieces of information closely related to both of the selected endpoint topics. Those provided topics can be anything contained in the topic map and therefore the user is not limited to information contained directly in the content to be searched. Based on the modeling of the topic map as a graph, and the usage of an introduced path generation algorithm, the calculation of results are also described for the case the user wants to provide multiple selected topics as input vector for a search request.

Before Sections 3.5.2-3.5.4 will show the underlying concept of the graph modeling and its application for the search algorithm inside digital knowledge, Section 3.5.1 will discuss the proposed approach against the classic query languages-driven way of dealing with Topic Maps.

3.5.1 Turning away from Query Languages

The key problem when trying to find information assisted by a system is to describe the information one is looking for. This problem also appears when accessing information stored at databases, where it is commonly solved by query languages. A similar approach has also been developed for Topic Maps with the query language Tolog[10]. Even a proof of concept implementation has demonstrated the usage in a mobile scenario, which is an important goal also for the ubiquitous information management discussed in this thesis. The application is running within a JAVA-enabled smartphone, providing mobile access to knowledge stored in Topic Maps [SKA05]. A second language TMRQL[11] is directly based on the relational database model, defining a core set of abstract relational views to utilize the capabilities of the SQL language for making search requests. The approach's aim is to provide better accessibility and usability for developers because of the well known SQL language and its sound support with tools and portability [MA05].

[10] Implementation by Ontopia of the requirements stated by ISO Topic Maps query language TMQL standardization effort

[11] Topic Map Relational Query Language

Both of these query languages and their applications do have much power due to their precise description capabilities of the desired result. However, the inherent complexity of specifying queries in these languages does also have some drawbacks for users. They have to know the correct syntax and at least some attributes of the data structure to produce meaningful queries. Thus, being designed for developers and technical experienced users, query languages can not be used for a user interface intended to help the average user to manage their stored information.

To overcome this, often other components that generate queries from user input are proposed to be implemented between the query language interface and the user interface, disburdening the user to know the query language syntax. An interesting example is the usage of the Query-by-Example[12] paradigm which has been demonstrated for Topic Maps in [WDDA07]. Based on the QBE paradigm, this approach involves a set of template queries in which the user fills in values and then asks the system to complete the table with the search results. This user-friendly interface, however, has the drawback of limiting search queries on the usage of predefined templates which forces the users to think in given rules, restricting their search possibilities. This is a direct consequence of the underlying conversion to the formal query language.

Proposition 3.5. *The reason to search for an alternative approach for the intended user interface is therefore an unbalanced trade-off in presented approaches regarding*

- *simplicity of the interaction to be usable by the average user*
- *integrability into the overall usage concept of navigation inside digital knowledge*
- *usability of correlation between personal digital and the user's knowledge*
- *flexibility of search possibilities*

Because of these reasons this thesis refrains from using query languages and instead uses a novel concept by applying graph theory as an underlying principle for searching in Topic Maps.

3.5.2 Graph Modeling

Although Topic Maps are often described as graphs when proposing the idea of storing knowledge by interconnected nodes, it is interesting to see that the application of graph theory was not considered noticeably for finding information in Topic Maps so far. Anyway, after designing a fitting model, using graph theory should provide the possibility to use both structure and semantic of a topic map together. This is possible

[12] The QBE paradigm was developed by M.M. Zloof at the IBM Yorktown Heights Research Laboratory in 1975. It is based on the idea that instead of only displaying the search results in a tabular form, the users rather should perform their requests by filling in parts of information into columns of the displayed tables [Zlo75].

because of the homogeneous mapping of the information into the representation, so one generic model can be used for all and every information domain. Driven by the idea of a personal information management, the existing information, now interconnected to the personal digital knowledge, can be used more intuitive by the individual who already has been in touch with the managed information. The underlying principles and algorithms, however, are hidden from the users. The only concept they have to understand is the path paradigm which can be well explained to the user with the quite natural way of following associations in mind when thinking of subjects.

Preconditions for the Topic Map Structure

The generated topic map contains resource references to data sources from where the actual information can be retrieved. The structure of the map therefore is designed to help users to find the desired references by preserving original structures they have given their data sources already. Additionally the redundancy of data in multiple sources is used to interlink information useful to enrich the representation. These are then accessible via a single interface.

Proposition 3.6. *From this point of view, building a general topic map representation with a homogeneous structure requires that*

- *unified Public Subject Indicators (PSI) are used for a global typing schema across all extractors, that is applied strictly to all topics,*
- *associations do also have a type to reflect their general semantic of a property, type or hierarchical relation for the global representation,*
- *there is no directionality inherent in an association as defined by XTM specification [Topo1] and*
- *the topic map has to be made consistent, making each entry unique in the knowledge space of a user.*

Transition between Topic Maps and the Graph Model

Unlike a complete modeling of the entire Topic Maps concept as presented with the τ-model [BS05], the following transition focuses on the inherent structure of a topic map only. This has been done for directed acyclic graphs in [MT05], but a model for personal information management has to consider undirected cyclic graphs because of the structure built by the extractors. The following model therefore represents the proposed transformation from Topic Maps to a graph model, designed for algorithms dealing with the concept of paths.

In this model, the graph G built by a topic map is described by the pair (V, E). V is the finite set of vertices mapped to topics one-to-one and E is a set of edges representing the undirected associations between the topics. Additionally, E explicitly contains the binary relations between topics and other Topic Maps' entities (e.g., topic types) to preserve this structural information. Therefore, graph G also contains a vertex in V for every type topic. Their aim is to connect all instances with its type and vice versa, without having to include additional associations into the topic map, which would be redundant information since the topics already contain typing information as defined in Proposition 3.6. Reification[13] is addressed by creating a new vertice in V, built out of the reified Topic Maps construct (e.g., an association), and including its relations from all involved topics into E.

Each edge $(v_i, v_j) \in E$ is given a constant configurable weight w_{ij} depending on the type of association and the search mode, which will be described further in Section 3.5.3. Based on this graph further definitions are required to describe paths for using them with algorithms. Since only direct connections between topics are of relevance for the search, only *simple* paths are considered, which means that these paths have no repeated vertices.

Definition 3.1. *A simple path of length k from topic t_a to topic t_b in graph $G = (V, E)$ is a sequence $\langle v_0, v_1, v_2, ..., v_k \rangle$ of distinct vertices such that $t_a = v_0, t_b = v_k$ and $(v_{m-1}, v_m) \in E$ for $m = 1, 2, ..., k$.*

In the later description there is the need for a notation in expressing relations between topics connected via a path. Since there may be multiple possible simple paths to connect two topics, the following notation introduces an index i to indicate a specific path out of all possible.

Definition 3.2. *The i-th path p_i connecting the topics t_a and t_b is written as*

$$(t_a \overset{p}{\leftrightarrow} t_b)_i. \tag{3.1}$$

Additionally there is the need for defining a construct to describe the set of all simple paths that coexists between two marked topics. This is important as all these paths together contain relevant information regarding the search interest of the user. The set is therefore used for the described approach to create search results.

Definition 3.3. *The set P_{ab} of all n paths connecting the topics t_a and t_b is*

$$P_{ab} := \{(t_a \overset{p}{\leftrightarrow} t_b)_i \mid 1 \leq i \leq n\} \tag{3.2}$$

[13] In Topic Maps, reification is defined as making a topic represent the subject of another Topic Maps construct, like creating a topic that represents the relationship represented by an association, to attach additional information such as occurrences (see [Into6a])

In this application field, it can be assumed that users are only interested in simple paths, so all paths in P_{ab} are defined to have distinct vertices to prevent cycles. This definition is comprehensible since a topic occurring more than once in a path would have two effects. Firstly, the resulting path would be in conflict with the approach looking for the closest relation, because it then would contain a cycle. Secondly, these paths would not increase the navigational options by presenting topics multiple times.

3.5.3 Search Algorithms for Digital Knowledge Graphs

Based on the graph model, this thesis proposes algorithms to calculate search results. First an approach for calculating paths between marked topics is proposed that establishes the functionality to derive relations. One application for these paths is the visualization of intermediate topics lined up by available relations. So the user is presented topics that otherwise have to be reached manually through topic to topic navigation. This approach is called *path search* in this thesis. Having marked two topics, paths are presented to the user, which probably contains the topics that represents the searched information.

The second application of the calculated paths is presented by an extended approach. Its purpose is the creation of search results even for arbitrary numbers of marked topics. In this approach the calculated paths are used to provide related topics regarding the search query. So the paths expands the information provided as search input with information taken from the knowledge representation. Thus, the generation of paths is deemed a generic way for addressing the search problem with graph theory.

Path Search Approach

A first simple approach proposed in this thesis only uses the topic map source as an equally weighted graph, using $w_{ij} = 1$ for all edges $(v_i, v_j) \in E$. The approach follows the general template for *Bidirectional Searches* described in [LaV06]. Two simultaneous *Breadth-first Searches* (BFS) create spreading waves starting at the selected topics t_a and t_b, until both waves meet at a joint vertex. At this point, the shortest path can be returned. However, this would create always only one path between two topics, which would neglect relation between topics not connected via that path. Therefore, the algorithm is modified to produce multiple paths. To produce further paths, one edge is removed from the graph per path calculation. The edge to remove is either one that is on an earlier path to an already visited vertex or the one that connects both waves.

Algorithm 3.6 describes the proposed adaption of the bidirectional search. Each time *nextPath()* is called, a new path is returned. If no further paths can be found an empty path is returned. The function maintains the variable $\pi[u]$ that stores the predecessor of vertex u, like in classic BFS (cp. [CLR90]). Additionally, the variable $h[u]$ stores which wave already has visited the vertex u. Two first-in-first-out queues Q_a and Q_b manages the set of vertexes to be visited in the next step by the corresponding wave. As long as vertices are available in one queue, the two breadth-first waves are alternately expanded by calling function *nextStep()*. It returns the vertex that connects both waves or otherwise *NIL* if no joint vertex was found in that step. Once per call of nextPath(), the function nextStep() also removes one edge in the search graph to facilitate finding a new path in a further call.

Proposition 3.7. *It is deemed not useful to calculate all possible i paths* $(t_a \overset{p}{\leftrightarrow} t_b)_i$ *since there are often multiple paths connecting the same topics via a different sequence of associations. Instead, calculated paths should provide the connected topics in meaningful relations for navigational inspection. Therefore the path algorithm have to take care to limit the paths to those consisting of distinct topics.*

The proposed algorithm significantly limits the amount of displayed paths compared to calculating all possible paths. This is a desirable effect in this case, as the user only wants to see an overview of close relations between the starting topics. In this case it is deemed sufficient if all topics in possible paths are presented at least once in anyone of the displayed path, preventing the display of all other possible combinations of topics in paths computable in a dense connected graph. This is provided by the proposed algorithm by removing only edges between topics already contained in calculated paths, so users get the chance to find any topic in a path at least once that is related to the marked topics t_a and t_b.

Weighted Path Search

The proposed path search approach can be further extended by taking the association types into account. These types are assigned for all associations, assured by the extractor framework described in Section 3.3.3. This way it is possible to distinguish for example between hierarchical relations and property relations during path calculation. By changing the weighting w_{ij} of edges for an association type, users are enabled to further specify their search interests.

For this extended approach, Algorithm 3.6 needs to be adapted to follow a *Uniform-Cost Search* (UCS) strategy (cp. [RN09]). Instead of expanding vertices in order of their depth from the root, now the vertices are expanded in order of their total weight from the root. At each step, the next vertex u to be expanded is one whose weight $g(u)$ is lowest, where $g(u)$ is the sum of the edge weights from the start vertex to vertex u. Therefore, the FIFO queues Q_a and Q_b are changed to priority queues, sorting $g(u)$ ascending.

When using the extended approach, the weights for the associations between topics have to be defined. An example for such a definition is shown in Table 3.3, which

Algorithm 3.6 Adapted Bidirectional Search Algorithm for multiple paths

```
function NEXTPATH($G$, $t_a$, $t_b$)
    for each vertex $u \in V[G]$ do
        $\pi[u] \leftarrow NIL$                          ▷ no predecessor of $u$ set
        $h[u] \leftarrow 0$                              ▷ vertex is unvisited
    $Q_a \leftarrow t_a$; $h[t_a] \leftarrow 1$          ▷ initialize queues and start vertexes
    $Q_b \leftarrow t_b$; $h[t_b] \leftarrow 2$
    while $Q_a \neq \emptyset$ AND $Q_b \neq \emptyset$ do
        if $Q_a \neq \emptyset$ then
            $hit_b \leftarrow$ NEXTSTEP($G$, $Q_a$, $\pi$, $h$, 1)
            if $hit_b \neq NIL$ then                       ▷ joint vertex found by wave $a$
                $part_a \leftarrow$ SUBPATH($head[Q_a]$, $\pi$)   ▷ use predecessors back to $t_a$
                $part_b \leftarrow$ SUBPATH($hit_b$, $\pi$)       ▷ use predecessors back to $t_b$
                break
        if $Q_b \neq \emptyset$ then
            $hit_a \leftarrow$ NEXTSTEP($G$, $Q_b$, $\pi$, $h$, 2)
            if $hit_a \neq NIL$ then                       ▷ joint vertex found by wave $b$
                $part_a \leftarrow$ SUBPATH($hit_a$, $\pi$)
                $part_b \leftarrow$ SUBPATH($head[Q_b]$, $\pi$)
                break
    CLEARCHANGEDFLAG($G$)
    return CONNECT($part_a$, $part_b$)
function NEXTSTEP($G$, $Q$, $\pi$, $h$, $id$)
    $u \leftarrow head[Q]$
    for each $v \in Adj[u]$ do
        if $\pi[v] = NIL$ then                             ▷ vertex was unvisited
            ENQUEUE($Q$, $v$)
            $\pi[v] \leftarrow u$
            $h[v] \leftarrow id$
        else if $h[v] = id$ then                 ▷ vertex already visited by same wave
            if GRAPHALREADYCHANGED($G$) $= false$ then        ▷ only once per path
                REMOVEEDGE($v$, $\pi[v]$)          ▷ remove edge that reached vertex
        else                                    ▷ found vertex visited by other wave
            if GRAPHALREADYCHANGED($G$) $= false$ then
                REMOVEEDGE($u$, $v$)           ▷ prevent connecting edge in next paths
            return $v$
        DEQUEUE($Q$)
    return $NIL$
```

presents a weight assignment for the association types *hierarchy (h)*, *property (p)* and *type (t)* to modify the search results. The shown values were chosen to deal with structures in the graph where multiple paths connect two vertices. Selecting one of the predefined weight assignment — i.e. one row of Table 3.3 — can then be used to modify the calculation of the most interesting path. This is useful in domains with a dense connection of topics, established via different association types.

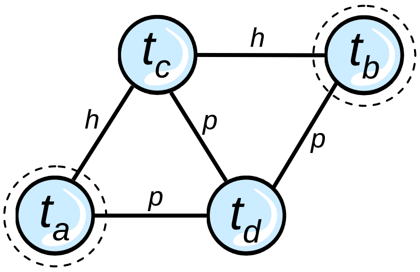


Figure 3.11: *Three-cornered Structure in Digital Knowledge*

Imagine such a structure consisting of three topics t_a, t_b and t_c with associations to the same property topic t_d and an additional hierarchical association between t_a, t_c and t_b as shown in Figure 3.11. By setting the weight $w(x)$ of one relation type x to $\frac{2}{3} < w(x) < 1$ and the weights for the other types to 2, the paths through one or more of these typical three-cornered structures are selectable. Of course, this is not accurate in all and every case one can imagine but these values have shown in many tests to work very well as an easy to use aid for the search process. By predefining these weights, the user only has to pick a descriptive mode of Table 3.3 to select the related weight assignment that expresses his search interest.

Table 3.3: *Example Weight Values for Path Search Modification [HS08]*

mode	$w(h)$	$w(p)$	$w(t)$	result
0	1.0	1.0	1.0	default case; shortest paths is calculated
1	0.7	2.0	2.0	hierarchical information in result path is desired
2	2.0	0.7	2.0	path should contain interconnection of properties
3	2.0	2.0	0.7	the interests in the relation of types is expressed

A practical example can be given by considering the structure of Figure 3.11 with given topics of a representation's section. The topic t_a then represents the folder "Security" which contains a subfolder "Mobility" denoted with t_c. This subfolder contains a document file "Bluetooth-Attacks.pdf" represented by t_b. Furthermore the topic t_d represents the date of creation connected via a property association to the other topics. The topics t_a and t_b have been marked by the user to find connecting paths that reveal information about their relation. Choosing *mode 1* for the path calculation then presents the path:

$$\texttt{Security} \xrightarrow{contains} \texttt{Mobility} \xrightarrow{contains} \texttt{Bluetooth-Attacks.pdf}$$

which shows the hierarchical relation between both selected topics. Therefore, in this case the marked topics and the chosen *search mode 1* describe the hierarchical search for the directory, which the marked file is located in, considered relative to the marked folder as the second topic. In contrast, choosing *mode 2* reveals the shared property t_d by resulting in the path:

$$\texttt{Security} \xrightarrow{\textit{has creation date}} \texttt{2009/03/12} \xrightarrow{\textit{is creation date of}} \texttt{Bluetooth-Attacks.pdf}$$

Further search strategies realizable with paths and the mode selection are described in Section 3.5.4. In those visualized paths also the type information of each topic is shown to the user. In the example above this was left off, as it has no impact on the calculation and therefore may only distract the reader from the underlying principle. Of course a user interface should present this information, either textual or by using a graphical representation.

Calculation of Intersections on Paths

This section deals with the case of a user who wants to provide information of the search interest with three or more marked topics, in order to specify the desired result more precisely. Instead of a path, the user then will be presented a set of topics that are deemed related regarding the search parameters.

To provide meaningful results in the case of multiple marked topics, this thesis proposes a usage of the already defined shortest paths as the underlying primitive. This is performed by first calculating all shortest paths between all m marked topics in $S := \{s_1, ..., s_m\}$ to extract the relations between the topics. The resulting paths are then transformed to bit vectors. Thus, all involved k vertices of all shortest paths found are indexed. These indexes are used for all paths p.

Definition 3.4. *The vector B^p_{ab} of length k indicates the presence or absence of vertices v_n in path p between vertice v_a and v_b for $0 < n \leq k$.*

$$B^p_{ab}[n] := \begin{cases} 1 & : \; v_n \overset{p}{\leftrightarrow} v_b \\ 0 & : \; v_n \overset{p}{\not\leftrightarrow} v_b \end{cases} \tag{3.3}$$

Operators on Bit-Vectors of Paths

Operations on the bit vectors representing paths have to be defined to provide the math for calculating results for multiple marked topics. Therefore the binary operations *OR* and *AND* are defined below for paths.

Definition 3.5. *The bitwise OR operation between two path vectors of lenght k is defined as*

$$B^p_{ab} \vee B^{p'}_{ab} := B^p_{ab}[j] \vee B^{p'}_{ab}[j] \text{ , } 0 < j \leq k \tag{3.4}$$

The AND operator is defined very similar to the definition above. It is used later to calculate an intersection between vertices of paths.

Definition 3.6. *The bitwise AND operation between two path vectors of lenght k is defined as*

$$B^{p}_{ab} \wedge B^{p'}_{ab} := B^{p}_{ab}[j] \wedge B^{p'}_{ab}[j] \;,\; 0 < j \leq k \tag{3.5}$$

With these two operations, the result of the query represented by the selected topics is calculated. First the OR operation is used on all bit vectors representing paths between two marked topics. The result is again a bit vector.

Definition 3.7. *The vector B_{ab} indicates the presence of topics that are contained in at least one of the n shortest paths between t_a and t_b*

$$B_{ab} := B^{1}_{ab} \vee B^{2}_{ab} \vee \cdots \vee B^{n}_{ab} \tag{3.6}$$

The OR operation is performed for all $\frac{m(m-1)}{2}$ pairs of the m marked topics in S. The resulting bit vectors of this step are then combined to a single one by applying a binary operation. In the default case this is the AND operation. The remaining topics are characterized by the closest relation — in terms of the lowest overall weight — to all marked topics. This characteristic is called *interestingness* in this thesis for the context of a specific search.

Definition 3.8. *The vector $B_{res(S)}$ represents the topics with a maximum of interestingness defined by the smallest weight of the paths to all of the m marked topics in the set of marked topics S.*

$$\begin{aligned} B_{res(S)} := B_{s_1s_2} \wedge B_{s_1s_3} \wedge B_{s_1s_4} \wedge \cdots \wedge B_{s_1s_m} \wedge \\ B_{s_2s_3} \wedge B_{s_2s_4} \wedge \cdots \wedge B_{s_2s_m} \wedge \\ B_{s_3s_4} \wedge \cdots \wedge B_{s_3s_m} \wedge \\ \vdots \quad \\ \wedge B_{s_{m-1}s_m} \end{aligned} \tag{3.7}$$

A user interface can then present the topics referenced by the bit vector $B_{res(S)}$ for further interaction. Presented topics have relations to all marked topics expressed by the existence of intersections in their combined shortest paths. With the Click and Cycle interface, introduced in Section 3.4.1, the user than could explore the displayed topics using them as navigational links. Using them provides additional facts, beyond the displayed name and type, set up by associations to other topics.

Choosing the Search Interest

Using the described set calculation together with the weight assignment for the underlying path calculation presented in Table 3.3, the user can further specify his search interest. By changing the weighting for association types the calculation of the shortest paths is modified. The characteristic of all shortest paths then also is reflected in the resulting intersection.

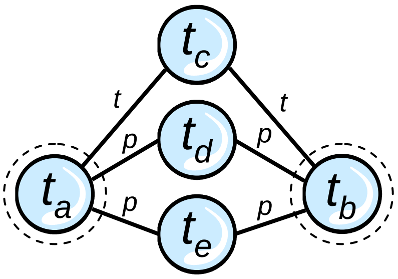


Figure 3.12: Application of Mode Selection for Set Calculation

Figure 3.12 shows the simplified cutout of an example structure between two marked topics t_a and t_b. For this example, an alphabetical ordering of the topic indices is used. The shortest path $p_1 = \langle t_a, t_e, t_b \rangle$ then is represented by bit vector $B^1_{ab} = \langle 1,1,0,0,1 \rangle$. A further shortest path $p_2 = \langle t_a, t_d, t_b \rangle$ is represented by $B^2_{ab} = \langle 1,1,0,1,0 \rangle$. If choosing mode 2, the also possible path $p_3 = \langle t_a, t_c, t_b \rangle$ is excluded because its weight disqualifies it as a shortest path, caused by the value assignment described for mode 2 that penalizes all typing associations. The combined vector B_{ab} then equals $\langle 1,1,0,1,1 \rangle$. Now imagine two additional bit vectors B_{ag} and B_{bg} calculated due to a third marked topic t_g not included in Figure 3.12 for the sake of clarity. The resulting vector $B_{res(S)}$ with $S := \{t_a, t_b, t_g\}$ then contains only topics contained in all of the vectors B_{ab}, B_{ag} and B_{bg}. All of these vectors represent the closest relation regarding the same selected mode. Therefore the resulting set $B_{res(S)}$ contains only topics with a close property relation to all marked topics in S.

Proposition 3.8. *The result set $B_{res(S)}$ can be modified based on one of the possible modes created by selection of a line in the weight assignment table.*

- *Mode 0, the default mode that only uses the structure of the topic map*
- *Mode 1, to focus the search on topics equally connected by hierarchy*
- *Mode 2, to focus the search on equal properties of marked topics*
- *Mode 3, to focus the search on equal types of marked topics*

Compared to query languages, this simple approach already produces valuable results and is easy to handle even for non-technical users, which is documented with examples in Section 3.5.4. But of course the transformation into bit vectors was done to have the ability to expand the possibilities for search requests to other binary operations. In the default case the ordering of the marked topics does not matter. In an advanced mode it can be used to prioritize their meaning by specifying a binary operation between sets of marked topics. The operation is then used on the resulting bit vectors of the sets of marked topics. Other binary operations such as *OR* and *XOR* are defined similar to the *AND* operation and produce according results. OR accumulates result topics whereas XOR presents the differences between the result sets. The main challenge of the extended approach with this functionality is to provide an easy to handle user interface. It should not swamp users with functions they rather rarely need. In the proof of concept chapter, Section 5.4 therefore describes the realized parts of the possibilities provided by the proposed applying of graph theory.

3.5.4 Usage Examples

The following examples should explain how marking topics to calculate paths and sets of topics can be used as a search technique. All examples presume a topic map autonomously extracted by MIDMAY from a personal digital organizer, filesystem folders used for storing documents and the personal email folder. Additionally, the company's LDAP[14] directory is extracted, which adds the projects' membership of employees and many other useful information about them to the digital knowledge of the user.

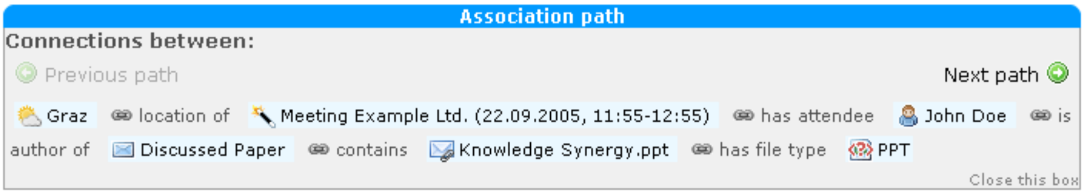
Leveraging Paths

In the first example the user Alice is looking for a document but can not remember its filename nor its author. However, she recalls the town where she attended the conference and met the person who emailed the document. Provided that the personal organizer's calendar application contains the meeting along with location and attendants information, the extractors have already included this knowledge into the topic map.

To use the path generation, Alice navigates to the topics that build the endpoints of the query. Because she is searching for a Microsoft PowerPoint presentation sent by someone she met at a meeting in the town Graz, Alice navigates to the location topic *Graz* and to the filetype topic *PPT*, using the concept described in Section 3.4, to mark both of them. Then she initiates the search process which generates shortest paths between the marked topics. Sorted by the weight, the computed paths (see Figure 3.13) now show the resulting relations between the marked topics. Alice can browse through the paths to locate the one containing the topic referencing the document she wants to find. In the example the shown path contains the email with the subject "Discussed Paper", which contains the attached PowerPoint presentation she is looking for. Having found this topic, Alice can browse through properties of the attachment, navigate to a related topic or open the attachment directly, if the system running the used client supports to display that file format. If this is not the case, like in the case of small mobile phones, Alice can instruct the knowledge representation to send this presentation to an appropriate recipient, such as a fax machine in her hotel. The access of data and its redirection is described in detail in Section 3.6.

There is also the possibility to narrow the search in a second step, in case too many alternative paths are found. This strategy uses topics presented in the paths to exclude not related options. So, if the location topic Graz is connected with many other meetings, Alice first locates a path containing the name of the person who had sent the attachment. Then she can replace the location topic with this person topic

[14] Lightweight Directory Access Protocol, an application protocol for querying directory services



Association path

Connections between:

Previous path Next path

Graz location of Meeting Example Ltd. (22.09.2005, 11:55-12:55) has attendee John Doe is author of Discussed Paper contains Knowledge Synergy.ppt has file type PPT

Close this box

Figure 3.13: Calculated shortest path between the location topic Graz and the filetype topics PPT. (Screenshot taken from MIDMAY web-interface) [HS08]

and re-initiate the path calculation. This excludes all paths containing other persons who also attended meetings in Graz.

Another strategy to find that person topic "John Doe" of Figure 3.13 would be to mark topic *Graz* and topic *Person*, which is the type representing the concept of a person in a topic. The resulting paths than contain all persons that are related to Graz. After picking the desired person topic in one calculated path, the search can be continued by marking other known aspects about the searched document.

Marking Multiple Topics to Search

Imagine Alice is trying to find authors of presentations that are related to a certain project. Again the required information to answer her question is already stored in maintained data sources. By letting the knowledge representation combine them in a topic map, Alice can use a single interface to mark topics that come to her mind when thinking of the desired result. So she marks the topic *Person* because the result should be closely related to the name of a person. Then she marks the topic *MIDMAY*, which represents the project she is interested in (see Figure 3.14). These two selected topics would result in paths containing all persons involved in the selected project. Additionally marking the topic *PPT*, which represents the type of PowerPoint presentations, the result is narrowed to persons that are also related to this file type. Then the result set caused by the three marked topics is calculated. For this example a single topic "Jens Heider" is displayed as result in the topic view section of the user interface. This topic represents the only person who is involved in the project MIDMAY and who has a close relation to the filetype PPT. In this case the relation exists via his authorship of presentations known to the system because of extracted metadata from PowerPoint files. Now Alice can navigate from the displayed topic to other associated topics to browse for additional related information, such as an email address or telephone number. Those information are interconnected from the LDAP directory. But before she gets in contact with Jens via these obtained contact information, she navigates to his presentations to see the covered aspects.

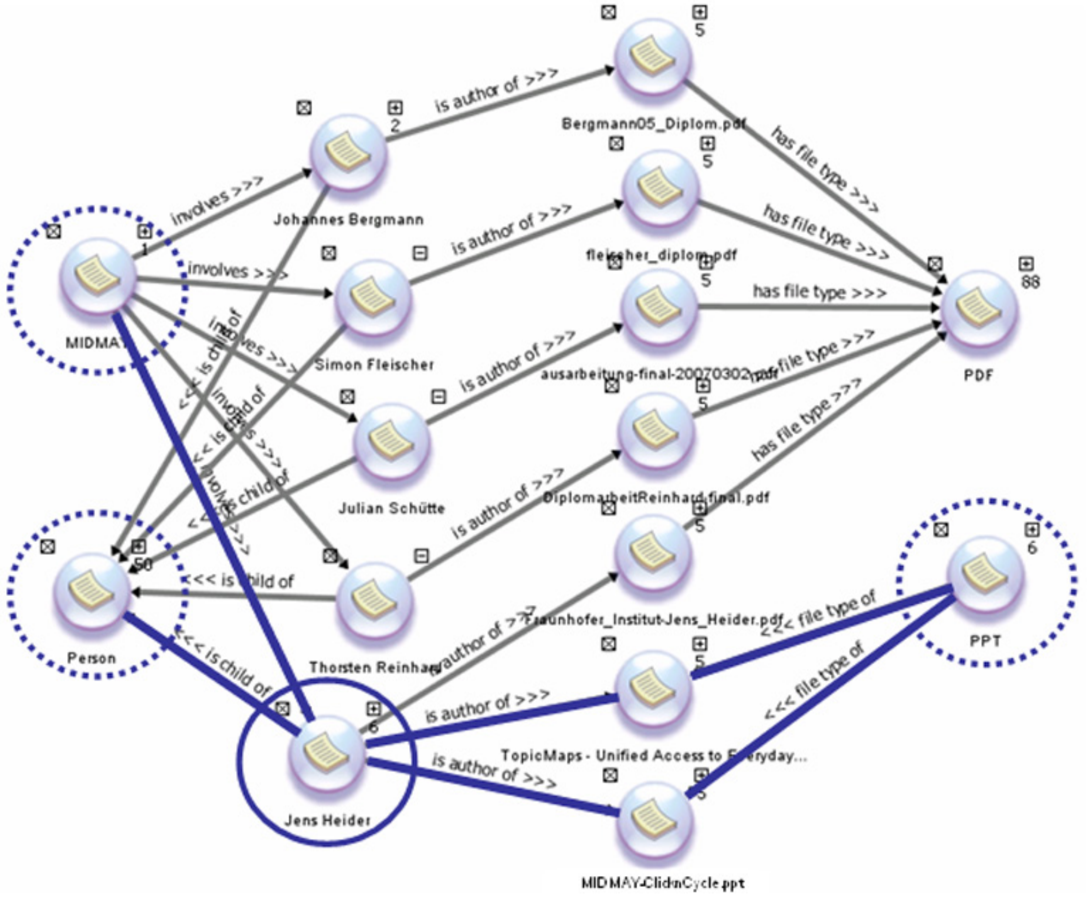
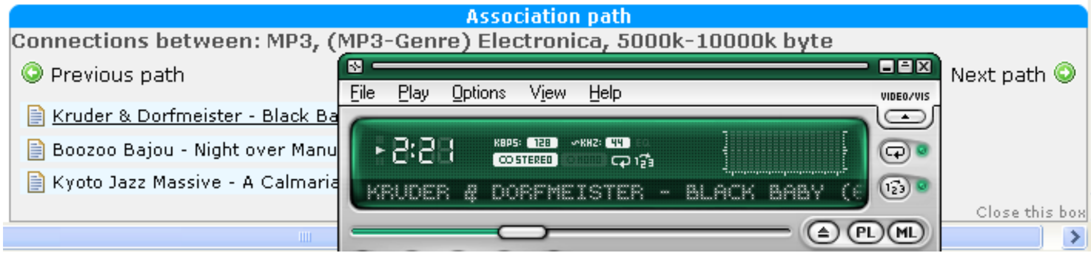


Figure 3.14: *Visualized cutout of a knowledge graph with highlighted result. The search request contains three marked topics in search mode 0. This visualization was created with an external Topic Maps viewer to provide an overview of the structure the search was performed on. Non-relevant nodes have been hidden, indicated by the numbers on the top right side of every node, to improve the readability. [HS08]*


Association path

Connections between: MP3, (MP3-Genre) Electronica, 5000k-10000k byte

Kruder & Dorfmeister - Black Ba

Boozoo Bajou - Night over Manu

Kyoto Jazz Massive - A Calmaria

Figure 3.15: *Search result of a request looking for mp3 files with genre Electronica and a filesize of 5-10MB. The files are directly accessible, if the system running the MIDMAY client supports opening the accessed filetype. In this example the MP3 player is directly invoked with the file retrieved via MIDMAY.*

After Alice has found a presentation, which provides the information she was looking for, and her talk with the author, she is looking for a certain music she would like to start the presentation with. Because Alice is only interested in MP3 music files, she first navigates to the *filetype* topic "MP3" via the type hierarchy, which lists all available filetype topics. After marking the topic *MP3*, she uses the term-search by entering the characters "elec" to find — among other terms starting with these characters — the term "Electronica", which is the genre she intents to limit her search to. The list of topics containing this term contains a *collection* topic "(MP3-Genre) Electronica". So she marks that topic to include it to her search query. Because she also knows that the searched file was quite big, she uses the type hierarchy again to navigate to the *filesize* topics to mark the "5000k-10000k byte" topic. In the calculated result Alice then browses through the list of MP3 files that are related to the given criteria and starts hearing some tracks (see Figure 3.15) to choose a fitting one. She does not notice, that some files where retrieved from her email inbox, whereas others reside in various folders in different filesystems. This way Alice can focus on her task of finding the right music. Other examples to do this are listed in Table 3.4 together with the intention of the presented search strategies. Although these examples are focused on metadata taken from music files, no additional code has to be implemented to provide these search functionalities. Any structure contained in the topic map can be used for searches exactly the same way.

Table 3.4: *Examples for Search Requests using multiple marked Topics*

Intention	Topic	Topic Type
Find MP3 files with a specific genre and filesize in a given directory	MP3 Electronica 5-10MB Music	Filetype Collection Filesize Directory
Find MP3 files belonging to a certain album, last modified on a certain date	MP3 Demon Days 2007-11-25	Filetype Collection Date
Find MP3 files contained in a subfolder with an arbitrary modification date	MP3 Downloads Date	Filetype Directory Type
Find MP3 files in filesystems from a certain artist	MP3 File Gorillaz	Filetype Type Person

3.6 Distributing Information

Information contained in the user's digital knowledge repository is also useful to be shared with others. This thesis uses the term *distribution* to describe the delivery of stored content that is linked from the knowledge repository. After the user has located a desired information (see Section 3.5), two kinds of follow-up tasks can be observed often. Either the information should be used for a personal demand, like working with the content, or the information was searched to be able to share it with friends, coworkers, and project partners. Up to now, the user has only located the topic that represents the wanted information, using its metadata stored inside the knowledge base. Now the same generic interface should be also used for information access.

Proposition 3.9. *Data referenced from personal digital knowledge implies the following use cases that have to be analyzed regarding requirements for user interface capabilities and for the design of information exchange between multiple nodes involved in the personal information workflow:*

- *Local access to stored data from current working environment*
- *Redirection of stored data to another device*
- *Distribution of stored data to other individuals*

In case the user wants to work with the actual physical piece of information, the related topics first has to be selected. The referenced data can be retrieved then from the data source as described in the following Section 3.6.1. But before retrieving, the user should be able to choose how to proceed with the referenced data. If the user wants to work with the information on the same device that was used for the interaction, the data has to be retrieved by the responsible extractor and can be transfered directly to the device to be displayed locally. However, some devices may not be able to display the data format directly, which should be addressed by redirecting the output of a document to another device that is capable of dealing with the format. The possibilities of redirection are described in Section 3.6.2.

Sharing the information with a closed group of recipients is the third alternative. A Personal Digital Knowledge System therefore provides the possibility to send chosen parts of information to selected recipients. The key issues for this functionality regarding communication are described in Section 3.6.3.

When designing the concept of transferring content referenced from digital knowledge, the question arises which possibilities are created by including additional context information. This context could be taken from the sender's knowledge representation and could be combined with the recipient's digital knowledge to provide an additional advantage for this type of information sharing. Therefore, the general concept of context generation and distribution are discussed in Section 3.6.4.

3.6.1 Selecting and Retrieving Information

Any topic inside the topic map can represent data stored at arbitrary physical locations. Therefore, all types of data can be referenced by digital knowledge. The granularity of these references, regarding the size of data to address for a unique access, depends on the data source and the used design for the extractors. So commonly the access ranges from complete entities of a data source, such as files in a filesystem, to single entries of data sources, such as a single appointment entry in a calendar file. In general any topic that contains a *Subject Identifier*[15] can be used to retrieve referenced data. The Subject Identifier points to the data source and uniquely identifies the part of it that is represented by the topic.

Selecting topics to retrieve data can be performed the same way as marking topic for search queries. First the user navigates to the topic representing the desired data to retrieve. This is done either directly via an appropriate entry point or through the search functionality. With the mark function of the interface then the current topic can be put into a virtual bag of accentuated topics. Having filled this bag, the user can decide about the way the topics are processed.

Accessing data through the Knowledge System

In case the data should be accessed directly on the current platform, the interface presents the subject identifiers of the selected topics to the extractor framework. The framework decides by inspecting the identifier which extractor instance is responsible for the requested data. Then the addressed extractor connects to the data source and retrieves the referenced portion of data. To make the data accessible for the user, it has to be transfered from the extractor component to the client software the user is currently working with.

At this point it can be transformed inside the framework to meet the capabilities of the client's platform. If data should be retrieved because of its informational value, then it is often sufficient to transform it to plain text, which has the advantage of being transferable and displayable efficiently. Examples are documents and entries, required because of included bits of information, such as financial numbers, text quotations or instructions to directly perform a current task. In cases where the data is needed to work with the content in its whole, the transformation would prevent further work on the data with the original application. In this case data should be delivered in its original form. Both options can be visualized by thinking of a calendar entry. Retrieved in plain text, it serves the purpose to present all details about a meeting, so the user is directly informed and can work with this information. However, sometimes it is also important to be able to retrieve the calendar entry in its original format, like for attaching it to an email in the ICAL format. This format then

[15] See topic identification schema in Proposition 3.3

provides the recipient's software the possibility of an automated processing. Giving users the option to choose between plain information and original data format, they do not have to switch between applications and can use a unified interface for search and retrieval, regardless of the current task.

3.6.2 Redirection of Output

Data that can not be displayed on the platform currently used should be made available to other output devices in the surrounding of the user. This has the advantage that the user can decide where to work with the data in a convenient representation, preventing drawbacks in case of a transformation to limited display capabilities. When thinking of common locations where mobile users might would like to access comprehensive data, many output devices already in place should be considered. Of course these scenarios are only applicable for data not considered strictly confidential, since the user can not fully rely on devices outside the own company regarding non-disclosure and authenticity of displayed content. In the end, this is a classical question of trust and law between individuals, not one solvable purely with technology.

Besides these security issues, the main challenges for redirection is an easy and secure identification of the output device and a compatible processing of the data for the output devices. Both can be addressed by involving broker components, specialized on content conversion, data transfer schemes, and service identification and discovery. Therefore, the following examples should visualize the concept that is driven by independent service providers, which will integrate with personal digital knowledge beneficially.

Access via Paper

In case the information to retrieve is only required for reading, but contains complex content not convertible to a convenient result, one option is to transfer it to paper. A common application for this are fax machines or other public output infrastructure such as printer pools. After the extractor has retrieved the data from its original data source, the content can be transmitted to a fitting output device. One common example is the usage of the fax machine at the reception of a hotel the user currently resides in. Having instructed the extractor to transmit the selected data to the corresponding telephone number, the access to important information can be performed spontaneously without the need for sophisticated hardware.

Access via Video

Sometimes the information to retrieve is intended for a bigger audience. In this case an output to a digital projector or a wall mounted display in the conference room is more convenient. As these devices currently do not have processing capabilities that would allow a direct communication of the content, they have to be connected to an intermediate component attached to the Internet. Via this connection the component communicates with the system delivering the content, controlled via the mobile device of the user. This content processor component is responsible for receiving the content, the interaction with the user, and the control of the output device. The interface for this intermediate component receives content in a standardized format, comparable to network printers, extended with access control to the content and remote control functionality. Using a common format for this interface like PostScript would make it as easy to use this type of output device as using network printers. Both, the conversion to this format, as well as its conversion back to a displayable form has reached a very sophisticated level and the integration as remote printing device is provided by almost any operating system.

Access via Substitute

Another way of access is necessary if the content should be also editable in its original format. In case the own client currently used to interact with the knowledge base is not capable of processing the required data, it can be also transfered to another device with enhanced capabilities. The information is then send encrypted from the knowledge base to this device, which by definition belongs to another person. If it would belong to the user, he would simply use his knowledge base directly from that device. With the secret displayed on the own client, the user gets access to the data after it has been received by the other device. Therefore the device plays a substitute role for the own device and identity, provided by a small network component that handles the secure communication for the download and a possible upload back to the data source, after the data has been edited.

This access method sounds similar to distributing data to other recipients described in the next section. However, it should be considered separately since the use case demands different functionalities. The main differences of a substitute access are the accessibility of all context information provided through a direct connection to the knowledge base, and the direct store function, which saves the edited content in a new version to the knowledge base.

3.6.3 Sending to Recipients

Today, sharing information is a common task when working in communication-based jobs. These tasks require distribution of information initiated by the worker. Sometimes this is caused by direct request of the recipient, in other cases the recipient already has acknowledged the interest in receiving communication of that kind for the chosen medium. The addressed class of tasks therefore is categorized by wanted information — explicitly excluding SPAM — directed to only a few recipients, and initiated by spontaneous actions, which therefore contrasts with distribution via a web server.

Having built personal digital knowledge, these tasks can be addressed by providing convenient and secure ways to exchange stored information with other individuals. Of course the sender can not presume a certain technology on receiver side, as that would limit the possible applications for the distribution of information. Hence, it is important to adapt the distribution process to a range of electronic communication technologies, including the most widely used forms, such as emails and content management platforms.

Using Email

Email attachments are a widely used method to distribute content to dedicated recipients. Client software to receive the attachments can be assumed to be available for nearly all recipients, letting this medium appear at first as an ideal way to share information. Having selected the information to distribute, the data is retrieved from its storage and attached to an email addressed to the recipient. The corresponding email address is entered either manually, or taken from the personal digital knowledge itself. In case the email address topic is also associated to a public key, the email can be sent encrypted.

However, attachments do have also disadvantages for sender and recipient that make it interesting to search for alternatives. First of all the usability of emails decrease with the size of the attachments, because of the necessary conversion time during fetching the attachment from the email server. Some servers do also limit the size per email, making it inevitable to split attachments to convenient portions, to ensure a proper processing for all recipients. If errors occur nevertheless, either during processing or reception by intermediate components, there is no secured channel back to the sender application in a standardized way to trustfully indicate a successful reception. In addition, the recipient has to organize the incoming content in his *inbox* manually by storing the attachments to useful folders or locations. Otherwise a search is only possible via the sender name or the date of receiving.

Addressing Content Management Platforms

A second commonly used method is the usage of web-based platform for managing, sharing and exchanging content. Though it is a common way, there are a broad range of different interfaces and the way they are structured ranges from personal storages to enterprise work spaces with support of collaborative demands. Because of a missing standard the direct interaction with them seems to be impracticable.

However, many of these web-based platforms supports the WebDAV[16] standard, which is a set of extensions to the HTTP protocol and allows components to manage files inside the platforms. In this case, a common transport medium and protocol is available. Unfortunately, this does not solve the problem completely, since all platforms do use different ways to organize their content. Therefore an upload to these system requires a set of information, including the URL pointing to the desired storing location and information about credentials for the identity used to upload the content.

This way of distributing content is therefore only applicable if the recipient has provided additional information along with his request. A spontaneous transmission initiated by the owner of the information is not possible. Besides these restrictions, the required effort for a proper configuration on sender and receiver side makes this communication method only an option for senders with frequent reasons for an exchange between the same users.

Direct Dispatch to Personal Digital Knowledge

A logical addition to the usage of transmission and storage standards is the consideration of directly adding the content to the digital knowledge of the recipient. This would have the benefit of a tight interaction between the knowledge representations and the possibility for an increased security through building a fitting security design for the use case. It would consider the value of the transmitted data, as well as the need to protect the recipient from unwanted communication; protecting the recipient from loosing information in the flood of unwanted data.

Since both communication ends are then designed to manage digital knowledge, there is also the opportunity to send context information to the recipient's system along with the transmitted content. Such context could contain valuable additional data for the recipient, which could then be integrated into his digital knowledge. The implication of the distribution such context information regarding benefits and security issues are discussed in the next section.

3.6.4 Context Generation and Distribution

In this section, possible benefits of context generation for the distribution to other digital knowledge systems are discussed and compared to privacy aspects. The extraction of a topic map fragment containing the metadata for the transmitted content can be performed in a most basic way by defining a radius of topics around the one referencing the transmitted content. Starting from this content topic, all contained references (which includes topic types for the radius zero and also topics connected by associations for a radius greater zero) are followed up to the defined

[16] Web-based Distributed Authoring and Versioning, http://webdav.org/

range and the reached topics are then included inside the topic map to be sent along with the content. This would include type topics that could help to sort the incoming content and could also add metadata not reproducible from the content. However, since the source topic map can bring together many data sources and the user is not always aware of the implicit connections that are created for transmitted content, also unintended metadata could be revealed to recipients, which would cause a violation of privacy aspects. A simple example is the sending of a document. It could be referenced by a topic that is associated with a topic representing the subject of an email. The *base name*[17] of this email topic would then be revealed to the recipient, if a radius of 1 is used. Therefore, from the privacy point of view, only context topics should be included in the describing topic map that could be also generated by the recipient, which would make it needless to send them.

A trade-off solution is to use a rule-based topic *inclusion*, in contrast to a rule-based *exclusion* that would have the common blacklist-inherent security problems. Such a whitelist approach specifies topic and association types that are save to be transmitted to foreign recipients. This requires to invest some effort in advance to define those rules. Another option without this effort is to display the contained topics and let the user decide in an opt-in approach which topics to expose to the recipient during the process of sending the content.

However, also the recipient side has to be considered. On this side mainly the trustworthiness of the submitted context is of interest. If any Topic Maps object contained in the received map is merged to the recipient's digital knowledge representation, a malicious sender could create incorrect relations. These relations could interfere with a beneficial usage of the overall system by polluting the recipient's representation with a multitude of incorrect or senseless connections between stored information. In worst case malicious relations could be also exploited for crafted topic references. This could lead to exposing confidential content to third parties, if such a topic is picked by the user to indicate the content to be transmitted without checking the connected content. Then other content is distributed unnoticed instead of the intended.

Another threat are attacks that try to reveal base names containing confidential information (e.g., email or event subjects). Topic type references from a transmitted topic to topics contained in the recipient's representation could create a malicious link to secret information. The process of sending this crafted topic to a recipient could collect the type topics and this way would have revealed also the base names to the recipient. Therefore all topic references contained in received topics that are directed to topics outside of the received topic map have to be removed before the merging process.

Proposition 3.10. *Summarizing the pros and cons for sender and recipients, the distribution of context information is only useful if it contains data that*

[17] A base name is a name or label for a subject, expressed as a string. See Proposition 3.3

- *can not be extracted by the recipient*
- *and does not contain confidential information*
- *and is not connected to other data of the recipient.*

Since the consequences of the aspects of Proposition 3.10 can only be decided individually by the users, a situational approach is suggested for handling the distribution of context information along with the actual piece of data to be shared with the recipient.

3.7 Summary

The intention of the novel concept described in this chapter was to design a generic concept for personal information management to improve the support of users in their daily tasks with personally stored information. The proposed approach is based on creating a digital knowledge representation from existing personal data sources to support the tasks of associative-like searching and accessing information scattered in different data sources as well as securely distributing information to dedicated recipients. The solution concept supports these tasks with the design of a universal interface that interlinks all stored information for a single point of access.

For interlinking data from different sources, the implicit dependency between the user's work with information and his stored data are used to create an information domain. Due to the concept of generic information domains, no assumptions regarding specific data structures or content have to be made. The solution concept is instead based on recurrent information pieces of the same type, created during the individual work of a user. These recurrent pieces are used in this work to create links between data sources. This way a network of links can be created on top of the already managed personal data structures that interconnects descriptive keywords with the entities that contain the actual data entries in the data sources.

The concept of a logic layer created by interlinking on top of the personal data then was further designed to be homogeneously represented with the Topic Maps technology. It is used to preserve structure and content together in a human interpretable form. However, it is not sufficient to only rebuild all structures with topic maps, since then data sources would sill stand isolated. Instead, identifying and unifying relations and content have shown to enrich the representation. The precondition for achieving a generic way to unify heterogeneous data structures was developed in this work by creating generic rules and patterns for arbitrary classes of data structures. With these rules, all data sources' relation types and structures can be mapped to the logic representation layer in a uniform way.

The proposed closed interaction cycle for the representation then demonstrated the benefits of a single unified access to the interlinked data sources. This interaction method showed how to make use of the representation in a content-independent way by using a trade-off between a graph-centric and a node-centric approach. Due to the simple and generic nature of the three staged cycle for the proposed path-centric approach it can be used uniformly on mobile and stationary clients. As intended, it fulfills the first goal of supporting seamless information accessibility across data sources. It supports new associative navigational strategies, described by the possibilities a user can put into practice with the proposed navigation elements on the representation.

Associative searches were addressed in a next step by taking the user's implicit way of organizing his data into account. These individual structures and correlations are preserved in the representation, which can help the user in finding connected data. Instead of having to enter a fitting keyword or manually following a paths to the desired information, the network of relations were used for a search algorithm to cover cases when a user may recall only associated information, such as locations, people, events, projects or any other information type relevant to his managed data. With the help of graph theory, a bidirectional Breadth-first Search algorithm was developed and combined with the navigation concept. This way the interface also is used to let the user choose the input nodes for the search. The different modes created for the search algorithm have shown various application ways beside a strict structural search, like providing searches for equal properties, types or relations. Corresponding examples for the application of the graph search have shown how and when these can expand conventional keyword approaches.

Besides creating new ways for accessing and searching, also the daily management of information have been incorporated into the proposed concept. Not only the possibility for remotely managing the data, but also the application of the concept to forward and display information in fitting alternative forms where discussed. This includes the common requirement of being able to securely distribute information to a closed dynamic group of dedicated individuals — in contrast to distributing it to large, static or open groups.

The question of whether additional context information, taken from the accumulated representation of the server, should also be additionally distributed were also analyzed. Although unquestionable advantages for the efficient interconnection of the transmitted information exist, also disadvantages in the form of security threats for senders and recipients have been identified. The coverage of distributed or received additional context information can only be decided by situation, as long as no model is found that can calculate the complex possibilities of effects on protection goals for senders and recipients.

The developed results have shown with a generic concept how to create a universal interface for daily work with information that is built on top of existing personal data sources. The challenges for IT security are addressed in the next chapter.

Chapter 4

Security Design

The effective protection of ubiquitous personal information management systems is a central aspect for their introduction into productive environments. This chapter describes the specific security design issues that reflect the special demands of the envisaged systems. These high security demands are created by potential risks exposed by the immanent single point of access design: A single interface empowers to transmit any stored data directly, passing through the protection mechanisms that secure the data's original storage system or environment. Compromising this single interface can be assumed to be of vital interest for attackers. Goal of the proposed design aspects is therefore to extend classic security measures of server-based services with additional lines of defense against attacks aiming at compromising and manipulating trusted components.

The starting point for extending the protection of the server-based services is created in Section 4.1 with the security considerations. As a result of analyzing the service's specific assets and threats, the proposed security objectives provide the foundation for the security design on a conceptual level. Since this work describes ubiquitous personal information management systems as a general concept, a specific instance developed from the concept should consider and adapt the security objectives for their security design.

However, the proposed security design should also take implementation flaws into account. Many of the everyday security bulletins clearly proof implementation flaws to be a serious threat to data confidentiality. Currently, the threats are addressed by trying to prevent the execution of injected malicious code, e.g. via the Data Execution Prevention (DEP) offered by Microsoft Windows [Mic06] and other stack-smashing protections [WC03]. However, there are already known circumvention methods [Ric02] and attackers can also introduce return-oriented payload attacks [BRSS08] by some other means than stack overflow such as heap corruption or a format string vulnerability. This way the attacker can still abuse existing code fragments rather than having to inject their own code. Corresponding counter measures like *Address Space Layout Randomization* (ASLR) that break the determinism of jump addresses usable for an attack have shown to be still circumventable [Mül08, SjGM^{+}04], too.

As a further line of defense additional to these protection mechanisms, this work motivates a tightened usage of the operating system's capabilities to prevent exploit code from violating protection goals. However, as long as services are designed as monolithic server architectures, one implementation flaw may endanger all stored data, because the service can not make use of fine grained operation system restrictions to prevent an exploit from manipulating the service. Therefore, an attacker can abuse any of the service's privileges to access arbitrary data. To mitigate this risk, an important objective proposed in this thesis is the separation of critical functions into components isolated by the operating system. The approach for creating separated components from a monolithic service — called *intra-service privilege borders* in this work — is intended to help create a protection that require an attacker to compromise and exploit at least *two* isolated service components to accomplish the mission of the attack successfully. With the contribution in Section 4.2, this novel approach for creating an architecture of isolated service components is proposed that require an attacker to defeat 2 of n components to succeed. As a further contribution, this approach is then used to design the actual isolated service components for ubiquitous personal information management systems (see Section 4.3).

4.1 Security Consideration

This section analyses the security problem space by analogy with Common Criteria (CC) protection profiles. As also other systems use remote devices for accessing information stored on servers, there are already protection profiles that deal with the underlying security aspects. For example the two CC protection profiles *Mobile Synchronisation Services (MSS PP)* [Bun08] and *Operating System Protection Profile (OSPP)* [Bun10] provide sound considerations for the evaluation of remote communication and proper operating system protection. Therefore, this work focuses on the additional specific security aspects on application level. These security aspects arise from the principle of operation that is introduced by the unified access to the complete personal data repository and the resulting increase for the demands in security.

Section 4.1.1 first describes on overview of the proposed generic system architecture to be protected by the security design, followed by the assumptions (Section 4.1.2) made for the environment in which the system is intended to be applied. Based on the functionality the architecture is envisaged to provide, Section 4.1.3 then defines the primary and secondary assets to be considered by the security objectives. Threats to these assets to be countered by security measures are described in Section 4.1.4.

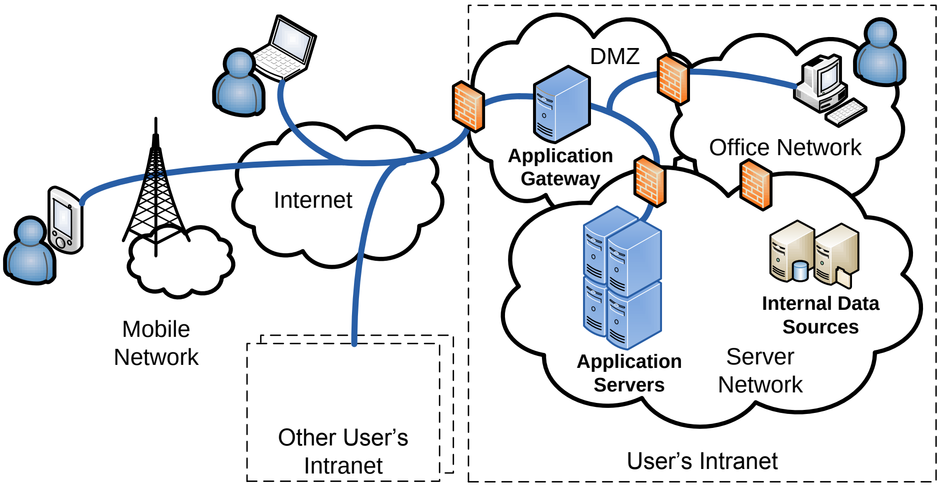


Figure 4.1: Generic System Architecture and Overview of possible Communication Paths

Finally Section 4.1.5 concludes the security objectives to counter the identified threats in the environment, which was defined by the assumptions.

4.1.1 System Architecture Overview

From a generic point of view, the overall system comprises the user's terminals together with the user's intranet and the foreign user's intranet that both may receive and send information upon requests of the user and the foreign user, respectively. The intranet by itself typically is divided into different networks for work stations, Internet servers in a demilitarized zone (DMZ) and further protected server networks. The resulting generic system architecture is shown in Figure 4.1. In this setup, it is common to introduce an application gateway in the DMZ to additionally protect the application servers. As a result, the blue lines indicate the communication paths between the entities that have to be considered for protection.

4.1.2 Assumptions

The following assumptions describe the security aspects of the environment in which the system is envisaged to be used.

Trusted Operator

The operator of the system is trusted to correctly initialize, configure, operate and decommission the service as intended.

Unshared Resources

No components of additional applications are installed on servers running the service on the application servers.

Intranet Security

User's intranet is operated under best practice security policy by the operator, such as separating the network sufficiently from Internet in a restrictive way (e.g., by firewalls).

Trustworthy Sources

The database services and repository sources added and used by the knowledge representation are trustworthy.

Trustworthy Operating System

The application server relies upon the trustworthy operating system to provide non-tampered functions such as file protection, domain separation, time stamps, non-bypassability and operating system user authentication.

Trustworthy Terminals

The client application for devices and browsers relies upon the trustworthy terminal platform (hardware and operating system) to provide data protection, domain separation and non-bypassability.

4.1.3 Assets

The assets are separated into primary and secondary assets to distinguish between data directly involved on service level and those involved in the protection of the service, respectively. The primary assets to be protected by the security design are shown in Table 4.1 together with their generic protection goals. These goals have to be protected by the security design to secure the usage of the service for the involved data. The same applies for the secondary assets shown in Table 4.2.

Table 4.1: Protection Goals of Primary Assets

Primary Asset	Definition	Generic protection goals
Knowledge Representation Data and Indexes	Content of user's data repositories, its interconnection and interpretation exchanged between the user's and foreign application servers or transfered to user's devices	Confidentiality Integrity Authenticity
Remote Control Commands	Instruction sequence issued by the user's devices to interact with the knowledge representation or to initiate an exchange with other users	Confidentiality Integrity Authenticity
Remote Menu Entries	Navigational elements extracted from the knowledge representation to visualize parts of available stored information	Confidentiality Integrity Authenticity
Data Source Credentials	Cryptographic material stored in the application server to access the user's databases	Confidentiality Integrity Authenticity
User-related Traffic Data	Data directly or indirectly indicating users (e.g., names, addresses, IDs, etc.)	Anonymity

4.1.4 Threats

The threats described in this section have to be countered by the security design independently or in collaboration with its operational environment. These threats result from the assets processed by the service and the method of the service how it uses them in the operational environment.

Channel Disclosing Forging

An attacker discloses or modifies knowledge representation data, remote control commands, remote menu entries or data source credentials while being transmitted between the user's terminal and the application server.

Personal Traffic Analysis

An attacker discloses user-related traffic data while being transmitted between a user's terminal and the application server.

Table 4.2: Protection Goals of Secondary Assets

Secondary Asset	Definition	Generic protection goals
Service immanent cryptographic keys	Cryptographic material used by the system to enforce its security functionality between components. The keys are kept on devices and application servers and can be exchanged between them over a trusted channel	Confidentiality Integrity Authenticity
User authentication data	Cryptographic material to authenticate legitimate users	Confidentiality Integrity Authenticity
User's cryptographic public keys	Cryptographic material to provide protected communication with other systems	Integrity Authenticity
Genuineness of system	Property of the system components to be authentic in order to provide proper security functionality	Integrity
Accessibility to administrative functions and data only for authorized subjects	Property of the system components to restrict administrative access to authorized subjects only	Availability

Channel Capture

An attacker masquerades as a valid service component by capturing the communication channel between the user's terminal and the application server in order to disclose or modify primary assets.

Service Misuse

An attacker misuses the service in order to gain access to the administrative functions or protected data of other users. Such an access can occur through the external interfaces of the user's terminal.

Service Manipulation

An attacker affects the genuineness of service components by modifying the executed code or security relevant data.

Information Leakage

An attacker exploits information that is leaking from the service while a user interacts with the service, in order to disclose confidential data (service immanent cryptographic keys or user authentication data). The information leakage may be inherent in the normal operation or caused by the attacker.

Terminal Manipulation

An attacker interferes with the software or hardware of the user's terminal in order (i) to disclose or to modify primary assets, or (ii) to weaken the security functions by manipulating code or by direct physical probing on secondary assets.

4.1.5 Security Objectives

As a result of the identified assets and threats, in this section the threats are countered by describing security objectives to be addressed by the security design of the service.

Channel Protection

The communication between the user's terminal and the application server should be considered a channel. This channel should protect transmitted data from one end to its other end against disclosure and modification. Only this channel should be used to exchange interface navigation data, remote control instructions, cryptographic keys and device management data between the user's terminal and the application server.

Traffic Anonymity

Network traffic is generated between the user's terminal, his application server and the recipients of the user's transmissions. This traffic should not contain application data that directly or indirectly would enable an intermediate party to learn user-related traffic data such as names, addresses and IDs. On application layer, this objective ensures traffic anonymity of all interactions with other users.

Channel Authenticity

All parts should verify the authenticity of the components the communication is established with. Any violations of authenticity for the established channel has to be detected to prevent a non-trusted channel. This way the authenticity of primary assets is ensured during its transmission within the channel.

Authorized Access

The service shall only provide access to administrative functions for authorized subjects. Authorized subjects are the service user and the system administrator, as well as a components of the underlying operating system. This access can be provided through the human interface of the service or through additional logic interfaces of the service.

Access Function Limitation

Even with administrative access, the authorized subjects should not be able by provided functionality to gain access to other users assets, to prevent exploitation of administrative access. The knowledge representation data and data source credentials shall be kept non-accessible, protected by user-defined secrets. User authentication data shall be stored as salted hashes to prevent exploiting the data for authenticating as another user.

System Integrity

A function to verify the integrity of all service components shall protect the integrity of cryptographic operations. This way simple modifications to those components that could weaken the protection shall be prevented.

Assignment Protection

The assignment is the process of linking an identity account to a specific knowledge representation instance. The assignment shall be protected cryptographically against modifications to ensure the interaction with the correct knowledge base for all users. This way, swapping an account of one assignment with another one shall not provide a working assignment without the interaction of both related users.

Ad-Hoc Key Exchange

An ad-hoc key exchange between mobile devices (the user's mobile terminal) is required for a secure spontaneous distribution of information to another user. The key exchange shall be easy and fast to perform, as otherwise the advantages of sent information out of the current situation are foiled and the user may be inveigled to use another unprotected communication. In addition, the key exchange shall detect manipulations in the used communication path to prevent an attacker from influencing the exchange.

Fortified Attack Resistance

Exploiting one implementation flaw in the service shall not provide an attacker the ability to violate (i) the confidentiality and integrity of stored and transmitted information, (ii) the confidentiality of cryptographic keys for data sources, (iii) the integrity and authenticity of remote commands and (iv) the integrity of indexes.

4.1.6 Results

Except for the last objectives, existing security technologies can create a fitting protection with approved and efficient algorithms and implementation principles. Regarding the objective *fortified attack resistance*, there is ongoing research targeted on improving the protection against unauthorized modifications of program flow and its detection, as discussed in the related technology chapter (see Section 2.3). As this work deals with the specific security implications of ubiquitous personal information management systems, the following sections introduces novel approaches for these challenges.

4.2 Intra-Service Privilege Borders (ISPB)

From a security perspective, operating systems offer the anchors for application security by enforcing restrictions to its resources such as file storage, memory, computing power and network. By granting privileges to an application, it can only operate inside the borders controlled by the operating system. However, if the application gets compromised by an attacker, he can abuse all given privileges, which makes it important to separate the functionality into distinct components to reduce the attack's potential in case of an exploitable implementation flaw. With functionality separated into distinct components, privileges can be tailored to the reduced requirements of each component, limiting the possibilities for an attacker of that component. However, without further protection mechanisms, the attacker would be still able to steal or alter data by changing the application's functionality inside the given privileges.

The next section introduces a approach that should help to design a secured architecture in a structured way that counteracts the aforementioned threats. The intention of the proposed *intra service privilege border* (ISPB) approach is the protection at runtime of processed data, even if one service component is completely controlled by an attacker. The idea for the underlying approach is comparable to the four-eye principle. It is applied in this work to guard the service components in case of a compromising attack. After describing the generic approach in the next section, the approach is applied in Section 4.3 to protect personal information management systems against actions of compromised components that would violate the protection goals.

4.2.1 Approach Principles

The precondition for the ISPB approach is a monolithic server-based service. It is assumed that its security design already sufficiently protects the defined assets against attacks on the logic and communication level. From this starting point, the approach describes a process for splitting up the monolithic service into multiple components. A component in this approach is an isolated executable with its own protected memory, preventing unauthorized interfering with the component's functionality. The goal of the approach is to strengthen the protection against attacks that compromise and manipulate service functionality by exploiting implementation flaws.

The approach is based on the following two principles:

- The separation of service functionality into components to reduce required privileges per component.
- Relevant operations for protection goals are mutually guarded by disjoint components to prevent exploiting a single point of attack.

Similar to the four-eye principle, the approach should enforce that at least two components have to be compromised to overcome the additional protection. This is considered a second line of defense, since compromising two independent components is considered more attack effort in terms of required knowledge, resources and attack concealment, which reduce in consequence the risks of successful attacks.

4.2.2 Approach Steps

The creation of the proposed protection architecture by the generic ISPB approach is expressed by pseudo code shown in Algorithm 4.1. The code is structured by labels for each step. Sub-functions are indicated by parenthesis enclosing assigned parameters. The number of square brackets pairs indicates the dimension of arrays and the characters between the brackets specifies the variable containing the associative index value for the arrays. The approach's mandatory four steps with the described sub-functions are motivated and explained in the following section.

Step 1 - Asset's Protection Goals: The starting point for this first step of the process shown in Algorithm 4.1 is a monolithic service that is already protected against attacks that work from the outside of the hosting server (see security considerations in Section 4.1). This service is then considered for the adaption process of the approach (Algorithm 4.2).

The step then inspects the possible attack targets of the service. For the second line of defense, all assets have to be identified that should be protected even if one component gets compromised (Algorithm 4.3). This includes assets that

Algorithm 4.1 Creation of Intra Service Privilege Borders

```
Step_1:
Service ← Get_Service_to_protect( );
Assets[] ← Identify_Assets(Service)
for each a in Assets[] do
    Protection_Goals[a][] ← Choose_Protection_Goals(Service, a)

Step_2:
Components[] ← Split_Functionality_into_Components(Service)
for each c in Components[] do
    Privileges[c][] ← Define_least_Privileges(c)
Enforce_with_OS(Privileges[][])

Step_3:
for each c in Components[] do
    for each a in Assets[] do
        if Asset_is_accessible_by_Component(a, c, Privileges[c][]) then
            for each g in Protection_Goals[a][] do
                Protection_Matrix[c][a][i++] ← g

Step_4:
for each c in Components[] do
    for each a in Assets[] do
        if Asset_is_accessible_by_Component(a, c, Privileges[c][]) then
            Operations[c][] ← get_Operations_with_Asset(a, c)

            Guarding_Operations[c][] ←
            match_Operations_to_Guard_Operation(Operations[c][], Protection_Matrix[c][a][],
            Guarding_Mapping_Table[][])
Guarding_Components[][] ← specify_Guarding_Components(Guarding_Operations[][],
Components[])
```

need to be protected additionally if a component would be compromised and assets that itself have a protective functionality.

For these assets the corresponding protection goals for compromised components are then aligned with those protection goals against outside attacks (cf. Section 4.1.3). Therefore, in this approach for each asset only those of the protection goals deemed necessary against outside attacks are considered that should also stay valid in case of a compromised component (Algorithm 4.4). The algorithm therefore chooses for each identified asset from the protection goals *confidentiality*, *integrity*, *availability*, *authenticity* and *anonymity*. Of course the dependencies between assets and their protection goals have to be considered for these decisions. For example, the confidentiality of cryptographic keys is in many situations a precondition for the confidentiality of encrypted assets. Therefore each asset have to be inspected for these dependencies of the considered service functionality.

Algorithm 4.2 Sub-Function: get_Service_to_protect()

```
function GET_SERVICE_TO_PROTECT
    service ← service already designed to withstand external attacks
    return service
```

Algorithm 4.3 Sub-Function: identify_Assets()

```
function IDENTIFY_ASSETS(Service)
    for each asset in Service do
        if PROTECT_FOR_COMPROMISED_COMPONENTS(asset) or IS_PROTECTIVE_ASSET(asset) then
            assets[] ← asset
    return assets[]                    ▷ set of assets to be protected in a second line of defense.
```

Step 2 - Service Segmentation: The goal of the approach's second step is to reduce the further effort for an additional protection of assets. There are only two possible attack vectors that an attacker could use with compromised components to violate protection goals of assets: (i) by abusing *intended*[1] access privileges to assets or (ii) by accessing them via other resources in *not intended* ways. The second attack vector is removed with this step. By applying operating system privilege on the separated components, assets not used by a component are no longer accessible, even in case the component gets compromised. Therefore only the remaining attack vector (i) — the abuse of intended access of components — has to be counteracted in step 3 by introducing guarding components.

Removing the attack vector (ii), the second step of Algorithm 4.1 splits the service to protect (see Algorithm 4.5) into components regarding their required resources by ensuring that any component only requires one system external resource (e.g., network, file storage, database, etc.). For example, a component with access to the network than does not have the privileges to access the file storage nor the database. Otherwise the operating system cannot prevent a compromised component in this approach from maliciously transferring data between resources, as the operation system has to grant access to all resources for the intended actions of the component. Instead by separating resource access, the operating system can limit each component's resource access separately by the least privilege paradigm. These least privileges are defined in the approach by Algorithm 4.6. The restrictions resulting from the least privilege paradigm prevent the access of a component to resources it is not intended to use. A single compromised component cannot bypass these defined restrictions as they are enforced by the operating system.

However, besides separating the resource access privileges, it is required to prevent components from influencing other components. Therefore the ap-

[1] It is intended because the access is required for the functionality of the component.

Algorithm 4.4 Sub-Function: choose_Protection_Goals()

```
function CHOOSE_PROTECTION_GOALS(Service, Asset)
    for each g in ['Confidentiality', 'Integrity', 'Availability', 'Authenticity', 'Anonymity'] do
        if SHOULD_BE_PROTECTED(Service, asset, g) then
            protection_goals[] ← g
    return protection_goals[]    ▷ these should be covered even in case of compromised components
```

proach also requires global restrictions of component privileges to ensure that all separated components have

- no write access to executables of other components, to prevent modification or exchange of executables,
- no access to memory of other components, to prevent stealing of assets and in-memory component modifications,
- allow only one external resource per component, to prevent attacks that bridges information between resources.

These combination of the individual privileges together with the global restrictions are then enforced by the operating system in Algorithm 4.7.

Algorithm 4.5 Sub-Function: split_Functionality_into_Components()

```
function SPLIT_FUNCTIONALITY_INTO_COMPONENTS(Service)
    for each function in Service do
        for each resource in function do
            functions_with_only_one_resource[] ← SPLIT(function, resource)
    components[] ← MERGE_FUNCTIONS_WITH_SAME_RESOURCE(functions_with_only_one_resource[])
    return components[]
```

Algorithm 4.6 Sub-Function: define_least_Privileges()

```
function DEFINE_LEAST_PRIVILEGES(Component)
    least_privileges ← DENY_ALL(Component)
    for each external_resource in Component do
        if ACCESS_REQUIRED(Component, external_resource) then
            least_privileges ← ALLOW(external_resource, Component)
    return least_privileges
```

Algorithm 4.7 Sub-Function: enforce_with_OS()

```
function ENFORCE_WITH_OS(Privileges[Components][Rules])
    for each component in Privileges[Component][] do
        OS.ENFORCE(component, "no write access to executables of other component")
        OS.ENFORCE(component, "no access to memory of other components")
        OS.ENFORCE(component, "allow only one external resource per component")
        OS.ENFORCE(component, Privileges[component][])
```

Step 3 - Component Guarding: The next modeling step is to identify for each component all assets that it is working with. To achieve this goal, the sub-function shown in Algorithm 4.8 decides whether an asset is accessible by a component with its given privileges. This is the case if (i) the component itself processes the asset in any way or (ii) the component has access to the asset via external resources. After identification of the assets per component that have to be considered for protection, for each component the corresponding asset's protection goals — identified in step 1 — are collected for the component. Each of these protection goals per component have to be guarded in the next step.

Algorithm 4.8 Sub-Function: is_accessible_by_Component()

```
function IS_ACCESSIBLE_BY_COMPONENT($Asset$, $Cmp$, $Privileges$[])
    if ASSET_PROCESSED_BY($Asset$, $Cmp$) or HAS_ACCESS_PRIVILEGES($Cmp$, $Asset$, $Privileges$[]) then
        return True
    else
        return False
```

Step 4 - Protection Mapping: In the final step, for each identified protection goal of an asset that is accessible by a primary component, another existing or additional guardian component is instructed to enforce the identified protection goals, for the case that the primary component gets compromised. The guardian component is therefore intended to prevent an attacker from violating protection goals of protected assets by a compromised component. If guardian components are applied this way for all protection goals of assets, an attacker has to compromise at least two components to accomplish an attack that could violate the guarded protection goals.

The starting point for this step is to collect the operations for each component that involves a specific asset. This involvement is checked by the sub-function in Algorithm 4.9. These operations are then matched against the mapping for guardian components. In the presented approach the five generic protection goals confidentiality, integrity, availability, authenticity and anonymity are considered for the mapping. Table 4.3 shows how to apply the guardian components for each asset to protect. This *Guarding Mapping Table* is used as a look-up for required guarding operations. The table is applied in Algorithm 4.10 to build a list of guard operations required to enforce the given protection goals. Then Algorithm 4.11 maps the guard operations to existing or new components.

Lets consider an example for the usage of the Guarding Mapping Table: If one of an asset's protection goal is *confidentiality*, then the entry of the first row in Table 4.3 shows that it has to be covered by a *control of the component's output (CCO)*. To explain this requirement, the table's middle column motivates the operation of the primary component and introduces related malicious actions in case of the component gets compromised. As a result, a guarding component can prevent the malicious action by following the rules of the third column. In the case of the given example, this would be the introduction of a guarding component that is capable of controlling the output of the primary component with the help of the operating system. If the primary component cannot leak data to the unprotected environment, then compromising of that component cannot violate confidentiality of the corresponding asset. This of course presumes that the rules for creating isolated components introduced in step 2 have been applied for all components.

Algorithm 4.9 Sub-Function: get_Operations_with_Asset()

```
function get_Operations_with_Asset(Asset, Component)
    operations[] $\leftarrow$ getOperations(Component)
    for each operation in operations[] do
        if asset_involved(operation, Asset) then
            asset_operation[] $\leftarrow$ operation
    return asset_operation[]
```

Algorithm 4.10 Sub-Function: match_Operations_to_Guard_Operation()

```
function match_Operations_to_Guard_Operation(Operations[], Protection_Goals[], Guarding_Mapping_Table[][])
    for each op in Operations[] do
        if PG_matches(Guarding_Mapping_Table[op][protection_goal], Protection_Goals[]) then
            guard_component_operations[] $\leftarrow$ Guarding_Mapping_Table[op][guard_operation]
    return guard_component_operations[]
```

Algorithm 4.11 Sub-Function: specify_Guarding_Components()

```
function specify_Guarding_Components(Guarding_Operations[][], Components[])
    for each component_to_protect in Guarding_Operations[][] do
        for each guard_op in Guarding_Operations[component_to_protect][] do
            components_with_operation[] $\leftarrow$ find_components_with_op(guard_op, Components[])
            for each c in components_with_operation[] do
                if c $\neq$ component_to_protect then          ▷ look for independent components
                    guarding_components[c][] $\leftarrow$ guard_op       ▷ use component for guard operation
                    use_existing_component $\leftarrow$ true
                    break
            if use_existing_component $\neq$ true then
                new_guard_component $\leftarrow$ define_new_Component(guard_op)
                Privileges[c][] $\leftarrow$ Define_least_Privileges(new_guard_component)
                Enforce_with_OS(Privileges[][])
                guarding_components[new_guard_component][] $\leftarrow$ guard_op
    return guarding_components[][]
```

Table 4.3: *Mapping Protection Goals to Guardian Components*

Protection Goal for Asset	Operation of primary Component with Asset	Operation of Guardian Component
Confidentiality	e.g. accessing, passing through or encrypting / decrypting asset to protect *(component could learn secrets that could be leaked by compromised component; in case of encryption operations a compromised component could also change the key material to leak information)*	**(CCO) Control of Component Output** has to ensure together with OS that output of primary component cannot leak data to the unprotected environment. This includes verifying the integrity of used keys for encryption operations
Integrity	e.g. hashing / verifying integrity of asset to protect *(component could produce fake hash value or fake validity)*	**(VIC) Verification of Integrity Correctness** has to reproduce hash from independent source
Availability	e.g. performing computation or providing access on asset to protect *(component could refuse functionality)*	**(SIF) Stand-In Functionality** has to provide equivalent functionality of primary component as replacement
Authenticity	e.g. signing / verifying authenticity of asset to protect *(component could sign malicious data or fake authenticity)*	**(ACA) Authenticity of primary Component Assets** has to verify authenticity of assets based on neutrally received data
Anonymity	e.g. replacing identifiers or generating non-relatable identifies for assets to protect *(component could violate anonymity by providing a relatable identity for asset)*	**(VIA) Verification of Intact Anonymity** has to validate anonymity function of primary component

4.3 Application of ISPB Approach

In addition to the protection against attacks from outside the server (cp. Section 4.1.5), the described generic approach of the previous section is applied in the following to protect the service side of the ubiquitous personal information management system against exploiting compromised service components.

4.3.1 Applying Step 1 : Asset's Protection Goals

Applying the approach starts by choosing a protected service design for ubiquitous personal information management, as it is described by the generic security considerations in Section 4.1. As input for the decision about the assets to be protected in a second line of defense, therefore, the primary and secondary assets listed in Table 4.1 and Table 4.2 are used. With Algorithm 4.3 then the assets *knowledge representation data and indexes*, *remote control commands*, *data source credentials*, *service immanent crypto-*

graphic keys, *user authentication data* and *user's cryptographic public keys* are chosen to be considered for the protection of a second line of defense. For these assets then the corresponding protection goals are defined with Algorithm 4.4. This way, in step 1 the following additional protection goals are defined that should not be violated in case of a compromised service:

- confidentiality and integrity of *knowledge representation data*
- confidentiality and integrity of *service immanent cryptographic keys*
- confidentiality and integrity of *user authentication data*
- confidentiality of *data source credentials*
- integrity of *user's cryptographic public keys*
- integrity and authenticity of *remote control commands*
- integrity of *indexes*

Note that these protection goals are chosen as a tradeoff between system complexity and security impact for users. A compromised component still may violate the goals of anonymity or availability, but the complexity rise required to compensate all identified threats also in a second line of defense would increase the likelihood of implementation flaws. This would neutralize the security gain for confidentiality, which was deemed of higher importance for the use cases in personal information management systems. As a result of the decision *is_protective_asset()* in Algorithm 4.4, the consideration of the protection goals for cryptographic keys and remote commands results from the dependencies to the protection goals for transmitted knowledge representation data. These assets are directly connected and a violation of their associated protection goals would directly empower an attacker to circumvent protection mechanism.

4.3.2 Applying Step 2 : Service Segmentation

For the second step the resources *(n)etwork*, *(k)ey database*, *(i)ndex* and *(u)ser's databases* are identified. Before service separation, the service contains the following functions which require the listed resources in parenthesis:

- User interface logic with user's terminal, (k,i,n)
- Index and relate personal information, (k,i,u)
- Distribute information, (k,i,n,u)
- Receive information, (k,i,n,u)

These functions are split with the help of Algorithm 4.5 into multiple functions that only require one resource. Then, functions requiring the same resource are merged to the following components (named A-F):

Component A: Information Handler performs the logic of the service. It creates and manages the user interface and offers functions for information distribution to other users. It only requires file access to the personal information index database.

Component B: Key Manager manages all cryptographic keys for the service:

- service immanent cryptographic keys
- user authentication data
- data source credentials
- foreign user's cryptographic public keys

The keys are send to component A, C, E and F when receiving a corresponding command via D. The component only requires file access to the protected key store.

Component C: Information Retriever retrieves the actual information from a specified data source and passes it to component A for further processing. The component only requires read access to the participating data sources.

Component D: Network Manager manages the communication with any remote component. Incoming data is passed to the components A, B, C, E and F. This component only requires access to the network.

Component E: Information Writer stores new received information into the corresponding data source. Therefore it only requires write access to the participating data sources.

Component F: Index Manager performs the indexing of metadata gathered from the personal information stored in data sources. The index data is forwarded to component A for further processing. The component F only requires read access to the participating data sources.

With the sharing of functionality across the service components A to F, for each component it is defined which component it requires to interact with. The components preliminary interaction is visualized in Figure 4.2. The arrows in the figure indicate the permitted communication direction between components. Other interactions have to be prevented by the operating system. Based on this service segmentation, the least privileges are derived for each component by Algorithm 4.6. The resulting privileges are shown in the right column of Table 4.4. As the outcome of this step,

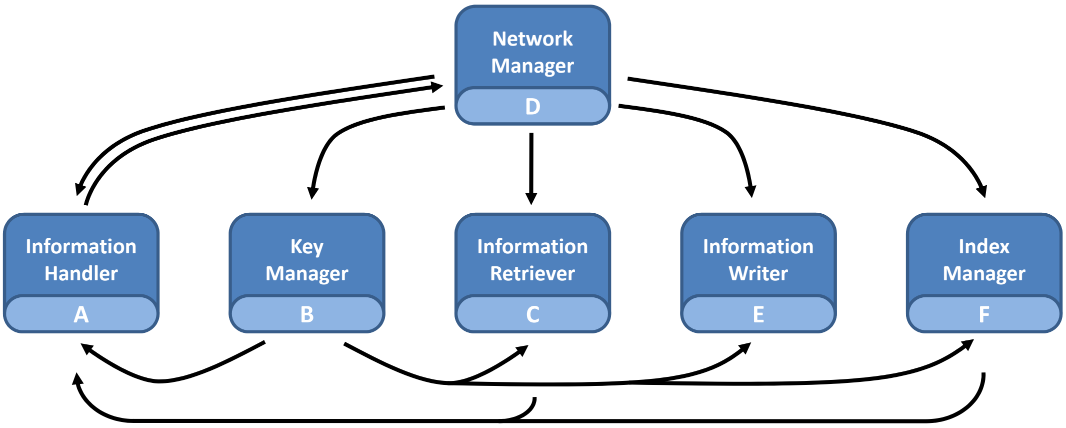


Figure 4.2: *Preliminary Interaction Graph between Service Components*

Table 4.4: *Preliminary Component Functionality and Privileges*

Comp.	Function	Privileges for Resources
A	Interface creation and information distribution	Deny: all Allow: Outbound Communication with D Allow: Inbound Communication from B, C, D, F Allow: Access to Index Store (RWA)
B	Manage personal keys for data sources	Deny: all Allow: Outbound Communication with A, C, E, F Allow: Inbound Communication from D Allow: Access to Key Store (RWA)
C	Retrieve information to be distributed	Deny: all Allow: Outbound Communication with A Allow: Inbound Communication from D, B Allow: Access to User's Databases (R)
D	Manage communication with external / non-server components	Deny: all Allow: Outbound Communication with A, B, C, E, F Allow: Inbound Communication from A Allow: Access to Network
E	Store received information	Deny: all Allow: Inbound Communication from D, E Allow: Access to User's Databases (W)
F	Crawl and index personal information	Deny: all Allow: Outbound Communication with A Allow: Inbound Communication from D, B Allow: Access to User's Databases (R)

these privileges have to be enforces by the operating system together with the general restrictions explained in the description of Step 2 (see Algorithm 4.7). File access privileges are granted for the actions *read (R)*, *write (W)* or *append (A)*. The intra component access is granted for *inbound* and *outbound* communication.

4.3.3 Applying Step 3 : Component Guarding

For the application of the generic approach, in Step 1 the following assets *knowledge representation data and indexes*, *remote control commands*, *data source credentials*, *service immanent cryptographic keys*, *user authentication data* and *user's cryptographic public keys* were identified to be considered for the protection of a second line of defense. The corresponding protection goals of these assets — also identified in Step 1 — are now collected for each component. The result is shown in Table 4.5. To ease the usage in Step 4, the IDs for each required guarding aspect are created by the following schema:

[component name].[asset number].[operation of guarding component]

These IDs are then used in Step 4 to mark the required guarding operations for each of the component's operations and to match actions to a component that actually performs the guarding of an operation.

Table 4.5: Required guarding for Assets of primary Components

Comp.	Assets	Protection Goal	Required Guard ID
A	Indexes	Integrity	A.1.VIC
	Knowledge Rep. Data	Confidentiality, Integrity	A.2.CCO, A.2.VIC
	Cryptographic Keys	Confidentiality	A.3.CCO
	Remote Commands	Authenticity, Integrity	A.4.ACA, A.4.VIC
B	Cryptographic Keys	Confidentiality, Integrity	B.1.CCO, B.1.VIC
	Remote Commands	Authenticity, Integrity	B.2.ACA, B.2.VIC
C	Knowledge Rep. Data	Confidentiality, Integrity	C.1.CCO, C.1.VIC
	Cryptographic Keys	Confidentiality	C.2.CCO
	Remote Commands	Authenticity, Integrity	C.3.ACA, C.3.VIC
D	Encrypted Communication Data	None	D.1.none
	Remote Commands	Authenticity, Integrity	D.2.ACA, D.2.VIC
E	Knowledge Rep. Data	Confidentiality, Integrity	E.1.CCO, E.1.VIC
	Cryptographic Keys	Confidentiality	E.2.CCO
	Remote Commands	Authenticity, Integrity	E.3.ACA, E.3.VIC
F	Knowledge Rep. Data	Confidentiality, Integrity	F.1.CCO, F.1.VIC
	Indexes	Integrity	F.2.VIC
	Cryptographic Keys	Confidentiality	F.3.CCO
	Remote Commands	Authenticity, Integrity	F.4.ACA,F.4.VIC

4.3.4 Applying Step 4 : Protection Mapping

In the final step for each component all operations are specified that it performs with the assets. Then for each operation a guarding is mapped with the Guarding Mapping Table (see Table 4.3). This mapping is shown in the following, based on the required guarding identified for each component in step 3 (see Table 4.5). The guard IDs of this table are used in this step to systematically build a second line of defense. For each of these guard IDs a fitting guarding operation has to be found. The result of this process is presented in a summary for each component. Table 4.7-4.11 show the result in the second column by listing the guarding operation's type (matching the assets ID), the guarding component's name and the specified guarding operation. The following paragraphs describe the found matching of guarding operations to components.

(B) Key Manager Guarding

For component B, the operations with the assets *Cryptographic Keys* and *Remote Commands* have to be guarded by an independent component. As shown in Table 4.6, the B.2.ACA and B.2.VIC guard for the remote commands does not need an additional component. It can be performed by the already existing component D: The incoming command data received by D is signed and its payload is encrypted. Before passing the data to addressed components for decrypting, the signature is verified by D.

The guard for B.1.CCO is done by preventing B from communication with untrusted components — preventing the leaking of keys. However, to prevent B from sending manipulated encryption keys (e.g. keys known to the attacker) to other components, a new components is created by Algorithm 4.11 to cover B.1.VIC. This new component G is the guardian component for B. It is named *Key Guard* and added to the list of components. Its introduced guarding operation stores hash values of all keys. This way components that use the keys from B for encryption can check if the provided keys have been tampered with.

The component B itself is used to act as guard of ACA and VIC for D: B detects modification of remote commands send by D by verifying the signature of all commands.

(A) Information Handler Guarding

For component A, the operations with the assets *Indexes*, *Knowledge Representation Data*, *Cryptographic Keys* and *Remote Commands* have to be guarded. In this case, the additional protection can be performed by the existing components D and C (see Table 4.7) . The component A itself is used to act as guard with the following operations:

- Guard ACA and VIC for D: A detects modification of remote commands send by D by verifying the signature of all commands.
- Guard CCO for B, C, F: A is the only possibility for guarded components to communicate with the environment outside the protected server.
- Guard VIC for C and F: When sending information, A checks that the information's independent hash values of C and F match.
- Guard VIC for B: A verifies hashes of encryption keys by checking it against the hashes provided by G

(C) Information Retriever Guarding

For component C, the operations with the assets *Knowledge Representation Data*, *Cryptographic Keys* and *Remote Commands* have to be guarded. In this case, the additional protection can be performed by the existing components D and A (see Table 4.8) . Component C itself is used to act as guard with the following operations:

- Guard ACA for D: C detects modification of remote commands send by D by verifying the signature of all commands.
- Guard CCO for B: C guards the communication originated from B to C.
- Guard CCO for A: C guards the communication originated from A via D.

(D) Network Manager Guarding

For component D, the operations with the assets *Encrypted Communication Data* and *Remote Commands* have to be guarded. In this case, the additional protection can be performed by the existing components A-G (see Table 4.9) . Component D itself is used to act as guard with the following operations:

- Guard ACA for A,B,C,E,F,G: verifies authenticity of commands and prevents compromised components A,B,C,E,F,G from accepting incorrect signatures.
- Guard VIC for A,B,C,E,F,G: detects modification of commands by external entities and prevents compromised components A,B,C,E,F,G from accepting invalid commands.
- Guard CCO for A: D is the only possibility for A to communicate with unprotected environment. A is not allowed to choose destination for communication. Instead D uses communication end-point contained in initial remote control command. D forwards encrypted outbound data from A to C. C decrypts it

Table 4.6: Protection Mapping for Component B - Key Manager

Operations of component B on assets with their guarding IDs	Guardian Component and its protection operation
receives and *verifies* signature of **remote commands** (B.2.ACA, B.2.VIC)	(ACA): D → verifies authenticity of received commands. (VIC): D → forwards only valid commands.
sends **keys** (B.1.CCO, B.1.VIC) to A,C,E,F	(CCO): A,C,E,F → prevents communication to alternative destinations by placing OS restrictions on B (VIC): A via G → A verifies hashes of encryption keys by checking it against the value provided by G

Table 4.7: Protection Mapping for Component A - Information Handler

Operations of component A on assets with their guarding IDs	Guardian Component and its protection operation
receives and *verifies* signature of **remote commands** (A.4.ACA, A.4.VIC)	(ACA): D → verifies authenticity of received commands. (VIC): D → forwards only valid commands.
receives **keys** (A.3.CCO) from B and *receives* **Knowledge Rep. Data** (A.2.CCO) from C	(CCO): D → prevents communication to alternative destinations by placing OS restrictions on A
receives new **indexes** (A.1.VIC) from F	(VIC): C → hashes verified by C before information is send out
send encrypted **Knowledge Rep. Data** (A.2.CCO, A.2.VIC) to D for communication with external components	(CCO): D → prevents communication to alternative destinations by placing OS restrictions on A. A is not allowed to choose destination for communication. Instead D uses communication endpoint contained in initial remote control command (VIC): C via D → receives encrypted outbound data from A via D. C decrypts it with the keys from B to verify that A has not altered or appended data. C authorizes D to send the data if metadata and hashes comply with own values.

Table 4.8: Protection Mapping for Component C - Information Retriever

Operations of component C on assets with their guarding IDs	Guardian Component and its protection operation
receives and *verifies* signature of **remote commands** (C.3.ACA, C.3.VIC)	(ACA): D → verifies authenticity of received commands. (VIC): D → forwards only valid commands.
receives **keys** (C.2.CCO) from B	(CCO): A → prevents communication to alternative destinations by placing OS restrictions on C
retrieves requested **Knowledge Rep. Data** (C.1.CCO, C.1.VIC) from data sources and *sends* it together with hashes to A	(VIC): A → compares metadata and hash of retrieved data with independent values provided by F (CCO): A → prevents communication to alternative destinations by placing OS restrictions on C

with the keys from B to verify that A has not altered or appended data. C authorizes D to send the data if metadata and hashes comply with own values.

Table 4.9: *Protection Mapping for Component D - Network Manager*

Operations of component D on assets with their guarding IDs	**Guardian Component and its protection operation**
receives and *verifies* signature of **remote commands** (D.2.ACA, D.2.VIC)	(ACA): A,B,C,E,F → recheck authenticity of forwarded commands (VIC): A,B,C,E,F → only processes valid commands
forwards incoming **encrypted communication data** (D.1.none) to A,B,C,E,F	no guard necessary

(E) Information Writer Guarding

For component E, the operations with the assets *Knowledge Representation Data*, *Cryptographic Keys* and *Remote Commands* have to be guarded. In this case, the additional protection can be performed by the existing components D and C (see Table 4.10) . Component E itself is used to act as guard with the following operations:

- Guard ACA for D: detects modification of commands by D or external entity
- Guard CCO for B: guards the communication originated from B to E.

Table 4.10: *Protection Mapping for Component E - Information Writer*

Operations of component E on assets with their guarding IDs	**Guardian Component and its protection operation**
receives and *verifies* signature of **remote commands** (E.3.ACA, E.3.VIC)	(ACA): D → verifies authenticity of received commands. (VIC): D → forwards only valid commands.
receives **keys** (E.2.CCO) from B	(CCO): C → no outbound communication allowed by OS restrictions
stores received **Knowledge Rep. Data** (E.1.CCO, E.1.VIC) received from D	(VIC): C → verifies that data stored by E complies with received data. (CCO): C → no outbound communication allowed by OS restrictions

(F) Index Manager Guarding

For component F, the operations with the assets *Knowledge Representation Data*, *Indexes*, *Cryptographic Keys* and *Remote Commands* have to be guarded. In this case, the additional protection can be performed by the existing components D and A (see Table 4.11) . Component F itself is used to act as guard with the following operations:

- Guard ACA for D: detects modification of commands by D or external entity

- Guard CCO for B: guards the communication originated from B to F.

Table 4.11: Protection Mapping for Component F - Index Manager

Operations of component F on assets with their guarding IDs	**Guardian Component and its protection operation**
receives and *verifies* signature of **remote commands** (F.4.ACA, F.4.VIC)	(ACA): D → verifies authenticity of received commands. (VIC): D → forwards only valid commands.
receives **keys** (F.3.CCO) from B	(CCO): A → prevents communication to alternative destinations by placing OS restrictions on F
creates and *sends* **indexes** (F.2.VIC) of **Knowledge Rep. Data** (F.1.CCO) together with hashes of **Knowledge Rep. Data** (F.1.VIC) to A	(VIC): A → before sending information out, component A compares metadata and hash with independent values provided by C when retrieving the information. (CCO): A → prevents communication to alternative destinations by placing OS restrictions on F

(G) Protection of additional component Key Guard

Component G is an additional component introduced by the approach for protecting the service against a malicious Key Manager component. Its operations with the assets *Cryptographic Keys* and *Remote Commands* have to be protected to prevent exploiting this component. The protection is performed by the components D and A (see Table 4.12) . Component G itself is used to provide hashes of keys to A so that A can verify the integrity of the keys send by B. This prevents a compromised component B from an undetected interfering with the keys used by A.

Table 4.12: Protection Mapping for Component G - Key Guard

Operations of component G on assets with their guarding IDs	**Guardian Component and its protection operation**
receives and *verifies* signature of **remote commands** (G.2.ACA, G.2.VIC)	(ACA): D → verifies authenticity of received commands. (VIC): D → forwards only valid commands.
creates and *stores* hashes from **keys** (G.1.CCO)	(CCO): A → prevents communication to alternative destinations by placing OS restrictions on G

4.3.5 Secured Design

After applying the approach, the service design now comprises seven isolated components. Based on the argumentation for the generic approach, a compromised component on its own cannot violate the specified protection goals anymore (see Section 4.3.1). In the modified design, all operations with the assets are now protected by another component. The final interaction graph is shown in Table 4.3. The final component functionality and privileges are shown in Table 4.13.

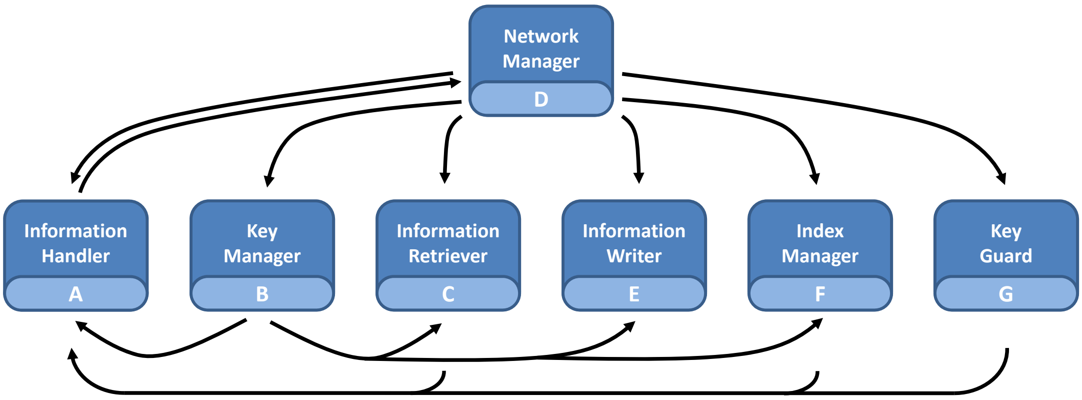


Figure 4.3: Final Interaction Graph between Service Components

Table 4.13: Final Overview of Component Functionality and Privileges

Comp.	Function	Privileges for Resources
A	Interface creation and information distribution	Deny: all Allow: Outbound Communication with D Allow: Inbound Communication from B, C, D, F, G Allow: Access to Index Store (RWA)
B	Manage personal keys for data sources	Deny: all Allow: Outbound Communication with A, C, E, F Allow: Inbound Communication from D Allow: Access to Key Store (RWA)
C	Retrieve information to be distributed	Deny: all Allow: Outbound Communication with A Allow: Inbound Communication from D, B Allow: Access to User's Databases (R)
D	Manage communication with external / non-server components	Deny: all Allow: Outbound Communication with A, B, C, E, F, G Allow: Inbound Communication from A Allow: Access to Network
E	Store received information	Deny: all Allow: Inbound Communication from D, E Allow: Access to User's Databases (W)
F	Crawl and index personal information	Deny: all Allow: Outbound Communication with A Allow: Inbound Communication from D, B Allow: Access to User's Databases (R)
G	Manage Key Hashes	Deny: all Allow: Outbound Communication with A Allow: Inbound Communication from D Allow: Access to Hash Store (RWA)

4.4 Summary

The first goal of this chapter was to identify the required protection for ubiquitous personal information management systems. This was achieved by inspecting the assets to protect in analogy with other fitting Common Criteria protection profiles but focused on additional protection aspects on application level. With this view, the described security objectives build the foundation for creating instances from the information management concept. These objectives can be achieved with existing cryptographic algorithms and common security principles, as long as implementation flaws are ruled out.

Since many of the daily security bulletins clearly proof implementation flaws to be a serious threat to data confidentiality, the second goal was to strengthen the security design with a further line of defense against attacks aiming at compromising and manipulating the service. The outcome is a generic approach that can be applied on information services to help increase the effectiveness of operating system protection. Starting from a monolithic service architecture, the approach guides to separate functionality into isolated components and to enforce mutual guarding of relevant operations. An attacker then has to compromise at least 2 of n components to overcome the privilege restrictions and component guarding. This is considered a second line of defense, since compromising two independent components is considerable more attack effort in terms of required knowledge, resources and attack concealment, which reduce in consequence the risks of successful attacks.

Having created a generic approach, the third goal was to create with it an actual security architecture instance for a service providing the functionality described in this work. The resulting service illustrates the possibility for the application scenario to arrange the functionality in components isolated by the operating system in the intended way. Together with the performed security considerations, the proposed architecture can be used for the protection of actual ubiquitous personal information management system implementations.

Chapter 5

Proof of Concept: MIDMAY

Having described the design for personal digital knowledge and the required security concepts, this chapter presents a proof of concept implementation called *MIDMAY* (Mobile Information Distribution, Management and Access for You!). The chapter therefore contains the implementation-specific aspects derived from the design presented in this work.

In Section 5.1 an overview of the implemented components is given, corresponding to the design described in Section 3.3 and the following. The section describes the steps taken to create a proof of concept implementation. The implemented core framework follows the information domain independent design, facilitated with the Topic Maps approach. The information extractor components, which implement the generic interfaces for retrieving information for data source's access protocols, were chosen to meet the requirements for the information worker domain, to research characteristics for a common application. An excerpt of these implemented extractors, describing the mapping to the information representation, is given in Section 5.2. It also shows how other data sources, even from other information domains, could be integrated into the extractor framework.

Section 5.3 provides insights about the implemented *information representation manager*, responsible for storing, maintaining and unifying the retrieved information. It describes the specific concept extensions for using databases to handle even large information representations, its required tasks and what is required to perform these efficiently.

Since the described concept is intended to help the *user* manage his information, Section 5.4 briefly discusses the implemented user interface, derived from the generic concept, which aims at an information domain independent interaction process. The same applies to the *information distribution manager*, described in Section 5.5.

5.1 Overview of Components

The proof of concept implementation closely follows the design concept and transfers it to an object-oriented architecture in the programming language Java. On a high-level view, the implementation can be summarized by the following components.

Information Retrieval Manager is responsible for controlling the framework of extractors, manages the requests for starting extractors and provides access via the extractors to its managed data source.

Information Representation Manager is the component that provides access to the unified information representation, created by MIDMAY. The initially independent parts created by the extractors inside the retrieval manager are presented by this component as a unified global representation through consistent merging. It manages the connection to the database via the Topic Maps access sub component to make the representation – and all sub parts – persistent.

Distribution Manager handles the secure communication with other user's information repositories to provide a convenient information exchange. It chooses between the possible transport options supported by both communication partners, negotiates communication settings according to security policies and passes though incoming data from other users to the representation manager.

Abstract User Interface uses the information representation manager to provide navigational access to the representation. It hides the underlying Topic Maps by offering a node based navigation together with interfaces for creating entry points to the representation. Additional interfaces for advanced features such as bookmarks, usage history, term search and similarity search extends the possibility for the user interfaces that implements this abstract user interface.

Application Gateway is located in the DMZ and protects the communication with the components residing in the Internet. Any incoming transmission from outside the perimeter is pasted through this verifying and authenticating component.

Remote Clients are connect via the application gateway to the abstract user interface, display the remotely stored content, and offers control options to authenticated users. It is the device-dependent face of MIDMAY to the user.

A major goal of these components is the separation of information retrieval from representation and visualization. This way the layers can be modified or exchanged if necessary without interfering with other components, as proposed by the design for ubiquitous personal information management. This applies also to the implementation aspects regarding security.

5.2 Information Retrieval Manager

The *Information Retrieval Manager* (IRM) represents the component with the connection to the data sources. It implements the extractor concept described in Section 3.3.3, which is capable of gathering metadata from arbitrary sources and which also provides the *Distribution Manager* a direct access to the stored content.

A range of useful extractors have been implemented, to verify the functionality of the concept and its acceptance by users. Providing a closer look into the concept, some of them will be introduced in short in the next sections as examples.

5.2.1 Extractor Framework

The extractor framework is designed with an open architecture to address the need for an easy integration of arbitrary information sources. Based on basic classes for one-dimensional, two-dimensional and hierarchical structures (following the rules presented in Section 3.3.2), new extractors can be built for new types of data sources by just implementing its access protocol. These extractors can be added at any time to the framework without changing implementation of other parts. It shields the differences of the data sources behind an extractor interface and integrates the results into the global representation.

Each extractor thereby maintains its own topic map via the interface to the *representation manager*, which keeps track of the maps and is responsible for merging the maps to a global one, as described in Section 5.3.2. Besides systematically harvesting a data source, the extractor concept also provides an interface for automatic updates. If a data source is capable of observing modification and insertion of data, the extractor can use this information to keep the topic map up-to-date.

The *representation manager* also provides the global topic types which are used across the different extractors. An excerpt of used topics is shown in Table 5.1. All identifier for the described topics are represented by *Published Subject Indicators* (PSI) and *Published Subject Identifier* (PSID).

Proposition 5.1. *The topic references are built with the following schema.*

- *Topic types are given a PSI that represents the concept of the type, like "http://www.project-midmay.de/midmay/psi/generic/#location" to specify location.*

- *Topics describing an instance of a topic type are given PSIs like "http://www.project-midmay.de/midmay/psi/generic/instances/location/#Berlin" to express the location Berlin.*

- *Topics representing referenced information are given PSIDs like "imap://user@mail.domain.tld/[folder]/[msgID]". Their type topics are given the type "Information-Object-Type" with the PSI* *"http://www.project-midmay.de/midmay/psi/generic/#information-object-type" to express that the typed topic represents a retrievable information object.*

- *Association types use PSIs like "http://www.project-midmay.de/midmay/psi/information/#object-author-relation" which is an association type between a topic playing the role* Information Object *and one which plays the role* Author*.*
- *Association type topics (such as the object-author-relation topic from the example above) are typed to describe their relation type. (e.g., a hierarchical relation is described by the PSIs proposed by Kal Ahmed [Ahm03]) "http://www.techquila.com/psi/hierarchy/#hierarchical-relation-type", whereas a property relation is described by the PSI "http://www.project-midmay.de/midmay/psi/property/#property-relation-type".*

Since each extractor instance maintains the connection to its data source, the PSIDs also identify the extractor that have to be asked to retrieve the PSID's content from the data source.

5.2.2 Filesystem Extractor

The *filesystem extractor* starts from a configured directory and collects metadata regarding all files and subdirectories recursively. Based on this basic extractor all file oriented repositories like networked file servers, web based document servers and even groupware solutions with external interfaces can be addressed.

Every file and directory is represented as a topic inside the topic map. To preserve the information which files and directors are contained in a directory, a *container-containee* association is defined and marked as hierarchical association. The topic types used for the representation are shown in Figure 5.1.

A facet for the structure is created to be able to find the root directory and the hierarchical association. Additionally, file properties provided by the filesystem — like the date of creation and the size of the file — are associated to the file topic. The crucial aspect in the creation of these property topics is the naming schema applied by the specific extractor. Information about a point in time offers a good clustering item, if a user remembers a special day or time frame. Therefore the global date topic is designed to be used as a universal timing reference. To be reusable for file dates, calendar entries and other contextual events, it is formated to describe the corresponding day in a uniform way. By default the user configurable date settings create the date topic basenames like "*20070802 / 2nd of August 2007 / Thursday*". This way, such basenames fulfill multiple purposes. The preceding *ISO 8601 basic format* provides the implicit sorting in lists. It is followed by the written out date to ease the search via *terms* (see 3.4.1) such as weekday, month, year or combinations of it. Date topics describing the same date then connects any other topic (events, files and

Table 5.1: *Examples of Relation Types and Roles in Prototype MIDMAY*

Relation Type	Involved Roles	Base Names
Address-Person Relation	Email Address Person	belongs to has mail address
Container-Containee Relation	Container Containee	contains is contained in
Event-EmailAddress Relation	Event Email Address	has attendee is attendee of
Facet-Hierarchy Relation	Information Object	belongs to
Facet-Root Relation	Information Object	belongs to
File-Filetype Relation	Filetype Information Object	is filetype of has filetype
Object-Author Relation	Author Information Object	is author of was written by
Object-Collection Relation	Information Object Collection	belongs to contains
Object-Date Relation	Information Object Date	has date is date of
Object-Filesize Relation	Information Object Filesize	has filesize is filesize of
Object-Identity Relation	Information Object Identity	has identity is identity of
Object-LastModified Relation	Date Information Object	is last modification date of was last modified on
Object-Location Relation	Information Object Location	is located at is location of
Object-Receiver Relation	Receiver Information Object	is receiver of was received by
Object-Sender Relation	Sender Information Object	is sender of was sent by
Object-Title Relation	Information Object Title	has title is title of
Parent-Child Relation	Child Parent	is child of is parent of
Topic-Project Relation	Project Artifact Project	belongs to involves

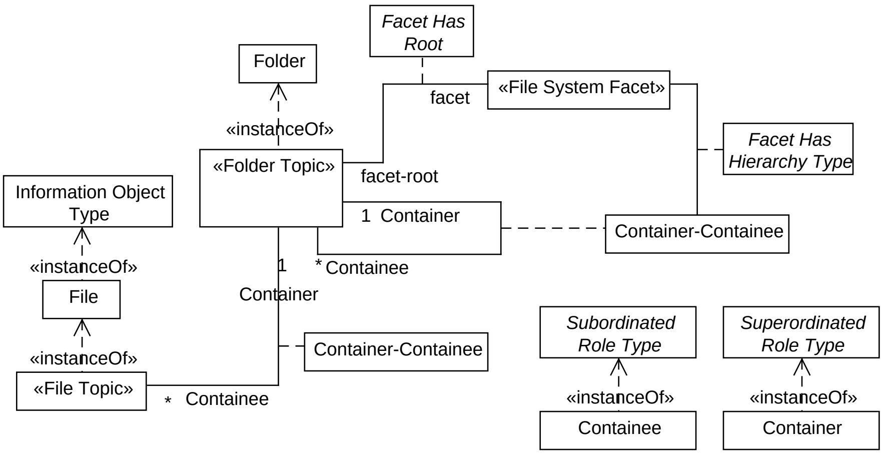


***Figure 5.1:** Extractor Model for Directories and Files in UML [Ber06]*

emails) associated to that date, independently of their original source and the creating extractor.

An equivalent name schema is used for the size of files. In this case a grouping of files with one of 6 predefined and configurable classes of file sizes is performed by the file extractor. Each file topic therefore is associated with one of the topics "*0-10k byte*", "*10-100k byte*", "*100k-1000k byte*", "*1000k-5000k byte*", "*5000k-10000k byte*" and "*above 10000k byte*", corresponding to the largest upper bound of a class that is not exceeded by the file size. Again the basenames are designed to ease the usage of search terms such as *10k*, *100k*, etc. This way the user can narrow down the search to a category of a file size he probably remembers, without having to recall the exact size of the file.

After finding the desired information inside the representation, the framework also needs to know where to find the original data to be accessible for the user who might want to access it or want to send it to someone. This is why the physical location of each information object is stored as *ResourceRef* of the corresponding topic by *file URL schema*[1]. Files referenced by this schema (like "*file:/d:/music/song.mp3*") can be retrieved by the responsible extractor, which is chosen by the framework depending on the registered PSID namespace for the extractor. At the same time the URL is also the unique identifier of the topic. This way, identical files found on different locations will have a unique link to its locations. Each file is also associated with its hash value (created via the file metadata extractor described in Section 5.2.5 to find identical files. This provides the possibility for the user to follow the association from a hash topic (representing the identity of a certain file) to all associated files (all identical files) regardless of their storage location or the required access protocol.

[1] Uniform Ressource Locators; defined in RFC 1738 (see `http://www.faqs.org/rfcs/rfc1738.html`)

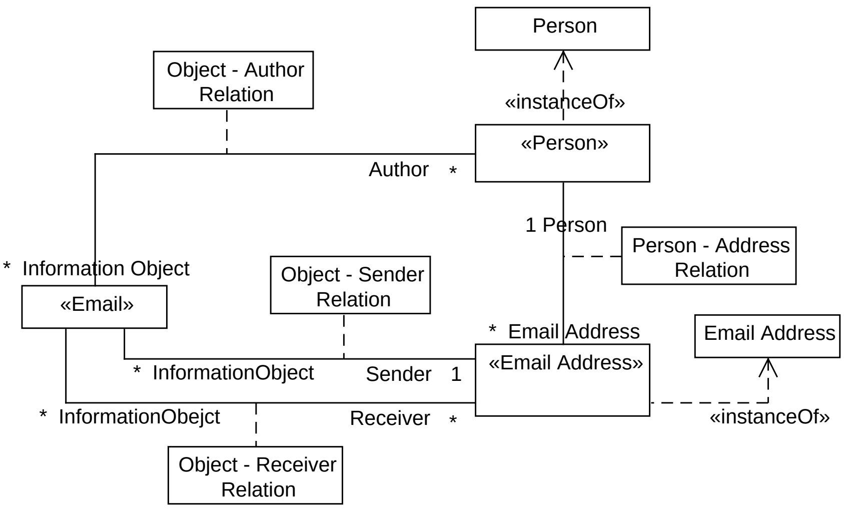


***Figure* 5.2:** *Extractor Model for Email Properties in UML [Ber06]*

This is useful to keep track of multiple locations used for storing identical files and to help the framework providing synchronization functionalities.

5.2.3 Email Extractor

Following the rules specified in Section 3.3.2, another extractor is created for email servers based on POP3 and IMAP protocols. Using the provided framework, it transfers the hierarchical structure of an email account into the representation applying *container-containee* relations (see Figure 5.2). The email subject is used as (not unique) basename for the topic representing the message instance, since it most likely contains information a user might remember. The topic is identified by the unique email's message ID inside the created subject identifiers: *imap://user@mail.domain.tld/[folder]/[msgID]*. With this identifier the extractor can retrieve the content of the email, so the user does not have to change to his email client to view the email.

Useful properties of the email, such as name of author, sender's email address, recipient's email address and the date of creation are represented by topics and associated with this email topic. Other associations are created that contain the hierarchical information like relation to email folders or attachments.

Contained attachments are processed also by the file metadata extractor (see Section 5.2.5) to enhance the representation with topics created from metadata about file types, authors, titles and hash identities. All this information is used to interconnect the knowledge to speed up the search and to provide a better overview of existing data that is accessible via the representation.

5.2.4 Organizer Extractor

A further example to be given here is an extractor for organizer content, collecting information about tasks, events, involved persons, locations and dates. It uses as input either iCalendar[2] files (provided by many organizer programs) or directly connects to a calendar server supporting the *Web Calendar Access Protocol* (WCAP) like the *Sun Java System Calendar Server*[3]. The advantage of a calendar server is twofold: it is accessible from a server-based extractor even if the user's desktop PC is not powered on; and the calendar server offers the functionality to synchronize data with other calendar client software like the Mozilla Thunderbird plugin called Lightning[4] or Microsoft Outlook[5], which makes the server an efficient and easy to maintain data source.

The iCalendar files are parsed and represented directly with the process described for two-dimensional sources (see Section 3.3.2), because the data is stored as a simple table of entries. The same applies to the information retrieved via WCAP. Topics with **subject indicators** are created for the contained locations, persons and dates, whereas the events and tasks are represented by topics with **subject identifiers** like: *WCAP://server.domain.tld:8003/[calendarID]/[entryID]*. Via these identifiers the full content for each entry can be retrieved as iCalendar object (RFC 2445), which is a common exchange format for these data types.

Especially this extractor binds together previously unrelated topics by leveraging the existing knowledge extracted from the organizer already maintained by the user. By connecting topics of persons, locations, dates of events/tasks implicitly through the used subject indicators with topics of files, emails and their properties, the context of each topic grows. Thus, the previously separated information represented by a single topic now forms an enriched picture of relations, which would be otherwise not usable for searching.

5.2.5 File Metadata Extractor

Files also often contain valuable metadata that can enrich the user's information representation. Therefore MIDMAY provides a file metadata extractor, which is invoked for ever source entry identified as file content, independent which extractor is accessing the file. The metadata extractor first identifies the type of the file

[2] Internet Calendaring and Scheduling Core Object Specification; defined in RFC 2445 (see http://www.ietf.org/rfc/rfc2445.txt)

[3] A.k.a: iPlanet Calendar Server and Sun ONE Calendar Server, see http://www.sun.com/software/products/calendar_srvr/

[4] Via the WCAP supporting version; see http://www.mozilla.org/projects/calendar/lightning/

[5] Via Sun Java System Connector; see http://www.sun.com/software/products/calendar_srvr/connector/ and office.microsoft.com/en-us/outlook/

with the help of *magic numbers*[6], by searching for recognizable structures inside the file, or by using the file extension, if the data format can not be identified otherwise. With this information the corresponding metadata extractor is invoked. These metadata extractors are registered with the framework to provide an easy extendable architecture. To prove this concept, extractors for common and metadata-rich filetypes, such as Microsoft Office documents (DOC, XLS, PPT), Adobe Portable Document Format (PDF) and the ID3-Tag contained in MPEG-1 Audio Layer 3 (MP3) files have been implemented for MIDMAY.

With the help of these metadata extractors, valuable information about subject, author, date and title of files are extracted and added to the representation. The globally defined subject indicator name schema again interconnects the topic representing the file with other topics associated to equal metadata topics. An author's name found inside file metadata will be represented by the same person topic that is also associated to any other related topics of that person, such as all the author's emails, tasks or events. Searching for this topic reveals the authorship for documents, emails, corresponding addresses and other concepts like the associations to meetings extracted from the personal organizer.

5.3 Information Representation Manager

For organizing the extracted information, the *information representation manager* offers interfaces for all other components. This component is responsible for managing the access to the representation and to provide the require functionality for a synchronized access of extractors and visualization components.

All collected information and the relations to each other are represented by Topic Maps. Using Topic Maps in a programming language requires a Topic Maps engine with a defined application program interface for accessing and manipulating data. The *Common Topic Map Application Programming Interface* (TMAPI)[7] defines such an interface to Topic Maps engines. TMAPI is designed to provide a single common API for Topic Maps developers to improve the portability from one engine to another with minimum effort. Currently open source implementations of TMAPI are available with TM4J[8] and TinyTIM[9].

The proof of concept implementation uses a modified version of TinyTIM together with an adjusted and extended version of the open source implementation XTM4XMLDB[10]. It is used for the efficient storage of topics in an XML database.

[6] A form of in-band signaling the data type by using the first bytes contained in many formats; e.g., PDF files start with "%PDF"

[7] See http://www.tmapi.org/

[8] Topic Maps For Java (see http://www.tm4j.org/)

[9] A tiny TMAPI in-memory implementation (see http://tinytim.sourceforge.net)

[10] A Topic Maps implementation for native XML database access (see http://sourceforge.net/projects/xtm4xmldb)

5.3.1 Storage Management

Even in small working environments, topic maps created by autonomous extractors fast grow over tens of thousands topics, which would limit a useful application to the capacity of the system's memory, if the complete map have to be held in memory. On the other hand, rolling out parts of the maps to cheap persistent storage requires a time consuming reloading in case of access. Therefore an efficient storage management for disk and memory usage is required to handle the huge number of topics, which can not be handled by pure in-memory concepts. Since the same problems are addressed in other fields by databases, the prototype also uses these already well engineered concepts of efficiently indexing data, cache management and transaction security, in combination with an application side access design suited for database access.

As database backend, the open source native XML database called *eXist*[11] is used. The database frontend (residing inside the Topic Maps engine) uses XPath[12] queries for an efficient database access on the XML representation of topic maps stored in the database.

The database uses collections to organize content in hierarchies. For MIDMAY, each user is given an own collection, which contains the collections created for each topic map. The representing topics of one topic map are stored as a collection of entries, containing the properties of a topic (see Figure 5.3). A second collection is maintained for entries of associations (see Figure 5.4). These entries build the leafs in the hierarchy of collections created in the database for each user. A path inside this hierarchy to a Topic Maps entity then looks like: */db/[userID]/[topicmap]/topics/[id]* for a topic instance, or */db/[userID]/[topicmap]/associations/[id]* in case of associations.

Besides leveraging powerful cache algorithms with *B+ Trees*, a further advantage of the database concept is the creation of application specific indexes.

Proposition 5.2. *Indexes that improve the query processing time for the use with Topic Maps in XML representation where identified as*

- *A* structural index *which contains the nodal structure to efficiently navigate inside the topic map by accessing dedicated entities. It is the default index in eXist.*
- *A* range index *over the attribute* id. *The index improves the access to topic nodes via their unique id in XPath queries. This is done e.g. during resolving references to type topics, scope topics and role topics. It also speeds up some merge operations that address and compare topics on database level via their ids. This index is also available by default in eXist.*

[11] Database management system entirely built on XML technology by storing XML data according to the XML data model and featuring index-based XQuery processing (see http://exist.sourceforge.net/)

[12] XML Path Language; language for selecting nodes from an XML document (see http://www.w3.org/TR/xpath)

```
<topic xmlns:xlink="http://www.w3.org/1999/xlink" id="id458404528909">
    <instanceOf>
        <topicRef xlink:href="#id458404528860"/>
    </instanceOf>
    <subjectIdentity>
        <resourceRef xlink:href="imap://user@mail.domain.tld/
                           INBOX.Sent/45F51BB7.1010405@companyg.de"/>
    </subjectIdentity>
    <baseName id="id458404529003">
        <baseNameString>Draft agreement</baseNameString>
    </baseName>
</topic>
```

Figure 5.3: *XML Representation of a Topic in the Database. The topic is an instance of the type referenced by the* xlink:href *attribute of the* topicRef *node inside the* instanceOf *node. (see [Into6b] for details of the Topic Maps XML syntax)*

- *A* range index *over the attribute* xlink:href*. Since this attribute is used to reference topic types, scope and roles, an index significantly speeds up all operations that need to access topics via these properties. Additionally it improves the access of topics via subject indicators and subject identifiers.*

An index configuration fulfilling these criteria is applied in eXist by creating a collection-specific standard XML document (see Figure 5.5), stored inside the system collection */db/system/config/*. Since configurations are shared by descendants in the hierarchy (unless they have their own configuration), only one configuration for the main collection have to be made to optimize the access for all contained topic maps.

However, an efficient access to the database is only the first step. The whole application has to be designed to be aware of the database access. Otherwise, algorithms designed for in-memory topic maps that use the provided interfaces would produce to many database calls that could be avoided by a better trade-off between memory consumption and function calls. Therefore, even if the usage of topic maps stored in databases is equal to in-memory topic maps regarding the access (because of TMAPI), the representation manager has to take care of efficiently using TMAPI access regarding access pattern, topic object re-usage and data lifetime. Otherwise, the access has shown to be far to slow for a direct user interface with the created topic map size.

The connection to the database is created by employing the *XML:DB* application program interface. This Java API performs the communication with the *XML-RPC* interface of the database, which provides the possibility to use dedicated database server. The round-trip time for queries (depending on their complexity) to the database ranges from 0.3 to 5 seconds, which is deemed usable, as it fits the usual

[13] See eXist index configuration, `http://exist.sourceforge.net/indexing.html#rangeidx`

```
<association xmlns:xlink="http://www.w3.org/1999/xlink" id="id803754262605">
    <instanceOf>
        <topicRef xlink:href="#id803754260098"/>
    </instanceOf>
    <member id="id803754262606">
        <roleSpec>
            <topicRef xlink:href="#id803754260100"/>
        </roleSpec>
        <topicRef xlink:href="#id458404528909"/>
    </member>
    <member id="id803754262607">
        <roleSpec>
            <topicRef xlink:href="#id803754260102"/>
        </roleSpec>
        <topicRef xlink:href="#id803754262588"/>
    </member>
</association>
```

Figure 5.4: *XML Representation of an Association in the Database. In this example the first member of the association is the topic shown in Figure 5.3, referenced by the* xlink:href *attribute of the* topicRef *node inside the* member *node. (see [Int06b] for further details of the Topic Maps XML syntax)*

```
<collection xmlns="http://exist-db.org/collection-config/1.0">
    <index xmlns:xlink="http://www.w3.org/1999/xlink">
        <create qname="@xlink:href" type="xs:string"/>
    </index>
</collection>
```

Figure 5.5: *Configuration of eXist Database Index for Topic Maps. All values of the attributes "xlink:href" are used for* range indexes[13] *independent of the position inside the XML hierarchy.*

response time of web applications. Especially in queries with a big result set the communication overhead takes a bigger part of the overall response time, compared to the time required to retrieve the data on database side. Therefore the embedded mode of *eXist* is an interesting option. In this mode, the database runs in the same Java virtual machine as the representation manager. It passes the data directly to the requesting component, without using the overhead of transmitting it though the TCP stack. In this mode, the response for big result sets is about twice as fast compared to the remote mode. So the embedded mode should be used if the advantage of shared resources for the database server and the accessibility of the stored topic maps from other applications is rated lesser than a fast read access of large result sets, which is deemed the case for the prototype implementation MIDMAY.

5.3.2 Merging Maps

The representation manager is also responsible for generating the unified representation of the user's data sources. This is performed by merging the single topic maps produced by each extractors into a global topic map, to keep the representation up to date as described in Proposition 3.4.

Merging of topic maps is a complex process performed by the Topic Maps engine, which normally does not require the developer to interfere with. However, because of the development status of the used XTM4XMLDB code, the merge process have to be redesigned to meet the requirements of MIDMAY regarding performance and database usage. The improved functionality, recapitulatory described below, improves the performance by a trade-off considering memory consumption, required database accesses and probability of certain situations in this process. The first optimization is sorting the maps that should be merged according their topic collection size and always merging the smaller map into the bigger one. The process of merging is described in Proposition 5.3.

Proposition 5.3. *The merging of a topic map B into a topic map A is performed by*

1. *Retrieving an index of all topics contained in topic map B and iterate of this index.*
   a) *Mark all topics M in B for which a topic exists in A that has an equivalent subject indicator or subject identifier.*
   b) *Merge all topics M with the equal topics of A by joining their properties and change the topic references to the IDs of topic map A.*
   c) *Copy all other topics from B to A*
2. *Copy associations in B to A*
   a) *Only check associations for a possible duplicate instance if one of its members is contained in M, hence the topic was merged with one of A.*
   b) *Change the topic references in copied associations to those of topic map A.*

The process described in Proposition 5.3(2a) is necessary to prevent duplicate association instances and to get a consistent representation after the merging. However, checking for consistency is a very time consuming process, so instead running over all associations in B, only associations using topics that have to be merged are tested. If a topic is not merged, an association connected in B to this topic cannot exist in A and therefore the time consuming testing for an equal association can be saved in this case.

5.4 User Interface

In this section MIDMAY's user interface is described. First the abstract functionality is explained before Section 5.4.1 and Section 5.4.2 present the implemented mobile client and the web interface, respectively.

The functionality of the abstract user interface offers the access to the stored representation of the user's data sources. Therefore the main interaction is done via the navigation interface. It provides entry points to the representation with a **term search** and the **access via types**. These entry points then can be used to follow the connected associations to other topics in a closed interaction cycle, as described in Section 3.4.1. The interface also provides direct access to any topic node via its *subject indicator* or *subject identifier*. This way the clients can implement **history functions** which display a list of nodes reached in previous interaction steps. By selecting such an entry, the interface can present the corresponding topic node with all interaction options. The most obvious further interaction is the **retrieval** of the represented data from the original data source, as described in Section 3.6.1 to be used in the current working environment. A similar functionality is the **distribution** of content to other users, either via email or via a direct communication with another MIDMAY server (see Section 3.6.3).

Any work with more then one topic is handled by the group interface. It offers the interface to a sorted collection, the user can fill will arbitrary topics by **marking topics** via their ID. The content can then be **resorted** and **removed**, depending on the actual use of the collection. One application for this group interface is the sending of multiple items to recipients, but the more advanced use of this interface is provided by the possibility to **calculate associated topics** to the given set of topics. Sample search queries with this functionality are shown in Section 3.5.4.

The remainder of the abstract user interface is dedicated to the **configuration** of MIDMAY components and to **display status information** of the server. This way extractors can be added, configured and started and the information about used resources and pending operations can be accessed.

5.4.1 Mobile Client

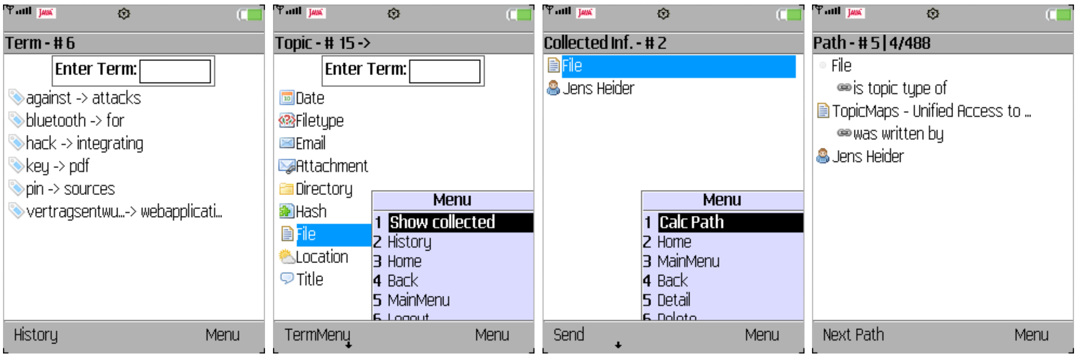
The mobile client provides the direct access to all personal information, which can be carried around in available mobile phones. It is based on the Java MIDP 2.0 technology with an additional support for Java Bluetooth access (JSR 82) and an access to the phonebook (via JSR 75). These requirements are fulfilled by the majority of todays mobile phones even in the sector below 100 Euro (without subsidization).

The implementation presented in this work is an enhancement of the client described in [Rei06]. Besides the integration into MIDMAY's security architecture and the improvement of the connectivity via the application gateway in the DMZ (see

Section 4.1.1), additional functionality to exchange MIDMAY identities via Bluetooth have been added. This exchange of contact data (containing also the public keys for the secure communication of the application servers; see Section 5.4.3) is based on the security framework *BlueLimes* [HS07, Sch07].

The user interface of the client implements MIDMAY's abstract interface (see Section 5.4) provides access by creating views for the states the user is currently situated in, according to the *click and cycle* interaction cycle (see Section 3.4.1). After connecting the client to the application gateway, the *type view* is presented, which lists all available type topics (see also Figure 3.10(a)). By selecting one type entry, a list of all topics of that type is presented. However, displaying a list with many items at once would be very inconvenient to use with the limited capabilities of mobile phones, since the user would have to spend some time to scroll down to the desired information. Therefore, in lists with many items, the topics are split-up into terms (see Section 3.4.1) that are used to cluster the topics in alphabetic groups in the *term view*, shown in Figure 5.6(a). These groups can have multiple levels, which are optimized for both: containing an equal number of entries in each group; and the number of lines should fit on the display screen for each level. Having reached the lowest level (because of the exponential capacity of this clustering, in common use cases only two or three levels are required to store all terms) or having used the direct term search field at the top of the view, selecting a term from the list presents the topics in a *topic view* (see also Figure 3.10(c)). In this view, all topic-related actions can be performed, as defined by the abstract user interface. To use a topic for any action, it is collected with the group interface. Then the collected topics can be accessed (see Figure 5.6(b) and Figure 5.6(c)) to choose the action to be performed. Besides retrieving the referenced information (converted to a simple text format, see Section 3.6.1), the main application is to instruct the application server to send the referenced documents in original format via the wired Internet to other recipients. The necessary contact information of recipients is directly accessible from the phone's address book. Either a stored email address of a recipients is used, or the phone already contains a previously exchanged MIDMAY identity for the desired recipient. The exchange of recipient information with other recipients via Bluetooth can be caught up also in the sending stage to ease the interaction process.

As shown in Figure 5.6(c) also the calculation of paths to perform advanced search queries (see Section 3.5.4) is performed via collecting desired topics with the group interface. The result of such a query is shown in a *path view* (see Figure 5.6(d)), which provides navigational options to reach all presented topics via selection. Of course all other views provide also the navigation interface (e.g., from any topic to the *associations view* shown in Figure 3.10(d)), so that the user experience the same interaction model regardless of the current view. This applies also for the *history view*, which provides navigational aid to topics previously visited by the user inside the current session. Frequently needed topics can be bookmarked and accessed later via



(a) Term Menu (b) Topic Menu (c) Collect Menu (d) Path Menu

Figure 5.6: Screenshots of mobile MIDMAY client

the *bookmark view*, for example to directly jump to an important folder, date or person topic as an entry point to the user's digital knowledge representation.

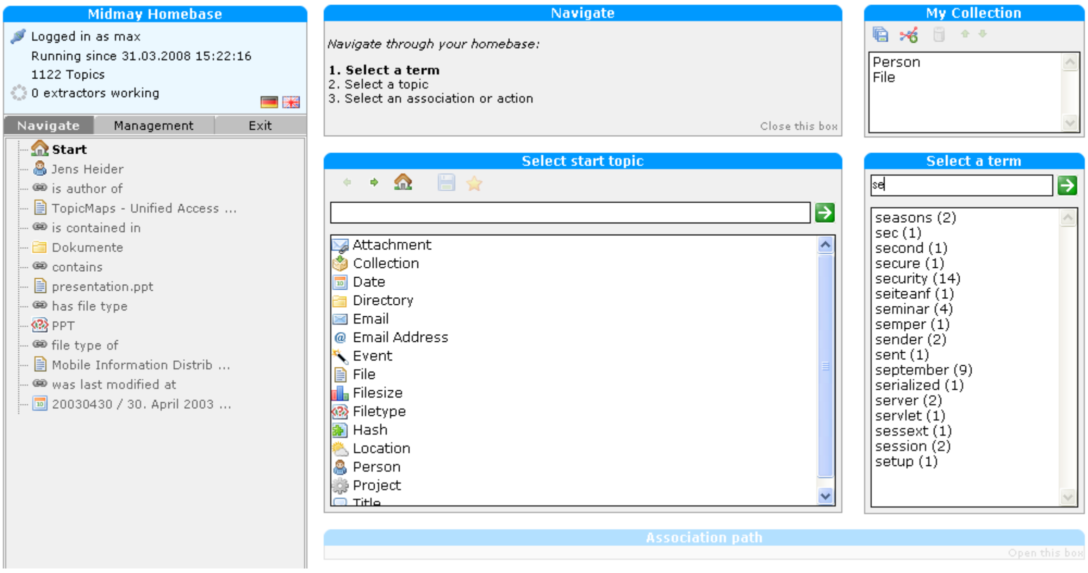
5.4.2 Web Management Interface

Daily work with information from the desktop is provided by MIDMAY's web interface. Its aim is to provide a single and convenient interface to all indexed data sources without the need to change between the interfaces normally required to access the data. This is especially helpful when tasks require the access of information stored in independent data sources.

The underlying technology for this web interface is AJAX[14], which provides bidirectional communication between application server and the web client. This way updated content to be displayed can be pushed by the server without requiring the user to update the web page. It is used in multiple ways to keep the user informed and to ease the interaction with the stored information.

The main interaction with this user interface is performed in the central canvas, shown in Figure 5.7. It displays lists of selectable items, which are used to navigate through the user's digital knowledge representation. Depending on the current state, the user is also guided by the window above about further actions he can perform at this point. With this central display, the user navigates from topics to associations, can retrieve referenced information like documents and other entries, sends referenced documents to other recipients and adds topics to the collection element at the upper right side. This collection then can be used to group actions like viewing or sending. Additionally, this element provides the key functionality to issue search queries related to the collected topics ((see Section 3.5.4)). When the server pushes a result to such a query back to the client, the path view at the bottom expands to a navigation

[14] "Asynchronous JavaScript and XML", a combination of web development techniques to build dynamic web pages on the client side.



Logged in as max
Running since 31.03.2008 15:22:16
1122 Topics
0 extractors working

Start
Jens Heider
is author of
TopicMaps - Unified Access ...
is contained in
Dokumente
contains
presentation.ppt
has file type
PPT
file type of
Mobile Information Distrib ...
was last modified at
20030430 / 30. April 2003 ...

Navigate through your homebase:

1. **Select a term**
2. Select a topic
3. Select an association or action

Person
File

Attachment
Collection
Date
Directory
Email
Email Address
Event
File
Filesize
Filetype
Hash
Location
Person
Project
Title

se

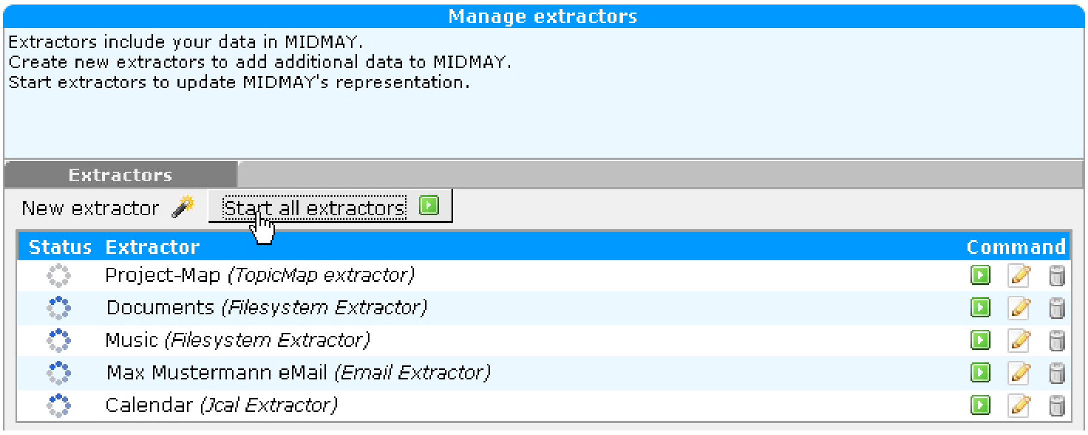
seasons (2)
sec (1)
second (1)
secure (1)
security (14)
seiteanf (1)
seminar (4)
semper (1)
sender (2)
sent (1)
september (9)
serialized (1)
server (2)
servlet (1)
sessext (1)
session (2)
setup (1)

Figure 5.7: Overview of MIDMAY's Web Interface. The left panel shows the navigation history, which can be switched to show management options. The main canvas in the middle handles the navigation through the representation, guided by the instructions shown above it. On the upper right side, the collected topics are handled and below this element the term search can be performed. The path view at the bottom is only accessible if path queries are issued (see Figure 3.13).

interface. It visualizes paths between topics, which can be clicked to jump to the returned topics inside the main view (see Figure 3.13). An equal interface technique is used for the history window on the left side. It records the visited nodes and provides a direct access to previous topics.

Besides using a type topic as entrance point, the term search window at the left side also provides the possibility to enter characters, which instantly produces a list of matching terms, after the amount of the calculated result falls below a threshold of displayable items. This way the user can see while typing which (and if) alternatives for the typed terms are available. Clicking a term in this window then shows all topics containing this term in the main canvas, which can be used for further interaction.

The left panel of the web user interface can be switched to the configuration menu. It contains status information about the current status of the server and holds the functionality to add, remove and manually start extractors (see Figure 5.8). Each extractor can be configures via this menu regarding the data source's connectivity properties and regarding options for the specification of extraction settings, such as data fields to use, date ranges and frequency. Additional extractors are added by first selecting the data source type, which creates the form of required and optional settings, and saving the filled out configuration.



Manage extractors

Extractors include your data in MIDMAY.
Create new extractors to add additional data to MIDMAY.
Start extractors to update MIDMAY's representation.

Status	Extractor	Command
○	Project-Map *(TopicMap extractor)*	▶ ✎ 🗑
○	Documents *(Filesystem Extractor)*	▶ ✎ 🗑
○	Music *(Filesystem Extractor)*	▶ ✎ 🗑
○	Max Mustermann eMail *(Email Extractor)*	▶ ✎ 🗑
○	Calendar *(Jcal Extractor)*	▶ ✎ 🗑

Figure 5.8: *Extractor List in MIDMAY Web Interface. Rotating gradual circles in front of the extractors indicate that the extractors are currently working.*

5.4.3 Secure Identity Exchange

The distribution of information between two MIDMAY users (as described in Chapter 3.6) requires the exchange of the authentic identity of both communicating parties. This identity contains the addresses for the information transmission but also public keys for the secure authentication. Therefore a wireless identity exchange between the mobile devices of the users have to be protected to prevent manipulation through man-in-the-middle attacks. The best suited technology for such a short range communication currently doubtless[15] is Bluetooth and there are various approaches to device authentication in ad-hoc situations that can detect manipulation attacks without having to activate the inconvenient build-in security (see Section 2.3.1). They take advantage of a common context of both communication parties. Usually, “context” is defined here as spatial proximity, but other interpretations are possible as well. In this work, additional hardware requirements in terms of phone cameras or IrDA interfaces is avoided to ensure a wide applicability.

The approach rather relies on the user himself as a trusted out-of-band-channel over which a limited amount of data can be transferred from one device to another. This channel is used here to compare the hash values of transmitted data on both communicating devices.

Proposition 5.4. *Because the user is turned into a part of the authentication protocol, some important points have to be taken care of, to avoid the protocol becoming insecure or unusable:*

[15] Regarding availability in devices, technology acceptance and communication characteristics. This consideration might change in the future, when Near Field Communication technology (see http://www.nfc-forum.org/aboutnfc/) will be available in the mobile device mass market.

- *The task, the user is set must be sufficiently easy to understand and fast to solve. Otherwise, users are overburdened and will make mistakes, which results in frustration and leaves the protocol prone to attacks.*
- *The usability has to remain constant, regardless of the security level in order to avoid users getting the impression that security is something that requires great effort and as a result opposing it.*
- *Conditioning users to insecure behavior must be circumvented in any case.*
- *The user must not authenticate a connection by mistake. Thus, erroneous inputs must always result in closing the connection.*

Although different kinds of user interaction are possible (see [Sch07]) the approach in this work uses a hash visualization method, because it best matches the requirements defined by ubiquitous personal information management.

Using unprotected Communication

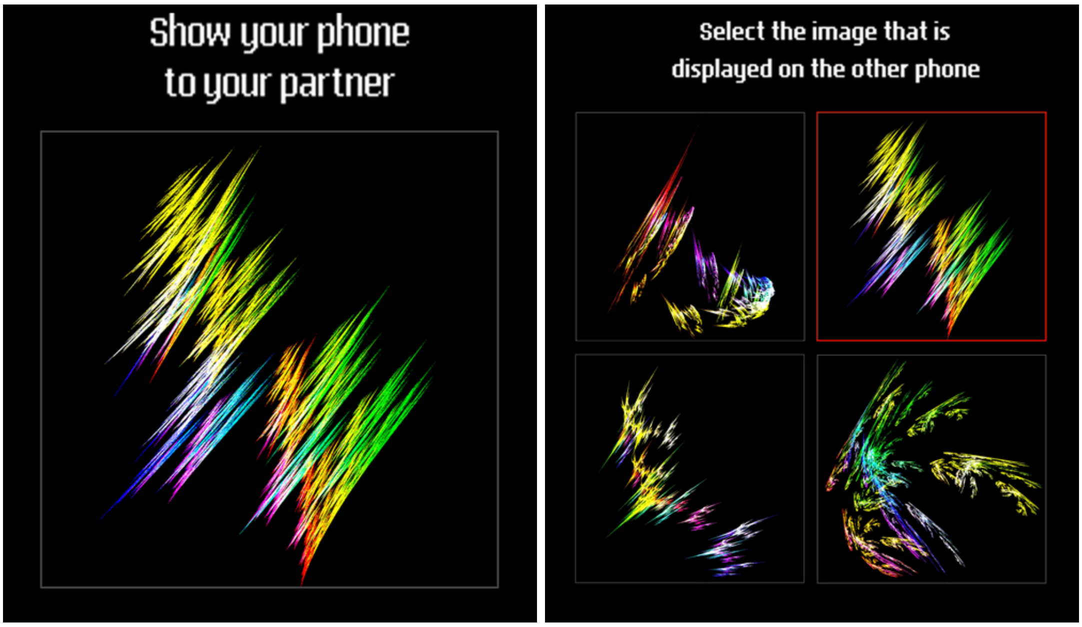
Usually securing the connection between Bluetooth devices is performed using the built-in security mechanisms, which require the user to conceive a secret PIN and enter it into both devices. However, Bluetooth pairing based on PINs is not only inconvenient, but is also prone to very effective passive attacks [SW05]. Additionally, as the enforcement of Bluetooth security parameters happens at link layer and depends on the overall device security mode, applications are not able to control if certain protection goals have been applied to their connections. Relying on approved higher-level approaches like SSL or IPSec is likewise inappropriate as their authentication methods are based on access to PKI or on pre-shared secrets.

Instead in the work presented, a communication session is established and security parameters are negotiated in an unprotected way. Then, a verifier module asserts in an authentic way that both devices have seen the same communication so far and thereby proves that no active attack has taken place.

Verifying the Integrity of Communication

In the verification phase, the hash value of the communicated data is visualized and presented to the user, who in turn has to check whether the other party's display shows the same value. Various visualization methods are conceivable, yet their computation has to be efficient enough to be used on mobile devices. The chosen fractals from *Iterated Function Systems* (IFS), pioneered by Michael Barnsley[16], appeared to provide the best usability while still being sufficiently effective computable (see

[16] In this work the Visprint-IFS (initially implemented by Ian Goldberg) is used as one instance of fractal visualization. See http://www.tastyrabbit.net/visprint



Show your phone to your partner

Select the image that is displayed on the other phone

(a) Display of Sending Device

(b) Display of Receiving Device

Figure 5.9: *Verifying the Integrity of transmitted Data with Hash Visualization [HS07]*

[HS07]). The necessary coefficients are derived in this approach from the hash value and the results of each equation is individually colored, to increase the criteria for distinguishing the hash values.

However, Proposition 5.4 demands that just showing an image on both devices and letting the user compare them is not secure enough. Non-manipulated connections are the usual case, therefore users would quickly become habituated to confirm the equality of the images. Hence, users have to be enforced to conduct the comparison and to answer truthful in the case of non-matching images. To achieve this goal, one device generates three additional fractals based on random data. These dummy fractals are displayed together with the actual fractal in random order, as shown in Figure 5.9(b). The receiving user is then asked to pick out the image he recognizes on the device of his communication partner. By selecting the matching image, he thereby confirms that both devices have calculated the same digest. In the case of an active attack, the user will not be able to find two matching images, which he states by choosing the "does not match" option in the user interface of his device.

Although the user becomes part of the authentication protocol, the interaction is kept quite simple and leads to a better usability, compared to Bluetooth PINs. It avoids conditioning effects because conscious decisions are required and thereby the risk of an unintentional confirmation of a manipulated connection is reduced.

This way the identity can be securely exchanged between two mobile devices. The contained public keys are then used by the user's server to authenticate each other during the distribution of content from one user to the server of the recipient, as described in the next section.

5.5 Information Distribution Manager

Distributing content to recipients is the aim of the *information distribution management* component. Depending on the given address type, this manager uses the *information retrieval manager* (see Section 5.2) to create an email with attachments for the recipients or directly connects to the referenced application server of the recipients for a direct upload of the retrieved information. In the proof of concept implementation only the second option (using exchanged MIDMAY identities, see Section 5.4.3) provides an encrypted communication, since no management for S/MIME certificates or PGP keys have been integrated so far. However, in an extended implementation these public keys could be retrieved from the user's representation as properties of an email address topic.

The protocol for the direct communication with another knowledge representation system applies the secure communication framework described in [Fle06]. It provides negotiation of encryption standards and offers schemes for mutual authentication. Both parties can specify the available communication protocol that can be used after establishing a secure session. The content and its describing map of additional context information (see discussion in Section 3.6.4) is therefore transmitted in an encrypted session with the MIDMAY Homebase Transport Protocol (MHTP) after the session was mutually authenticated using the MIDMAY identities of both sides.

On the recipient side, the distribution manger is responsible for validating the received content. The received describing map is checked for consistency and for the absence of external references, to prevent security impacts on the own representation. Then the metadata extractor is invoked and additional metadata created from the transmission event is added to the describing map to enrich the context of the content and to provide additional search paths to it for the user. After this step, the content is stored in the filesystem and its describing map is merged with the topic map holding all received information topics. Then the leveled merging process is started, as shown in Figure 3.6 in Section 3.3.3.

Chapter 6

Prototype Results

The proof of concept implementation provided the chance to analyze the characteristics of the concepts applied to the information worker domain. In this chapter, these results are described and interpreted regarding expectations and assumptions made for the underlying concept. Measurable properties of the implementation are presented and evaluated for productive environments in Section 6.1, to decide in which fields further research is beneficial. This analysis is split up into a consideration of the creation and maintenance of the digital knowledge representation (Section 6.1.1), the survey of its graph's characteristics (Section 6.1.2) and the system's mobile access performance (Section 6.1.3). These are crucial aspects when discussing the benefits of the concept for users, since it significantly determines the feasibility of the digital knowledge concept.

Also results are made during this research whose accurate measurement exceeded the scope of this work. These findings are nevertheless presented in Section 6.2 to show first impacts on organizing information work and search strategies (Section 6.2.1), as well as security implications found during using the implementation (Section 6.2.2).

6.1 Properties of the Implementation

The proof of concept implementation is independent from the used information domain, because no domain specific aspects are assumed or integrated and the applied Topic Maps concept is universally applicable. In contrast, the data source specific extractors with their implicit ontology does of course have an individual effect on the results. Therefore the intention of this section is to evaluate the feasibility of one possible implementation instance derived from the personal digital knowledge concept and to give insights about what expectations can be made regarding computational effort and characteristics of the results. All this, of course, should be considered as experimental data that indicates the tendency for the evaluated aspects.

An even greater impact on the results than the implementation may be caused by the used personal data, since it is also their characteristics that influences the results. But even if the used testing data would have been normalized or would have been created by a model aiming at describing how *average* personal data might look like or is organized, the individual effect for the user and his personal way of organizing his stored information would not be reflected. The used ways to deal with information are simply to different as shown by William Jones et al. in the publication about organizing personal information to get things done [JPGB05]. For this reason, it is deemed accurate to focus on real life examples of personal data and to look then for generalizable effecting characteristics that increases or decreases the benefits for applying the concept.

The data used for the tests comprises a two-year's snapshot of personal data of the author, accumulated during the typical desk work occurring in the lifecycle of research projects. It therefore contains rich email communication and organizer usage, as well as large repositories of documents, which is deemed typical for the domain of information workers. This data is further specified in the next Section 6.1.1, which uses the data characteristics to evaluate its processing effort that is required to create the networked representation in such an information domain.

6.1.1 Data Processing Performance

Initially the Topic Maps engine TM4J[1] was used together with in-memory processing. All topic map data were persistently stored in XML files, which are loaded into memory during server startup. The advantage of this approach was a very fast read and write access during topic map modification, but the amount of generated topics soon exceeded the economic limits given by memory restrictions. Although the use of virtual memory potentially increases the size of topic maps to some degree — causing also thrashing during the process of memory swapping, because of the decentral access to data portions scattered over many different memory pages —, the memory size would have still become a non-acceptable limiting factor for data capacities of realistic working environment. Therefore the proof of concept implementation was extended to only work with a small amount of in-memory data, keeping the rest of the data persistently in a database (see Section 5.3.1). Even with applied caching techniques, the overall read and write performance now becomes a crucial aspect for the intended usage pattern. Therefore it is necessary to measure access times to evaluate the effort and complexity for creating and accessing the knowledge representation with the concepts that should benefit from it. Especially read access pattern while navigating inside the representation, graph-based search queries and calculation of entrance points to the representation require an optimization for

[1] Topic Maps For Java (see http://www.tm4j.org/)

Table 6.1: *Metrics of origin Data Sources*

data source	items	time-frame	topics	assoc.
(M1) emails	8442	Jan. 2006 - Dec. 2007	15684	64243
(M2) organizer	675	Jan. 2006 - Dec. 2007	1287	2729
(M3) documents (papers)	1055	Jan. 2006 - Dec. 2007	2774	5637
(M4) documents (projects)	3234	Jan. 2006 - Dec. 2007	7000	15583

database processing of the query and an application-side trade-off between database access latency and in-memory storage limitations.

The metrics of data used for measuring are shown in Table 6.1. In this table, the number of items denotes to the characteristic information objects in each data source the extraction is iterated over, such as emails, tasks, events, files and folders. Table entries with the generated topics and associations give a first impression about size and complexity of the created topic maps before the content is merged to a global representation.

Before the performance results are described, first the used example data sources are presented. The data source M1 (an IMAP folder with subfolders) contains received and sent emails within the shown time-frame. SPAM was filtered out via server side rules and client based actions, before the extraction took place. All contained attachments are processed by the metadata extractor. Added topics and associations of this extractor are included in the shown values.

From data source M2 (a WCAP compatible calendar server) the author's organizer data was retrieved. It contains dates of appointments together with locations and email addresses of attendees, as commonly used in iCalendar[2] conform applications.

The next data source M3 is a local filesystem directory and its subfolders, which contains a collection of gathered scientific documents. Formats comprise of Microsoft Office files (DOC, PPT, XLS), Portable Document Format (PDF) files, PostScript files, plain text files and compressed ZIP files. From these files, the Microsoft Office files and PDF files are processed by the metadata extractor to retrieve information about authors, titles and keywords.

Data source M4 is also a repository of documents. Its main difference to data source M4 is its content location and authorship. In contrast to M3, the documents are stored in a remote document repository containing project documentation the author was involved in.

[2] Internet Calendaring and Scheduling Core Object Specification; defined in RFC 2445 (see http://www.ietf.org/rfc/rfc2445.txt)

Representation Creation

A crucial aspect of using a digital knowledge representation is the calculation effort required to create it. The main focus in this analysis is therefore to verify that the computational problems contained in the used concepts and technologies do not limit its application, like an exponential growth of the required effort would soon cause a low efficiency for realistic data source sizes. All tests were performed with the *eXist* database embedded mode running in a *SUN Java Standard Edition Runtime Environment (JRE) version 6* on a Windows XP 512MB notebook powered by a 1.2Ghz Celeron M processor. This low performance system is chosen to make sure the system can also be used on older computers and does not require expensive new hardware.

The tests in this section were performed with the calendar server extractor representative also for other extractors. Measured times for extraction of calendar entries in Table 6.2 also contain the time for accessing the calendar server via intranet, because topic creation and data source access is performed in parallel. Because the response time of data sources varies, the times are measured multiple times during different daytimes to collect a realistic average processing time. For these tests, the calendar entries were created randomly to produce a virtual calendar with 1000 event entries per year, which was repeated to fill the calendar for 5 years. These entries are then accessed in ranges for 1, 3 and 5 years to compare the extraction performance for three sizes of calendars.

It should be noted that the absolute processing times significantly depend on the amount of redundant information inside the calendar data (e.g., equal attendees, locations etc.), because it determines the total amount of generated/stored unique topics. However, comparing the results for different sizes of calendars that were built with the same redundancy patterns for one year provides the possibility to analyze how the extraction performance changes if the size of the data source increases. This comparison is therefore independent of the data structure and from the amount of redundant information inside the calendar data.

Table 6.2: *Effort for creating Representation Topic Map*

action	items	topics	assoc.	$t_{avg}/item$
	1000	1449	4998	114.23 ms
extracting data	3000	4347	14996	236.53 ms
	5000	7245	24992	414.75 ms
	500	250		9.23 ms
merging extractor maps	5000	2500		23.81 ms
	50000	25000		361.60 ms

One might expect the extraction time to grow proportional to the amount of data entries, which would require that all contained operations can be performed in a constant time on average, independent of the amount of data already stored in the database. However, the *extraction process* contains a processing step that checks for the existence of equal topics and associations, which have to be done for every new topic and association to keep the topic map consistent. Therefore the overall extraction time mainly depends on write and lookup performance of the database.

As shown in Table 6.2, the average time for processing one item increases in case of the proof of concept implementation. This is caused by an increasing response time for a lookup of topics via its subject indicator (despite the proven efficiency of the used database index shown in Figure 5.5) for growing datasets inside the used representation database *eXist*, whose lookup of a single topic nevertheless still can be considered as fast for xml-based databases. However, for this type of application, the huge amount of small update and lookup operations during the extraction phase accumulate to the shown values. This accumulation of increasing access times can constitute a problem if tens of thousands single lookup or write operations have to be performed in a given time frame.

As a result, the time for extracting data mainly depends on the organization of the database. Therefore a database strategy should be chosen for the digital knowledge representation that is also optimized for write and lookup times in case of large sets of topics, to provide an adequate extraction time for large data sources.

The *merging process* mainly depends on the write performance that is necessary to store the merged topics efficiently. In this process multiple topics are retrieved from the database in one request to be compared with those of the topic map to be merged in. For the considered example, two maps of equal size are merged. This is the worst case for this process since the algorithm compares all topics of the smaller topic map with those of the larger. In Table 6.2 the following measured values are shown: the total amount of extracted items for both topic maps (the source items are equally split-up to both maps); the amount of topics after merging them together; and the resulting time of this process per item on average. For this worst case simulation, all topics of the first topic map will be merged with a topic from the second topic map. As a result in this worst case all topics in the destination map have to be updated.

Like in the tests for the extraction process, the measured times show as one result that the merging process mainly depends on the performance of the database. For large representations it is therefore important to use a database that is capable of providing fast write and lookup times even for large sets of topics, to speed up the merge process for larger representations. For the representation created from the two year's snapshot of personal information (see Table 6.1), the database implementation is deemed sufficient nevertheless. Especially because the extraction of a complete data source and its full merging only has to be performed once. After this step, only newly added entries will be extracted and merged into the representation, which have shown to be very efficient for the considered amount of new entries per day. However, the decreasing write performance of the used database still denotes a limit for the scalability. Additional counter measures therefore have to be performed on database level, not on the concepts of personal digital knowledge.

6.1.2 Representation Characteristics

The characteristics of the digital knowledge representation are a further result of this work. Using metrics known from graph theory, the autonomously created representation is compared in this section with the conceptional description of Chapter 3. The data sources used for creating the representation are described in Table 6.1. The structure and redundancy contained in the described data sources are taken from a real working environment and therefore should be analyzed for characteristics of digital knowledge.

The graph used to perform the analysis is built from the representation with the model presented in Section 3.5.2. Therefore, topics represent vertices, and edges of the graph are created by Topic Maps associations and additionally by references of topics to other topics, like in case of type topics. This virtual graph built from the representation is then used to derive[3] information about its connectivity and global structure.

A major aspect on digital knowledge is the *connectivity* between the information parts it is built of. A high connectivity is one indicator for a rich dependency between data contained in different data sources, and thus also would be a sign for a successful process of bringing together pieces of information which were initially split apart because of separated ways of handling its parts. In this section, the following graph metrics are applied on knowledge representations:

- degree of connections to other topics
- clustering coefficients of topics
- articulation points in the representation
- distances between topics
- similarity to *small-world networks*
- redundancy in data source and merged representation

[3] The free Java Universal Network/Graph Framework (JUNG; see http://jung.sourceforge.net/) was used to perform the calculations.

Degree of Connections

A first connectivity metric to be applied to the representation is to count the connections of every topic to other topics. Calculating this value for every topic (i.e., calculating its degree) and building the average over all values creates the *average degree* of the graph. It shows the average number of topics each single topic is connected with. This average degree of a representation can be compared to other representation and to its changes when these data sources are merged (see Table 6.3).

As shown in Table 6.3, the average degree significantly varies for the sample extractors. This is caused by the unequal amount of metadata extracted from the sources, but also from a few heavily interconnected vertices. These hubs are created by hierarchical structures (e.g., many entries of a directory) and the accumulation of relations to equal properties. This is also indicated by the high values of the standard derivation. The decreasing average degree of the merged graphs is caused by the differences of its source graphs. Therefore, this observation can not be used to indicate properties of the created connectivity between data sources. Even if the average degree would have increased for the merged graphs it would have been only a weak indicator that equal topics contained in different data sources are merged together. During merging, the connections of the merged topics to other topics are accumulated on one topic instance, which increases the degree of the resulting topic. Thus, the redundancy of topics between representation and added data sources can also have an effect on the average degree.

However, the average degree of the representation would also increase if the included topics have a higher degree on average, but are not merged with existing topics of the representation. Therefore this metric alone is only applicable, if the added topics have a comparable average degree. But the impact of merging on graph structures can also be visualized directly by calculating the redundancy of data between the representation and newly-added data source. It is directly computable during the merge process, discussed later in the subsection *leveraged redundancies*.

Clustering Coefficients

For a further analysis of the connectivity between data sources, the topology of the representation has to be considered, too. A common metric for the topology is the *clustering coefficient* [JUT07].

Definition 6.1. *The* local clustering coefficient C_i *for a vertex* v_i *is given by the proportion of edges between the vertices within its neighborhood, divided by the number of edges that could potentially exist between them. The set* S *contains the neighbors of vertex* v_i *in an undirected Graph* $G = (V, E)$*. The neighbors of vertex* v_i *can have* $\frac{|S|(|S|-1)}{2}$ *edges among each other. The neighborhood* N_i *for a vertex* v_i *is therefore* $N_i = \{v_j \mid e_{ij} \in E\}$*. Then the local clustering coefficient* C_i *of* v_i *is defined to* 0*, if* $|S| = 0$*. If* $|S| = 1$*,* C_i *is defined to* 1*.*

For $|S| > 1$,

$$C_i = \frac{2|\{e_{ij}\}|}{|S|(|S|-1)} : v_j, v_k \in N_i \text{ , } e_{jk} \in E \tag{6.1}$$

In Definition 6.1, the local clustering coefficient is the fraction of a vertex's neighbors that are also neighbors of each other. For the representation this implies that a high local clustering coefficient is the result, if many topics connected to the considered topic also have associations between each other. This in turn would indicate the conjunction of different relations types (e.g., hierarchical, type and property relations) to and between the adjacent topics. This is explicable because examples are very rare for situation in which a single relation type creates a cycle over three or more topics (hence the same relation type would connect multiple adjacent topics among themselves and also with the considered topic). Therefore, a high local clustering coefficient in digital knowledge representations indicates multiple relation types among itself and topics of its neighborhood.

For the analysis of a complete representation, this metric can also be extended for the whole graph, which is then called *average clustering coefficient*.

Definition 6.2. *The average clustering coefficient (ACC) is defined as*

$$\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \tag{6.2}$$

The results in Table 6.3 show a very low ACC that decreases if more data sources are included into the representation. Considering also the low values for the standard deviation, the majority of the structure contains no short cycles, because the majority of the adjacent topics are not directly connected to each other. Instead they are connected via one additional topic (e.g., an equal type or property), reached from another topic. This close structure is also the reason for the low value for the average shortest path, as shown below.

A second conclusion of the low ACC is that the topology formed by the graph is also tree-like by the majority. This is caused on the one hand by the strong data source hierarchies (e.g., folders and sub-hierarchies), preserved for the user to serve for an additional navigation. Additionally, the properties of information object entries often form child nodes not directly connected to its siblings.

Articulation Points

As shown in the previous section, the observed graph has characteristics of hierarchy structures. So, if the graph would be *only* connected by this hierarchy, there should exist topics called *articulation points* that build a bridge between two or more subgraphs containing *biconnected components*.

Definition 6.3. *Let $G = (V, E)$ be a connected, undirected graph. An* articulation point *of G is a vertex whose removal disconnects G. A* biconnected component *of G is a maximal set of edges such that any two edges in the set lie on a* common simple cycle*.*

Although the graph contained many hierarchical relations shown in the analysis of clustering coefficients, no articulation points are contained in the graph, as shown in Table 6.3. This indicates that at least one additional path outside the hierarchical relations must exist, caused by the interconnection between properties and types of data sources.

Distance between Topics

A further necessary metric for the connectivity of the described graph characteristics is to calculate the average distance of the graph, also known as the *characteristic path length*.

Definition 6.4. *The average distance $\mu(G)$ of a connected, undirected graph $G = (V, E)$ is the average of the shortest distances between all pairs of vertices of G. The shortest distance $\delta(s, v)$ from a given source vertex $s \in V$ to a vertex v is the minimum number of edges in any path from s to v, or else ∞ if there is no path from s to v.*

The first result derivable from the computation of this metric is that the graph is connected, since, if topics can not be reached from all other topics via a path, the result would have been an infinite value (cp. Table 6.3). This underlines the compliance to the facet design described in Section 3.3.1, which connects all extracted data sources with a designated root.

Small-World Networks

For the connectivity, the low value of shortest paths in Table 6.3 indicates a strong interconnection (via more than one frequently used node, because otherwise there would have been articulation points) between all included topics. This issues the question if the graph can be compared with other types of strongly connected graphs. A fitting model was found with the *small-world networks*. These are generally characterized by a high global connectivity and a high local clustering [Xu07, Wat99]. A random generated graph G_{sw} that is built with small-world network characteristics[4] of a similar size (lattice size set to 125 — representing a graph with 15625 vertices — and α set to 1.05) shows a metric value of 0.043 ± 0.001 for the average over the ACC of G_{sw} and a low standard deviation across 100 generated graph instances. Comparing this metric with the values of the graph M1 in Table 6.3 indicates a similarity of the overall topology, even though M1 is much more interconnected, which is visible from

[4] Generation of a graph with the JUNG Java API implementation of the Kleinberg small-world graph model, as published in [Kle00]

Table 6.3: The table shows the results of the connectivity metrics: (i) articulation points (art. pts.), (ii) average degree (avg. deg.), (iii) average clustering coefficient ($\bar{C}$) and (iv) average distance (μ). The standard derivation is indicated after the ± symbol for all values.

data sources	art. pts.	avg. deg.	$\bar{C}$	μ
(M1) emails	0	10.18 ± 102.45	0.043 ± 0.064	2.97 ± 0.85
(M2) organizer	0	6.31 ± 35.05	0.003 ± 0.056	2.67 ± 0.64
(M3) documents (papers)	0	6.10 ± 30.95	0.007 ± 0.063	3.01 ± 0.76
(M4) documents (projects)	0	6.47 ± 60.74	0.006 ± 0.048	2.79 ± 0.69
(M1+M2)	0	10.13 ± 100.56	0.041 ± 0.064	3.09 ± 0.93
(M1+M2+M3)	0	9.60 ± 95.56	0.036 ± 0.064	3.31 ± 1.01
(M1+M2+M3+M4)	0	8.85 ± 94.17	0.028 ± 0.061	3.49 ± 1.04

the lower shortest paths lengths compared to this metric of G_{sw} with values beyond 8. However, the topology doubtless has similarities to the local and global connectivity characteristics of small-world graphs, which therefore can be used beneficial for further search and navigation interfaces. This result also shows that the responsible extractors for M2-M4 do not create the same interconnectivity between extracted facts. This might be caused by their plain data structures, a lack of usable metadata and a missing interpretation of facts between single data source entries.

Connectivity Characteristics

A further result of Table 6.3 is the conclusion that the individual abilities of the extractors to create highly connected topics maps affect the metrics of the overall connectivity. In the actual case, the sources M2-M4 can not keep up with the interconnected structures of M1 and therefore have lowered also the connectivity metric for the merged representation. However, the merged representation M1-M4 still has a beneficial structure. Its short average path length (together with its low standard derivation) shows the successful combination of heterogeneous data sources into a homogeneous representation. A much shorter value would have made the meaning of calculated paths meaningless (connecting nearly every topic to all other), whereas a higher value decreases the relation between the start and end vertices. The absence of articulation points indicate the absence of isolated sub-graphs and the values for the clustering coefficients shows the dominance of multidimensional tree-like structures, which provides good search characteristics and no dead ends during navigation and browsing inside the graph.

Summarizing the observed graph characteristics, the following general statements can be derived so far from the graph's topology:

- The graph G created by the presented design is connected in any case, which provides the foundation for a global navigation inside the graph. The higher the average degree of vertices, the richer every information object is described by extracted properties that itself are also connected to other information objects.
- The topology of the representation is mainly formed tree-like by the combined multi-hierarchies collected by the extractors. No cliques are formed, as the majority of topics are not directly connected to its siblings, which decreases the connectivity. But on the lower end of connectivity also no articulation points are inherent to the design. Therefore any contained clusters are multiply connected to other parts of the graph, which increases the opportunities to access wanted information via such a path.
- Similarities to small-world network characteristics can be also found in personal digital knowledge graphs. This indicates possibilities to apply research results of that field also to these graphs. A concrete example is the ability to identify the connection to topics that represents a first branch inside a short path that lead out of the local cluster, by using only a local knowledge approach. These topics are of interest to the user because they represent a relation to information of another cluster and its context.
- The observed short average lengths of paths provide very fast computation results for the path search, because it increases the probability that only a fraction of the graph has to be inspected to find the path. Additionally, the local context character of short paths between any possible combination of input values also increases the value of uncovered connected topics, in contrast to those topics that might be on a longer, lesser related path.

Leveraged Redundancies

Besides the characteristics of the graph's topology, two further important metrics for digital knowledge representations have to be considered:

- the amount of data that has been merged inside an extractor, and
- the amount of data merged from different sources.

The first metric indicates redundancy inside the data source that is used by the extractors to strengthen the visibility of relations between information types in the considered data source. Whereas the second type of redundancy is an indicator for relations between information that would be not usable by the user without the digital knowledge approach. Often this information was once entered and stored during a single process, but split-up into parts because of the separation of the used data sources (like the received email confirmation of a meeting organized in the separated personal calendar). But it applies also to information previously not related at first glance, which nevertheless now can help the user to find related items (like the extracted author information from a stored document in a filesystem data source, merged to the author information of received emails).

Definition 6.5. *The redundancy R is defined in this work as the number r of items that can be removed from the data source without reducing the contained information, divided by the number n of all items contained in the data source. As a global metric for Topic Maps, n is set to the sum of all unique topics u_t and unique associations u_a, plus the sum r of removable topics r_t and removable associations r_a.*

$$R = \frac{r}{n} * 100 = \frac{r}{r+u} * 100 = \frac{r_t + r_a}{r_t + r_a + u_r + u_a} * 100 \tag{6.3}$$

An analysis of increased connectivity therefore is performed by measuring the local and global redundancy of data sources. This way the created representation after the merging process can be compared to the previously plain data sources. This statistical data of the merging process is directly accessible in the proof of concept implementation.

The *local redundancy* is calculated by counting creation attempts of redundant topics and associations during the extraction process before they are replaced by the single instance representing the unique holder of this information part. Results of this analysis for the data sources described in Table 6.1 are shown in the first four rows of Table 6.4.

The *global redundancy* is calculated during the merging process of topic maps. It expresses the percentage of information that is contained in both maps to be combined and therefore is being merged to one instance. Since every merged instance provides an interconnection between both data sources, the global redundancy indicates the connectivity between information of different data sources. Statistical results of this merging process are shown in the lower four rows of Table 6.4. These rows show the linear process of merging the extracted topic maps one by one into the representation. The symbol + denotes the merge operation and the identifiers inside parentheses represent the content of a merged topic map.

In Table 6.4, source M1 clearly offers the most local redundancy. This is quite rational since email folders contain reoccurring information in every entry, such as the relation between the name of a person and its email address. In contrast the other sources do not contain such simply derivable information, visible in the lower values for the local redundancy and a missing redundancy of associations listed in the column r_a.

The global redundancy values indicate the amount of information that is used to interconnect separated data sources. Although the values are much lower than those of the local redundancy, they show the existence of content that can be merged. As

Table 6.4: *The table shows the usable local and global redundancy in the denoted data sources. For each topic map the amount of unique topics (u_t), unique associations (u_a), removable topics (r_t) and removable associations (r_a) is shown. The first four rows show the local redundancy (R), whereas the following four rows show the global redundancy after merging.*

topic maps	u_t	u_a	r_t	r_a	R
(M1) emails	15684	64243	62731	22345	51.51 %
(M2) organizer	1287	2729	1481	0	26.94 %
(M3) documents (papers)	2774	5637	2908	0	25.69 %
(M4) documents (projects)	7000	15583	8628	0	27.64 %
(M1) + M2	16440	66972	531	0	0.63 %
(M1+M2) + M3	19086	72609	128	0	0.14 %
(M1+M2+M3) + M4	25708	88192	378	0	0.33 %
(M1) + (M2+M3+M4)	25708	88192	1037	0	0.90 %

an example, about 0.9 percent of the content contained in the merged representation of M2+M3+M4 is also contained in M1. Thus, by merging the data sources into a homogeneous representation, new relations are created for 1037 information entities, regardless of the sequence of merging. This enriches the possibilities to find these entities, but also those in a close context. Therefore, the overall aggregated value of 1.1 percent[5] for the three merging processes is already a justifiable value when considering that the user does not have invested any additional effort and the created interconnections are usable across all data sources.

Derived Characteristics

Form these experimental results, general characteristics can be derived regarding supporting or unfavorable effects of personal organizing methods and the kind of processed information. As shown in the analysis above, the redundancy between data sources also depends on quality and quantity of metadata contained in the information items. Especially the metadata contained in documents have shown to create a strong impact on the interconnection of data sources in the digital knowledge representation. For the inspected data, currently only the minority of those files make fully use of this functionality. In their current working environment, most authors do not have any advantages from investing additional effort for filling out these metadata fields during content creation, because this metadata is not used actively and is only of an increased interestingness for documents of other authors. With an automated processing by digital knowledge systems this situation changes, because in this case any additional metadata induces a direct improvement of the

[5] Overall redundancy value during merges as of Table 6.4: 0.63% + 0.14% + 0.33% = 1.1%

own digital representation with an observable benefit for participating users. This creates a motivation to pay attention on setting up document creation software to include correct and meaningful metadata into own documents; with the benefits also for recipients of these documents.

A major unfavorable effect on interconnecting information parts is created by file-based encryption. This important security technique, which is necessary to protect information in situations where no other specific protection is available during information communication or storage, prevents the efficient extraction of metadata from encrypted documents as well. The common usage of PGP encrypted attachments leaves a user only two options: either those content is only partly integrated into the knowledge representation (currently the case in the proof of concept implementation), which makes it harder to find it, or the extractors have to be extended with decryption support and an access to the private key ring of the knowledge representation owner, at the cost of an increased resource consumption. Therefore, the general usage of file-based encryption in the daily management of information decreases the advantages achievable with digital knowledge management.

However, also the way metadata is created by the user effects its automated processing. Although the described design does not required the user to adhere to a special convention, using an individual loose naming schema for names of locations, events or subjects increases the advantages of its implicit created interconnection. This does not require the user to strictly follow his own rules. It is more like a tendency to use similar words in reoccurring situations. Even if the user does this with low accuracy, it increases the connectivity and with it the advantages for searching. In this case, clusters of topics with similar descriptions are created, connected to each other via relations to other shared properties. Via these relations, the interconnected clusters help the user to navigate and search inside the representation. Therefore even a low accuracy in using an individual naming schema increases the observable advantages for the user and induces a motivation to stick to that schema.

As a final conclusion of this section, the experimental results have shown to support the Assumptions 3.1 and 3.2: There are recurrent information types in the original data sources which describe recurrent data and the usage of this redundancy has created a usable benefit for the structure of the autonomously created representation.

6.1.3 Communication Characteristics

The mobile use of the prototype provided insights about the characteristics of latency and volume of the communication. Of course, the latency for wireless networks is well known for the pure network level, but for a user the whole round trip time is of much greater importance when using a service with a corporate backend from their mobile device. It is the response time for an issued command from the mobile device, via gateway and corporate server, back to the device. These times are measured for typical use cases with the example data described in Section 6.1.1. Because of the different complexity for the corporate server caused by various commands, a standard use case was created that contains all typical interactions with the server in a single session. Then this scenario was executed multiple times in various places and during different times of the day, since the network throughput also depends on these factors. The calculated result for a single interaction amounts to 3.17 ± 1.21 seconds on average, such as retrieving a non-cached document list or submitting a request to distribute a document. These times were deemed acceptable, since the majority of tasks in mobile scenarios only require 5 to 7 interactions with the prototype implementation and therefore the overall time only amounts to less then 30 seconds for experienced users. Additionally, there is still room for improvement on the server side, which could affect about 50% of the meassured roundtrip time.

The same use case scenario was used to measure the effective volume of transmitted data. During the typical scenario of searching and distributing a file to a recipient, 3015 bytes were sent and 7430 bytes were received on average. Looking at the communicated raw data of 5662 bytes on average, an overhead of about 50% was observed, mainly caused by SSL/TCP/IP overhead. This could be lowered with the use of UDP and a customized encryption schema that avoids exchaning the somewhat bulky certificates in the case of TLS. However, the use of UDP would also require a wireless network that does not prevent two-way UDP communication and the use of a customized encryption schema could also affect the trust into the applied security. Furthermore, the observed data volume is very low compared to the amount of data that would have been caused by the classic way. In this case the complete data would have to be retrieved from the original data source via VPN including the caused overhead by a classic transfer protocol and then it would have been sent from the mobile device to the recipient.

6.2 Abstract Findings

In contrast to the findings of the previous section, this section presents findings whose exact measurement is out of scope of this work. These findings are nevertheless presented to document first observed impacts on organizing information work, applied search strategies, as well as security implications found during using the implementation. These abstract findings are derived from the work with the prototype and should serve as basic observations from practical experience.

6.2.1 Search Strategies

This section describes example use cases, solved with the proof of concept implementation. It is intended to provide insights into applied use cases performed during this research. It shows benefits for test users of the proof of concept implementation focused on the information worker domain.

While Working on the Desk

Many of the common tasks in the office worker domain are bound to the information item "time". It arranges events in a chronological order, which is a naturally usable dimension for searching information. In MIDMAY the dimension time is directly accessible even across data sources. So questions like *"What happened on 1st of October 2008"* can be answered by navigating to the topic that represents this date. Via the displayed associations the user than has access to all information items connected to that date. In this scenario a user could also use the available search interface of the original data source or a desktop search engine, so MIDMAY offers the same possibilities with the advantage of a one-stop interface. The main advantages are introduced by MIDMAY if the question is *"What happened on 1st of October 2008 to project YAMDIM?"*. Searching is as easy as for the first example, since now only a second topic (representing the project YAMDIM) is added to the search. Given this search request, the user can explore all items that are related to the project and to the given date. Again this works across all data sources (e.g., related emails, files, attachments, meetings, etc.) just by MIDMAY's autonomous combining the affiliation of persons extracted from the LDAP directory with all the other registered data sources. But there is also the possibility to narrow the search to one data source by just additionally adding its representing data source topic to the search request.

Another dimension that frequently came useful during the tests where related to persons. Questions of the kind *"What do I know about John Doe?"* were found to be answered fast with MIDMAY. Besides the retrieval of related information, also the history of activity with that person and the correlation to other facts can be gained from the created knowledge representation. This way of exploring the relations often reveals other interesting information besides the searched information.

All these powerful queries produces meaningful results because of the preserved context of the combined data sources. On the other hand, search functions from single data sources or global full text search engines do also offer possibilities to assist the user in finding answers to the above questions, but require the user to use filtering attributes that describe the searched content. In a direct comparison, searching in MIDMAY is about asking oneself *"What is related to the searched information?"*, whereas other search interfaces require the user to think of the *content* of the searched information. Therefore these approaches well extend each other, depending on the current search task and the recallable facts.

Other benefits for working processes where identified to result from a single interface navigation. It provides a universal interface to browse though any represented structures. These are user-created in the original data source and could have been used also there for the information retrieval and search. However, the presence of a single interface for these tasks was recognized as a more convenient way, especially because it avoids the need to authenticate against several data sources.

For the desk work, the possibility to distribute information while browsing for other things was observed as a major speedup for this type of task. However, the speedup is also partly attributable to the missing possibility in the current implementation to include additional comments for the recipient of the sent information, as one would do in the case of classic email communication.

While Meeting People

Another observed benefit from MIDMAY was caused while in the situation of meeting people. MIDMAY users are motivated to ask attendees for their electronic business cards, sometimes already prepared in their mobile phones. This motivation was caused by the fact that it makes the further communication much more easier for MIDMAY users than for those how would have to sync their phones with a notebook to be able to use the email address for a transmission of documents, which is deemed one reason why the business card exchange via Bluetooth currently does not play a major role during meetings. Using the opportunity to receive authentic contact information together with the certainty that it will be accessible immediately inside the own knowledge representation also encouraged users to take down these information inside their mobile phone instead of just writing it down somewhere. This changes also the search strategy applied for email addresses, since the representation also offers more accumulated facts from other sources than classic email software (e.g., related meetings with the owner of the address or authored documents of him).

The main difference in strategies where caused in this situations by the possibility to directly look for documents fitting to the current discussions during breaks, even if it is not stored on the mobile device. Attendees then received the new input via email on their mobile device and did not have to wait until the sender was back on his desk. Of course this requires the user to organize his data foresighted within data sources personally registered to the MIDMAY extractor framework.

While on Train

The prototype was also used while traveling. Beyond already described use cases, in this situation, benefits from MIDMAY are observed by providing the possibility to look for details of upcoming appointments, such as the location address one is heading to, and adjacent events. As this normally is a favored functionality of mobile device, of course also the various calendar functionalities available for this purpose could have be used, if synchronized recently offline or online. So one reason for using this functionality from MIDMAY is the possibility to perform also other tasks in parallel in the same interface, like submitting relevant documents for the meeting in advance or checking if new emails or documents have arrived in the meantime from attendees without having to search for this information. These are all directly accessible via paths from the topic representing the event. Using this connectivity autonomously organize also newly received information, regardless of its data source type. Especially during travels this can be a smart help to keep informed about certain activities of certain project processes.

6.2.2 Security Implications

The proof of concept implementation also provided the opportunity to evaluate the security design in real life situations. Since no other systems with an equal functionality exist, the implementation is compared to mobile VPN solutions, which provide a comparable access to stored content on remote servers, even though the VPN functionality requires additional technologies for the actual information access. What risks are inherent to current VPN usage and does MIDMAY expose managed information to other new risks? These questions are focused on the security on usage level rather than a technology evaluation, which makes it necessary to first specify how VPN is commonly applied for corporate networks.

From a user's perspective, access to the corporate network is granted via VPN by providing some credentials (passwords, one-time codes, asymmetric keys on a smartcard, etc.). Access to the desired information is then possible via additional credentials for the hosting data source. An attacker with knowledge of the VPN credentials therefore can reach all services provided in the internal network, but requires additional credentials for each data source to access stored information, assuming the credentials of the attacked VPN user account are not used for other services, too. Since this can not be ruled out in general and single-sign-on solutions are often used to simplify the access, the information's security depends on the strength of the initial login credential and its unavailability to the attacker.

On this level, the same applies to the security concept of MIDMAY, since it can be also seen as a single-sign-on approach to the managed data sources. The initial password and the possession of the registered mobile device grands access to the managed information sources via a single interaction interface. From a security perspective, however, this user level access is a major difference to the VPN's broad network access, which can only be tightened by applying firewall rules on network and application protocol level, but not on user level. A server made accessible via VPN for a certain protocol is therefore directly reachable from all users (at least via that protocol), regardless if a user account on that server exists or not. In MIDMAY these intranet servers are only reachable if they are registered for the considered user and, even more important, the access is only possible on the logic level. A manipulation of network packets to exploit server vulnerabilities is therefore ruled out. This increases the protection of the overall intranet infrastructure.

On the other hand it has to be noted that the ubiquitous usage of the information access also requires the user to improve his security awareness. Passwords entered in crowded public places have shown to be prone to be spied out and the additional protection provided by the possession of the mobile device has to be considered in conjunction with the increasing risk of an unnoticed theft the smaller the device is sized. Therefore one advantage of the presented security design is that the amount of transaction allowed per day can be restricted on application level, to limit the risk during the time a device theft remained unnoticed and the device credentials are not revoked on server side. This way the attacker would be not able to steal all managed information at once. However, this restriction is currently not fully effective against well equipped attackers, since these still could extract the device credentials (because of the missing secure storage on current mobile devices) and then return the device before the victim notices the theft. Of course the usage of external security tokens could mitigate this threat, but they are considered inconvenient for spontaneous actions with typically short interaction times, which is necessary to make a direct settlement of a current information management task reasonable while not on the working desk. Further research has to address this protection of user authentication with the integration of trusted and authentic mobile hardware.

Because of the advanced interaction control between the user and his information and the protection against network level attacks targeted at intranet services or infrastructure, the applied security concept of MIDMAY is considered being suited better for pure information access than the provided security by VPN and the various necessary application protocols on top of it. However, any kind of centralized information management access demands a strict security evaluation on software implementation and on network configuration level, which was out of scope of this work. Like for other enterprise solutions with access to information of potential high value for attackers, a continuous observation of the system's activity has to be performed anyway to reveal flaws and attacks.

Chapter 7

Conclusion

In this final chapter the proposed approach and the developed results are summarized in Section 7.1. Identified aspects for further work are presented in Section 7.2.

7.1 Summary

This thesis addresses the problem space of daily work with information in mobile and stationary scenarios. The proposed solution concept *"Ubiquitous Personal Information Management"* is based on the seamless combination of technologies for information management, mobility and IT security, with the goal to complement and support each other. In this combination, the proposed approach shows the benefits of designing homogeneous structures from existing distributed information sources to offer a convenient universal interface even on limited devices.

As foundation, autonomously creating an interlinked representation of available information in arbitrary data sources was achieved on the conceptual level by unifying hierarchical structures with the help of Topic Maps design patterns. Despite their task of unification, the patterns have also shown to preserve existing user-given structures. The next step in this work was to specify generic rules for the extraction of information that is beneficial for representing the user's stored content. These proposed rules select information that: (i) the user may recall, (ii) interconnects the data or (iii) classifies it to enrich the representation. This way the user can map parts of his recollection with the digital representation to find related information.

The necessary visualization of the representation and the interaction model was built on top of a generic path-centric concept. This way, all tasks with the representation can be performed with the same path-based interface. The closed interaction cycle on these paths is used for navigation, selecting search input, distributing information and for all other actions specified in the representation. Because of this recurrent concept, the user can perceive the interaction as a single process of remote controlling the representation. This combined interface for browsing, managing and searching was a first step to extend the possibilities users can gain from their stored data, without investing additional organizational effort.

The proposed novel concept of applying graph theory on knowledge graphs allows for a further improvement: an associative search functionality inside the autonomously created representation that takes the user's implicit way of organizing his data into account. The approach thereby leverages existing metadata and inherent structures across data sources, created by data occurring in multiple sources. In contrast to a keyword-based search, the user marks related information to reveal desired content. Thus, the universal interface is also used to let the user choose the input nodes for the search. This way, the proposed bidirectional Breadth-first Search algorithm provides an additional way of getting access to stored information. The different modes created for the search algorithm have shown various application ways beside a strict structural search, such as providing searches for equal properties, types or relations. Corresponding examples for the application of the graph search have shown how and when these can expand conventional keyword approaches.

The protection of the user's combined digital knowledge is a major aspect for ubiquitous personal information management. It demands the goal of a user-friendly security design that is both: capable of managing ad-hoc situations and on the other hand also uses the potential created by mobility, to withstand common attack scenarios. Therefore, the required protection was described on a conceptional level in analogy with Common Criteria protection profiles. By inspecting the assets to protect, the additional aspects for the solution concept on application level were identified. With this view, the described security objectives build the foundation for creating instances from the information management concept. As a result, these objectives can be achieved with existing cryptographic algorithms and common security principles, as long as implementation flaws are ruled out. However, since many of the daily security bulletins clearly proof implementation flaws to be a serious threat to data confidentiality, the security design is strengthen with a further line of defense against attacks aiming at compromising and manipulating the service. The outcome is a generic approach that can be applied on information services to help increase the effectiveness of operating system protection. Starting from a monolithic service architecture, the approach guides to separate functionality into isolated components and to enforce mutual guarding of relevant operations. An attacker then has to compromise at least 2 of n components to overcome the privilege restrictions and component guarding. This is considered a second line of defense, since compromising two independent components is considerable more attack effort in terms of required knowledge, resources and attack concealment, which reduce in consequence the risks of successful attacks. With this generic approach, an actual security architecture instance were created for the service providing the personal information management. The resulting service illustrates the possibility for the application scenario to arrange the functionality in components isolated by the operating system in the intended way. Together with the performed security considerations, the proposed architecture can be used for the protection of actual ubiquitous personal information management system implementations.

On client-side the special threats caused by mobility are considered by taking into account the increased risks of manipulation and theft. Although, currently some of them can only be counteracted by awareness training of the users, also the advantages of mobile device ubiquity are leveraged to provide a convenient way for secure information distribution to dedicated recipients. This way the mobile device combines the functionality of a remote control to personal information with the possibility to collect and manage authentic identities of known recipients.

Having implemented the design, results about the representation's topology have been collected and interpreted regarding expectations and assumptions made for the underlying concept. These are crucial aspects when discussing the benefits of the concept for users, since it significantly determines the feasibility of the digital knowledge concept. Regarding the data processing the current database implementation was deemed sufficient for the example data comprising a two year's snapshot of personal information. The representation created from this data have been analyzed and interpreted. This showed insights about the topology and other graph metrics, which can be used to further improve the representation's generation, but it also shows further possibilities for its interpretation to increase the benefits for the user. Also the measured communication characteristics showed the feasibility of the mobility aspects, since latency and volume remains in useful ranges.

With these results the concept of ubiquitous personal information management have shown its feasibility. The analysis has identified determining factors that influences the benefits users can expect. These are all realizable without having to wait for improvements of new common hardware and without having to invest additional management effort by the user or even having to adapt to a predetermined way of organizing personal information. Therefore, ubiquitous personal information management creates an additional way of helping the user to tackle his daily work with information.

7.2 Future Work

Major current stumbling blocks have been identified with missing trusted mobile hardware and the still improvable databases suitable for the demanding access patterns and the data volume created by representations of personal information collections.

Regarding the security aspect, already efforts are made to integrate trustworthy hardware components into mobile devices, since also other applications will benefit from a trusted security anchor inside the device. The Trusted Computing Group has proposed the Mobile Trusted Module that may be available in future devices to help protect the data's confidentiality, integrity and authenticity. With the availability of devices employing such modules in the mass market, mobile application could benefit from the security anchor if the modules potential can be brought up flawlessly to the applications by applying a fitting security design to increase the effort for an unauthorized access via physical attacks. The security design presented in this work then could be extended with a secure storage to keep the credentials for the remote access protected even in case of device theft by sophisticated attackers. The discussed attacks caused by software manipulation could maybe also be addressed this way, if the device — and with it the software — could authenticate itself to the user in a secure way.

Considering the conceptional aspects, the prototype already showed promising results for an improved management of information. Especially the concept of applying graph theory to search in personal repositories can be extended further in terms of flexibility and comprehensibility of the search process. The key aspect in comprehensibility is then combining the universality of the node concept together with novel interpretations of algorithms on graphs, such as similarity clustering with a dynamic weighting of connections. In contrast to the proposed static weight sets for different search modes, this dynamic adjustment of the weights calculated can improve the focus for the given context. Such an adjustment could be calculated for example using the local and average clustering coefficients.

Additional to the interpretation of preexisting structures also the generation of further context data — automatically generated by preexisting knowledge or ambient sensor devices — could enrich the representation with context information not contained in the original data sources. An example would be the use of the personal GPS-enabled mobile phone to automatically submit current position, velocity and direction of the user while working with the remote interface. This information could be correlated with already stored locations and with their interconnected date information from digital organizers. The mobile device then would become a context sensor that may also could perform local preprocessing of other data, like analyzing taken images or creating relations between persons via connection data of messaging and incoming/outgoing calls. The merging of these context information inside the representation with additional ontology graphs to increase the relation between similar human-created identifiers could further improve the accuracy of results by providing a semantic to commonly used words. Together with an intuitive user interface that provides suitable interaction with these functionalities, this paradigm will show more new ways to find information in personal collections.

Bibliography

[ACGSGS+05] Francisco Alvarez-Cavazos, Roberto Garcia-Sanchez, David Garza-Salazar, Juan C. Lavariega, Lorena G. Gomez, and Martha Sordia. Universal access architecture for digital libraries. In *CASCON '05: Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research*, pages 12–28. IBM Press, 2005.

[ACGSLJ05] Francisco Alvarez-Cavazos, David A. Garza-Salazar, and Juan C. Lavariega-Jarquin. PDLib: personal digital libraries with universal access. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 365–365, New York, NY, USA, 2005. ACM Press.

[ADJ05] Naresh Apte, Keith Deutsch, and Ravi Jain. Wireless SOAP: optimizations for mobile wireless web services. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1178–1179, New York, NY, USA, 2005. ACM Press.

[AGS97] Rakesh Agrawal, A. Gupta, and Sunita Sarawagi. Modeling Multidimensional Databases. In *Proc. 13th Int. Conf. Data Engineering, ICDE*, pages 232–243. IEEE Computer Society, 7–11 1997.

[Aha07] Na'ama Aharony. *On Ranking Techniques for Desktop Search*. PhD thesis, Technion - Israel Institute of Technology, Haifa, March 2007.

[Ahm03] Kal Ahmed. Beyond PSIs : Topic map design patterns. In *Extreme Markup Languages*, 2003.

[AKG+07] N. Asokan, Kari Kostiainen, Philip Ginzboorg, Jörg Ott, and Cheng Luo. Applicability of identity-based cryptography for disruption-tolerant networking. In *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 52–56, New York, NY, USA, 2007. ACM.

[AKS99] E. Adar, D. Karger, and L. Stein. Haystack: Per-User Information Environments. In *Proceedings of the 1999 Conference on Information and Knowledge Management, CIKM*, 1999.

[ATAdL06] Trevor Armstrong, Olivier Trescases, Cristiana Amza, and Eyal de Lara. Efficient and transparent dynamic content updates for mobile clients. In *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 56–68, New York, NY, USA, 2006. ACM Press.

[Bar05] Robert Barta. TMIP, A RESTful Topic Maps Interaction Protocol. In *Extreme Markup*, 2005.

[Ber06] Johannes Bergmann. Einheitliche Repraesentation heterogener Datenquellen mit Topic Maps. Master's thesis, Technische Universität Darmstadt, Germany, January 2006.

[BF03] Dan Boneh and Matthew Franklin. Identity-Based Encryption from the Weil Pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[BHH06] Atta Badii, Mario Hoffmann, and Jens Heider. MobiPETS-GRID - context-aware mobile service provisioning framework deploying enhanced personalisation and privacy and security technologies. In *Proceedings SOFTPLATFORMS*, 2006.

[BL06] Tim Berners-Lee. An readable language for data on the Web. http://www.w3.org/DesignIssues/Notation3.html, March 2006.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lasilla. The Semantic Web. *Scientific American*, pages 34–43, May 2001.

[Blu07] Bluetooth Special Interest Group. Specification of the Bluetooth System, Core Version 2.1 + EDR. http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/, July 2007.

[Bor00] Christian Borgelt. *Data Mining with Graphical Models*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Germany, 2000.

[BPH09] Raluca Budiu, Peter Pirolli, and Lichan Hong. Remembrance of things tagged: how tagging effort affects tag production and human memory. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 615–624, New York, NY, USA, 2009. ACM.

[BRSS08] Erik Buchanan, Ryan Roemer, Hovav Shacham, and Stefan Savage. When Good Instructions Go Bad: Generalizing Return-Oriented Programming to RISC. In Paul Syverson and Somesh Jha, editors, *Proceedings of CCS 2008*, pages 27–38. ACM Press, October 2008.

[BS05] Robert Barta and Gernot Salzer. The Tau Model, Formalizing Topic Maps. In Sven Hartmann and Markus Stumptner, editors, *Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005)*, volume 43 of *CRPIT*, pages 37–42, Newcastle, Australia, 2005. ACS.

[Bun08] Bundesamt für Sicherheit in der Informationstechnik. Mobile Synchronisation Services (MSS PP), Version 1.9. https://www.bsi.bund.de/cae/servlet/contentblob/480252/publicationFile/29289/pp0048b_pdf.pdf, 2008.

[Bun10] Bundesamt für Sicherheit in der Informationstechnik. Operating System Protection Profile (OSPP) Version 2.0. https://www.bsi.bund.de/cae/servlet/contentblob/1098082/publicationFile/88584/pp0067b_pdf.pdf, 2010.

[Bus45] Vannevar Bush. As We May Think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

[BVKKS08] Michael Bernstein, Max Van Kleek, David Karger, and M. C. Schraefel. Information scraps: How and why information eludes our personal information management tools. *ACM Trans. Inf. Syst.*, 26(4):1–46, 2008.

[CBH03] S. Capkun, L. Buttyan, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *Mobile Computing, IEEE Transactions on*, 2(1):52–64, Jan.-March 2003.

[CCH06] M. Cagalj, S. Capkun, and J. P. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, 94(2):467–478, 2006.

[CD06] Edward Cutrell and Susan T. Dumais. Exploring personal information. *Communications of the ACM*, 49(4):50–51, 2006.

[CDZ07] Sara Cohen, Carmel Domshlak, and Na'ama Zwerdling. On Ranking Techniques for Desktop Search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1183–1184, New York, NY, USA, 2007. ACM Press.

[CHB06] S. Capkun, J.P. Hubaux, and L. Buttyan. Mobility helps peer-to-peer security. *Mobile Computing, IEEE Transactions on*, 5(1):43–51, Jan. 2006.

[CHJ+03] Yih-Farn Chen, Huale Huang, Rittwik Jana, Trevor Jim, Matti Hiltunen, Sam John, Serban Jora, Radhakrishnan Muthumanickam, and Bin Wei. iMobile EE: an enterprise mobile service platform. *Wirel. Netw.*, 9(4):283–297, 2003.

[Cla99] Roger Clark. Fundamentals of 'Information Systems'. http://www.rogerclarke.com/SOS/ISFundas.html, August 1999.

[CLR90] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*, chapter 23, pages 469–471. The MIT Press, Cambridge, MA, 1990.

[DA04] Hongmei Deng and Dharma P. Agrawal. Tids: threshold and identity-based security scheme for wireless ad hoc networks. *Ad Hoc Networks*, 2(3):291–307, 2004.

[DHSW02] Erik Duval, Wayne Hodgins, Stuart Sutton, and Stuart L. Weibel. Metadata Principles and Practicalities. *D-Lib Magazine*, 8(4):1–16, April 2002.

[Die07] Kurt Dietrich. An integrated architecture for trusted computing for java enabled embedded devices. In *STC '07: Proceedings of the 2007 ACM workshop on Scalable trusted computing*, pages 2–6, New York, NY, USA, 2007. ACM.

[DKF+03] Pratik Dave, Unmil P. Karadkar, Richard Furuta, Luis Francisco-Revilla, Frank Shipman, Suvendu Dash, and Zubin Dalal. Browsing intricately interconnected paths. In *HYPERTEXT '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 95–103, New York, NY, USA, 2003. ACM Press.

[DSS93] Randall Davis, Howard E. Shrobe, and Peter Szolovits. What Is a Knowledge Representation? *AI Magazine*, 14(1):17–33, 1993.

[DSTZ05] M. Debbabi, M. Saleh, C. Talhi, and S. Zhioua. Java for mobile devices: a security study. *Computer Security Applications Conference, 21st Annual*, pages 10 pp.–, 5-9 Dec. 2005.

[DSTZ06] Mourad Debbabi, Mohamed Saleh, Chamseddine Talhi, and Sami Zhioua. Security Evaluation of J2ME CLDC Embedded Java Platform. *Journal of Object Technology*, 5(2):125–154, 2006.

[FDMC05] S. Feldman, J. Duhl, J. R. Marobella, and A. Crawford. The Hidden Costs of Information Work. IDC Information and Data, USA, http://www.factiva.com/factivaforum/2005/frankfurt/TheHiddenCostsOfInformationWork.pdf, 2005.

[Fie00] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, UNIVERSITY OF CALIFORNIA, IRVINE, 2000.

[Fle06] Simon Fleischer. Framework zur sicheren verteilten Kommunikation ohne PKI Unterstützung. Master's thesis, Technische Universität Darmstadt, Germany, November 2006.

[Flo03] Luciano Floridi, editor. *Blackwell Guide to the Philosophy of Computing and Information*, chapter The data-based definition of information. Blackwell Publishers, Inc., Cambridge, MA, USA, 2003.

[FPJea00] M. Flynn, D. Pendlebury, C. Jones, and et al. The Satchel system architecture: Mobile Access to documents and services. *Mobile Networks and Applications*, 5:243 – 258, 2000.

[Fre02] Eric Freese. So why aren't Topic Maps ruling the world? In *Extreme Markup Languages*, 2002.

[FS03] Niels Ferguson and Bruce Schneier. *Practical Cryptography*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[FV06] S. Feldman and R. L. Villars. The Information Lifecycle Management Imperative. IDC Information and Data, USA, July 2006.

[FWWe93] Tim Finin, Jay Weber, Gio Wiederhold, and et.al. Specification of the KMQL Agent Communication Language. http://www.cs.umbc.edu/kqml/papers/kqmlspec.pdf, June 1993.

[Gar03] Lars Marius Garshol. Living with Topic Maps and RDF. http://www.ontopia.net/topicmaps/materials/tmrdf.html, 2003.

[Gar04] Lars Marius Garshol. Metadata? Thesauri? Taxonomies? Topic Maps! *Journal of Information Science*, 30(4):378–391, 2004.

[Gar05] Lars Marius Garshol. TMRAP - Topic Maps Remote Access Protocol. http://www.garshol.priv.no/download/text/tmrap.pdf, 2005.

[GB04] G. Gehlen and R. Bergs. Performance of mobile Web Service Access using the Wireless Application Protocol (WAP). In *Proceedings of World Wireless Congress. Sanfransico, USA*, 2004.

[GBL$^+$02] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. Mylifebits: fulfilling the memex vision. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 235–238, New York, NY, USA, 2002. ACM Press.

[GLLR06] Lawrence A. Gordon, Martin P. Loeb, William Lucyshyn, and Robert Richardson. Computer crime and security survey 2006. http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2006.pdf, 2006.

[Hal04] Benjamin Halpert. Mobile device security. In *InfoSecCD '04: Proceedings of the 1st annual conference on Information security curriculum development*, pages 99–101, New York, NY, USA, 2004. ACM.

[Hay92] R. Hayes. The measurement of information. In Pertti Vakkari and Blaise Cronin, editors, *Conceptions of Library and Information Science*, pages 268–285. Taylor Graham, London, 1992.

[HB06] Jens Heider and Johannes Bergmann. TopicMaps: Unified Access to Everyday Data. In *Proceedings of I-KNOW '06, Graz, Austria*, pages 473–480, 2006.

[HB11] Jens Heider and Matthias Boll. Lost iPhone? Lost Passwords! Practical Consideration of iOS Device Encryption Security. http://www.sit.fraunhofer.de/Images/sc_iPhone%20Passwords_tcm501-80443.pdf, February 9 2011.

[Hei04] Jens Heider. Vision und Realisierung einer sicheren mobilen Informations-Verteilung, Verwaltung und Abfrage. In *Multikonferenz Wirtschaftsinformatik (MKWI) 2004. Bd 3. Mobile Business Systems*, 2004.

[HKQ02] D. Huynh, D. Karger, and D. Quan. Haystack: A Platform for Creating, Organizing and Visualizing Information Using RDF, 2002.

[HS07] Jens Heider and Julian Schütte. Security made easy: Achieving user-friendly communication protection in ad-hoc situations. In *Proceedings SECURWARE 2007*. IEEE Computer Society Press, 2007.

[HS08] Jens Heider and Julian Schütte. On Path-Centric Navigation and Search Techniques for Personal Knowledge Stored in Topic Maps. In Lutz Maicher and Lars Marius Garshol, editors, *Scaling Topic Maps*, 2008.

[HTMP02] Uwe Hansmann, Peter Thompson, Riku M. Mettala, and Apratim Purakayastha. *SyncML: Synchronizing Your Mobile Data*. Prentice Hall Professional Technical Reference, 2002.

[Int02] International Organization for Standardisation. Guide to the topic map standards. http://www.y12.doe.gov/sgml/sc34/document/0323.htm, 2002.

[Int06a] International Organization for Standardisation. ISO 13250-2: Topic Maps – Data Model. http://www.isotopicmaps.org/sam/sam-model/, June 2006.

[Int06b] International Organization for Standardisation. ISO 13250-3: Topic Maps – XML Syntax. http://www.isotopicmaps.org/sam/sam-xtm/, June 2006.

[JP03] William C. Janssen and Kris Popat. Uplib: a universal personal digital library system. In *DocEng '03: Proceedings of the 2003 ACM symposium on Document engineering*, pages 234–242, New York, NY, USA, 2003. ACM Press.

[JPGB05] William Jones, Ammy Jiranida Phuwanartnurak, Rajdeep Gill, and Harry Bruce. Don't take my folders away!: organizing personal information to get things done. In *CHI '05: extended abstracts on Human factors in computing systems*, pages 1505–1508, New York, NY, USA, 2005. ACM Press.

[JT07] William Jones and Jaime Teevan, editors. *Personal Information Management*, chapter 1, page 13. Seattle, USA: University of Washington Press, 2007.

[JUT07] A. Jamakovic, S. Uhlig, and I. Theisler. On the relationships between topological metrics in real-world networks. In *European Conference on Complex Systems*, 2007.

[KB03] Rudolf Kruse and Christian Borgelt. Information Mining: Editorial. *Int. Journal of Approximate Reasoning*, 32:63–65, 2003.

[KH06] Stan Kurkovsky and Karthik Harihar. Using ubiquitous computing in interactive mobile marketing. *Personal Ubiquitous Comput.*, 10(4):227–240, 2006.

[Kle00] J. Kleinberg. Navigation in a small world. *Nature*, 406:845, August 2000.

[LaV06] S. M. LaValle. *Planning Algorithms*, pages 42, figure 2.7. Cambridge University Press, Cambridge, U.K., 2006. http://planning.cs.uiuc.edu/.

[LE01] C. Ladas and R. Edwards. Use of wireless application protocol service configuration provision over the short messaging system for nomadic device adaptation. In *2nd PGNET Symposium*, 2001.

[LEF$^+$00] Mik Lamming, Marge Eldridge, Mike Flynn, Chris Jones, and David Pendlebury. Satchel: providing access to any document, any time, anywhere. *ACM Transactions on Computer-Human Interaction*, 7(3):322–352, 2000.

[LJ96] David D. Lewis and Karen Sparck Jones. Natural Language Processing for Information Retrieval. *Communications of the ACM*, 39(1):92–101, 1996.

[MA05] Graham Moore and Kal Ahmed. Topic Map Relational Query Language - TMRQL. http://www.networkedplanet.com/download/TMRQL.pdf, 2005.

[Mah04] Qusay H. Mahmoud. Getting Started with Data Synchronization Using SyncML. http://developers.sun.com/mobility/midp/articles/syncml/, September 2004.

[Mai07] Lutz Maicher. *Autonome Topic Maps zur dezentralen Erstellung von implizit und explizit vernetzten Topic Maps in semantisch heterogenen Umgebungen.* PhD thesis, Universität Leipzig, 2007.

[Mas09] Massachusetts Institute of Technology. Metadata Reference Guide. http://libraries.mit.edu/guides/subjects/metadata/standards.html, 2009.

[May08] Rene Mayrhofer. Ubiquitous computing security: Authenticating spontaneous interactions, September 2008. Habilitation thesis, submitted to University of Vienna.

[MCE04] Cecilia Mascolo, Licia Capra, and Wolfgang Emmerich. *Middleware for Communications*, chapter 12: Principles of Mobile Computing Middleware. Wiley Publishing, June 2004.

[MDM07] Johann Van Der Merwe, Dawoud Dawoud, and Stephen McDonald. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Comput. Surv.*, 39(1):1, 2007.

[MG07] R. Mayrhofer and H. Gellersen. Shake well before use: two implementations for implicit context authentication. In *UbiComp 2007: Proceedings of the 9th International Conference on Ubiquitous Computing*. Springer, September 2007.

[Mic06] Microsoft. A detailed description of the Data Execution Prevention (DEP) feature in Windows XP Service Pack 2, Windows XP Tablet PC Edition 2005, and Windows Server 2003. http://support.microsoft.com/?scid=kb%3Ben-us%3B875352, September 2006.

[Mül08] Tilo Müller. ASLR Smack & Laugh Reference, Seminar on Advanced Exploitation Techniques. http://www-users.rwth-aachen.de/Tilo.Mueller/ASLRpaper.pdf, February 2008.

[MLRM04] Ana Maguitman, David Leake, Thomas Reichherzer, and Filippo Menczer. Dynamic extraction topic descriptors and discriminators: towards automatic context-based topic search. In *CIKM '04: Proc. of the 13th ACM international conference on Information and knowledge management*, pages 463–472. ACM Press, 2004.

[MMBB02] Sanjay Kumar Madria, Mukesh Mohania, Sourav S. Bhowmick, and Bharat Bhargava. Mobile data and transaction management. *Inf. Sci. Inf. Comput. Sci.*, 141(3-4):279–309, 2002.

[Mos02] Marie-Luise Moschgath. *Kontextabhängige Zugriffskontrolle für Anwendungen im Ubiquitous Computing*. PhD thesis, TU Darmstadt, Fachbereich Informatik, 2002.

[MPR05] J.M. McCune, A. Perrig, and M.K. Reiter. Seeing-is-believing: using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, pages 110–124, May 2005.

[MT05] Qiang Ma and Katsumi Tanaka. Topic-structure-based complementary information retrieval and its application. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(4):475–503, 2005.

[Nai82] John Naisbitt. *Megatrends*. Warner Books, 1982.

[NP05] Wolfgang Nejdl and Raluca Paiu. I know I stored it somewhere - Contextual Information and Ranking on our Desktop. In *Proceedings of 8th International Workshop of the EU Network of Excellence DELOS on Future Digital Library Management Systems*, 2005.

[OF04] Sangyoon Oh and Geoffrey C. Fox. A Customizable Reliable Message Service for Hand Held Devices. Technical report, University Indiana, USA, May 2004.

[OKAD04] Bahattin Ozen, Ozgur Kilic, Mehmet Altinel, and Asuman Dogac. Highly personalized information delivery to mobile clients. *Wirel. Netw.*, 10(6):665–683, 2004.

[OPK$^+$05] Sangyoon Oh, Sangmi Lee Pallickara, Sunghoon Ko, Jai-Hoon Kim, and Geoffrey Fox. Publish/Subscribe Systems on Node and Link Error Prone Mobile Environments. In *Lecture Notes in Computer Science (Proc. of the Wireless and Mobile Systems Workshop at ICCS 2005)*, pages 576–584, May 2005.

[Oxf98] Oxford. *The New Oxford Dictionary of English*. Oxford University Press, UK, 1998.

[Pep00] Steve Pepper. The TAO of Topic Maps - finding the way in the age of infoglut. http://www.ontopia.net/topicmaps/materials/tao.html, 2000.

[PG02] Steve Pepper and Geir Ove Grønmo. Towards a General Theory of Scope. http://www.ontopia.net/topicmaps/materials/scope.htm, 2002.

[PHJ02] I. Podnar, M. Hauswirth, and M. Jazayeri. Mobile Push: Delivering Content to Mobile Users. In *Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS'02)*, pages 563–570, Vienna, Austria, 2002.

[PK00] Joonah Park and Jinwoo Kim. Effects of contextual navigation aids on browsing diverse web systems. In *CHI '00: Proc. of the SIGCHI conference on Human factors in computing systems*. ACM Press, 2000.

[PTB06] Khoi Anh Phan, Zahir Tari, and Peter Bertok. A benchmark on SOAP's transport protocols performance for mobile applications. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 1139–1144, New York, NY, USA, 2006. ACM Press.

[Rei06] Thorsten Reinhard. Sichere Informationsverteilung und -visualisierung mit mobilen Endgeräten. Master's thesis, Technische Universität Darmstadt, Germany, September 2006.

[Ric02] Gerardo Richarte. Four different tricks to bypass stackshield and stackguard protection. http://www.cs.purdue.edu/~xyzhang/spring07/Papers/, April 2002.

[RMB01] William A. Ruh, Francis X. Maginnis, and William J. Brown. *Enterprise Application Integration - A Wiley Tech Brief*. John Wiley & Sons, Inc, 2001.

[RN09] Stuart J. Russell and Peter Norvig. *Artificial intelligence : a modern approach*, pages 83–84. Prentice Hall series in artificial intelligence. Prentice Hall, 3 edition, December 2009.

[SA99] Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Security Protocols, 7th International Workshop Proceedings*, pages 172–194, 1999.

[Sau05] Leo Sauermann. The Gnowsis Semantic Desktop for Information Integration. In *IOA Workshop of the WM2005 Conference*, 2005.

[SBD05] Leo Sauermann, Ansgar Bernardi, and Andreas Dengel. Overview and outlook on the semantic desktop. In *In Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference*, 2005.

[SBLH06] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.

[Sch07] Julian Schütte. Sichere Bluetooth-Kommunikation in Ad-hoc-Szenarien. Master's thesis, Technische Universität Darmstadt, Germany, March 2007.

[SE08] Frederic Stumpf and Claudia Eckert. Enhancing trusted platform modules with hardware-based virtualization techniques. In *Proceedings of the Second International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2008)*, pages 1–9, Cap Esterel, France, August 25-31 2008. IEEE Computer Society.

[SG02] Thomas Schwotzer and Kurt Geihs. Shark - a System for Management, Synchronization and Exchange of Knowledge in Mobile User Groups. *2nd International Conference on Knowledge Management (I-KNOW '02)*, pages 149–156, 2002. Graz, Austria, July 2002.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, November 1979.

[SHF01] Gary Stoneburner, Clark Hayden, and Alexis Feringa. Engineering Principles for Information Technology Security (A Baseline for Achieving Security). NIST Special Publication 800-27, June 2001.

[SjGM+04] Hovav Shacham, Eu jin Goh, Nagendra Modadugu, Ben Pfaff, and Dan Boneh. On the effectiveness of address-space randomization. In *In CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, pages 298–307. ACM Press, 2004.

[SKA05] Stefan Seedorf, Axel Korthaus, and Markus Aleksy. Creating a topic map query tool for mobile devices using J2ME and XML. In *Proceedings of the 4th international symposium on Information and communication technologies WISICT*, 2005.

[SKK08] Andreas U. Schmidt, Nicolai Kuntze, and Michael Kasper. On the deployment of mobile trusted modules. In *Proceedings of the Wireless Communications and Networking Conference, WCNC 2008, Las Vegas, USA, 31 March - 2 April 2008*, pages 3163–3168. IEEE, 2008.

[SvED07] Leo Sauermann, Ludger van Elst, and Andreas Dengel. PIMO - a Framework for Representing Personal Information Models. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics 2007*, pages pp. 270–277. JUCS, 2007.

[SW05] Yaniv Shaked and Avishai Wool. Cracking the bluetooth pin. In *MobiSys 05: Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 39–50, New York, NY, USA, 2005. ACM Press.

[Swi04] Peter P. Swire. A Model for When Disclosure Helps Security: What is Different About Computer and Network Security? *Journal on Telecommunications and High Technology Law*, Vol. 2, 2004.

[Tay03] Chris Taylor. An Introduction to Metadata. http://www.library.uq.edu.au/iad/ctmeta4.html, July 2003.

[TJB06] Jaime Teevan, William Jones, and Benjamin B. Bederson. Special Issue on Personal Information Management. *Commun. ACM*, 49:40–43, January 2006.

[Top01] Topicmaps.org. XML Topic Maps (XTM) 1.0 Specification. http://www.topicmaps.org/xtm/1.0, 2001.

[Tru07] Trusted Computing Group. TPM Main Specification Level 2 Version 1.2, Revision 103. http://www.trustedcomputinggroup.org/resources/tpm_main_specification, July 2007.

[Tse07] Yuh-Min Tseng. A heterogeneous-network aided public-key management scheme for mobile ad hoc networks. *Int. Journal of Network Management*, 17(1):3–15, 2007.

[Vat04] Bernard Vatant. Ontology-driven topic maps. In *Proceedings of XML Europe 2004*, Amsterdam, Netherlands, April 2004.

[W3C01] W3C. DAML+OIL Reference Description. http://www.w3.org/TR/daml+oil-reference, December 2001.

[W3C04a] W3C. OWL Web Ontology Language Overview. http://www.w3.org/TR/2004/REC-owl-features-20040210/, February 2004.

[W3C04b] W3C. RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/, February 2004.

[W3C04c] W3C. RDF/XML Syntax Specification (Revised). http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/, February 2004.

[W3C04d] W3C. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, February 2004.

[WAP99] WAP Forum, Ltd. Wireless Application Environment Overview. http://www.wapforum.org/what/technical/SPEC-WAEOverview-19991104.pdf, 1999.

[WAP01] WAP Forum, Ltd. Push OTA Protocol. http://www.openmobilealliance.org/tech/affiliates/wap/wap-235-pushota-20010425-a.pdf, 2001.

[Wat99] D. J. Watts. Networks, dynamics, and the small-world phenomenon. *The American Journal of Sociology*, 105(2):493–527, 1999.

[WC03] Perry Wagle and Crispin Cowan. Stackguard: Simple stack smash protection for gcc. In *Proc. of the GCC Developers Summit*, pages 243–255, 2003.

[WDDA07] Dandan Wang, Darina Dicheva, Christo Dichev, and Jerry Akouala. Retrieving information in topic maps: the case of TM4L. In *ACM-SE 45: Proceedings of the 45th annual southeast regional conference*, pages 88–93, New York, NY, USA, 2007. ACM Press.

[Whe03] William Wheeler. *Integrating Wireless Technology in the Enterprise: PDAs, Blackberries, and Mobile Devices*. Butterworth-Heinemann, Newton, MA, USA, 2003.

[WKLW98] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. RFC 2413: Dublin Core Metadata for Resource Discovery. http://www.ietf.org/rfc/rfc2413.txt, September 1998.

[WW08] Wolfgang Woerndl and Maximilian Woehrl. SeMoDesk: Towards a Mobile Semantic Desktop. In *Proceedings Personal Information Management (PIM) Workshop, CHI 2008 Conference, Florence, Italy*, April 2008.

[Xu07] Zengwang Xu. *Small-world characteristics in geographic, epidemic, and virtual spaces : a comparative study*. PhD thesis, Texas A&M University, 2007.

[ZH99] Lidong Zhou and Zygmunt J. Haas. Securing Ad Hoc Networks. *IEEE Network*, 13(6):24–30, 1999.

[Zlo75] Moshé M. Zloof. Query By Example. In *AFIPS National Computer Conference*, volume 44, pages 431–438, 1975.

List of Figures

List of Tables

List of Algorithms