Privacy Management in Smart Environments

Dissertation
zur
Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
der Fakultät für Informatik und Elektrotechnik
der Universität Rostock

Vorgelegt von
Dipl-Inf. Christian Bünnig
geboren am 4.4.1980 in Nauen
aus Potsdam

Rostock, 2012

urn:nbn:de:gbv:28-diss2013-0065-3

Gutachter
Prof. Dr. rer. nat. Clemens H. Cap, Universiät Rostock
Prof. Dr.-Ing. Thomas Kirste, Universität Rostock
Prof. Dr.-Ing. Andreas Schrader, Universität zu Lübeck

Datum der Einreichung: 07.09.2012
Datum der Verteidung: 18.12.2012

Acknowledgements

This thesis has been one of the most challenging works in my life. It would not have been completed with the support of several people.

First of all I want express my gratitude to Prof. Dr. Clemens Cap, who gave me the opportunity to extensively research the topic of my thesis and who regularly assisted me with critical and motivating feedback. Moreover I want to thank Prof. Dr. Thomas Kirste and Prof. Dr. Anderas Schrader for reviewing my thesis. I am especially grateful to Prof. Dr. Thomas Kirste for his feedback concerning the machine learning aspects of my thesis.

Many thanks for insightful discussions and helpful sparring go to my former colleagues from the Chair for Information and Communication Technologies as well as the other PhD students from the MuSAMA project.

Finally, a deep thank you is forwarded to my family for their patience and for assisting me in finishing my work.

Abstract

Smart environments are physical spaces, enriched with various sensors, devices, and services which aim to support inhabitants in their activities. A typical application area are environments which support collaboration, enhance productivity, or boost creative activities – for instance smart meeting rooms. User adapted behavior and personalization are central themes of smart environments. However, next to improving the overall user experience these features and their underlying technologies are a rich source of privacy issues. Privacy is a broad term which is used for different concepts. Especially in collaboration scenarios the concept of interpersonal privacy plays an important role. It affects the communication of personal information within social interactions and aims to support social norms and boundaries. The manifold and pervasive communication modalities in smart environments easily may disrupt social interactions instead of supporting them. In contrast to privacy preferences in face of abstract entities like governments or corporations, information disclosure decisions in social interactions tend to be more dynamic and intuitive, rather than static and rational, which results in more complex disclosure parameters whose values are harder to predict and often not easy to describe formally. Existing solutions to protect and manage privacy in smart environments do not sufficiently consider the specific characteristics of interpersonal privacy within smart environments. In order to objectively address the generally vague concept of interpersonal privacy, this work introduces a model to express disclosure decisions in social interactions within smart environments. The model is used to develop several set-based patterns of privacy management which make it possible to programmatically manage interpersonal information exchange. The practical relevance of these patterns is evaluated in a corresponding user study. As there is no sole disclosure control method which allows intuitive and individual privacy control, this work further presents a reference model of a composite disclosure control system, which stages and integrates various control methods. The benefit of this system is twofold. First, single components complement each other, and second, users are able to practice privacy according to their individual needs and preferences. One specific control method, predicting information disclosure with the help of machine learning techniques, is addressed in detail. Several existing learning methods as well as a novel one, based on the previously developed privacy patterns, are presented and evaluated. Additionally, pattern-based prediction validators are developed which reduce the impact of wrong predictions. Evaluation results show that learning methods are able to correctly predict disclosures in many cases while still providing workload-reducing suggestions in cases where predictions are not correct. The novel learning method is a competitive alternative to existing methods with special properties especially useful for interpersonal privacy control. This work is completed with general guidelines how to support interpersonal privacy when designing and engineering smart environments.

Kurzfassung

Smart Environments sind physische Räume, die mit diversen Sensoren, Geräten und Diensten ausgestattet sind, um deren Benutzer in alltäglichen Aufgaben zu unterstützen. Ein typischer Anwendungsbereich sind *Smart Meeting Rooms*, die Teamarbeit unterstützen, Produktivität steigern und kreative Aktivitäten fördern sollen. Nutzer-zentriertes Verhalten, Personalisierung und Informationsaustausch sind dabei zentrale Ideen, die aber auch vielfältige Probleme im Bereich *Privacy* bergen. Der Begriff Privacy ist sehr weitreichend und wird für verschiedene Konzepte benutzt. Speziell in kollaborativen Szenarien spielt das Konzept *Interpersonal Privacy* eine wichtige Rolle. Es bezieht sich auf die Kommunikation von persönlichen Informationen bei der sozialen Interaktion und dient der Unterstützung sozialer Normen und Grenzen. Die vielfältigen und teilweise versteckten Modalitäten für die Kommunikation in Smart Environments können soziale Interaktionen schnell entstellen anstatt sie zu unterstützen. Während Entscheidungen über die Freigabe persönlicher Informationen gegenüber abstrakten Einheiten (Regierungen, Unternehmen) meist statisch und rational sind, sind Entscheidungen in sozialen Interaktionen deutlich dynamischer und intuitiver, so dass Freigabeparameter schwerer zu erfassen und zu formalisieren sind. Die speziellen Eigenschaften von Interpersonal Privacy werden in vorhandenen Lösungen zum Thema Privacy in Smart Environments nicht ausreichend berücksichtigt. Um das unscharfe Konzept von Interpersonal Privacy objektiv zu behandeln, wird in dieser Arbeit ein formales Modell für Freigabeentscheidungen in sozialen Interaktionen eingeführt und darauf aufbauend bestimmte Freigabemuster erarbeitet. Diese Muster ermöglichen eine programmatische Herangehensweise an die Handhabung persönlicher Informationen. Die praktische Relevanz der Muster wird in einer empirischen Studie evaluiert. Da es keine Methode der Freigabekontrolle gibt, die allein eine intuitive als auch individualisierte Handhabung persönlicher Informationen erlaubt, wird des Weiteren ein Referenzmodell eines kombinierten Systems für die Freigabekontrolle erarbeitet, welches verschiedene Methoden so miteinander integriert, dass sie sich gegenseitig ergänzen und Nutzern ein individuelles Praktizieren von Privacy ermöglichen. Eine spezifische Methode, die Vorhersage von Freigaben mit Hilfe maschineller Lernverfahren, wird ausführlicher behandelt. Es werden verschiedene existente und ein neues Lernverfahren vorgestellt und evaluiert. Das neue Verfahren nutzt die zuvor erarbeiteten Freigabemuster. Diese werden auch zur Validierung von vorhergesagten Freigaben genutzt, wodurch der negative Einfluss falscher Vorhersagen deutlich reduziert wird. Die Evaluierungen zeigen, dass maschinelle Lernverfahren Freigabeentscheidungen oft korrekt vorhersagen, während falsche Vorhersagen immer noch hilfreiche Vorschläge für manuelle Entscheidungen darstellen. Die Performanz des neuen Lernverfahrens ist in vielen Fällen besser als vorhandene Verfahren und hat speziell für den Bereich Interpersonal Privacy günstige Eigenschaften. Diese Arbeit wird abgerundet durch allgemeine Richtlinien und Empfehlungen für Entwickler von Smart Environments bezüglich der Unterstützung von Interpersonal Privacy.

Contents

1 Introduction

1.1 Motivation

Smart environments are a specific application area of the paradigm of ubiquitous computing. They are physical spaces, enriched with various sensors, devices and services which aim to support inhabitants of the environment in their activities. Following the vision of ubiquitous computing, underlying technology transparently integrates into facilities of the environment and services fulfill their tasks in a user-centric manner. A typical application area are environments which support collaboration, enhance productivity, or boost creative activities – for instance smart meeting rooms. User adopted behavior and personalization are central themes of ubiquitous computing respectively smart environments. However, next to improving the overall user experience these features and their underlying technologies are a rich source of privacy issues.

Privacy is a flexible term, depending on individual needs and perceptions. It is not a pure technical issue but involves legal and social aspects. In order to determine privacy issues and develop corresponding solutions, it is helpful to distinguish different concepts of privacy (see section 2.1). The most traditional one is *privacy in private*, sometimes also described as the right to be let alone. It deals with invasions to and the protection of private spaces. Issues like surveillance fall into this concept. Problems related to the processing and storage of personal information outside private spaces by third parties are covered by the concept of *privacy in public*. It aims to empower individuals to control the flow of their information and to prevent malicious usage of private data. Finally there is the concept of *interpersonal privacy*. It affects the communication of personal information within social interactions and aims to support social norms and boundaries. Violations of this concept appear to be less threatening than the first two, but they fundamentally conflict with the envisioned user experience of smart environments and thus degrade any other benefits.

Smart environments have the potential to interfere with each of the three concepts (see sections 2.2 and 2.3). They may cross natural borders between private and public spaces, track and distribute personal information without the knowledge of users and beyond their

control capabilities, and they may disrupt social interactions mediated by the environment. Preventing these problems and providing efficient means to cope with them are crucial factors for the success of smart environments.

1.2 Problem Statement

A generally successful approach to protect privacy is to limit the amount of communicated personal information – both by quantitative and qualitative means, e.g. by waiving, blurring or anonymizing data (see section 3.3.1). This often already prevents invasions to private spaces and avoids the need to manage personal information in public spaces. Still, in many cases users *want* to disclose personal information for the sake of improved services or as part of social interactions. Typically, the decision which personal information to communicate with services in smart environments is regulated by negotiations between service-side privacy policies and user-side privacy preferences (see section 3.3.2). Policies describe what data a service consumes and how respectively for which purpose it is used. With reference to policy parameters, preferences specify which information to disclose to which service under which conditions. Such negotiations are suitable for the management of privacy in public, but for several reasons they do not work similarly for interpersonal privacy. Information disclosure decisions in social interactions tend to be more dynamic and intuitive, rather than static and rational. They are highly individual and have a stronger link to the current situation, including social context. This results in different, potentially more complex disclosure parameters whose values are harder to predict and often not easy to describe formally. Supporting users in managing interpersonal privacy requires more intuitive mechanisms to express privacy preferences while recognizing individual patterns of information disclosure. Approaches to manage privacy in public often are backed up by legal regulations, which do not directly prevent but discourage privacy invasions. Such regulations do not apply to interpersonal privacy – no laws could force an interaction partner to ignore an unintentionally communicated information.

Current solutions to protect and manage privacy in smart environments do not sufficiently consider these specific characteristics of interpersonal privacy within smart environments (see sections 3.3.3 and 3.4). Though user adaption and seamless interaction are core concepts of smart environments, there is currently no solution for interpersonal privacy management which provides *intuitive* control mechanisms tailored to *individual* ways of practicing privacy.

In particular there are the following open issues: (I) consideration of social aspects in interpersonal privacy management and resulting patterns in information disclosure decisions, (II) suitable disclosure control mechanisms that match these patterns, as well as

their orchestration, (III) automating disclosure control in a user-adaptive but easy to manage fashion, and (IV) general guidelines and principles to develop interpersonal privacy sensitive smart environments and to evaluate corresponding solutions.

1.3 Contributions

This thesis elaborates techniques for smart environments to manage privacy in social interactions diligently, but in an unobtrusive and seamless way – as generally aspired by smart environments. More precise – in response to the open issues mentioned in the previous section – it makes the following specific contributions:

Social aspects and patterns in interpersonal privacy management

A **formal model** of information disclosure emphasizing social factors is developed. Based on this model, different **patterns** of information disclosure are identified. These patterns indicate specific types of information disclosure behavior and thus allow disclosure control assistance mechanisms to be improved and tailored to individual users. A **survey system** for empirical user studies on the relevance and usage of these patterns in context of different scenarios, i.e. interpersonal information disclosure situations, is developed. Pattern usage results from an **exemplary study** and implications for disclosure control assistance mechanism are presented.

Suitable disclosure control methods and their orchestration

A reference model for a **composite disclosure control system** is developed which combines existing approaches to information disclosure control. Next to providing **multiple control methods** covering different capabilities and temporal schemes, this system's advantages are **mutual supplementation** of individual components as well as **user adaptive** selection of most suitable components.

User-adaptive automation of disclosure decisions

Disclosure decisions may be automated by applying machine learning technologies to observed disclosure behavior. The corresponding **learning problem** is analyzed and matching learning methods are presented. This involves existing standard methods as well as new **enhanced methods** incorporating disclosure patterns. In order to evaluate these methods with regard to different scenarios and patterns of interpersonal information exchange, a

corresponding learning method **evaluation system** is developed. The evaluation **results of exemplary scenarios** are presented and discussed with regard to their generalizability to certain scenario characteristics. Additionally, specific **integration possibilities** of individual methods into a composite disclosure control are described.

Design principles for interpersonal privacy sensitive environments

Existing guidelines and design principles concerning privacy in ubiquitous computing environments do not sufficiently consider interpersonal privacy issues. Filling this gap, a list of missing general **design principles** is compiled. It **combines and generalizes** multiple aspects of this work, including the previously mentioned disclosure patterns and related study results, the concepts involved in a composite disclosure control system, and the findings about machine learning driven automatic disclosures.

1.4 Overview

Following is an overview about the content of this thesis. Each paragraph provides a brief summary of a subsequent chapter. The main contributions of this work are covered by the chapters 4, 5, 6, and 7.

Chapter 2 - Basics. This chapter introduces basic knowledge and declares a common terminology required for the understanding of this work. Especially privacy is a term referring to a wide range of concepts and issues. Determining concerns of and developing solutions for privacy requires to examine this term more closely. For that purpose different concepts of privacy are described. Similarly, the term smart environment covers various types of technologies and applications. This chapter positions the concepts of smart environments within the general research topic of ubiquitous computing and determines the technical characteristics of smart environments relevant for this work. Eventually, both topics are merged to point out specific privacy problems in smart environments.

Chapter 3 - Related Work. Privacy has been an issue since the vision of ubiquitous computing became popular. With reference to the different privacy concepts and issues identified in the preceding chapter, this chapter reviews previous work about privacy in ubiquitous computing in general and smart environments in particular. This includes theoretical frameworks and design principles for privacy sensitive ubiquitous computing environments, empirical studies analyzing patterns of privacy management, and specific technical approaches to protect and manage privacy. Finally, it highlights open issues and narrows them down to those focused in this work.

Chapter 4 - Information Disclosure Patterns. This chapter elaborates patterns of interpersonal information disclosure not yet considered by related work. It defines a formal model for disclosure decisions and analyzes deducible set-based structural information and resulting patterns of information disclosure. Subsequent chapters utilize these patterns in order to improve disclosure assistance mechanisms. A concept and implementation of a survey system for capturing disclosure behavior in social interactions is presented. Its purpose is to explore the practical relevance of the elaborated patterns. The survey system allows to perform empirical studies for different scenarios respectively domains. A specific study conducted with this survey system is described and corresponding pattern usage results are shown and discussed.

Chapter 5 - Composite Disclosure Control. As indicated by the review of related work and by the study results of the foregoing chapter there is no optimal way to control the disclosure of personal information: the way of choice depends on user preferences, type or sensitivity of potentially shared information and circumstances accompanying disclosures. This chapter presents different general approaches for privacy management. This is followed by an orchestration of these approaches to a reference model of composite disclosure control system. Particularly, this chapter discusses the integration of different disclosure control mechanisms in order to complement each other and for the sake of providing user specific optimal control mechanisms.

Chapter 6 - Learning Disclosure Decisions. One component of the presented composite disclosure control system is supposed to conceptualize and automate disclosure decisions by observing users and applying machine learning techniques to predict or suggest information disclosures in new situations. This chapter analyzes the actual corresponding learning problem and presents related standard learning methods as well as a novel method based on the patterns investigated in chapter 4. These patterns are also used to develop novel prediction validation methods. Further, this chapter describes a developed system to extensively evaluate these learning methods in context of different scenarios respectively domains. This evaluation system is applied to manually composed scenarios and empirically captured data from the study in chapter 4. Corresponding results, i.e. the performance of different learning mechanisms, are shown and discussed with respect to their suitability to automate disclosure decisions.

Chapter 7 - Guidelines for Interpersonal Privacy. This chapter combines the various findings of this work to a generic framework of design principles for privacy sensitive smart environments in context of social interactions. It extends existing frameworks presented in chapter 3 by putting a stronger focus on interpersonal privacy and by including the findings about disclosure patterns from chapter 4. Additionally, it generalizes the rather

technical concepts and results of chapters 5 and 6 to more abstract design concepts for disclosure assistance systems in ubiquitous computing environments.

Chapter 8 - Conclusion and Future Work. This chapter concludes this thesis with a summary of its results and findings and by stating remaining challenges for future work.

2 Basics

Privacy and smart environments are malleable terms referring to a wide range of concepts and issues. For the sake of disambiguation this chapter examines these terms more closely. It describes different concepts of privacy, positions the notion of smart environments within the general research topic of ubiquitous computing and determines the technical characteristics of smart environments relevant for this work. Finally it merges both topics to point out specific privacy problems in smart environments.

2.1 Concepts of Privacy

Privacy is the main issue driving this thesis, but what actually is privacy? It is a widely honored but rarely understood issue. In order to develop systems and technologies which protect privacy and help individuals in managing privacy one needs a clear understanding of what actually is considered as privacy and in which ways it could be violated.

Looking back in history, privacy initially has been a physical issue, protecting a person's home from public invasion and its body from being exposed or touched unwillingly. This is also known as territorial and bodily privacy. References to the latter one already can be found in the Justice of the Peace Act in England from 1361 which mentions the arresting of peeping toms and eavesdroppers (Pratt, 1979, page 54). A historical reference for territorial privacy is given by a speech of the English parliamentarian William Pitt in the 18th century: "The poorest man may in his cottage bid defiance to all the force of the Crown. It may be frail; its roof may shake; the wind may blow through it; the storms may enter, the rain may enter – but the King of England cannot enter; all his forces dare not cross the threshold of the ruined tenement!" (quoted in Knowles, 1999, page 576).

However, this work is about informational privacy, i.e. concerning the access to and distribution respectively exchange of personal information. In this regard this section elaborates how privacy has evolved over the years and which corresponding concepts of privacy have been developed.

2.1.1 Privacy in Private

Concerning today's industrial societies, one of the oldest and influential publications promoting privacy is the article "The Right to Privacy" by Warren & Brandeis (1890) which discusses a judicial manifestation of a right to privacy in the United States. This article faces invasions of private life caused by photography and newspapers. These days photographic equipment became small and cheap enough to be used by the general public. At the same time sensational newspapers (yellow press) started to become very popular. Information about personal life, previously restricted to domestic circles and oral distribution, now could easily spread to a wide public. Warren & Brandeis mainly discuss judicial aspects of privacy. However, they also promote privacy as a concept in that they mention it as "the right to be let alone"[1].

International privacy legislation in the 20th century mainly adopted the concept of privacy as noted by Warren & Brandeis. For instance the Universal Declaration of Human Rights (UDHR) by the UN General Assembly (1948) mentions privacy in article 12: "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks." In contrast to Warren & Brandeis, who mainly aim to protect personal information from unauthorized publication, the UDHR affects *any* invasion into private life, be it public media, the government or a single person. The philosopher Ferdinand Schoeman (1984) proposes a similar understanding but describes privacy as a *state* (rather than as a right): "A person has privacy to the extent that others have limited access to information about him, limited access to the intimacies of his life, or limited access to his thoughts or his body".

These concepts focus on "privacy in private" and propose to "limit the ability of others to penetrate your private space" (Lessig, 2006, chap. 11). They express a "binary understanding of privacy" (Solove, 2007, page 163) distinguishing between dedicated public and private spaces. Individuals practice privacy as a kind of withdrawal from public space.

2.1.2 Privacy in Public

In the 1960s and 1970s, when governments started to employ nation-wide automatic data processing facilities for managing information about their citizens, Warren & Brandeis's concept of privacy did not seem to be sufficient anymore. The increased and long-lasting placement of personal information outside the private space raised the issue of "privacy

[1] Warren & Brandeis are not the first to express this concept of privacy but they successfully popularized it.

in public" (Lessig, 2006). An influential work facing the new concerns about privacy is *Privacy and Freedom* by Westin (1967) which considers privacy as "the right of the individual to decide what information about himself should be communicated to others and under what circumstances". Westin's work has been used as a base for the US Privacy Act of 1974 (United States Code) which defines fair information practices for governmental use of personal information.

The German constitution mentions privacy in article 2, 10 and 13 (Deutscher Bundestag, 1949). Article 2 affects the development of one's own personality. Article 10 is about communication privacy (mail, telephony) and article 13 assures territorial privacy. In 1983, in the context of a national census, the German Federal Constitutional Court consolidated these articles to a general right to informational self-determination (based on the German term "informationelle Selbstbestimmung"). Basically this right matches the concept of privacy given by Westin.

Another notable legislative manifestation of privacy is the Directive 95/46/EC "on the protection of individuals with regard to the processing of personal data and on the free movement of such data" of the European Parliament and Council of the European Union (1995) and its revised version 2002/58/EC from 2002. The directive responds to an increased processing of digital data, especially in the business sector (previous laws mainly covered governmental privacy invasions). From a legislative perspective it is an influential rework of the fair information practices mentioned above. Conceptually it is similar in that it expresses Westin's understanding of privacy.

Similar concepts of privacy can be found in various other works about privacy with regard to digitally and potentially centralized as well as permanently stored data within a – usually impersonal – public space. The common idea is to practice privacy by consciously controlling the access by other entities to one's personal information.

2.1.3 Interpersonal Privacy

The concept of privacy in public mainly addresses privacy in face of rather abstract entities, e.g. organizations, corporations or governments. As highlighted by the social psychologist Irwin Altman (1975) this concept does not really apply to interpersonal information disclosure. He describes privacy as a *process* of dialectic and dynamic boundary regulation, emphasizing social interaction and the environment as driving regulation factors. In his framework privacy is about regulating *interaction* with others. This includes several behavioral mechanisms, i.e. verbal communication, gestures, visual appearance and territorial responses. According to Altman social interaction is composed of *inputs* and *outputs*. Inputs refer to information retrieved from others and outputs mean information

communicated to others. In this regard managing privacy can be described as adjusting the desired levels of inputs and outputs using different behavioral mechanisms. It is important to note that, in contrast to privacy concepts described above, Altman does not put the focus on personal information but on the process of *providing* information and receiving feedback. In this process the environment (involving social expectations and norms) is a significant parameter for information disclosure. In other words, privacy is not only about hiding information but about reaching a desired state of participation in a social environment by selectively disclosing information.

Today computers increasingly are used to mediate communication which motivated Palen & Dourish (2003) to convey Altman's privacy framework to socio-technical environments. Technical components used to communicate information are an additional privacy regulation mechanism, that is an additional medium which influences the levels of *inputs* and *outputs*. As such it is a rather complex medium, potentially hard to understand and control. To shift Altman's boundary concept to technical settings Palen & Dourish consider three types of boundaries: disclosure, identity and temporal boundary. The *disclosure boundary* addresses privacy as the selective disclosure of information according to circumstances like social norms, allegiance, self-representation and expectations one might put on a social interaction. A noteworthy aspect of the disclosure boundary is that keeping certain information private may require other information to be disclosed. For instance to disguise unappreciated information about oneself somewhere in the web, one could set up an own personal website, which provides intended personal information. Especially if one has only limited control about what information is public the only way to reach a desired level of privacy may be to show more information. This illustrates that privacy in context of the disclosure boundary is not defined by access to individual information but by a composition of hidden and disclosed information. The *identity boundary* deals with the tension between different roles one might take on. Palen & Dourish highlight that privacy is not bound to one persona per individual. For instance acting as a representative for an affiliation usually changes the assumed persona which in turn impacts the desired level of privacy - both for oneself and persons one interacts with. Regulating the identity boundary in technical environments may fail because the facilities used for social interaction may hide or alter intended respectively perceived personae. Loss of context over time is one example for an unintentionally changed personae. This issue of asynchronous communication is also addressed by the *temporal boundary* which regards the fact that privacy regulation additionally is affected by information disclosed in the past and in the future as well as potential perception of this information in the future.

Summarized, the privacy concept elaborated by Altman and Palen & Dourish shifts the focus from information and associated rights to interaction and associated processes. Similar to the concept of privacy in public it expresses the need to properly control the flow of

and access to information. In contrast, interpersonal privacy applies to social interactions and corresponding unique disclosure factors and implications.

2.1.4 Zero Privacy

The idea of zero privacy is a rather extreme solution to cope with the huge amount of personal information which nowadays is communicated and stored within the digital world. It expresses the consensus that digital sharing or processing of personal information and privacy are contradicting concepts which cannot be realized both without negatively interfering each other. Hence, any information disclosed within a *certain* public should be considered to be *completely* public. Most popularly this has been expressed by Scott McNealy: "You have zero privacy anyway. Get over it" (quoted in Lucky, 2008). More recently Facebook founder Mark Zuckerberg argued that "people have really gotten comfortable not only sharing more information and different kinds, but more openly and with more people. [...] That social norm is just something that has evolved over time" (quoted in The Guardian Online, 2010). These seem to be rather simple answers to the problem of privacy in the age of information technology – they solve privacy issues by negating them. On the other hand there is something to be said for them. Cochrane (2000) reasons that "All this secrecy is making life harder, more expensive, dangerous and less serendipitous". He sees privacy as a barrier to what technically is possible with regard to exchange of information and that privacy is a burden of the past one should get rid of – honest people do not have to fear anything anyway. Etzioni (1999) argues that each privacy right mostly is shadowed by a higher valued societal right. For instance he confronts the privacy of HIV infected persons with demands of public health. Here privacy is seen as a barrier to transparency.

These thoughts are debatable and still not erase all privacy problems. Any published information looses parts of its original context or may be erroneous. This could fatally impact a persons integration within the society. Further, societal and technical benefits do not diminish or reduce the problem of misuse of published information. Consequently, even if someone is willing to disclose most (if not all) personal information, it still needs some kind of protection. Eventually it is unclear if zero privacy indeed should be considered as a real concept. It is still unknown if it is accepted on a wider basis. At first glance the younger generation which grew up with social networking sites and which generally acts far more public than older generations appear to be a relevant societal group following a concept of zero privacy. However, this does not imply that young persons intentionally chose to drop any privacy concerns. With regard to social networking sites, providers generally have little interest in strong privacy support as personal user data is seen as a capital resource. Thus the providers generally motivate site members to share more information rather than less and often impede defensive information sharing behavior (Schneier, 2010). Switching

to a more privacy sensitive information sharing platform mostly is no option for social reasons – if the majority of friends acts on Facebook, a switch probably would disconnect relations in the analogue world. Additionally, the unconcerned privacy behavior of the young generation might not be a societal evolution at all but an age-based phenomenon. Young persons generally did not yet experience the fact that life often is not free of rough edges and flaws but involves mistakes and shifts of opinion which may raise the need to revoke certain aspects of one's life from a general public access.

Concluding, it remains to be seen if the idea of zero privacy must be considered as a real concept of privacy or if it is a phenomenon yielded by technical constraints and limited to a *not yet* concerned youth generation. For that reason it is not considered further within this work but has been elaborated here for the sake of conceptual completeness.

2.1.5 Conclusion

This section described different concepts of privacy in relation to developments in information technology. They have been elaborated from an individual's point of view on how to protect and manage privacy. It is worth noting that different perspectives yield other interesting conceptualizations of privacy. In this regard Flaherty (1992), Lessig (2006), Diffie & Landau (2007), and Solove *et al.* (2008) are insightful references for further reading.

The different understandings of privacy listed here are an important prerequisite when investigating privacy issues and corresponding solutions. Especially in smart environments, the application area focused by this work, there may be privacy issues concerning more than one concept of privacy. In this regard the next but one section gives an overview about which privacy related problems in smart environments exist. First, however, the next section introduces the research area of smart environments and ubiquitous computing in general.

2.2 The Paradigm of Ubiquitous Computing

This thesis addresses privacy issues specific to smart environments. Similar to privacy the term *smart environment* lacks a generally valid definition and partly overlaps with other ubiquitous computing related terms like pervasive computing or ambient intelligence. For the purpose of disambiguation this section provides a brief overview about ubiquitous computing and derived paradigms – for instance smart environments – and concludes what kind of smart environments and corresponding aspects of ubiquitous computing are in the focus of this work.

2.2.1 Ubiquitous Computing

In 1991 Mark Weiser, then working at XEROX Palo Alto Research Center, described his vision of "the computer of the 21st century" where technologies "weave themselves into the fabric of everyday life until they are indistinguishable from it". He describes his vision as ubiquitous computing, a "new way of thinking about computers in the world, one that takes into account the natural human environment and allows the computers themselves to vanish into the background". In these early years ubiquitous computing mainly was about miniaturizing office equipment and its integration into already existing patterns of working life. Later on, when mobile phones became popular, the focus moved to home entertainment and mobile computing (Ronzani, 2009). Today ubiquitous computing generally refers to technologies where many distributed and mobile devices, integrated into everyday artifacts, seamlessly and spontaneously interact with each other and with users in a context-driven manner. The central theme is to support users in work and other activities by assisting in complex processes and automating tedious repetitive tasks. Popular and illustrative examples are kitchens which assist in cooking based on available food or wardrobe mirrors which provide whether forecasts and suggest a suitable dress.

To some extent Weiser's vision has been realized already today. Computers no more are only available as heavy weighted desktop systems. Devices continuously shrink in size which implicitly raises more and more mobile usage patterns. Cars, cell phones and home appliances nowadays often are able to communicate with other devices and increasingly gain a certain level of "intelligence". However, it's still a long way to really match the idea of ubiquitous computing. Based on Weiser's vision various related paradigms have been developed, e.g. pervasive computing, the Internet of things, ambient intelligence, and smart environments – each emphasizing specific aspects of ubiquitous computing.

2.2.2 Pervasive Computing

In the 1990s the term pervasive computing mainly has been propagated by Novell and IBM. It has been a first step in realizing the paradigm of ubiquitous computing and initially was centered around connecting people and information (Novell) and mobile devices (IBM) (Ronzani, 2009). After the turn of the millennium pervasive computing has been used in a broader context. According to Hansmann *et al.* (2003) pervasive computing follows the paradigm of ubiquitous computing with regard to user experience but has a stronger focus on underlying technologies, mainly interoperability and seamless interconnectivity (Satyanarayanan, 2001).

Today pervasive computing mostly is used as a synonym for ubiquitous computing – each preferred by certain research communities and companies (Saha & Mukherjee, 2003; Ronzani, 2009). Within this work the terms pervasive and ubiquitous computing are used interchangeably.

2.2.3 The Internet of Things

The Internet of things – sometimes also referred to as physical computing – is a ubiquitous computing related research area centered around smart and interconnected everyday objects. An early work is that of Barrett & Maglio (1998) about virtually attaching arbitrary information to physical objects. Actual information is stored in the web while objects are marked with information retrieval identifiers (URLs). The main targeted scenario, though not limited to, is that of exchanging documents by handing out physical objects, a process which hides network and protocol details involved in the transfer of documents. Existing network technologies are used in a more user friendly way by using modalities from the real world. Here objects simply act as ID carriers. Today the exchange of documents via the Internet is far less complex (using one of the numerous file sharing services) and does not necessarily require physical objects to provide a user friendly experience. However, objects with unique identifiers motivate other interesting scenarios and research challenges. Consider the wardrobe example mentioned above which requires wirelessly readable unique identifiers on garments. Mattern (2003) lists further examples, e.g. operating instructions "attached" to technical facilities. Though objects themselves only carry identifiers, they appear to be "smart" if used with appropriate reading devices. Next to simply identified artifacts the Internet of things also involves more powerful objects which perceive and manipulate their surrounding context (Siegemund, 2004) and which interact with each other in order to complete the knowledge about their environment and in order to coordinate their behavior (Kortuem *et al.*, 2010).

With regard to the ubiquitous computing vision the Internet of things tackles the integration of computers into everyday artifacts and the interaction with hidden computing devices using modalities from the "real" world.

2.2.4 Ambient Intelligence

The term Ambient Intelligence (AmI) has been coined by Phillips in the late 1990s (Zelkha & Epstein, 1998). It describes a vision for digital systems in the years 2010 to 2020. Aarts & Encarnação (2006) describe the AmI world as one in which "devices operate collectively using information and intelligence that is hidden in the network connecting the devices. Lighting, sound, vision, domestic appliance, and personal health care products

all cooperate seamlessly with one another to improve the total user experience through the support of natural and intuitive user interfaces". The notion *ambient* describes the integration of technology into everyday objects of the environment while *intelligence* refers to social interaction by recognizing persons in the environment, adapting to, learning from and potentially acting on behalf of them.

AmI stresses a more user centric view of the ubiquitous computing paradigm and tries to answer the question how to cope with "myriad devices and the ever-present ubiquitous and pervasive infrastructure" (Shadbolt, 2003). Augusto (2007) highlights that an AmI system "proactively, but sensibly, supports people in their daily lives" which makes human centered design and artificial intelligence required to anticipate individual needs of users to core research areas. The vision of AmI has been taken on by the European Commission to describe a future information society (Ducatel *et al.*, 2001) which shows that research on AmI also covers societal aspects and integration issues (Burgelman & Punie, 2006) of ubiquitous computing scenarios.

2.2.5 Smart Environments

Smart environments can be considered as a specific application of AmI to limited physical spaces. According to Dey *et al.* (1999) "one of the goals of a smart environment is that it supports and enhances the abilities of its occupants in executing tasks". Cook & Das (2005) define *smart* as the ability to autonomously acquire and apply knowledge and *environment* as a subject's surrounding. Therefore they define a smart environment as one which has and applies knowledge about itself and which adapts to its inhabitants for the purpose of improved user experience, for instance by automating processes performed frequently in an environment. Expected experiences vary among users and specific environments and evolve with the capabilities of smart environments. Cook & Das describe their understanding of a smart environment by characterizing its main technical features. The most basic feature is the remote and automatic control of devices via wireless or existing cable connections (e.g. power line). This allows devices to visually diminish and to be integrated into facilities of the environment while still letting them communicate with each other and users. Device communication is another core feature of smart environments which requires certain standards for information exchange and which is the base for combining distributed device specific information to higher level information and for letting devices react upon events in other devices. For instance light and temperature in an environment may automatically adjust in response to an incoming inhabitant's preferences. This also implicates distributed sensors which are able to recognize the state of the environment and activities of inhabitants. More powerful and intelligent devices utilize the communication infrastructure and aggregated information to provide services which actually make the environment smart. Such services usually have prediction and decision

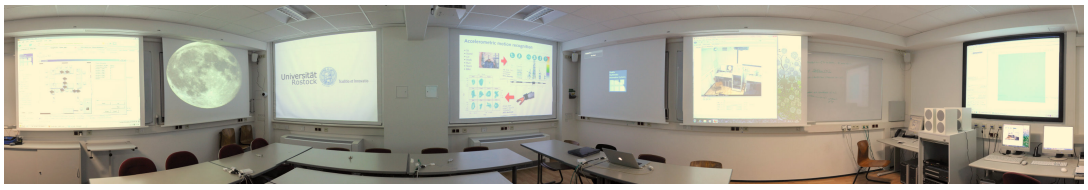

Figure 2.1: The Smart Appliance Lab of the MuSAMA project is a testbed for smart environments.

making capabilities in order to automate processes, not only based on predefined strategies but also based on dynamic individual preferences and past activities of inhabitants and groups of them.

Kirste (2006) complements these rather implementation specific characteristics with a more user centric definition: "Smart environments are physical spaces that are able to react to the activities of users, in a way that assists the users in achieving their objectives in this environments". Here the term *smart* refers to the proper selection and combination of environment actions according to user objectives which requires "a certain level of understanding of the user's view of the world". With regard to underlying technologies, to a large extent most smart environment scenarios can already be realized today. However, Kirste highlights that detecting user objectives and providing meaningful actions in *unforeseen* scenarios still is a big challenge in smart environments. Indeed this specific problem of ad hoc scenarios is a core research area of the project MuSAMA[2]. Figure 2.1 shows an experimental smart environment used by the MuSAMA project.

2.2.6 Conclusion

The initial paradigm of ubiquitous computing has triggered various further research areas. To a large extent these areas overlap, mainly concerning the aimed user experience. On the other hand each area emphasizes specific aspects of ubiquitous computing, e.g. targeted scenarios, functional focus or abstraction of technologies. This work, as part of the MuSAMA[2] project, primarily deals with smart environments whose main application scenario is that of smart meeting rooms, i.e. environments designed to assist in collaborative work. However, since related ubiquitous computing research areas are not completely disjoint, it is not strictly limited to dedicated smart meeting room use cases. Within the scope of this work a smart environment primarily is considered as a physical environment with various integrated devices whose functionality is driven by services. Services are applications running on the environment's devices and usually consume information from

[2]The MuSAMA project: `www.musama.de`

their users to fulfill a dedicated task. Typically these tasks implicate a perceivable change within the environment but may also induce processes outside the environment. Hence, from a privacy point of view the relevant aspects are service-driven processing of personal information and the implicated disclosure of personal information in different forms to different entities. The following section elaborates which privacy issues ubiquitous computing respectively smart environments raise.

2.3 Privacy in Smart Environments

Core features of smart environments are their adaption to preferences and behavior of users as well as the support of collaboration processes or leisure activities. Obviously these features require environment inhabitants to disclose a certain amount of personal information to the components utilizing this information. This affects various types of personal information, e.g. user preferences and sensor data required to set up the environment, tracks of behavioral information used to anticipate intended activities, or any arbitrary data involved in collaboration or entertainment motivated information exchange. The diversity of types of personal information, the various and partly unobservable modalities used to communicate information as well as the effects of processing personal data within the environment raise different types of privacy issues. To a certain level these issues already existed in context of traditional desktop based computing. However, ubiquitous computing aggravates them because it communicates much more personal information and it is present everywhere and anytime – there's no physical boundary and no power button which disconnects the digital and analogue worlds. Smart environments are limited in terms of physical space, but this limitation diminishes if each room in a building is a smart environment or if one spends the majority of a day within smart environments. This section describes which types of privacy issue may occur in smart environments, with reference to the privacy concepts presented in section 2.1.

2.3.1 Invading Private Space

The various sensors and devices within smart environments are, by definition, able to detect inhabitants, their activities and their state. The integration of computing technologies into facilities of the environment makes this happen silently and transparently. These capabilities bear the risk to blur natural borders traditionally assumed to separate private and public spheres. For instance physical borders like walls do not necessarily block information from leaving a room. Sensors capturing emotional conditions cross bodily borders, e.g. facial expressions. Persistently stored and searchable information dissolves spatial and temporal borders which divide different parts of one's life. Smart environments have to ensure to respect such natural borders which separate private and public areas. Otherwise

they violate the concept of privacy in private. Additionally – considering cognitive resources as a property of private space – excessive interaction requests by the environment or other awareness snatching events violate this privacy concept in the form of unwanted distraction.

Obviously the invasion of private spaces conflicts with a user's expectation that a smart environment make his life easier. Besides that, Cas (2005) states a more general, societal reason for preserving private spaces: "Pervasive surveillance creates enormous pressure to behave in a 'normal' way and not to leave the standardized paths of widely accepted social behavior. On the other hand, social innovation requires deviations by members of the society". Still, in some cases it is useful or necessary to let information cross natural borders, i.e. to let it enter a certain public space. However, here it is important to clearly communicate this fact to allow users to estimate their current state of privacy and, this directly leads to the next section, to regulate their degree of privacy.

2.3.2 Exploiting Personal Information

Given that environment inhabitants are willing to communicate information to devices and services in the environment and given that they are aware of the fact that this information potentially crosses natural borders, users should know what information is communicated for what purpose and under which conditions it is used by whom. This follows the concept of privacy in public. It is not easy to accomplish these requirements. User adopted behavior may need tracked and aggregated information from various sources which requires the storing of information by different parties for a longer period of time. Even if the parties involved in the communication and processing of personal information guarantee a sane handling of this data (possibly according to agreed policies) one might still have a bad feeling about it. Policies do not prevent human or technical errors. Such errors may leak information to unintended recipients or falsify personal information. In the latter case users had to cope with misunderstandings which may be almost impossible to clarify once they have been propagated (Garfinkel, 2000, page 28). Especially wrong medical or financial data may crucially harm a person's societal status. In fact these problems also violate the concept of zero privacy.

Another potential issue is that of searching possibilities on stored user data. Users may agree to disclose certain pieces of information but may not be aware of the fact that this information could be combined with other, possibly historic data and thus reveals much more information as originally intended. The same applies to potential future uses of disclosed personal information. Next to combination of separately harmless information the loss of context is another problem persistently stored data holds. A sentence said today in a specific situation in front of a specific group of persons may have a totally

different meaning if replayed outside of this setting. It is nearly impossible to guess such transitions of information.

Further, the increasing amount of personal information persistently available at various locations elicits interests in usage of this information besides its original, agreed purpose. A (mis)use for marketing purposes usually is not perceived as a strong privacy violation. In contrast, temptations to use this information for law enforcement may lead to more critical privacy issues in the long run. Already today a lot of digitally tracked information is used for crime investigations, for instance server log files (Diffie & Landau (2007) provide staggering insights on this topic). Why not use information tracked in smart environments too, or – utilizing a smart environment's anticipating capabilities – why not use predicted *intentions*? If so, shouldn't it be prohibited to delete such information because it is a helpful utility in crime preventions and investigations (Bohn *et al.*, 2003)? Such thoughts seem to be impossible today but it is likely that once such data is widely available, corresponding discussions will arise.

All these issues impede the management of privacy in public. The distribution, aggregation and longevity of information disclosed outside an explicit private space make it hard to understand and control who has access to which data for what purpose. Indirectly, unintended disclosures of personal information could also implicate invasions into private life, e.g. by being inundated with advertisements or by being forced to defend imputations caused by misinterpreted or wrong personal information.

2.3.3 Interfering Social Interaction

Smart environment are not only designed to assist individual users but also to support interaction within person groups, for instance collaborative work or social leisure activities. In this regard services in the environment provide various modalities to communicate and exchange information between inhabitants, i.e. the disclosure of information is a dedicated feature. The potential privacy issue here is not the communication of data in principle but the (in)appropriate respectively (un)intended mediation of information. Interacting with other persons in a smart environment is a social process which involves the exchange of information on a give-and-take basis. Not only the persons one interacts with but also the current situation, including expectations and social norms affect what information a person is willing to disclose within the environment. For instance a formal meeting situation and a casual get-together usually have different implications on information exchange, even in context of identical person groups. A smart environment which disregards such differences has the potential to violate interpersonal privacy.

Moreover, the various modalities of information exchange provided by smart environments complicate interpersonal privacy management, for instance due to automatic distribution of information or due to semantic transformations of information during a communication process (see figure 2.2). Even today, in comparably simply equipped technological environments, one can find such examples. Unwanted e-mail arrival notifications popping up while giving a talk is one of them. Feature rich smart environments as described in section 2.2.5 bear much more: medical information may be revealed indirectly due to activated assistive technologies, emotional conditions might leak because of a corresponding acoustic or visual personalization of the environment, and collaboration environments may display inappropriate documents. While not necessarily violating one's dignity such disruptions still alter a person's originally intended boundary between itself and others and thus could negatively influence any subsequent interaction. Consider a contract negotiation meeting between representatives of a software company and a client corporation as an example. During such negotiations each party tries to carry its point as much as possible. Next to bare facts (pricing, set of features) negotiations are driven by potentially sophisticated social nuances adjusted by the selective disclosure and withholding of information. An unintentionally disclosed (or withheld) information might not be perceived as such by an interaction partner, but it could corrupt an interaction from the perspective of the (involuntary) information provider. The problem of semantic transformations may occur due to improper modalities used to communicate information. An environment which allows to distribute information to other inhabitants could do this, for instance, via a shared large display or by forwarding this information to the inhabitant's mobile devices. Either one has different implications on social interaction. Own contact data shown in full screen mode on a large display wall instead of being sent to mobile devices appears rather priggish. The other way around, there may be cases when a document preferable is displayed on a shared screen where each person retrieves the information uniformly and only for a limited time.

These issues show that smart environments – actually intended to seamlessly support a multimodal exchange of information – are also able to derogate interpersonal communication if they disregard the fact that "privacy management is a dynamic response to circumstance rather than a static enforcement of rules" (Palen & Dourish, 2003). Privacy issues concerning interpersonal communication probably are the most challenging ones. They depend on a variety of – partly diffuse – factors and are highly individual (Ackerman, 2000). One problem is the lack of general patterns. Even the defensive scheme to disclose as less information as possible could be the wrong decision if a social context advocates the opposite.

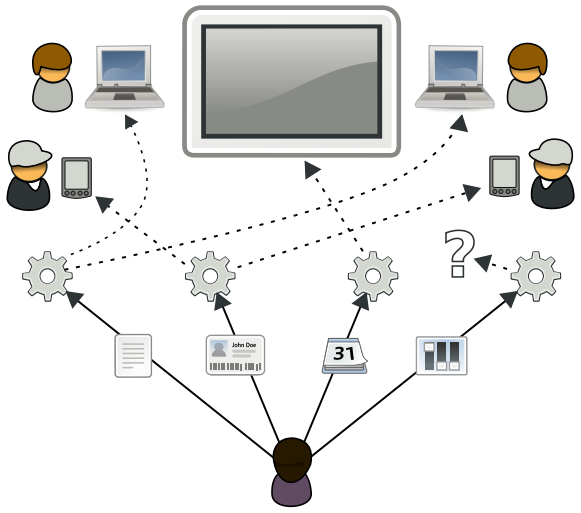

Figure 2.2: Services in ubiquitous computing environments may distribute various personal information (documents, contacts, schedules or arbitrary environment configurations). Users have a hard time in understanding and controlling related interpersonal privacy implications.

2.3.4 Conclusion

Smart environments have the potential to violate each of the privacy concepts presented in section 2.1. The described issues often correlate with each other. For instance blurred natural borders also raise problems with regard to social interaction and misused information within a public space could result in invasions into a private space. Hence, resolving one issue may obliterate another one. Still, some issues cannot be eliminated solely by technical means from the perspective of the environment. They may require legal regulations, social norms or utilities which empower users to cope with them individually. This directly leads to the next chapter, which reviews existing work on protecting and managing privacy in ubiquitous computing in general and smart environments in particular.

3 Related Work

The previous chapter described various privacy issues one can find in smart environments. This one investigates to which extent current research is able to handle them. The first section presents some general high level guidelines and principles elaborated by other researchers. Subsequently, section 3.2 reviews existing work about privacy needs and patterns, based on conducted surveys and "lessons learned" publications. Section 3.3 provides an overview about specific technical solutions to protect and manage privacy. Finally, section 3.4 determines still open issues and narrows them down to those covered by this work.

3.1 General Guidelines and Design Principles

Privacy issues in ubiquitous computing scenarios are a lively research area since the vision of ubiquitous computing has been expressed. Hence, a number of theoretical frameworks have been developed which suggest general guidelines and design principles for privacy protection and management.

3.1.1 Feedback and Control

An early work is that of Bellotti & Sellen (1993). It discusses privacy issues and possible solutions in context of RAVE (Gaver *et al.*, 1992), a media space equipped with technologies to capture and communicate audio and video data for the purpose of improved collaboration and interaction. Their central message is that a sane handling of private information requires *feedback* about when and what information is captured and to whom it is made available as well as *control* over what information one projects to whom. In particular these mechanisms should provide appropriate means for the following affairs: *capture* (what information is being picked up), *construction* (how information is being processed, including storing, encryption, or combination with other sources), *accessibility* (who has access to my information), and *purpose* (for what purpose information is being used, now as well as in the future). In addition they suggest several evaluation criteria to be used as a checklist when evaluating how ubiquitous computing applications realize

the proposed scheme of feedback and control. Among others these include *trustworthiness* (reliability, confidence and understandability of mechanisms), *timing* (feedback should be provided when control is required and most effective), *perceptibility* (feedback should be noticeable), *unobtrusiveness* (feedback should not distract or annoy users), *intrusiveness* (feedback should not invade the privacy of others), *flexibility* (control should adopt to individual privacy needs), and *effort* (control should require as few actions as possible). Often it is not possible to respect all criteria, e.g. a low effort tends to reduce flexibility.

3.1.2 Fair Information Practices

Based on the fair information practices expressed by the Directive 95/46/EC of the European Parliament and Council of the European Union (1995), Langheinrich (2001) has compiled a list of corresponding design principles for privacy-aware ubiquitous computing applications. These principles have a broader perspective than the framework of Bellotti & Sellen, which mainly focuses user interfaces. Following is a short summary:

Notice: This principle expresses the fundamental requirement that any data collection should not take place without being noticed by the monitored subject. This matches the above mentioned concept of feedback.

Choice and Consent: In response to proper notifications users should be able to choose which information others may capture and use. There should not be any data collection without explicit user consent. This basically refers to what Bellotti & Sellen call control. Langheinrich highlights that the interaction modalities in ubiquitous computing often conflict with this principle.

Anonymity and Pseudonymity: The need for a (practically often hard to realize) consent can be avoided by utilizing anonymization techniques. Applications which require some form of authentication or which provide personalization should, if possible, incorporate pseudonymity, a weaker form of anonymity.

Proximity and Locality: This principle aims to balance practical shortcomings of the previous ones. If neither of them is reasonable, limiting the spatial distribution of information still preserves a certain level of protection, for instance by limiting the access to some information to the physical location where it originates.

Adequate Security: A necessary (but not sufficient) requirement for privacy protection is the secure processing of information. Though, it is not a practical solution in all cases, e.g when involved devices do not have enough resources or lack interfaces required to integrate user-side secrets. The notion *adequate* proposes to deploy security models which

are reasonable with regard to the sensitivity of communicated information and the cost-value ratio of an attack.

Access and Recourse: Privacy protection cannot be realized solely by technical means but requires appropriate social conventions and legal regulations. However, systems should incorporate technologies to express, enforce and evaluate corresponding policies, e.g. a data collection and usage limitation policy.

3.1.3 Genres of Disclosure

Palen & Dourish (2003) brought up the term *genres of disclosures* to describe socially constructed patterns of privacy management. A genre refers to common negotiations of the boundaries disclosure, identity, and time (which have been described more detailed in section 2.1.3) within a specific social context. Systems should be designed to align with known genres of disclosures, otherwise they are likely to cause privacy violations. For instance an availability and activity notification system might be useful and accepted within certain working environments but does not similarly apply to home environments where the sixteen year old daughter refuses to let her parents always know what she is doing with whom. Both cases differ in their typical disclosure boundary. An example for a mismatch with the identity boundary is given by a location tracking system used in a company. The system was supposed to track persons in their professional role for the purpose of improved collaboration. However, only staff actually moving around a lot perceived it as such. Staff working most of the time in their office could not map the tracking functionality to their professional role – there was no obvious use. Hence, they perceived it as a system monitoring them as a private person. Of course, new technologies sometimes require and provoke new genres of disclosure. Though, respecting existing genres greatly simplifies the realization of proper feedback (or notice) and control (or consent) mechanisms because privacy related implications usually are already known and accepted.

3.1.4 Design Pitfalls

Complementing general guidelines, Lederer *et al.* (2004) have compiled a list of pitfalls designers of privacy-affective systems (mainly in the field of ubiquitous computing) should take care of. These pitfalls relate to a user's *understanding* of potential privacy issues and available *actions* to handle them (in that they match the notions of feedback and control given by Bellotti & Sellen). They can be summarized as follows:

Obscuring potential information flow: In order to let users decide the usage of a system, it should not conceal what kind of *potential* disclosures may occur. This includes missing as well as ambiguous privacy declarations. For instance anonymization techniques usually only apply to certain communication layers, but not so well informed users could assume to completely act anonymously.

Obscuring actual information flow: In a related manner designs should not obscure *actual* information disclosures but display them as immediate as possible. Otherwise users cannot estimate their current degree of privacy respectively how they appear to whom in situ. This is also important in order to properly react to accidental disclosures.

Emphasizing configuration over action: Systems should avoid comprehensive but complex privacy settings – they tend to be ignored or not understood by the majority of users. Instead privacy management should be integrated into already existing actions and assume safe defaults. For instance, smart environments should not activate privacy-affective services unless an incoming person holds a badge near a receiver at the entry door.

Lacking coarse-grained control: If the latter one is hard to realize, system designs should not miss simple top-level mechanisms for disclosure control. Examples are interfaces which adapt the semantics of power buttons, e.g. to hide one's location, or simple ordinal controls, e.g. to adjust the precision of location data. Such interfaces offer direct feedback and simple control at the same time.

Inhibiting established practice: Systems should not disregard existing technical and social conventions (like genres of disclosures as described above) but try to map them to new technologies. For instance, context-aware phones which are able to communicate a user's activity to a caller in order to explain why a call is not accepted, should not do this automatically as it violates the convention of plausible deniability.

3.1.5 Further Objectives

The presented frameworks are only a selection which covers the most important issues. Other frameworks often overlap with these ones but emphasize specific perspectives or add further insightful principles. For instance in an own work a list of objectives to diminish user concerns about privacy in smart environments has been compiled (Bünnig & Cap, 2007). These objectives partially overlap with Langheinrich's principles but assume the perspective of a user interacting with smart environment services. Additionally, they have a stronger focus on technical approaches. One of its findings to add here is that services processing personal information should support user-driven *audit* mechanisms, i.e. methods which enable users to autonomously verify policy compliance. Institutional validation

mechanisms probably are not able to handle the potential vast amount of services and service providers in ubiquitous computing environments. This mainly affects the concept of privacy in public. Audit mechanisms do not eliminate corresponding privacy violations but tend to repress them because violators can be held accountable.

Kobsa (2007) mentions two further noteworthy objectives. First, stored personal information should be *distributed* to several interacting clusters, each covering only a limited amount of users and a limited portion of user information. Further, he recommends to shift the storing and processing of information to facilities owned by users. Applications which require aggregated user information could utilize homomorphic encryption techniques. These objectives technically limit respectively disable violations of the concept of privacy in public.

Jiang *et al.* (2002) propose the principle of *minimum asymmetry*. Information flows in ubiquitous computing should be designed to "minimize the asymmetry between data owners and data collectors and data users". With reference to privacy in public, this mostly means that information flows from owners to collectors need to be reduced while they should be increased the other way around. With respect to privacy in public this means, for instance, that each information item communicated to an entity should be responded with information about how it is being used. An example in face of interpersonal privacy is that a receiver of some contact information should provide the same information in return. Esquivel *et al.* (2007) promote a similar idea using the "fair trade" metaphor. In practice such a symmetry is hard to realize as the sensitivity or value of information may be perceived differently among individual information sharing entities.

3.1.6 Summary and Open Issues

These guidelines and principles do not only help in the design phase of ubiquitous computing applications respectively smart environments, but also function as evaluation frameworks for existing systems. The presented objectives often affect more than one concept of privacy. However, most comprehensively they deal with issues concerning privacy in public (and implicitly help in preserving private spaces). The general notions of feedback, control and audit also apply to interpersonal privacy, but do not sufficiently cover it. Here the genres of disclosure proposed by Palen & Dourish and the approach of minimum asymmetry by Jiang *et al.* are important principles. Chapter 7 (Guidelines for Interpersonal Privacy) extends the guidelines presented here by explicitly addressing interpersonal privacy management.

3.2 Patterns of Privacy Management

Next to high level guidelines and principles, designing privacy management solutions requires an understanding of *how* users practice privacy, i.e. by which needs it is driven and which patterns it reveals. This section reviews corresponding work. As this mainly affects information leaving a private space, the review is limited to the concepts *privacy in public* and *interpersonal privacy*. Note that the following grouping by these two concepts is a loose one – often patterns affect both privacy concepts.

3.2.1 Privacy in Public

Ackerman *et al.* (1999) conducted a survey to understand privacy concerns of users of e-commerce web sites. Though not directly related to ubiquitous computing scenarios, it provides some general insight on how users practice privacy in public. The authors were able to distinguish three general clusters of users: privacy fundamentalists, marginally concerned users and – the majority – pragmatic users. The latter group mainly balances information disclosure with corresponding benefits. This corresponds to the observations made by Grudin & Horvitz (2003) who identify the general privacy management strategies of *pessimistic*, *optimistic*, and *mixed* access control. Here pessimistic refers to preventive control, optimistic to retrospective control, and mixed to on-demand or in-situ control. Other findings by Ackerman *et al.* are how comfortably users feel about sharing specific types of information and which feedback and control possibilities users considered most important. In general most users dislike the sharing of financial and medical information but are more relaxed with regard to personal preferences – probably because they mostly provide useful personalization and do not have an obvious potential for a crucial misuse. Concerning feedback and control users mostly care about the factors with whom disclosed information is shared, for what purpose it is used, and if it is possible to opt out. The declaration of privacy and data retention policies were comparably less relevant factors. The authors guess this could be due to a lack of trust in and understanding of such warrants.

West (2008) claims that the decision to disclose information is strongly influenced by the perceptibility of positive implications. Users expect to gain something when making decisions about information disclosure, i.e. they want to be rewarded for making good decisions. The point is that providing less information theoretically protects privacy but practically yields less perceivable rewards. In contrast, providing more information usually has immediate effects. As an example, consider a service which requests some personal information in order to work and which does nothing when no information is provided. To also reward a defensive information handling users should be noticed about their current

strong privacy level and services should provide some fallback activities which do not require personal information. However, while West's arguments are commendable in terms of privacy, service providers probably are less motivated to reward a defensive information sharing behavior.

Another interesting factor influencing information disclosure is given by Huberman *et al.* (2005). They conducted a survey where participants exchanged personal information in the style of a reverse second-price auction (each piece of information was valued with real money). One of their findings was that participants valued information by how strong it deviates from the average among all participants. There are two reasons why uncommon information is handled more sensitively: it is easier to deduce the owner of the information (identification issue) and there is a risk of embarrassment (social issue). In this respect the norm deviation of personal information actually is also a pattern in the management of interpersonal privacy.

3.2.2 Interpersonal Privacy

Adams (2000) investigated privacy issues in multimedia spaces, i.e. environments using audio and video facilities for interpersonal communication. She highlights that primarily communicated information is accompanied by several second level information – spoken words also disclose voice intonations and faces usually expresses emotional conditions. Hence, Adams not only states information sensitivity, recipients and usage as important factors but how users *perceive* them, based on recognized secondary level information. Joinson *et al.* (2006) refer to such personal perceptions as dispositional disclosure variables (in contrast to situational ones). These works show that privacy is a highly individual affair and that it cannot be managed solely based on the primarily communicated information itself but with regard to how it is communicated and which further information it mediates.

Lederer *et al.* (2003b) investigated the relative importance of the recipient of an information item and the item itself – primarily the current situation one is in – when deciding its disclosure. For that purpose they conducted a survey which asked participants to choose the accuracy (true, vague and blank) of two possible activities and locations (working lunch at downtown and social evening in a live club) when requested by an inquirer (given as an abstract role, e.g. employer or stranger). The main result is that the information recipient is a more important factor than the communicated information. Participants more often chose the same accuracy for different situations but identical inquirers than the other way around. Though, for the person role employer, the information itself appeared to be an equally important disclosure parameter.

Cadiz & Gupta (2001) ran a study in order to examine how people make privacy decisions when sharing various information (e.g. contact and calendar information, medical data, hobbies, music, and current activities) with other persons (in contrast to impersonal entities). Persons are referred to by abstract roles, e.g. family members, friends and co-workers (scaled by how close relationships are). Participants were able to constrain information sharing by enforcing notices when an item actually is shared and by limiting it to certain time frames. One finding of their qualitative analysis is that these features have been endorsed by most participants but practically they did not influence disclosure decisions in most cases. Another result is that disclosure decisions are driven by a pattern of four questions:

- Does this person already have this information?
- Does this person need to know this information?
- Do I care if this persons has this information?
- Is this person trustworthy?

The importance of these questions varies among participants. Some only considered if they care about sharing an information item with a specific person. Others mainly asked if the person in question needs an information item at all. The study also shows that there is no common strategy of openness for specific groups (e.g. family members) or person types (e.g. a friend). Some treated all family members the same while others did not. Similarly, even close friends not always implicate openness, e.g. in case of an intimate but gossip friend. Finally, when managing information disclosure, Cadiz & Gupta suggest to distinguish between *fast* versus *slow changing* information and *descriptive* versus *communicative* information. Fast changing information quickly gets invalid and thus mostly is disclosed with less concern. In contrast, rather static information cannot easily be revoked once it has been disclosed. Descriptive information (e.g. job, age, preferences) is perceived to be more personal than communicative information (e.g. phone number, location), which does not primarily say something about a person but is used to get in touch with each other. However, this distinction is not sharp. Communicative information may involve personal data (e.g. an e-mail address can reveal the company one works at) or it may induce descriptive information when aggregated (e.g. location tracking). The different information types should be recognized when designing means to manage their disclosure. Slow changing information should be handled more carefully and communicative information should be prevented from becoming descriptive and thus unexpectedly revealing personal details.

Prabaker *et al.* (2007) analyzed how users tend to specify disclosure rules in context of a location sharing application. Rules could be parametrized by the location itself, time and information recipients. When applying these rules, users were asked to justify how well automatic disclosures match actual disclosures they would choose manually and ad

hoc. Initial definitions of rules did not perform well in practice and had to be revised multiple time. Still, final revisions only reached an accuracy of 60% to 80%. Toninelli *et al.* (2009) came to similar results concerning phone call acceptance policies. Users have strategies in mind when to accept or reject phone calls and when to annotate rejects with explanatory context information. Though, they have problems to define them formally. The authors mention two reasons why users fail in accurately defining disclosure rules. First, predefined disclosure decisions face hypothetical situations which might mismatch real situations or do not sufficiently cover them. Second, intuitive strategies cannot easily be mapped to formal rule systems because they use different vocabularies and because it is likely that the formal rule system is not equally powerful, especially when it comes to exceptional cases and fine-grained disclosure parameters. Though, rules defined ad hoc (i.e. on information request) or retrospectively usually are more accurate than predefined ones.

3.2.3 Summary and Open Issues

Several patterns have been identified. None of them appears to be valid for all users. Users follow one of three three general approaches to privacy (pessimistic, pragmatic, and optimistic) while individual information disclosure decisions may be influenced by multiple factors:

- Does the receiver already have this information?
- Does the receiver need to know this information?
- Do I care if the receiver has this information?
- Is the receiver trustworthy?
- Is this information fast or slow changing?
- Is this information descriptive or communicative?
- What is the type of this information?
- How does the receiver perceive this information?
- How does the information deviate from the average?
- What are the immediate positive effects of (not) disclosing this information?

Obviously the decision process to disclose some personal information may get very complex and easily diverges among individual users. Not only that different users concentrate on different factors, even identical answers to one of the above questions might eventually result in different disclosures. For instance some users do not feel comfortable with disclosing information which deviates from a social norm while others explicitly prefer such deviations. Hence, it is hard to derive general rules from these patterns.

The reviewed work mainly considers information recipients, information itself and how it is used. But, surveys did not investigate the influence of circumstances accompanying disclosures, e.g. one's own role and the social context (besides the general social relationship to a single information receiver). Chapter 4 (Information Disclosure Patterns) deals with these aspects.

3.3 Specific Technical Solutions

Of course related work did not only elaborate theoretical frameworks and conducted empirical studies. There also exist various technical solutions to tackle privacy issues in smart environments. In relation to the privacy concepts presented in section 2.1, the following subsections provide an overview about these solutions.

3.3.1 Staying Private

Supporting the concept of privacy in private means to help users in preserving natural borders between private and public spaces or – in other words – to help them in preventing private information entering a public space (which obviates the need to manage privacy in public). Specific technical solutions for staying private typically utilize anonymization and pseudonymization, but may also explicitly provide means to define borders between private and public spaces.

Boundary Regulation and Awareness

In context of smart environments, an example for border management is the *virtual wall* proposed by Kapadia *et al.* (2007), an intuitive mechanism to control the spatial range of visibility of sensor information. Inhabitants of an environment set up a virtual wall to decide which information captured by sensors *within* an environment may be visible *outside* the environment, i.e. behind the virtual wall. The wall can be configured to be opaque or partly respectively completely transparent. For instance a partly transparent wall could show that and how much persons are in an environment, but not which persons. The virtual wall is a straightforward concept with a twofold purpose. First, it makes inhabitants aware of the fact that the real boundary between private and public space might not match the natural one assumed by physical borders. Second, it provides a natural metaphor to align it with personal preferences. The virtual wall does not apply to all aspects of information sharing potentially occurring in smart environments but for its specific use case it is a commendable solution.

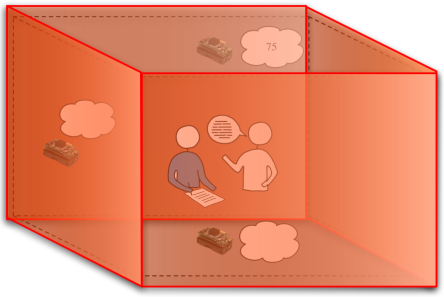

Figure 3.1: The transparency of a *virtual wall* decides the visibility of certain sensor information outside an environment. *Source:* Kapadia *et al.* (2007)

Similar to the virtual wall, any solutions promoting intuitive awareness systems help users in staying private in that they are able to distinguish private from public spaces and thus not accidentally reveal personal information or mistakenly feel their private space invaded although this is not the case (Beckwith, 2003). For instance EuroPARC's RAVE, an early example of a ubiquitous system, used audio notifications, in particular the sound of an opening door, to notify inhabitants of an environment that some external person is going to virtually drop by using the environment's camera (Gaver *et al.*, 1992). Constantly running cameras in turn may be highlighted by being mounted on self-explanatory objects, e.g. a cameraman statue (Bellotti & Sellen, 1993).

Anonymization and Pseudonymization

There are several solutions using anonymization and pseudonymization. One noteworthy example is the decentralized and pseudonym-based context exchange scheme by Evans *et al.* (2007). It is designed for a polling-based communication where *Bob* wants to check if *Alice* is in a specific context C. Alice uses her context and identification, and a key shared with Bob to generate a pseudonym which is then placed in a database accessible by Bob. He can then check if Alice is in context C (i.e. he must have some assumption which contexts to ask for). Pseudonyms do not have to be stored centrally but just need to be accessible by both communication parties. Preferably they are stored in a database physically bound to a context. Evans *et al.* primarily focus transport applications but their scheme may also be used for asynchronous information exchange in smart environments. It provides a private communication channel but may only be used for scenarios where a polling-based communication is reasonable.

Al-Muhtadi *et al.* (2002a,b, 2006) and Kobsa & Schreck (2003) describe anonymization techniques to authenticate users and to handle user profiles (for user-adapted services) in

such a way that the infrastructure provider as well as the participating services cannot link user information to real identities. The common basic concept is to use *mix techniques* as originally described by Chaum (1981). An anonymous service authentication scheme using capability-based authentication and partially blind signatures is given by Konidala *et al.* (2005). Beresford & Stajano (2003) apply mix techniques to location-based services where users interact with services depending on their location but without allowing service-providers to track individual users. Though, the middleware implementing the *mixing* and mediating between services and users has be to be trusted. Nohara *et al.* (2005) and Bessler & Jorns (2005) also use a trusted third party (e.g. the wireless carrier of a mobile user) to allow pseudonymous service interaction. Effectively such third-party–dependent pseudonymization techniques do not provide completely private communication channels, but already require users to put some linkable personal information into a public space. Hence, such solutions only partially help users in staying private. Actually they are special ways to manage privacy in public.

A worthwhile further reading about technical solutions aiming to protect privacy along the notion of privacy in private is Wright *et al.* (2008, section 5.1).

3.3.2 Managing the Publicity of Information

Whenever the usage of a service requires to put linkable personal information into a public space (where public does not mean general public but a certain set of other persons or entities), users face the problem of managing privacy in public. Basically this means to regulate access to published information as well as altering or revoking it. Hence, technical solutions for managing privacy in public have to provide mechanisms to express privacy preferences which regulate information access as well as mechanisms to enforce these preferences. Different but often combined concepts are used to accomplish this. Information receiver accessing and processing personal information may provide policies how and for which purpose information is used. In turn, users define preferences (i.e. rules) which are mapped to such policies in order to decide information disclosure. Another concept is that users express privacy preferences by defining context-based rules. To specify the actual set of information to disclose, identity-management systems are a common approach. Abstract roles often are also used to categorize information receiver, let it be institutions or natural persons. Finally, personal agents, information management repositories and audit mechanisms are used to enforce disclosure preferences. The subsequently reviewed technical solutions demonstrate the application of these these concepts.

Concerning privacy policies, Langheinrich (2002) describes *pawS*, a privacy management system using Platform for Privacy Preferences (P3P) (W3C, 2002b) and P3P Preference Exchange Language (APPEL) (W3C, 2002a). Originally these languages have been de-

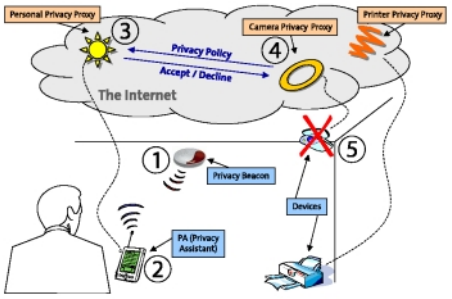

Figure 3.2: Illustration of *PawS*, a P3P-based privacy management system for ubiquitous computing environments. Services and devices in an environment announce their privacy policies while a user's personal assistant sets up the services to align them with configured privacy preferences. *Source:* Langheinrich (2002).

signed for web privacy management. Langheinrich introduces some extensions for special requirements in ubiquitous computing scenarios. Environment services use P3P to announce how they handle personal information. In turn users can specify personal preferences using APPEL. Preferences and policies may then be used to decide which information is disclosed to or captured by the environment. Services provide their policies either implicitly on service interaction or periodically with so called privacy beacons.

Myles *et al.* (2003) describe a policy-based approach for exchanging location information. A central component, the *location server*, keeps location information of users and handles their distribution. Information requesters have to specify a usage purpose for a location request. Users are able to setup *validators* which get contacted by the *location server* before any location information is disclosed. Validators are configured with rule templates which decide for whom and for which purpose a location may be disclosed.

User-friendly interfaces to set up preferences to negotiate information disclosure with regard to policies are a crucial requirement for the efficiency of such solutions. Especially P3P and APPEL are not suitable for direct user interaction but require appropriate tools to set up preferences. Even for website-related privacy management only, this issue is still subject of research (Cranor *et al.*, 2006; Besmer *et al.*, 2010). In ubiquitous computing environments the lack of suitable tools is even worse. In any case, cooperative policy authors are required. Pollach (2007) claims that policies for institutional information exchange in practice often are compiled to protect institutions from legal issues – instead of supporting actual privacy needs of users. For instance they often use ambiguous or downplaying terms to obfuscate actual collection and usage of information. To set up rules for the validators used in the location sharing system presented by Myles *et al.*, users are provided with rule templates representing common disclosure patterns. Still, it is likely

Figure 3.3: Role-based information disclosure control using the "Privacy Manager". *Source:* Lederer *et al.* (2003a).

that such precompiled template repositories are no practical solution for managing general information exchanged in smart environments. Some researches propose to use learning methods to formally conceptualize privacy preferences (Saleh *et al.*, 2007; Zhang *et al.*, 2007). However, both still require users to abstract privacy preferences in advance in that information receiver as well as the set of disclosed information must be described using roles, identities or generalized linear degrees of disclosure accuracy.

An illustrative example for identity-based privacy management is the *Privacy Manager* presented by Lederer *et al.* (2003b). Here users define a set of *faces* and control the disclosure of personal information by linking information recipients, situations and faces (see figure 3.3). Every face describes a subset of personal information or a degree of information accuracy to disclose. Related approaches are given by Jendricke *et al.* (2002), Clauß *et al.* (2002), and Maibaum *et al.* (2002). The basic common idea is to abstract a specific set of personal information to a role or identity, e.g. "anonymous", "private", "job", or "public". Roles are supposed to provide an easy way of managing personal information. However, role concepts always conflict between simple but too general and subtle but too complex. Indeed Lederer highlights this problem of generality in a subsequent work (Lederer *et al.*, 2004). The tension between abstraction and specialization also applies to information recipients modeled as abstract personae, e.g. "friend" or "employer".

The disclosure management solutions described above rely on social or legal regulations and audit (Dekker *et al.*, 2007) to ensure that disclosed information is handled according to a users stated preferences respectively a receiver's stated policy. The *Confab* toolkit, a privacy management solution given by Hong & Landay (2004) *technically* enforces a proper handling. Here personal information is organized in so called *info spaces*. For instance an individual's info space could contain a current location and activity. Info spaces are managed by servers which run on user-owned devices. Any information exchange is handled by these servers, i.e. if two parties share some information, both must use an info space server for sharing. The rationale is that information items in info spaces are annotated with privacy tags which describe under which conditions an information owner

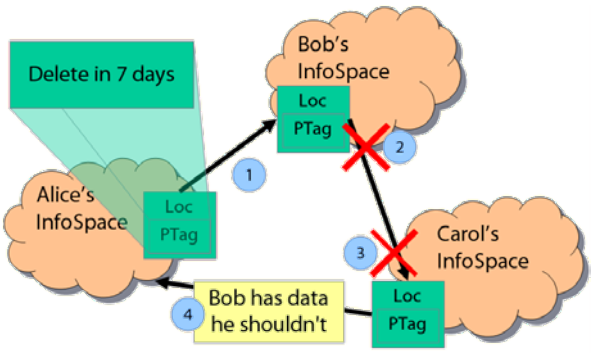

Figure 3.4: Interaction of info spaces in *Confab*. Personal information items are annotated with privacy tags to regulate the access and processing of these items. Information exchange is managed by *info spaces* which recognize and enforce these annotations. *Source:* Hong & Landay (2004).

is willing to disclose the information. Information requests as well as information items (including privacy annotations) are encoded in a Confab-specific XML dialect providing any information needed for automatic data handling. Confab supports pessimistic, on-demand, and optimistic disclosure control. For pessimistic control, users may define precise conditions by whom, in which context, for what purpose and possibly how long an item may be accessed. In case of on-demand control, an information owner is requested to provide an acknowledgment whenever a corresponding item from her info space is queried. Optimistic control just logs any access while possibly raising an alarm event when certain access limits are exceeded. Confab mainly addresses the sharing of context information whose capturing and processing tightly integrates with info-spaces. Technically it could also be used for any other personal information but this would require users to convert and annotate them so that they can be handled by info space servers. In any case, users must be trusted not to use communication channels outside of info spaces – otherwise privacy preferences could not be recognized and enforced. While this may be a practical assumption for context information (where processing applications are well-integrated with info spaces), it shouldn't be assumed for arbitrary other shared information.

Although some of the solutions presented here also address interpersonal information exchange, they do not fully support the concept of interpersonal privacy for reasons elaborated in the next section.

3.3.3 Supporting Interpersonal Privacy

As reasoned in section 2.1.3, managing privacy in context of interpersonal interaction has its specific characteristics. It does not focus information and associated rights but inter-

action and corresponding social relations, norms, and expectations which drive mutual information exchange. Obviously a technical assistance for such a privacy management is more difficult to achieve. The assistance has to respect the dynamic nature of social relations and context as well as the increased individuality of privacy preferences (compared to the management of general public access to personal information).

At first glance solutions to manage privacy in public appear to be similarly applicable for interpersonal privacy management. However, these solutions primarily focus the exchange of personal information with abstract entities (i.e. institutions or generalized personae) and generally require users to express their preferences in advance within a given formal system for expressing privacy preferences (i.e. using identities, policy negotiation preferences, disclosure rules, or personae). In that there are suited well for scenarios where users have a rather clear understanding of whom to give which information under which conditions and where users are able (in both qualitative and quantitative terms) to express this understanding. However, when a user's disclosure behavior is more complex and harder to describe formally, e.g. during social interactions within smart environments, these concepts aren't practically. Whichever identities for oneself and personae for others one has defined, it is likely that they aren't sufficient. Even if they are, people tend to move interaction partners seamlessly among different abstract personae (Ackerman, 2000). Finally, the existing user interfaces hardly incorporate interaction modalities available in smart environments.

To some extent the mentioned *Confab* toolkit by Hong & Landay aligns with interpersonal privacy management. Next to a priori disclosure control it provides dynamic situational control as well as posterior control. Additionally, it does not abstract information recipients to generalized roles but allows to express and deploy preferences tailored to specific interaction partners. Still, information management primarily is focused on asynchronous access rights (in contrast to direct interaction) and requires information to be exchanged within the *info space* driven infrastructure which requires personal information explicitly integrated into that infrastructure in advance.

A promising approach to automate disclosures while still allowing dynamic handling of information exchange is given by Prabaker *et al.* (2007). In their already mentioned location sharing system (see section 3.2.2) users often failed in expressing precise rules for disclosing their location to other persons. In contrast, automated disclosures learned using case-based reasoning (CBR) could improve the general accuracy of disclosure decisions. Unfortunately they do not provide further details here, e.g. why particularly CBR has been used and how its performance evolves over time. Further, disclosure decisions are binary only: hide or show a location – a rather simple model compared to the potential diversity of information exchanged in smart environments. Another limitation is that disclosure

rules consider social relations to the information receiver but not the social context of information exchange – primarily because information is exchanged remotely. The same applies to the "socially aware access control policy model" by Toninelli *et al.* (2009). However, they suggest to assist users in defining disclosure rules based on templates and additionally allow users to take individual perspectives when defining such rules, i.e. users may define access rights based on information items, information receivers, or request situations – whichever *parameter* comes first in a user's mindset. Again, the proposed policy model concentrates on a specific type of personal information, in particular status information (availability, location). On the other hand, to some extent their policy model also affects interpersonal privacy management within direct social interactions in smart environments by controlling when a user's phone should ring or not, depending on the caller and the current context. The authors do not explicitly highlight it, but privacy management in this case is twofold. First, it controls information exchange with a caller. Second, it controls the callee's self-representation to social interaction partners by deciding if to interrupt an interaction in favor of accepting a call.

The bottom line is that interpersonal privacy management with regard to direct interactions in smart environment is addressed rarely. If so, it focuses specific information types and does not fully support the specific characteristics of interpersonal privacy (primarily dynamics and individuality) described in section 2.1.3 and the corresponding management patterns described in section 3.2.2.

3.3.4 Summary and Open Issues

There are a number of technical solutions aiming to solve privacy problems in smart environments. Mechanisms to support users to practice privacy in private, i.e. being aware of and controlling borders between private and pubic spaces, are quite mature and comprehensive. Similarly, various techniques have been proposed to manage privacy in public – although appropriate user interfaces are still a subject of research. To some extent the broad coverage of these solutions is due to the fact that there is a significant overlapping with concepts designed for "traditional" computing scenarios, mainly online activities. In contrast, solutions explicitly addressing privacy management in direct social interactions in smart environments are not satisfactory. This open issue is dealt with in the chapters 5 (Composite Disclosure Control) and 6 (Learning Disclosure Decisions).

3.4 Conclusion

This chapter reviewed existing work on privacy in smart environments with regard to general guidelines, privacy management patterns, and specific technical solutions to support

users in practicing privacy. It has been shown that each of these aspects lacks some work particularly focusing interpersonal privacy management during direct social interactions in smart environments. As anticipated in section 1.2, the following open issues are tackled by this work: (I) consideration of social aspects in interpersonal privacy management and resulting patterns in information disclosure decisions, (II) suitable disclosure control mechanisms that match these patterns, as well as their orchestration, (III) automating disclosure control in a user-adaptive but easy to manage fashion, and (IV) general guidelines and principles to develop interpersonal privacy sensitive smart environments and to evaluate corresponding solutions.

4 Information Disclosure Patterns

Smart environments should assist users in handling interpersonal privacy (with respect to modalities used to communicate information and possible social factors) by automating as much disclosure decisions as possible while minimizing the steps required to configure the automatism. Such an assistance obviously needs a certain understanding about how users practice privacy, i.e. by which patterns it is driven. Section 3.2.2 already presented some analytical work about patterns of privacy management in interpersonal communication. However, some questions remain, for instance the impact of a social context to disclosure decisions. One might have a clear and solid concept of whom to disclose one's e-mail address, independent of a social context. In contrast the disclosure of documents in a meeting may depend on the role one assumes or expectations others have on the meeting and on oneself. Furthermore, privacy management mechanisms usually assume that a growing information recipient group implicates a smaller set of disclosed information. This is a helpful characteristic for an automated disclosure assistance, but does it really apply to all disclosures? Or do there exist other *patterns* which may be valuable input to disclosure assistance mechanisms? It is likely that different patterns exist and that each one motivates specific disclosure assistance mechanisms.

An obvious approach to investigate these aspects of interpersonal information disclosure is to observe the disclosure behavior of users in smart environment settings. However, as those environments still are work in progress (most existing environments are prototypes in research institutes) and not yet integrated into everyday life it is often impossible to observe disclosure decisions in real life. Even if it would be possible to observe such disclosure behavior it is questionable if users are willing to reveal their disclosure decisions because they must be considered as sensitive personal data. An alternative which circumvents these issues is to capture disclosure decisions by conducting a survey. Here it is possible to express disclosure situations without necessarily having a productively used smart environment. Additionally it is easier to respect the sensitivity of disclosure decisions by providing a reasonable level of anonymity.

This chapter aims to get an understanding of the mentioned aspects of interpersonal privacy management with the help of a survey system. In that it deals with the specific issue (I) mentioned in section 1.2. First, section 4.1 presents a way to model disclosure

decisions, describes which structural information about disclosure habits may be extracted from a set of disclosure decisions expressed with that model and elaborates resulting implications for disclosure assistance mechanisms. Subsequently, section 4.2 describes how to capture disclosure decisions with the help of a survey which applies the model and evaluation methods from the previous section. It compiles a list of general requirements such a survey system has to meet and presents DiHabs, a corresponding specific survey system implementation. Section 4.3 describes the setup and results of an example survey conducted with the developed survey system and evaluates the results with regard to implications for disclosure assistance mechanisms in smart environments. Finally, the findings of these chapter are summarized in section 4.4.

4.1 Modeling and Evaluating Disclosures

This section describes a model to express disclosure decisions during social interactions in smart environments. Subsequently it elaborates which structural information a collection of disclosure decisions (represented by that model) provides, which potential privacy related patterns it reveals, and how these patterns influence disclosure assistance mechanisms. Within the scope of this section the patterns are compiled in an *explorative* manner. Their practical relevance is investigated with the help of the survey system presented in the next section. Their contribution to automating disclosures is evaluated in chapter 6.

4.1.1 Disclosure Model

The main factors driving interpersonal information disclosure are the disclosure situation and information recipients which usually are other persons one interacts with in an environment. A disclosure situation may be a *meeting in room X in context of project Y* or *coffee break chatting in Bob's office*, i.e. a combination of physical and social context. Especially in smart environments there is a third important factor influencing disclosure decisions: the modality used to mediate information. For instance a document might be displayed on a shared large screen or on devices of other persons in the room. In the latter case information recipients are able to view the document more thoroughly compared to a limited time display on a shared screen. The modality may also encode constraints how information recipients may use a disclosed information (e.g. a temporal access limits).

Based on this an interpersonal information disclosure within smart environment scenarios may be modeled by the following mapping τ:

$$\tau : S \times M \times \mathcal{P}(P) \rightarrow \mathcal{P}(I) \tag{4.1}$$

where S is a set of situations, M a set of disclosure modalities, P a set of persons potentially present within the environment and I a set of possible personal information items to disclose (e.g. documents, contact information, etc.). $\mathcal{P}(X)$ denotes the powerset of X. The precise content of S and M depend on a specific smart environment domain. Additionally S is likely to be specific to individual users, depending on how they *perceive* a situation Adams (2000). For now specific values of S and M are not elaborated in detail but assumed to be aggregated high-level parameters. This chapter puts the main focus on disclosure patterns yielded by relations between $\mathcal{P}(P)$ and $\mathcal{P}(I)$.

Alternative Models

Next to this model one can also think of alternative formalisms. For instance a state machine could be used to express states of disclosures with contextual conditions when to move from one state to another. This allows to express disclosure sequences (e.g. if one communicates information A, and if then event X happens, information B is exchanged next). Furthermore a memorization model could be used to encode which interaction partner already received an information item in the past in order to incrementally set up a database of access rights. However, state machine models quickly get very complex and bear the risk of overfitting as the specific dependency of a prior to a new disclosure situation reduces the generality of individual situations. The dependency on past situations, which even more applies to memorization models, also conflicts with the dynamics of privacy preferences in social interactions.

In contrast, the presented model aligns well with the characteristics of privacy management when interacting with other persons in smart environments. It represents highly dynamic decisions which primarily depend on the current setting (i.e. they do not only depend on information items and receivers and they mostly are independent from previous interactions). The model also allows to express disclosures triggered by an explicit request of an interaction partner (information is pulled by a receiver) as well as triggered by an impersonal event of the environment, e.g. entering the environment or start of a meeting (information pushed by the owner). Especially the latter case is quite common in collaboration-oriented smart environments where subjects rather *contribute* than *request* documents. In that the chosen model matches the claim of Palen & Dourish that "[interpersonal] privacy management is a dynamic response to circumstance rather than a static enforcement of rules" (2003).

4.1.2 Evaluation and Implications

Disclosure decisions made by a person constitute a subset of the graph of τ. It is possible to aggregate various structural information about disclosure decisions from such a subset. Patterns deduced from that structural information may be used to choose appropriate disclosure control methods and to enhance automatic disclosure mechanisms (as shown in chapters 5 and 6). For now only subsets of the graph of τ with a fixed situation $s \in S$ and a fixed modality $m \in M$ are considered, i.e. only the persons and disclosures vary. Which additional structural information may be aggregated from graph subsets with varying s and m is discussed at the end of section 4.1.2. Subsequently such a subset of the graph of τ is referred to as T. This set represents the disclosure decisions made by a particular person and thus expresses this person's disclosure behavior:

$$T := \big\{(s, m, g_1, \tau(s, m, g_1)), \ldots, (s, m, g_n, \tau(s, m, g_n))\big\}$$

with $s \in S$, $m \in M$ and $g_i \in \mathcal{P}(P)$ while $i \in \{1, \ldots, n\} \wedge i \neq j \Rightarrow g_i \neq g_j$. For this specific sub-graph of τ, D forms the set of disclosed information item sets only:

$$D := \big\{\tau(s, m, g_1), \ldots, \tau(s, m, g_n)\big\}$$

Similar G refers to the person groups[1] only:

$$G := \big\{g_1, \ldots, g_n\big\}$$

Note that $|D| \leq |G| = |T|$. With reference to T, D and G one can extract various information which describe different patterns of privacy preferences. The following subsections elaborate these patterns and deduce implications for information disclosure assistance mechanisms.

Number of Unique Disclosures

The size of the set D, which is at least 1 and at most $min(|T|, |\mathcal{P}(I)|)$, is a simple metric for the complexity of disclosure decisions. A small number of unique disclosures indicate a rather simple privacy preferences concept while many unique disclosures display a more sophisticated management of personal information. For instance a person who only distinguishes professional and private situations respectively information and who does not consider other aspects when deciding a disclosure is likely to generate only a few unique disclosures, two in an extreme case (private and public information). In contrast, someone who is very eager in disclosing very specific information sets based on a variety of factors

[1] In this chapter the term *group* does not denote a mathematical group but is used informally as in common language, i.e. in context of persons it actually refers to a *set* of persons.

(e.g. the precise constellation of persons receiving the information and multiple characteristics of the current situation), the resulting number of unique disclosures is comparatively high, potentially similar to the number of disclosure decision instances $|T|$. For only a few unique disclosures it may be possible to express the underlying disclosure behavior with a small, manageable set of rules. When there is a high number of unique disclosures, rules easily get very complex so that most users are not able or willing to express and maintain them. Automating disclosures still may be possible using machine learning based approaches, depending on the next structural parameter.

Number of Uses of Individual Disclosures

Even persons who make rather specific disclosure decisions which are closely related to various context information and thus result in a high number of unique disclosures may repeatedly disclose one or a few general purpose information sets in different situations. Examples are generic contact data or a default set of slides one shows at different presentations. A disclosure behavior which results in a repeated use of one disclosure allows an assistance mechanism to utilize this as a fallback disclosure. A high number of occurrences of individual disclosures in general increases the input machine learning driven assistance mechanisms have to conceptualize disclosure behavior. In contrast, if all disclosures are used only once or a few times, an automated disclosure mechanism lacks sufficient information to model disclosure decisions.

Poset Characteristics of D

The set of disclosures D is a subset of $\mathcal{P}(I)$ and thus it forms a poset (partially ordered set) with regard to the binary relation $\subseteq$ (see figure 4.1). The width of the poset (i.e. the cardinality of the poset's maximum antichain[2]) is another structural information describing the complexity of disclosure decisions. For instance a totally ordered disclosure set D has a poset width of 1 and indicates a simple linear privacy concept. In contrast, a D with a poset width equal to $|D|$ indicates quite complex privacy preferences where individual disclosures do not relate to each other (with regard to set comparability). A totally ordered set of disclosures mean that in every situation, the set of information to disclose either is identical, a superset or a subset of another set of disclosed information. More illustrative this means that disclosures move along one *path*. This fact

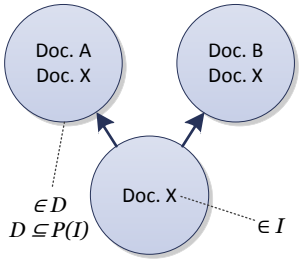

Figure 4.1: Poset graph representation of a set of disclosures D

[2] An antichain is a subset of a poset in which any two elements are incomparable. A maximum antichain is an antichain with a cardinality at least as large as every other antichain.

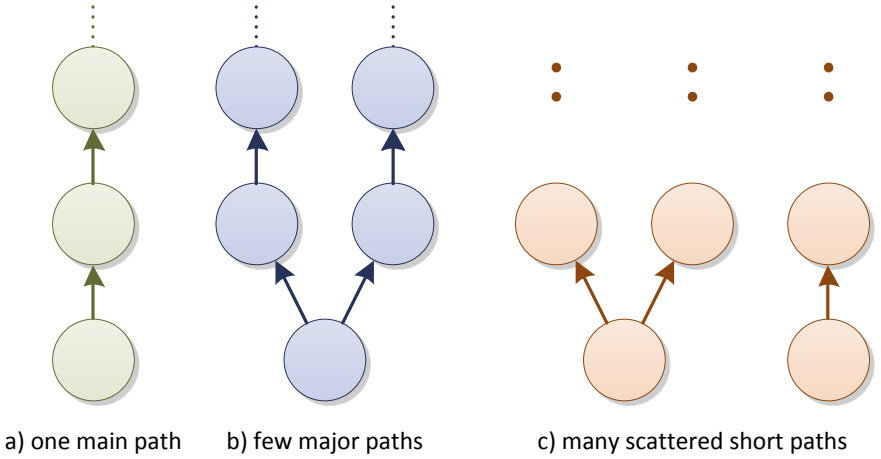

Figure 4.2: Possible disclosure graphs, given by the transitive reduction of the poset D. Disclosures may move along one or a few main paths (a,b) but may also miss any major paths (c).

greatly simplifies automated disclosures, let it be rules defined by users or machine learning based approaches. To some extent this is also valid for posets with a width greater than one. Each maximal chain[3] in a poset can be seen as an individual disclosure *path*. The smaller the poset width and the greater the lengths of the maximal chains, the more applies the pattern that disclosures move along a few certain major paths.

Different path structures are illustrated in figure 4.2. A practical example for one main path like 4.2.a is when a person decides disclosures solely based on how close relationships to information recipients are, e.g. ranging from *intimate* to *stranger*. Disclosures reflect this linear structure, e.g. they may range from *all* to *no* information. A scattered graph like 4.2.c could result from disclosure decisions driven by a variety of orthogonal factors which, in composition, have no linear order. For instance, when assessing information recipients not only on individual relationships but also on their constellation as a group and on social roles (of oneself and others), disclosed information sets tend to be distinct and pairwise incomparable. A small number of main paths like in graph 4.2.b could result from a mainly relationship driven disclosure behavior with minor influence of other factors, e.g. if one acts in a professional or private role.

The existence of major disclosure paths reduces the number of possible disclosures and thus simplifies user defined disclosure rules or increases the chance that disclosures successfully get predicted by machine learning driven assistance mechanisms.

[3] A maximal chain is a totally ordered subset of a poset where no elements can be added without loosing the property of being totally ordered.

Order Mapping of Groups and Disclosures

Similar to D, the set of all person groups[1] G forms a poset. Inspecting and comparing order relations in both D and G reveals if greater person sets either implicate smaller sets of disclosed information items (i.e. for a specific situation s and modality m the disclosure function τ is order-reversing, or antitone, with respect to its person set argument), if the opposite is true (τ is order-preserving, or isotone), if it results in identical sets of disclosed information items (τ is order-ignoring, or constant), or if it does not implicate any subset relations between the corresponding disclosed information item sets (τ is order-loosing). Figure 4.3 illustrates these types of order mappings.

Based on the set of disclosure decision instances T, the number of occurrences of each type of order mapping formally can be described as follows. Let R be the set of disclosure decision instance pairs for which the subset relation applies to their person group[1] arguments:

$$R := \left\{ \left((s, m, g_1, \tau(s, m, g_1)), (s, m, g_2, \tau(s, m, g_2)) \right) \in T \times T \mid g_1 \subset g_2 \right\} \tag{4.2}$$

Then R_p, R_r, R_i and R_l are defined as the sets of disclosure decision pairs with a preserving, reversing, ignoring, respectively loosing order mapping:

$$\begin{aligned} R_p &:= \{(t_1, t_2) \in R \mid \pi_4(t_1) \subset \pi_4(t_2)\} \\ R_r &:= \{(t_1, t_2) \in R \mid \pi_4(t_1) \supset \pi_4(t_2)\} \\ R_i &:= \{(t_1, t_2) \in R \mid \pi_4(t_1) = \pi_4(t_2)\} \\ R_l &:= \{(t_1, t_2) \in R \mid \pi_4(t_1) \parallel \pi_4(t_2)\} \end{aligned}$$

Here $\pi_n(t)$ projects to the n-th element in tuple t and $\parallel$ denotes the relation of unequal sets where neither one is a subset of the other, i.e. $A \parallel B \Leftrightarrow A \not\subseteq B \wedge A \not\supseteq B$.

Most privacy management systems assume an order-reversing pattern, that is a greater information recipient group results in a smaller set of disclosed information. In that case most pairs of disclosure decisions, i.e. elements from R, are contained in R_r. While this is true for many cases there also may exist situations where contradicting relations make sense. For instance when interacting with a greater group of persons one might decide to disclose *more* information in reply to growing expectations within the social context. This illustrates the fact that privacy not only is about hiding information but about providing an appropriate set of information to reach a desired level of social participation (see section 2.1.3). Knowing if disclosures generally either are order-reversing, order-preserving, order-ignoring, order-loosing, or if there is no consistent order mapping, is a valuable input for

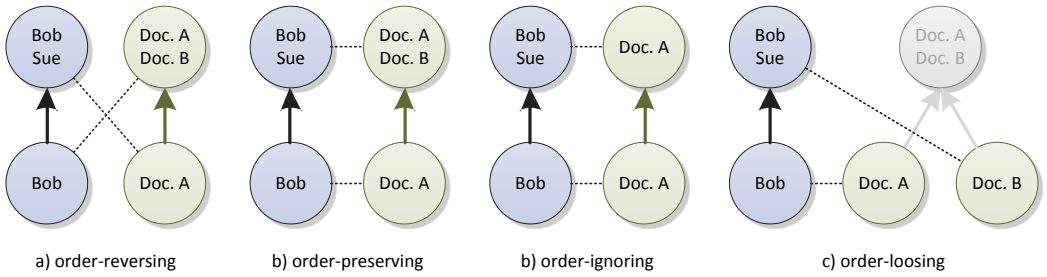

Figure 4.3: Order mapping types of person sets and corresponding sets of disclosed information item. Order-reversing disclosures, for instance, reflect the pattern that more recipients implicate less information. Order-preserving disclosures may result from the intention that a broader audience expects a more comprehensive set of information. Order-loosing disclosures may be given when distinct information recipient groups are supposed to get specific individual sets of information.

an disclosure assistance mechanism. It potentially limits the number of possible disclosure predictions in that only those which do not break order mappings of their set-related neighbors need to be considered.

Other Privacy Related Characteristics

Until now only disclosure decision instances for a fixed situation $s \in S$ and a fixed modality $m \in M$ have been evaluated. Looking at varying situations and modalities one can extract further information. However, as the contents of S and M are specific to certain domains of interpersonal information disclosure and because a semantic ascertainment of S is a complex field of research on its own and heavily depends on individual users, patterns related to varying S and M are not yet considered here (nevertheless section 6.1 provides specific examples of situations and modalities). Of course, in case it is possible to define orders on S and M it is interesting to investigate order mappings as it has been done for the person set argument. Possible orders on modalities may be given by the duration information is visible or by permissions to read, edit or overwrite information items.

Summary

Summarized the structural information gained from a set of disclosure decision instances reveals the following patterns: 1) general complexity of disclosure behavior based on the number of unique disclosures, 2) occurrence of general purpose or main disclosure, 3) existence of major disclosure *paths* indicating confidentiality scales, and 4) general types of order mapping between person groups and disclosed sets of information.

These patterns are utilized by components of the composite disclosure control system presented in chapter 5 and by a novel machine learning method presented in chapter 6. As a short preview, they serve the following purposes: The first two patterns may be used to assess if automatic disclosures in general and manual disclosure rules in particular are suitable to handle a user's disclosure behavior. A low number of unique disclosures and the existence of general purpose disclosures increase the performance and maintainability of automated decision components. This knowledge allows users to be offered sensible control methods, suitable for their privacy preferences. The third pattern, major disclosure paths, limits the possible decisions a machine learning based component has to consider. A learning component which recognizes such paths potentially performs better than one ignoring them. The forth pattern, set order mappings, also helps to limit possible decisions a machine learning component has to regard in that only predictions which do not break order mappings of their set-related neighbors need to be taken into account. Additionally order-mapping–based patterns are used to predict disclosures by interpolations. In case no automatic decision is possible, patterns 2 to 4 provide hints for meaningful disclosure suggestions shown to a user for an explicit manual decision. Finally all patterns come into action for *validating* disclosures predicted by a machine learning mechanism.

4.2 Capturing Disclosure Decisions

As reasoned in the chapter's introduction capturing disclosure decisions *in the wild* often is not possible. An alternative is to capture such decisions using a survey which allows to express disclosure situations without necessarily having a productively used smart environment and which makes it easier to respect the sensitivity of disclosure decisions by providing a reasonable level of anonymity. The following subsection compiles a generic list of requirements for a survey system supposed to capture disclosure decisions in context of social interaction. Afterwards a specific survey system implementation which follows these requirements is presented.

4.2.1 Survey Requirements

Capturing privacy preferences respectively disclosure decisions for analysis purposes is a problematic issue because those preferences must be considered private too. In order to gain knowledge about privacy preferences one has to ensure that users reveal their real preferences and not those they would reveal to others (i.e. the persons conducting the survey) or those they feel to be expected of them. Consequently the survey system needs to decouple structural information from identifying content in privacy preferences and it needs to credibly communicate this fact to participants.

Another problem a survey system has to catch is that of hypothetical privacy preferences. That is if participants are asked for their preferences out of a context they can map to personal experiences. Especially when capturing privacy preferences related to smart environment scenarios this is a difficult challenge as those environments often have a visionary character and might not be portable to situations participants experienced in real life. More precise this means that questions in the survey has to refer to situations, modalities, persons and information items users are familiar with. Consequently one cannot compile a static list of questions to capture privacy preferences. Instead questions have to be adapted dynamically to the individual background of participants.

A third requirement is to be as less suggestive as possible when asking survey participants about their disclosure habits. Looking at existing privacy management systems, often the management system itself forces to express preferences in a specific structure, for instance to classify potential information recipients and disclosure situations into (possibly limited) groups and to describe disclosure rules according to these groups. This might not reflect real preferences and it is important not to motivate any specific pattern.

Finally, as smart environments often are designed for domain specific use cases, the survey system needs to be able to address privacy preferences specific for that use cases. For instance information communicated in environments designed for entertainment purposes differs from those designed to assist collaborative work, e.g. smart meeting rooms. A survey system to understand privacy preferences related to a specific smart environment domain needs to be adjustable for that specific domain.

Summarized a survey system for capturing user privacy preferences needs to 1) anonymize captured privacy preferences, 2) adapt questions to personal background of participants, 3) allow participants to express privacy preferences independent of a formalism which suggests specific patterns of preferences, and 4) be adjustable for different smart environment domains.

4.2.2 DiHabs Survey System

Following the requirements in the previous section and according to the disclosure modeling and evaluation approach described in section 4.1 the online survey system DiHabs[4], which captures and evaluates disclosure habits, has been developed. Online interviews are preferable to traditional pen and paper based questionnaires because questionnaires need to be customized to individual participants. One could still do dynamically adapted verbal

[4]The software and data related to this system is available at the Open Science Repository of the Computer Science Department at Rostock University: `http://opsci.informatik.uni-rostock.de/index.php/DiHabs`. Further information can be found in appendix A.

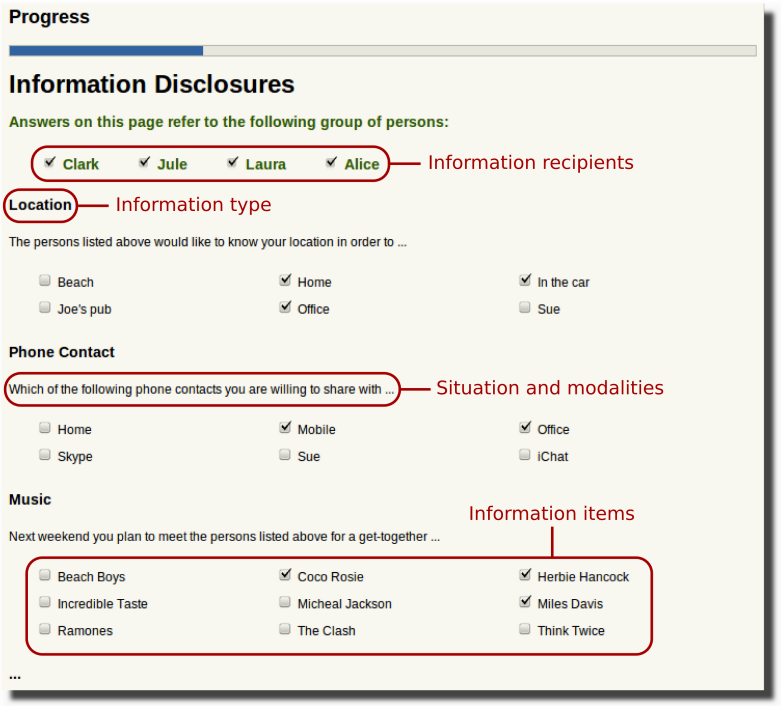

Figure 4.4: Condensed screenshot of the third part of an interview, asking for which information to disclose in context of a specific situation and group of persons. Person names and information items have been given by the participant in the first two parts of the interview.

interviews but this makes it hard to reproduce and analyze survey results. Additionally that would still conflict with the requirement to only gather anonymized information as an analogue anonymization layer between participants and interviewers seems highly impracticable. Finally, online (browser based) interviews[5] allow participants to answer questions at their home or office, i.e. at a familiar location which reduces the feeling of being observed and which increases the willingness to provide *private* privacy preferences.

DiHabs interviews are structured in three parts. As mentioned above privacy is mainly considered from a social point of view. For that reason information about the social environment of participants is needed. The first part of the interview collects this information by asking participants to name some persons from their social environment, including family, friends, colleagues and so on. The second part requests participants to specify a set of personal information items for different information types. For instance the information items *mobile*, *home*, *office* and *skype* may be possible inputs for the information type *phone number*. Participants are not asked to enter real values (e.g. actual phone numbers) but

[5]Traditionally the term *interview* refers to face-to-face consultations. Within this work it used as a shortcut for *online interviews*.

descriptive names (e.g. *mobile*). The purpose of these two first parts of an interview is not to collect the entered information but to use it for questions about information disclosure in the third part of the interview in order to meet the second and third requirement from section 4.2.1. Without the information participants provide in the first two parts, interviewers had either to guess persons and information items relevant for participants (which might be wrong in that participants are not able to map them to personal experiences) or they had to use generic roles for persons and information items (which tends to be suggestive). The third part of an interview captures information disclosure decisions, i.e. this part provides the actual data to be surveyed. Based on the given person names a set of person groups[1] is generated. For each generated group participants are asked to decide the disclosure of their information concerning specific situations and disclosure effects (details on this are given in section 4.2.2). Figure 4.4 is a condensed and annotated screenshot of a disclosure question from the example survey described in section 4.3.

To meet the first requirement from section 4.2.1 (anonymize personal identifying information) DiHabs does not record actual person names and information item names. The survey to conduct with DiHabs strive for structural knowledge only. Hence, it is sufficient to store person and item names internally as sequential numbers, i.e. the first person name is represented as a 1, the second as a 2 and so on. Item names are handled similarly. This anonymous representation does not reduce the structural information one can retrieve from interview responses.

Interview Specifications

In order to be adjustable for specific smart environment domains (see requirement 4 from section 4.2.1) DiHabs makes it easy to compile domain specific interviews. An interview specification consists of some general attributes and multiple sections for specific types of personal information. The general attributes specify how many persons should be given by participants and which groups of these persons should be used for disclosure questions (third part of the interview). Each type of personal information results in an *information type* section as seen in figure 4.4. Table 4.1 and 4.2 describe the most important attributes which make up a specific interview. It is also possible to describe multiple interviews and let DiHabs choose one randomly for each participant. This allows to validate interviews regarding suggestive phrases by, for instance, using slightly different situation descriptions and comparing result differences in participant answers. More details and an example interview specification can be found in appendix A.

Attribute	Description and example value
`npersons`	Minimum and maximum number of persons participants may specify. Example: (10,20)
`groups`	Person groups to ask disclosures for, specified as index sets. Indices refer to names provided by participants in the first part of the interview. May be generated randomly. Example: [(1), (2), (1,2), (1,3,4), ...]
`ngdups`	Number of groups to use twice to check for inconsistencies in answers. Example: 2
`start`, `finish`	Introduction and closing text for the interview. Example: Welcome to ...

Table 4.1: General interview attributes.

Attribute	Description and example value
`name`	Name of the type used in interview questions. Example: Taste in music
`nvalues_min`, `nvalues_max`	Minimum and maximum number of information items participants may specify. Example: 5, 10
`desc_input`	Descriptive text to use when asking participants to enter exemplary items. Example: Please specify some artists or albums of your music library. Ideally your selection reflects a broad spectrum of your taste in music ...
`desc_select`	Text describing a situation (incl. social context and modalities) in which information of this type might get disclosed (specifying multiple situations is possible). Example: Tonight you're meeting for a get-together ... Everyone is asked to contribute some music. What's your choice?

Table 4.2: Attributes for specific information types.

Result Analysis

For each information type and situation used in an interview, DiHabs aggregates responses and extracts structural information and inferable privacy characteristics as described in section 4.1.2. DiHabs uses some of them to draw graphs and generate plots which help to validate and understand the aggregated structural information about privacy preferences in a more illustrative fashion. Section 4.3 describes the result analysis more practically based on the outcome of an example survey.

Implementation Details

DiHabs is implemented as a web application. A thoroughly description of DiHabs's implementation as well as an exemplary complete interview specification can be found in appendix A. However, one implementation detail to mention here is how DiHabs ensures to meet requirement 1 from section 4.2.1 (respect the privacy of privacy preferences).

Person and information item names given by participants in the first two interview parts must be considered as sensitive and identifying data. As described above these names internally are replaced by their index numbers which ensures that finally stored results are free of identifying data. To prevent sensitive data from being saved server-side as part of session data, person and item names are stored as cookies in participant's browsers. Of course these names still reach the server but there they only exist in volatile memory during a request-response cycle. This ensures that interrupted sessions or server-side crashes do not accidentally leak personal data from participants.

4.3 Example Survey

This sections presents the result of an exemplary survey conducted with DiHabs. The purpose of this survey was to validate the disclosure patterns described in section 4.1.2 and to gain first insights about related implications for disclosure assistance mechanisms. Additionally it was supposed to generally evaluate the developed survey system DiHabs.

4.3.1 Survey Setup

This survey has been done with 74 participants, students and coworkers at the author's institute. Though DiHabs is an online survey system, interviewers accompanied the first 16 participants in order to assist in possibly unclear questions and to retrieve direct feedback on how to improve interviews. Participants still did the survey in front of their own computers, hidden from the eyes of the interviewer. The feedback of these 16 accompanied interviews has been used to improve the survey interface with regard to usability and privacy concerns so that subsequent interviews may be conducted unattended. Hence, the remaining 58 participants did the survey completely online. Results from the first 16 and the remaining interviews did not reveal obvious differences, except that online-only interviews had to be filtered to exclude interviews obviously not taken seriously. Such interviews could be detected by the time participants spent on individual disclosure decisions. A regular participant usually allowed at least 25 seconds for an answer, often more. Interviews where this time constraint did not apply have been excluded. Finally there were 59 suitable interviews left.

Participants have been asked to decide the disclosure of 5 different information types: *location*, *phone* numbers, *e-mail* addresses, taste in *music* and taste in *movies*. For the first three types they have been asked for their disclosure in general (i.e. the disclosure question did not involve S and M, see section 4.1.1). Disclosures for the last two types were asked in context of a specific situation and modality. For instance participants were asked to decide the disclosure of a subset of their music library in case they are respon-

sible for the background music of a get-together. Similarly participants had to select possible movies to watch with a certain group of persons. Though this thesis primarily deals with collaboration oriented smart environments, the conducted survey uses rather technology-independent situations and modalities because most, if not all, of the participants already experienced similar situation. Nevertheless the selected disclosure situations already revealed different privacy patterns which may exist similarly in collaborative smart environment scenarios.

The interview specification used for the survey as well as a corresponding screenshot-based walk through the resulting online interview can be found in the appendix section A.4.

4.3.2 Results

This subsection presents and discusses the survey results, with regard to the structural information and patterns elaborated in section 4.1.2.

Figure 4.5 illustrates how many participants provided specific numbers of unique disclosures. The results show that the information types *phone* and *location*, which have been asked for without a reference to a specific situation and modality, in most cases result in 5 or less distinct disclosures. In contrast requests for music and movie taste related information resulted in 4 to 8 unique disclosures in the majority of cases. This shows that privacy management in context of a specific situation (when the disclosed information is a significant element of a social interaction) is more complex. On the other side privacy preferences for location and contact information may be expressed using a small and maintainable set of identities (e.g. public, business and private contact information)[6].

Figure 4.6 displays the widths of the poset given by the set of all disclosures respectively how many interviews resulted in which width. These results comply with the findings based on the number of unique disclosures. It shows that most *phone* and *location* information disclosures move along one or two paths. In contrast disclosure sets for the *music* and *movie* taste related questions often have a greater width, i.e. they have a higher number of order-unrelated disclosures and indicate a more sophisticated underlying privacy concept.

Figures 4.7 and 4.8 illustrate the occurrences of the different order mapping types. Specifically, figure 4.7 shows for how many participants a certain order mapping type occurred in the majority of cases. It shows that there is no general order mapping type which is valid

[6] Actually the differences may originate in both information type and situation, i.e. the results do not indicate a correlation between patterns and either information type or situatiuon alone. However, this bias of the experimental setting is less relevant here since not a specific correlation but the practical relevance of patterns in general was the main objective of the survey.

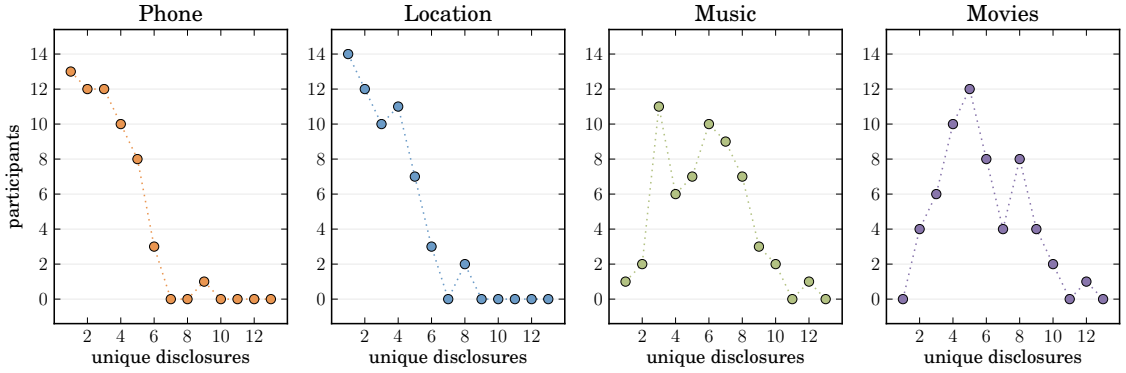

Figure 4.5: Possible numbers of unique disclosures per information type (respectively situation) and the number of participants for which disclosures resulted in each of these numbers.

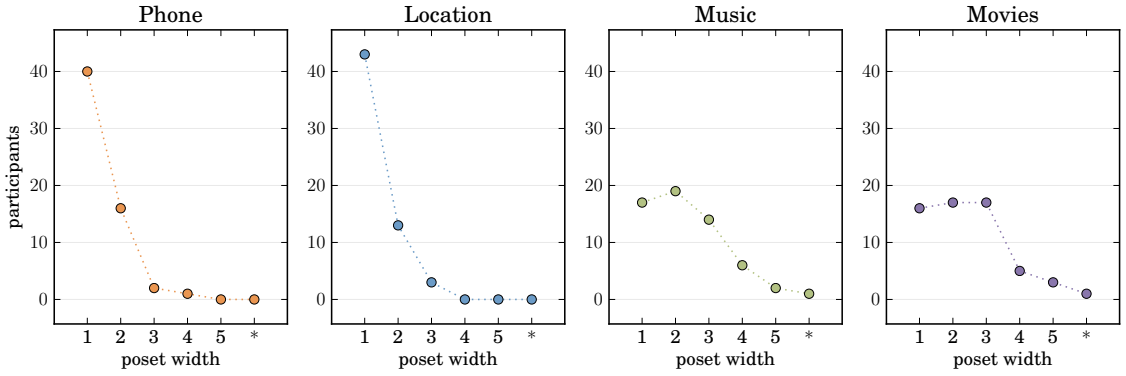

Figure 4.6: Possible poset widths per information type (respectively situation) and the number of participants for which disclosures resulted in each of these widths. The asterisk catches all widths greater than 5.

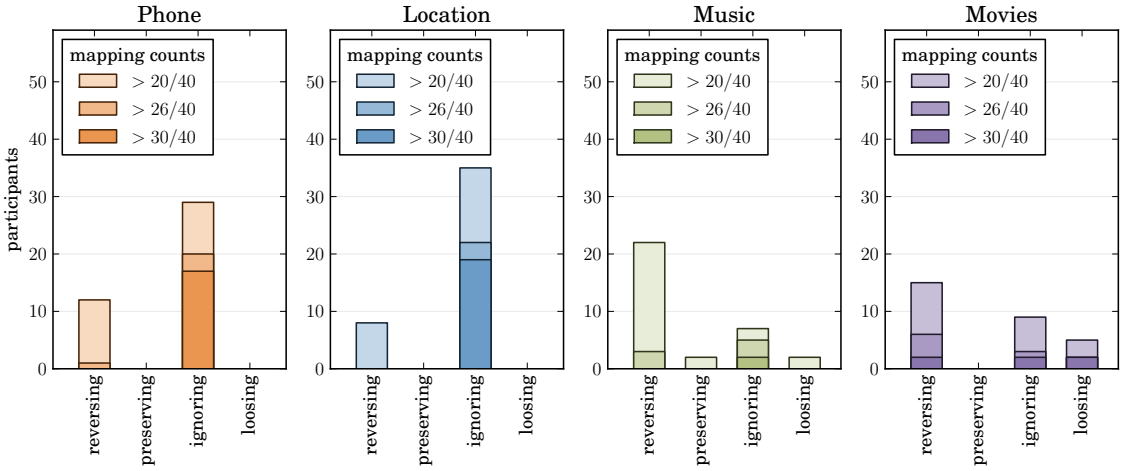

Figure 4.7: Possible order mapping types per information type (respectively situation) and the number of participants for which at least a certain number of disclosure decision pairs (referring to the set R from section 4.1.2) have the corresponding type. In other words, these plots illustrate for how many participants a certain order mapping type occurred in the majority (more than 50%, 66%, or 75%) of cases.

for all 59 participants. Though there are users who generally follow an order-reversing or order-ignoring disclosure behavior, a few disclose information in an order-preserving or order-loosing manner. In any case there is a significant portion of participants who follow *no* general order mapping type (depending on information type and definition of *majority*, this significant portion ranges from roughly 30% to 90%). These results do not support the intuitive expectation that greater information recipient groups implicate smaller sets of disclosed information. Instead they highlight that disclosure assistance mechanisms not only need to be designed depending on information types and disclosure situations but also with regard to individual behavior of users. Figure 4.8 provides another view about the occurrences of order mapping types. It shows the distribution of mapping type occurrence numbers among all 59 participants. For most participants, order-reversing and order-ignoring mappings occur most often but for information type *music* and *movies* this difference diminishes. This corresponds with the results displayed in figures 4.5 and 4.6.

4.4 Conclusion

This chapter elaborated patterns of privacy management in social interactions. The patterns are based on structural information gained from the mapping of information recipient groups to disclosed sets of information items. The practical existence of the theoretical patterns have been backed up by an online survey which captured disclosure decisions

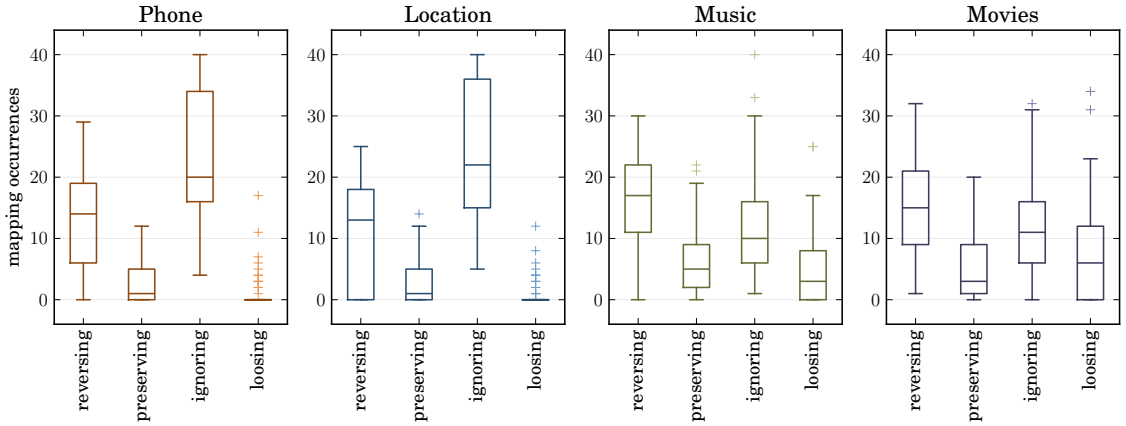

Figure 4.8: Possible order mapping types per information type (respectively situation) and the number of their occurrences per participant, displayed as box plots with whiskers representing the highest and lowest values within the 1.5 interquartile range (IQR) and outliers (+).

for different situations and information types. The survey results showed that no single pattern applies to all (or most) users. Instead the usage of patterns significantly varies among information types (and disclosure situation) as well as individual users.

The contribution of this chapter is an exploratory one in that it provided novel views on how users manage their personal information. It solved a part of issue (I) by analyzing structural relations between information receivers and information items. However, this chapter only briefly touched how the patterns help in developing privacy management mechanisms for smart environments. These aspects are part of the following two chapters.

5 Composite Disclosure Control

Users practice privacy individually. Personal needs and perceptions result in different patterns of information disclosure, as shown by previous work (see section 3.2) and chapter 4. Even a specific user might have different strategies how to manage her information, depending on the information type and situation of disclosure. As shown in the previous chapter, disclosure patterns may be rather simple, easily to express by a small set of rules, or they may be more complex, so that users cannot express them formally. Still, it may be possible to assist users in abstracting their privacy preferences, but it is also possible that disclosure decisions do not follow a concept which can be expressed within a formal system: In that case ad hoc user decisions are the only way to manage disclosures accurately. Hence – picking up issue (II) from section 1.2 – users should have the option to utilize multiple mechanisms to express and enforce information disclosure. Ideally, a disclosure management solution adapts to individual users in that it automatically suggests proper mechanisms to control their information. However, a composition of multiple disclosure control techniques does not only allow to choose an optimal one for certain disclosure patterns. It can also be used to let different mechanisms interfere with each other. For instance ad hoc decisions can be used to learn disclosure decisions or to assist users in setting up manual disclosure rules. Similarly, a community based disclosure recommendation system can be used to back up disclosure predictions of a learning based mechanism or to provide sensible defaults for ad hoc decisions. Last but not least, the decision which information to disclose can be made by staging different components. Consequently, combined multiple disclosure mechanisms have two main benefits, they provide user-adapted control techniques and complement each other.

The main challenges in realizing such an approach are how to integrate the different mechanisms in terms of mutual interaction and supplementation as well as how to choose appropriate default mechanisms. This chapter deals with these challenges. First, section 5.1 presents and discusses different mechanisms that can be used to manage the disclosure of personal information. Afterwards, section 5.2 describes how these mechanisms can be combined and integrated with each other to a composite disclosure control system.

5.1 Components

Managing the communication and disclosure of personal information (or information in general) can be done in a variety of ways, each targeting specific information types, individual user needs, and modalities to enforce information exchange policies. Some mechanisms are not directly related to privacy management but still make sense to be used to decide the disclosure of personal information.

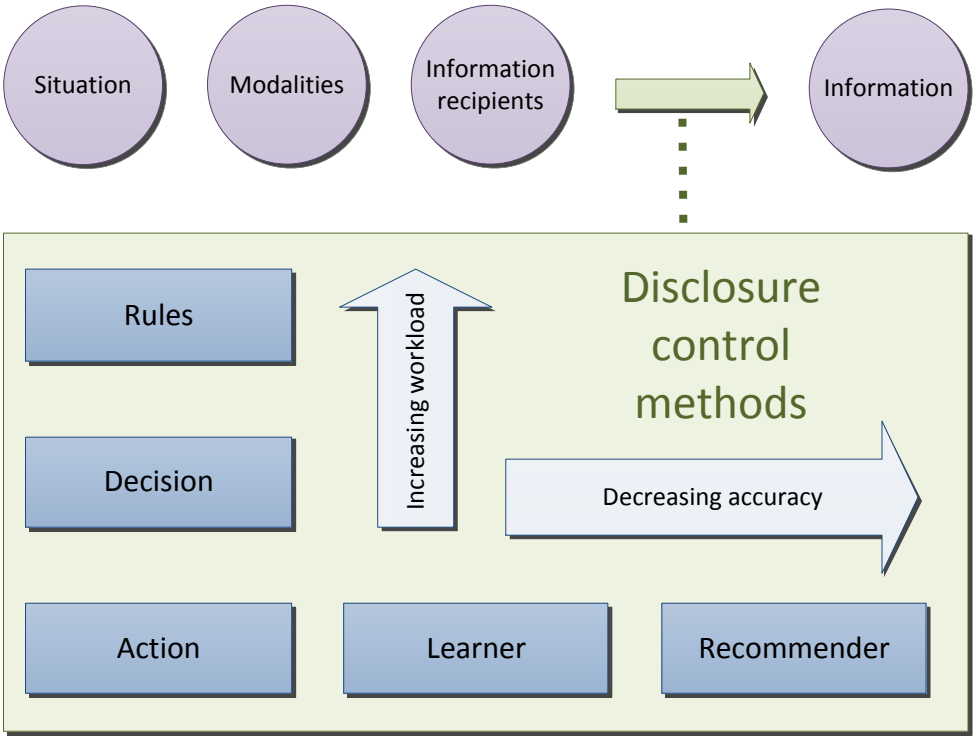

Figure 5.1: Overview of a composite disclosure control. Several components contribute to the decision which information to disclose in a situation with specific disclosure modalities and information recipients. The components are arranged according to their required management workload and potential accuracy in deciding disclosures in compliance with a user's actually intended disclosure decision.

Figure 5.1 provides an overview about different approaches to control the communication of information. These approaches can be arranged according to the workload demanded from users and their degree of accurately expressing actual disclosure intentions. For instance formally expressed rules, ad hoc user decisions as well as arbitrary actions linked to information disclosure have the potential to completely match actual user intentions. In contrast, a learning-based automatic disclosure system as well as more impersonal

recommender systems are not able to always correctly model the information exchange preferences of a specific individual.

At first sight it seems that regular actions with implicit information disclosures are the method of choice. However, as always each method has certain shortcomings and specific advantages. The following sections elaborate these special characteristics.

5.1.1 User Decisions

The technically most simple method to control information disclosures are explicit, ad hoc user decisions. Here users manually decide which information to communicate in the moment when it is requested. In that the significant advantage of this approach is that decisions are always correct in terms of current user intentions (assuming a user knows what she does). A disadvantage is that they demand a user's attention, potentially distracting her from actual tasks she wants to perform within a smart environment. User decisions are the method of choice when other, less obtrusive methods fail or are not available.

Example

Actually current smart environments mostly utilize explicit ad hoc information disclosures where users are faced with an information request and manually choose the information to communicate to a service. Consider a meeting where a participant is requested to show some presentation slides as an example. While the request and distribution may be automated, there is usually no reasoning mechanism which information from a user is communicated to the presentation service. Users actively select and push the documents to display.

5.1.2 Implicit Actions

A disclosure control method which aligns well with the user interface goals of smart environment are implicit actions. Here privacy control is integrated into already existing processes. In fact such a disclosure control implements physical metaphors as proposed by the idea of Internet of Things (IoT). They also are good candidates to follow genres of disclosures (see section 3.1.3), as long as physical actions map to correspondingly expected disclosures.

Example

Straight forward examples of such actions are closing respectively opening a door or a notebook. In the first case a general room-wide policy may be derived which prohibits the communication of information exchanged *within* a room to parties *outside* of the room. The notebook example is a more personal action and could be used to generally enable or disable the disclosure of personal information within a smart environment. Similar actions are putting a smartphone on a desk with the display turned downwards – the smartphone then could refuse any information requests by the environment or other persons.

As these examples illustrate, action-based implicit disclosure is an intuitive way to manage privacy. On the other side it only provides coarse-grained control.

5.1.3 Disclosure Rules

After ad hoc disclosure decisions, precompiled disclosure rules expressing which information to communicate to whom under which conditions are the most precise method. Given that all potential conditions can be sensed and evaluated programmatically, they theoretically are able to completely express the disclosure behavior of an individual. However, these requirements often are not met in real life. As stated before, privacy is influenced by a variety of factors, including gut feelings and complex preferences. In that not all factors can be evaluated using rules and rules easily get too complex to be maintainable or even expressible by regular users.

Still, within the realms of its capabilities, manually compiled rules are a useful method to control information disclosure. They are suitable for simple disclosure patterns where users have clear concept of how to communicate information to whom. Complex disclosure circumstances usually are not well expressible using rules since they often are not known in advance and are likely to be insufficient respectively too static. In that rules are a perfect match for managing rather sensitive information where a disclosure decision primarily is influenced by the information recipient. In fact the time-consuming compilation and maintenance of rules are a significant drawback of this approach.

Example

Outside smart environments, an example for a rule system is the Platform for Privacy Preferences (P3P) and its companion P3P Preference Exchange Language (APPEL). It is mainly used to manage which identifying and communicational information users are willing to hand over to websites. Indeed these systems have been ported to smart envi-

ronment use cases, as presented in section 3.3.2. A more specific example not linked to a specific system, is a rule which prevents the disclosure of certain documents within a smart environment whenever a yet unknown person enters the environment. More complex rules can be arranged by mapping information items to allowed information recipients (which mimics an access control list).

5.1.4 Learned Disclosure Concepts

Rules theoretically are able to express complex privacy preferences but practically are hard to maintain. This issue can be balanced by using machine learning techniques which model repetitive disclosure decisions. Here users do not need to formalize their privacy preferences manually in advance. Instead preferences are formalized gradually, based on ad hoc disclosure decisions. This process not only releases users from manually expressing rules, the input for the modeling mechanism is also given ad hoc on information request, i.e. at a time when it is easier to estimate potential privacy implications. Additionally this prevents users from setting up privacy preferences for theoretical disclosure situations which never occur in practice. Learned disclosure concepts are able to balance the drawbacks of ad hoc decisions and precompiled rules – more on this follows in the upcoming section 5.2.

Example

Supposed Bob regularly meets with other persons to work on different projects. Each project is associated with certain persons participating in meetings and documents shared within the meeting. This association is a perfect candidate which could be modeled by a learning mechanism. Compared to rules a learning based approach even gets more beneficial when more parameters are included in disclosure decisions, e.g. in which room or at which time a meeting takes place.

5.1.5 Recommendations

Recommendation systems generally are not directly linked to the management of private information. Instead they often are used to collaboratively assess the value or trustworthiness of entities (e.g. persons, services, or organizations). Consider product reviews or seller ratings on shopping related sites as examples. Still, the general concept may also be used to let a community vote for information sharing practices when interacting with smart environment services. Obviously a group of individuals cannot agree on which information items a particular individual should share via a certain service. However, a more

coarse grained recommendation can be built for which information types and which sharing modalities should be used during service interaction. In other words, not specific pieces of information are *recommended* to be disclosed but disclosure parameters portable across different users. DiGioia & Dourish (2005) propose such concepts as *social navigation*.

Example

In context of interpersonal privacy management in smart environment scenarios, a recommender systems could assist users in deciding which information to disclose by providing hints like: "Other users generally exchange contact information by transmitting them to mobile devices of the information recipients. They rarely share them on the display wall". Next to type and modality, context information like the location or number of information recipients may also be used to parametrize recommendations.

5.1.6 Summary

The disclosure control methods presented here can be classified according to several characteristics:

Pessimistic or optimistic: Pessimistic control express privacy preferences in advance and target rather sensitive information. Optimistic methods control information disclosure on demand or posthumously when automatic disclosures failed.

Automatic or manual: Disclosures may be decided automatically by a reasoning component or manually by the information holder.

Individual or generic: Some methods allow to handle disclosures specifically tailored to an individual while others only support generic disclosure behavior.

Implicit or explicit: The obtrusiveness of a control method depends on whether it discloses information implicitly as part of already existing activities or explicitly as separate control actions.

Coarse or fine grained: Coarse grained control only allows to generally disable information exchange (simply speaking) while fine grained control also allows to depict specific information items to disclose.

These characteristics not always are mutually exclusive or orthogonal. For instance adjusting information disclosure by opening or closing a door can be both an implicit and explicit action. Also, every control method principally is able to support coarse grained control. However, each method specifically targets certain characteristics, as illustrated in table 5.1. This table once more highlights that a comprehensive privacy control system

	pess.	opt.	autom.	man.	indiv.	gener.	impl.	expl.	coarse	fine
User decisions		✓		✓	✓			✓		✓
Implicit actions		✓		✓			✓		✓	
Disclosure rules	✓		✓		✓			✓		✓
Learned concepts		✓	✓		✓					✓
Recommendations			✓			✓				

Table 5.1: Disclosure control methods and their main characteristics.

requires a composition of multiple methods in order to meet individual privacy needs. Users should be able to practice privacy according to the characteristics they prefer.

The next section elaborates how these disclosure control methods can be combined to a composite disclosure control system. The approach to learn disclosure concepts is explicitly dealt with in chapter 6. Compared to the other methods, it is the least covered one in current research about smart environments.

5.2 Integration

Combining multiple disclosure control components not only has the advantage that they support multiple approaches to information disclosure. It also allows different components to reference each other for improved accuracy. For instance a learning-based component can validate predictions using a recommendation system. Another example is the creation or extension of precompiled disclosure rules using a learned disclosure model.

The next section 5.2.1 describes the temporal scheme in which disclosure control components interact. Section 5.2.2 presents a concept how to stage components to a powerful decision process and how individual components complement each other.

5.2.1 Temporal Scheme

Disclosure control components come into action at different points in time. Generally one has to distinguish preliminary, situational, and retrospective controls, as illustrated in figure 5.2.

Preliminary control covers the compilation of disclosure rules. It fits well for regulating the disclosure of sensitive information. In that it is a pessimistic control method for users which have clear conception of privacy preferences for certain information items.

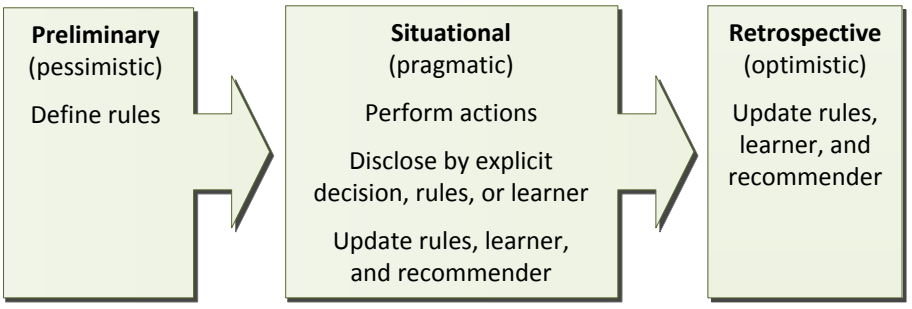

Figure 5.2: Temporal scheme of a composite disclosure control management. A pessimistic management defines disclosure rules in advance. In contrast, an optimistic management audits disclosures retrospectively and adjusts automatic disclosure components accordingly. A pragmatic situational management involves actions implicitly disclosing information, explicit user decisions and corresponding updates of the learner and recommender component as well as an optional immediate update of disclosure rules.

Situational control involves implicit actions, ad hoc (explicit) user decisions, automatic disclosures by a reasoning component and immediate manual adjustments of a reasoning component (in case users want to correct automatic decisions). It is a pragmatic approach where users manage their privacy on demand, i.e. in the moment when it actually is necessary and when it is easy to guess potential privacy implications. The obtrusiveness of situational control depends on the capabilities of the used reasoning components (ideally users rarely have to manually decide disclosures, only when new situations occur which are not yet known to the used reasoning components).

Retrospective control follows an optimistic approach to privacy management. It resembles an audit where users adjust automatic disclosure components more comprehensively than during the situational control. For instance users could mark bad predictions by a learner, edit rules which caused wrong disclosures, or selectively revoke previous decisions (e.g. invalidate all decisions in context of a specific information recipient) when a user's privacy preferences change.

5.2.2 Decision Process

Combining disclosure control components requires some kind of orchestration to an overall decision process. The flowchart in figure 5.3 presents such an orchestration concept. It displays the staging and interaction of different components. The arrangement is based on temporal, functional, and data representation characteristics of individual components

and resulting integration possibilities. The operation of individual components should be quite obvious. Here only the interesting integration parts are explained more detailed.

Suggestions

Disclosure decisions by a recommendation and learning component may not only be used for automatic disclosures but also for suggesting disclosure. While such suggestion still require user interventions, they reduce the workload a user has to face when managing information exchange. In fact recommendations should only be used for suggestions since they do not support individual privacy preferences and only provide hints on information types (or classes) and disclosure modalities. Disclosures predicted by a learning component may be used for suggestions when their prediction does not provide a certain confidence.

User Veto

Any automatic disclosure decision is not applied immediately but delayed by a configurable time in order to allow users to veto a disclosure. This is an important step in the decision process as no automatic mechanism will always perform correct disclosures. It also provides some level of transparency to users in that nothing happens completely automatically without the possibility to express or revoke consent. The time a decision is delayed and the degree to which the veto possibility is alerted to a user may be dynamical, depending on the confidence a reasoning component has in its disclosure prediction.

Rule Templates

Rule templates allow to utilize rules in a more pragmatic (ad hoc) way (Bünnig, 2009b). Whenever users manually decide a disclosure, users have the option to generate a disclosure rule for future situations. This is especially useful for exceptional (i.e. less frequent and thus not learnable) but important decisions which are worth to be manifested in rules. Rule templates allow to make use of the beneficial characteristics of disclosure rules while avoiding some of its downsides, namely the abstraction of future situations and the formal compilation of disclosure parameters to a rule. By providing templates (e.g. by the learning component) users are released from most of the awkward part of rule compilation. Still, rules practically only work if their overall size respectively count does not exceed a certain limit. Additionally, rule templates bear the risk to unexpectedly influence automatic disclosure decisions for other situations too. Rule templates might tempt users to set up complex privacy preferences but still rules should only be used to formalize rather simple, straight forward preferences.

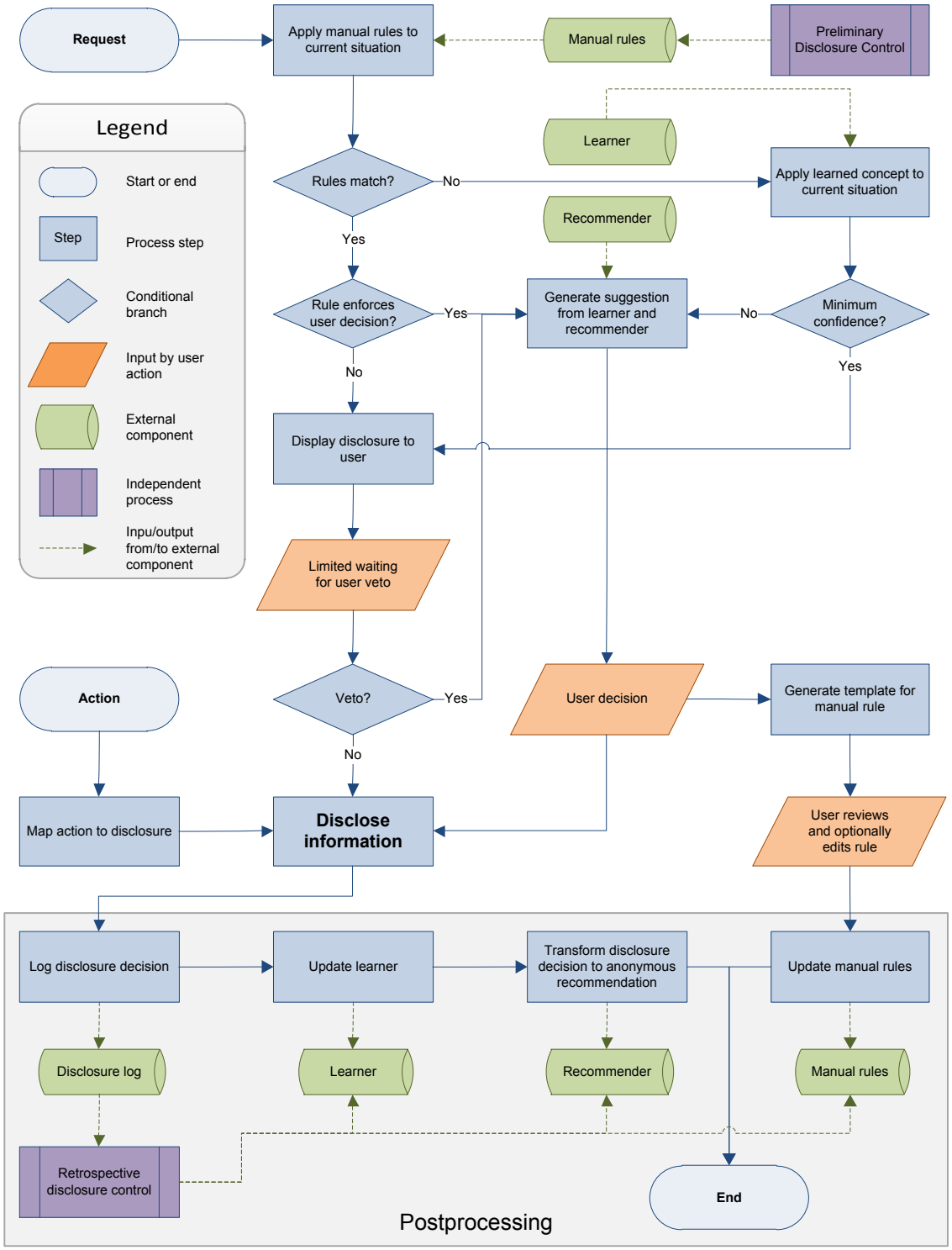

Figure 5.3: Potential steps involved in an information disclosure driven by a composite disclosure control system.

Postprocessing

An important part of the decision process are the postprocessing steps. They include automatic as well as optional manual actions. First, any decision is stored in a disclosure log. This allows retrospective privacy control at an any later point in time. Second, the learner component is updated with the new situation (i.e. the training set is extended). Third, the current disclosure is anonymized and transformed to a general recommendation to feed back to the recommendation component.

The most simple form of retrospective control is to inspect the disclosure log and remove or alter previous disclosure situations. This implicitly updates the recommendation and learning component. A more subtle retrospective control also allows to directly interact with the recommendation and learning component.

The specific interaction possibilities with the learning component depends on the used learning method. For so called black box learner, which encode their disclosure decision model in a way which is not mappable to human decision processes, the only form of interaction is to inspect and weight previous disclosure situations which make up the learners training set. For instance users might remove exceptional cases or increase the weight of exemplary cases (where they see an elementary representation of their privacy preferences). White box learners, whose model encodes decisions in a human readable format, provide further interaction possibilities. Chapter 6, which deals with particular learning techniques, investigates learner-specific interaction possibilities in section 6.7.

5.3 Deployment

Obviously privacy management should be taken into account early when engineering smart environments. However, this should not lead to the misconception of tightly incorporating information management components into the environment infrastructure. The environment should be seen as a tool to accomplish certain tasks, not as a platform to host and manage personal information. Users have to manage the communication of their data within different systems. In that the management should stand on its own and not depend on one particular environment it may be be used in. Figure 5.4 illustrates this directive. Here the information management is not part of the environment but composed of independent mechanisms which may also be used when interacting with(in) other systems. Similarly, the repositories which hold personal information are not part of the environment. In fact there may be various repositories. This aligns with the recommendation of Kobsa (2007) to keep information distributed. For the most part the information management is not directly linked to a particular environment. Points of contacts are information request

events, implicit actions, context information describing a situation, and recommendations evaluated by the recommender component (recommendations should be part of the environment as they apply to information sharing events within a specific environment and also should be sharable among multiple persons acting within the environment).

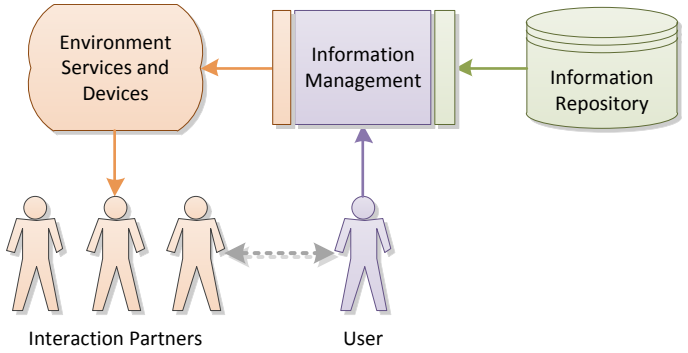

Figure 5.4: Decoupling of the environment infrastructure and user-associated information management mechanisms. The weak environment and strong user association makes these mechanisms more portable across different systems where users have to manage their information. It also increases the trust users have into the management mechanisms.

5.4 Conclusion

This sections presented an abstract concept for a composite disclosure control system. Several components contributing to this system have been presented. Further, a decision process has been developed which incorporates these components to a reference model of a comprehensive disclosure control. The process covers preliminary, situational, and retrospective privacy control. It stages different disclosure control components and integrates them with each other. Finally, a generic deployment strategy for a composite disclosure control system has been presented. In that this chapter provides a conceptual blueprint to solve issue (II) from section 1.2. The next chapter deals with a specific component of the proposed system: the automation and suggestion of disclosure decisions using machine learning methods.

6 Learning Disclosure Decisions

One component of the composite disclosure control system presented in chapter 5 assists in making disclosure decisions based on a learned disclosure concept (see section 5.1.4). The basic idea is to conceptualize the privacy preferences of a user gradually by observing and learning her disclosure decisions. Though it is unlikely that every aspect of a user's concept of privacy management can be grasped by a learning component, it is well suited to conceptualize disclosure decisions following a stable pattern – without the need for users to abstract and express preferences manually in advance. In that a learning component has the potential to remedy issue (III) mentioned in section 1.2.

This chapter investigates the problem of learning disclosure decisions. First, it specifies the actual learning problem to handle. Subsequently, section 6.2 analyzes which existing methods are suitable for such a learning problem. Section 6.3 presents a new interpolation-based learning method which uses disclosure patterns as discussed in section 3.2 and elaborated in section 4.1.2. How these disclosure patterns may be used to validate predictions made by a learning method (in order to prevent as much wrong predictions as possible while supporting as much correct predictions as possible) is described in section 6.4. Further improvements of learning methods may be possible by utilizing scenario-specific semantic knowledge. This idea is considered in section 6.5. The presented general learning methods, the new interpolation-based learning method as well as the validation methods are evaluated in section 6.6. It presents a scenario-independent learning method evaluation system, corresponding results for example scenarios, and a mapping of generic scenario characteristics to suitable learning methods. Finally, section 6.7 analyzes specific possibilities to integrate different learning methods within a composite disclosure control system (in addition to generic integration approaches presented in section 5.2).

6.1 Learning Problem

In section 4.1 a disclosure decision has been modeled as a mapping of a situation S, modalities M and information recipients $\mathcal{P}(P)$ to a set of correspondingly disclosed information $\mathcal{P}(I)$:

$$\tau : S \times M \times \mathcal{P}(P) \rightarrow \mathcal{P}(I) \tag{4.1}$$

In order to learn a model which conceptualizes a set of instances of this mapping, it has to be transformed to a learning problem. This requires to transform the instances of the mapping's domain to a feature vector and values from the co-domain to labels. First, this section investigates the feature space (i.e. the mapping's domain). Afterwards, it analyzes characteristics of the label space (i.e. the mapping's co-domain). Finally, it concludes these finding to a specification of the learning problem.

6.1.1 Features

Situations

In practice, a situation is an individual factor which might be perceived differently across users, in extreme cases even based on a gut feeling (see section 3.2.2). However, for the sake of learning such a mapping, situation descriptions have to be limited to measurable context information. Location and time are the most prominent ones, though often there is more measurable context information available, for instance situation type information like *"meeting"* or topic information like *"project xy"* – those may be available as annotations or attributes of entries in a user's calendar or in the booking schedule of a smart environment. To some extend they also can be detected automatically: Christoph Burghardt, another researcher within the MuSAMA project, explicitly deals with the aggregation of low-level sensor information to high level situation context (Burghardt & Kirste, 2008; Burghardt *et al.*, 2011). There also might be measurable *individual* context information, for instance personal calender item annotations like *"confidential"* or *"negotiations"*. The semantics and availability of context information describing a situation significantly depends on concrete scenarios and the participating users. Hence, a situation has to be represented by a varying number of multiple features, each referring to a specific context information. Feature values may be numeric (time) or nominal (location), as well as sets of them (personal annotations attached to a calendar entry).

Modalities

Modalities describe how information is going to be disclosed. This includes the event firing a disclosure as well as the paths disclosed information is communicated. Possible firing events could be *start of meeting* or *enter environment*. Practically such an event can be seen as the core request to disclose information. Examples for communication paths

are *show on display wall* or *distribute to nearby mobile devices*. Different communication paths implicate how information is perceived by the recipients. For instance large shared screens display information prominently, and detailed, but temporarily and identically for all recipients. In contrast, if information is distributed to the devices of the other persons in the environment, it is shown in different ways and the time and duration recipients consume this information is out of control of the information provider. Thus, a modality requires multiple features too – one for the firing event (with nominal values as event identifiers) and one or more describing the medium used to communicate information. Possible medium related features are device identifiers, medium type or medium characteristics. Again, the set of features describing a modality should be considered to be scenario-dependent.

In practice there might be different possible communication paths where a user has to choose one. In that case, the chosen modality is also part of a user's disclosure decision – section 6.1.2 gets back to this issue.

Persons

The recipients of information items usually are the persons one interacts with in an environment, possibly limited by modalities. Obviously they are the most important factor influencing which information should be disclosed (see section 3.2.2). Persons can be represented by whatever identity a presence sensing system provides – as long as they are unique and identical across different situations[1]. Bijective identifications are preferable compared to abstract role schemes for the reasons elaborated in sections 3.3.2 and 3.3.3 (either role assignments are too generic or hard to manage). Hence, this mapping parameter can be represented by a single feature which lists all persons acting as information recipients (i.e. the feature value is a set of nominal values).

[1] Ensuring that a specific person always is detected as the same one is a challenge on its own – consider different mobile devices or sensing technologies as examples. This work does not cover this technical issue.

Example Values

```
room: C102
weekday: 2
type: meeting
topic: project-x
tags: {confidential, important}

trigger: start-of-meeting
medium: display-wall

persons: {alice, bob, clark, dent}
```

6.1.2 Labels

The labels to predict by a learned model which conceptualizes a set of instances of the mapping 4.1 are sets of information items to disclose. Effectively this makes the learning problem a multi-label problem (Tsoumakas & Katakis, 2007) where each information item is an individual label. If individual items are part of a hierarchically structured class taxonomy, the learning problem additionally is a hierarchical one (Wu *et al.*, 2005). For instance in one situation a person could disclose *all* documents related to a project "X" while in another situation she discloses only final versions of documents from that project (i.e. no draft material). The information disclosed in the latter case is a subset (or specialization) of the one disclosed in the first one. As indicated above the communication path, or medium, may also be part of a disclosures decision. While the chosen medium actually is not an information item, it can be encoded as such when integrating it in the hierarchical structure of disclosures, e.g. as a root or leaf element. Further hierarchical structures could be given by permissions. For instance when displaying information on a shared screen, permissions could regulate if the environment is allowed to store the displayed information for later reference. Different scenarios would suggest different types and degrees of hierarchical structures. The benefit of utilizing hierarchical structures is that a learned model may fall back to generalized disclosure predictions if exact predictions are likely to be wrong.

Following is an example disclosure where the items' path in the underlying hierarchical class taxonomy is encoded using dot-separators:

```
documents.project-x.final.displaywall
documents.project-y.displaywall
```

This example describes the disclosure of final documents from project X and any documents related to project Y, using the "displaywall" as the disclosure medium. The medium also could haven been integrated as a root element (which variant performs best is part of the evaluations in section 6.6). The semantics of individual path elements of an item depend on a specific scenario and how a user manages his information. This is not a problem as standard learning methods utilize structural information only (section 6.5 briefly discusses enhanced learning methods which utilize semantic knowledge about disclosure values).

6.1.3 Conclusion

Predicting disclosures according to mapping 4.1 is a supervised hierarchical multi-label classification problem with features whose values may be numeric, nominal, or sets of numeric respectively nominal values. The number of features depends on how much context information is available to describe a situation and on how detailed modalities are described (the scenarios considered within this work have eight or less features). Labels are sets of nominal items, each an element of a hierarchical class taxonomy. The number of different labels and their distribution depends on individual users and the scenario they are part of.

The training data to learn from grows incrementally, which results in an online learning problem. However, the distinction between online and offline learning is not that relevant here as the size of the complete training data as well as the frequency of new cases to learn is small enough to allow – concerning computational costs – offline learning methods imitating online learning.

Most learning methods assume features and labels to be of a specific type. It is straightforward to convert features from a nominal to a numeric space (and vice versa). The same applies to set-based values (which can be flattened to multiple scalar values). In contrast, handling multiple labels and their hierarchical structures is more sophisticated and usually requires a special combination of non-hierarchical single-label learning methods. The next section deals with these issues.

6.2 General Learning Methods

Hierarchical multi-label learning problems usually are handled by special wrappers on top of basic learners, i.e. learners which expect non-hierarchical single-label problems. This section presents common basic learners and wrapping methods suitable for the learning problem at hand. The presented approaches are referenced later in section 6.6 to evaluate

their ability in predicting disclosures, and in section 6.7 to evaluate their characteristics concerning an integration into a composite disclosure control system as presented in chapter 5.

6.2.1 Multi-Label Learning

A common approach to handle multi-label learning problems is to transform the learning problem so that it can be applied to single-label learning methods. The next section gives an overview about different transformation methods, which is followed by a description of the corresponding implications on performance metrics and their meaning in context of the prediction of information disclosures.

Problem Transformation Methods

The two most simple problem transformation methods are to reduce each multi-label to a randomly or otherwise chosen single item or by discarding any sample whose label contains multiple items. Obviously these methods potentially discard a lot of useful information which cannot be used to conceptualize the underlying mapping from features to labels. Another method which discards less information is to replicate samples in that a sample with a multi-label containing n single-label items is transformed to n samples with identical features but each mapping to one of the single-items only. Optionally these samples may be weighted depending on the number of replications (e.g. $w = 1/n$). While this method does not discard single-label items or samples, it looses correlation information of individual labels. More promising methods are the so called *powerset* and *binary relevance* transformation methods (Tsoumakas *et al.*, 2010).

Powerset method. Practically this method considers each distinct multi-label, i.e. each distinct set of single-label items, as an individual single-label. It is called powerset method because the theoretical domain of the mapping from features to labels is the powerset of all single-label items. However, while this method makes the learning problem applicable to learning methods expecting single-item labels without discarding available information, it does not exploit any multi-label–related information.

Binary relevance method. This method indeed utilizes multi-label–related information by aggregating multiple binary learners – one for each single-label item contained in multi-labels. When predicting the label, i.e. disclosure, for a new situation, each binary learner is run independently and the prediction is a multi-label containing all single-label items

whose binary learner has a positive prediction. In contrast to the powerset method, this method is able to predict multi-labels which haven't occurred yet in the training set.

Performance Metrics

Learning methods applied to multi-label problems require different metrics than those used for single-label problems. The most common single-label metric is the ratio of the number of exactly matching predictions to the number of all predictions (subsequently this is called the *match* metric). However, concerning multi-label problems, this is a rather strict metric which does not measure to which extent a wrong prediction differs from the true label. Instead, metrics for multi-label learning problems have to consider set differences. Tsoumakas *et al.* (2010) state the following most important multi-label metrics: *Hamming Loss*, *Accuracy*, *Precision*, and *Recall*. They are defined as follows.

Let L be a non-empty set of single-label items. Further, let X be a set of evaluation samples (f_i, L_i) with $1 \leq i \leq |X|$ and $L_i \subseteq L$, applied to a multi-label learner C which maps features f_i to predictions Y_i, i.e. $Y_i = C(f_i)$. Then the **hamming loss** is defined as:

$$HammingLoss(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \frac{|L_i \triangle Y_i|}{|L|}$$

This metric considers false predicted and true not predicted single-label items, i.e. it actually is a measure of failure. In contrast, the **accuracy** is defined as the averaged Jaccard index:

$$Accuracy(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \begin{cases} 1 & \text{if } L_i = Y_i \\ \dfrac{|L_i \cap Y_i|}{|L_i \cup Y_i|} & \text{otherwise} \end{cases}$$

It is a general measure of how *close* a predicted label has been to a true label in average. **Precision** and **recall** are more specific metrics which express how many of the predicted single-label items actually are part of the true label (precision) and how many of the single-label items contained in the true label actually have been predicted (recall) on average:

$$Precision(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \begin{cases} 1 & \text{if } L_i = Y_i = \emptyset \\ 0 & \text{if } L_i \neq Y_i = \emptyset \\ \dfrac{|L_i \cap Y_i|}{|Y_i|} & \text{otherwise} \end{cases}$$

$$Recall(C,X) = \frac{1}{|X|}\sum_{i=1}^{|X|}\begin{cases}1 & \text{if } L_i = Y_i = \emptyset \\ 0 & \text{if } L_i = \emptyset \neq Y_i \\ \dfrac{|L_i \cap Y_i|}{|L_i|} & \text{otherwise}\end{cases}$$

The strict **match** metric mentioned introductorily is defined as:

$$Match(C,X) = \frac{1}{|X|}\sum_{i=1}^{|X|}\begin{cases}1 & \text{if } L_i = Y_i \\ 0 & \text{otherwise}\end{cases}$$

Next to these standard metrics, the application context of privacy management advocates a further metric. A subset-tolerant match metric, which is zero if the predicted label contains any item which is not part of the true label and which evaluates to the Jaccard index in all other cases, is a good measure for a sensitive information management. It is a combination of the metrics *match* and *accuracy*: more tolerant than a strict *match* but also more defensive than *accuracy* in that it only acknowledges wrong predictions which are subsets of the true label.

$$Submatch(C,X) = \frac{1}{|X|}\sum_{i=1}^{|X|}\begin{cases}1 & \text{if } Y_i = L_i \\ \dfrac{|L_i \cap Y_i|}{|L_i \cup Y_i|} & \text{if } Y_i \subset L_i \\ 0 & \text{otherwise}\end{cases}$$

The most relevant metrics for privacy-related investigations targeted here are *match*, *accuracy*, and *submatch*. The first one indicates how often a predicted disclosure decision equals an actually intended one. Accuracy is a more tolerant measure which indicates how much a user would have to adjust the predictions in order to equal the actually intended disclosures. The latter one, submatch, is a defensive measure of accuracy which does not accept predictions containing information items not part of the true label.

6.2.2 Hierarchical Learning

Similar to multi-label learning problems, hierarchical problems are coped with by transforming the problem to be applicable to non-hierarchical learning methods. Different methods are briefly described in the following section, followed by corresponding implications on performance metrics and their meaning in context of predicting the disclosure of information.

Problem Transformation Methods

Technically the **powerset transformation method** used for multi-label problems can be used similarly for hierarchically structured labels where each path is considered a distinct single-label item. However, obviously it does not utilize any hierarchy information contained in labels. A more advanced method is **binarized structured label learning** (Wu *et al.*, 2005), sometimes also referred to as **hierarchical binary relevance** (Tsoumakas *et al.*, 2010). Similar to the binary relevance method for multi-label learning, a binary base classifier is instantiated for each partial path starting from the root of the label taxonomy. These classifiers are arranged in a similar hierarchy as the (partial) single-label items they represent. When predicting a label, these classifiers are used in a top-down manner according to their hierarchical structure. If a classifier provides a positive prediction for its label, then its child-classifiers are consulted too. In case of a negative prediction or if there are no child classifiers, the (partial) single-label item corresponding to the current classifier is included in the predicted multi-label.

Tsoumakas *et al.* (2010) and Wu *et al.* (2005) provide more detailed information about and illustrations of the method described here. Further, they reference more specific hierarchical learning methods. However, most methods are based on the hierarchical binary relevance method used here and, while they perform better in specific situations, there appears to be no critical performance difference which advocates their additional evaluation here in the first place.

Hierarchy-based prediction validation. Next to exploiting hierarchical information from labels, the binarized structured label learning method has the advantage of being able to express a kind of *uncertainty*, when it predicts a partial single-label item but if it is unsure about possible hierarchical complements. This knowledge of uncertainty may be used to decide if a prediction is used for an automatic disclosure or just as a template for a manual one. However, to make use of this potential feature, the single-label items (which actually are paths in the overall taxonomy of previously disclosed information) have to be extended with a virtual leaf node. Then, if any single-label item in a predicted multi-label does not have this virtual leaf, the prediction is marked as *uncertain* and is respectively used as a suggestion for a manual disclosure. Figure 6.1 illustrates this approach. It shows two disclosures of previous situations, where each single-label item (i.e. a path from root to a leaf) is annotated with a virtual leaf node. Additionally it shows a predicted label. Without the virtual leaf nodes, it would be impossible to decide if the nodes A and X do not have children because the learner neither predicted A (respectively X) alone nor one of its children or because the learner explicitly predicted A (respectively X) alone. However, using the virtual leaves, it is clear that the learner neither predicted A alone nor $A.B$ nor $A.C$ but explicitly predicted X alone.

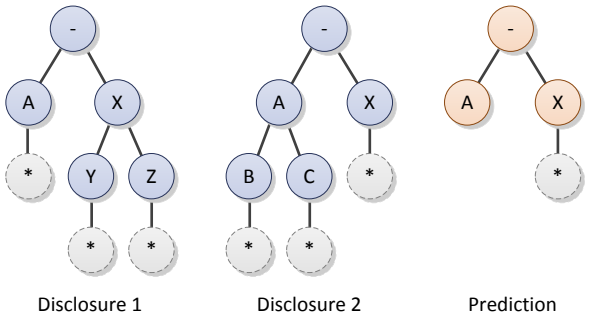

Figure 6.1: Hierarchical labels extended with dummy leaves. Each circle represents a node in the hierarchical taxonomy of personal information. The left two labels are those of previous situations while the right one is the predicted label of a new situation. The dashed circles are virtual nodes added to each leaf node of a disclosure. The existence of these virtual leaves in a prediction indicates whether partial paths are predicted explicitly or because of uncertainty.

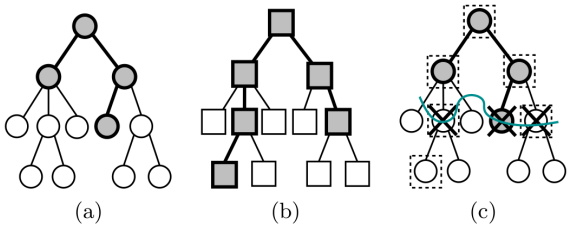

Figure 6.2: H-loss metric for hierarchical multi-label learning. Each tree represents the same label taxonomy of a given data set. The gray nodes in tree (a) represent a predicted hierarchical multi-label while the gray nodes in tree (b) illustrate the true label. The H-loss considers the top-most mismatching nodes as indicated by the checked nodes in tree (c). *Source:* Cesa-Bianchi *et al.* (2004)

Performance Metrics

Clearly a hierarchical multi-label learning also requires specific performance metrics in order to not only account for partial matches of single-label items but also partial matches of hierarchy paths of single-label items.

A similar notation as for multi-label learner metrics is used, with the difference that the elements in L (single-label items) correspond to root-starting paths in a given hierarchical information item structure. That means they are tuples of node elements $(e_1, \ldots, e_v)$ where e_1 is a root node and e_v is a leaf node of a hierarchical mutli-label item. This implies that

redundant sub-path elements are not contained in a label L_i ($i \in \{1, \ldots, |X|\}$), i.e. for any path lengths $1 \leq v < w$ one has

$$(e_1, \ldots, e_v, \ldots, e_w) \in L_i \Rightarrow (e_1, \ldots, e_v) \notin L_i$$

In contrast, let the primed version of a label also contain any sub-path elements:

$$(e_1, \ldots, e_v, \ldots) \in L_i \Rightarrow (e_1, \ldots, e_v) \in L'_i$$

Note that C now refers to a hierarchical multi-label learner, i.e. a learner which follows one of the previously mentioned problem transformation methods.

An often used metric is the **H-loss** proposed by Cesa-Bianchi *et al.* (2004). The general idea is that for a given class taxonomy, each node is processed in a top-down manner (preorder) and whenever a node is present in the predicted label but not in the true label – or vice versa – the H-loss is increased by a cost value related to the current node. However, children nodes of a mismatching node are not processed further, i.e. only the top-most failures are counted, not their derived ones. This scheme is illustrated in figure 6.2 using an example taxonomy with a predicted and true label. For a predicted label Y_i and a true label L_i, the set of nodes Z_i which contribute to the loss are given by:

$$Z_i = \left\{(e_1, \ldots, e_v) \in L'_i \triangle Y'_i \mid \forall w < v \;\; \nexists (e^*_1, \ldots, e^*_w) \in L'_i \triangle Y'_i : e_1 = e^*_1 \wedge \ldots \wedge e_w = e^*_w\right\}$$

Then the H-loss metric is defined as:

$$H\text{-}loss(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \sum_{z \in Z_i} \zeta(z)$$

Concerning the cost function ζ, a simple setting is to let it always evaluate to 1. The problem here is that this favors short-path predictions. To circumvent this, Cesa-Bianchi *et al.* suggest to use node-specific values depending on the number of a node's siblings and its level within the tree, i.e. the cost $\zeta(z)$ for a node z is $\zeta(parent(z))/|children(parent(z))|$ where *parent* and *children* evaluate to the corresponding node (set). Nodes without a parent have a cost value of 1. Given a taxonomy with k root nodes, this cost scheme normalizes the H-loss within $[0, k]$ and charges errors near the top (or root) more than those near the bottom (or leafs).

Still, it is questionable if these recursively reduced error costs yield a useful loss metric in terms of privacy management, where the costs of misclassifications are more related to semantical but not structural context.

An alternative metric with less influence of structural context is to simply relate the number of matching nodes to the number of nodes occurring in both the predicted and true label. Practically this is a hierarchical variant of the *accuracy* metric described for multi-label learners and thus denoted as **H-accuracy**. It is defined as:

$$H\text{-}accuracy(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \begin{cases} 1 & \text{if } L_i = Y_i \\ \frac{|L_i' \cap Y_i'|}{|L_i' \cup Y_i'|} & \text{otherwise} \end{cases}$$

In a similar manner one can define a hierarchical equivalent of the *submatch* metric defined for multi-label problems, the **H-submatch** metric. However, here one has to consider a specialized subset-relation $\subseteq^h$ since shorter paths mean more general information selections and thus correspond to *more* information (in contrast to smaller primed labels):

$$L_1 \subseteq^h L_2 \Leftrightarrow \forall (e_1, \ldots, e_v) \in L_1 \, \exists w \leq v : (e_1, \ldots, e_w) \in L_2$$

Then the hierarchical submatch metric is defined as:

$$H\text{-}submatch(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \begin{cases} 1 & \text{if } Y_i = L_i = \emptyset \\ \frac{|L_i' \cap Y_i'|}{|L_i' \cup Y_i'|} & \text{if } Y_i \subseteq^h L_i \\ 0 & \text{otherwise} \end{cases}$$

In the context of privacy management, another useful metric is that of adjustment costs. If a learner is not only used to predict disclosures but also to provide suggestions or templates for final manual disclosures, the relevant metric here would be how much work a user had to face when adjusting the suggested disclosure to match the actually intended one. To some extent the adjustment costs are calculated similarly to the H-loss, i.e. the nodes in the given class taxonomy are processed in preorder. In contrast, here each mismatch is charged equally, i.e. also children nodes of mismatching nodes are considered and each failure increases the costs by 1. For comparison reasons the metric can be normalized nevertheless by setting an upper bound β of mismatching nodes. In this case the metric is defined to be 1 when the number of mismatching nodes reaches or exceeds β while it is divided by β in all other cases. Such an adjustment costs metric, here denoted as **strain**, assumes a hierarchically organized user interface to (de)select information items to disclose. It is defined as follows:

$$Strain(C, X) = \frac{1}{|X|} \sum_{i=1}^{|X|} \min\left(\frac{|L_i' \triangle Y_i'|}{\beta}, 1\right)$$

Reasonable values for the upper bound β depend on the actual user interface used to adjust suggested disclosures in order to align them with intended ones. However, for initial evaluation purposes $\beta = 10$ is an appropriate bound as more than 10 adjustments can be seen as in unacceptable strain in any case.

6.2.3 Base Learners

The problem transformation methods described above actually are wrappers around single-label base learners. This section presents the base learners evaluated within this work. The list of learners is not supposed to be complete but aims to cover the most popular ones and to cover the most general concepts used for classification: (non)linear discrimination, rule systems, and density estimation (Henery, 1994). In particular the following methods are considered: k-Nearest Neighbors (k-NN), Naive Bayes, Support Vector Machines (SVMs), Rule Induction, and Decision Trees. Besides potential performance differences, these methods differ in the resources required for training and prediction as well as in their transparency concerning automatic decisions and possibilities for manual adjustments. With regard to the application context at hand, privacy management, resource requirements are no crucial factors. In contrast, user interaction possibilities are more relevant – section 6.7 deals with these aspects more thoroughly. The particular algorithms used for the mentioned base learners are described more detailed in section 6.6.4.

6.2.4 Confidence-Based Prediction Validation

Each learning method may annotate predictions with a confidence in the correctness of predictions. Confidence values have different origins respectively meanings depending on the used learning method. Some methods provide confidences based on class distributions while others have independent confidence values for each class. The main point is that confidence values cannot be compared across multiple learners but only across predictions of the same methods.

Confidence values may be used to validate a prediction made by a learner. Next to metrics like accuracy and match, an almost similarly important performance metric of a learning method is its ability to decide if its predictions are likely to be correct. Confidence values may be used for validation along two general strategies. One strategy is to set a static minimum confidence value per learner which must be reached in order to support (i.e. accept) or prevent (i.e. reject) a prediction. The other strategy is to calculate a

confidence threshold dynamically. For instance, the evaluation system presented in section 6.6 calculates a dynamic confidence threshold as follows[2]:

1. Confidence values of previous predictions are grouped into two sets, one containing values of positive (i.e. correct) predictions, and one containing values of negative (i.e. wrong) predictions.
2. Candidates for confidence thresholds are the medians of any two subsequent confidence values among all (numerically sorted) confidence values of previous predictions (including the median of 0 and the smallest confidence value, and the median of 1 and the greatest confidence value).
3. For each candidate, an error is calculated. The error is given by the number of smaller confidence values in the set corresponding to positive predictions plus the number of greater confidence values in the set corresponding to the negative predictions. From all thresholds with a minimal error, the one preventing the most negative predictions is chosen.

6.3 Disclosure Interpolation Based on Order Mappings

Section 4.1.2 elaborated several order-theory–related patterns of information disclosure. One pattern is based on types of order mappings of disclosure situation pairs. For instance, two disclosure situations are considered to be *order reversing* if the person (i.e. information recipient) group from one situation is a subset of the person group from the other situation while the subset relation is inverse for the disclosed sets of information. Practically this reflects the case that a greater information recipient group implicates a smaller set of disclosed information. Other order-mapping types are order *preserving*, *ignoring*, and *loosing* (section 4.1.2 defines these types more formally and provides further illustrative examples). A learning method which has to predict the disclosure for a new disclosure situation may refer to order-mapping types of previous (i.e. already seen) situations to interpolate a disclosure for a new situation. As an example, if there are two previous situations, one with a smaller and one with a greater information recipient group than the one of a new situation, and if the disclosed sets of information items follow the same order (i.e. one is a subset of the other), then a similar order can be assumed for the disclosure to predict for the new situation. Obviously such an interpolation only applies to situations which differ only in the persons group context (i.e. features expressing the presence of persons, see section 6.1.1) – otherwise the order of disclosures cannot directly be linked to the order of person groups. Additionally, an interpolation is only needed when the

[2]The corresponding implementation can be found in the DiLES software (Bünnig, 2011b), in particular in the function `confsep` within the package `diles.learn.util`.

person group of a new situation actually differs from all previous situations – otherwise the disclosure of an identical previous situation could be used directly.

6.3.1 Formal Groundwork

More formally the interpolation requirements and components can be described as follows. Supposed disclosure situations are described by n features and a label (a set of information items disclosed in that situation) while the first feature is assumed to be a set of person identifiers (information recipients). Then let S be a set of m distinct previous, already seen situations:

$$S := \left\{(g^1, f_1^1, \ldots, f_{n-1}^1, l^1), \ldots, (g^m, f_1^m, \ldots, f_{n-1}^m, l^m)\right\}$$

Hence, situation j with $1 \leq j \leq m$ is described by the person group g^j, followed by $n-1$ arbitrary other feature values f_i^j with $i \in \{1, \ldots, n-1\}$ and a label l^j. Similarly, let

$$x := (g^*, f_1^*, \ldots, f_{n-1}^*, l^*)$$

be a new situation, i.e. its feature values do not occur in S in that constellation and its label l^* still has to be predicted (respectively interpolated). Then the set of situations whose features differ from x only in the person group and thus may be used for interpolation is given by the situation filter function Φ:

$$\Phi(S, x) = \left\{(g, f_1, \ldots, f_{n-1}, l) \in S \mid \forall i \in \{1, \ldots, n-1\} : f_i = f_i^*\right\}$$

Note that the set of situations given by $\Phi(S, x)$ semantically corresponds to the set of disclosure decisions T, i.e. instances of the mapping τ, defined in section 4.1.2. However, it differs syntactically in that the domain and codomain of τ here are encoded in one tuple: τ's *person* parameter is represented by g, the *situation* and *modalities* parameters are represented by $f_1, \ldots, f_{n-1}$, and τ's output is represented by l. The different notation used here has been chosen because it better corresponds to the format used for data processed by machine learning methods. Further, let ρ be a function providing situation pairs which are suitable for order-mapping inspections with regard to x:

$$\begin{aligned}\rho(S, x) = \big\{\big((g^1, f_1^1, \ldots, f_{n-1}^1, l^1), (g^2, f_1^2, \ldots, f_{n-1}^2, l^2)\big) \in S \times S \\ \mid g^1 \subset g^2 \wedge \forall i \in \{1, \ldots, n-1\} : f_i^1 = f_i^2\big\}\end{aligned} \tag{6.1}$$

The situation pairs produced by ρ semantically match the set R from definition 4.2 in section 4.1.2, i.e. it contains situation pairs where the person group features have a subset relation while all other features equal. In a similar correspondence the following derived

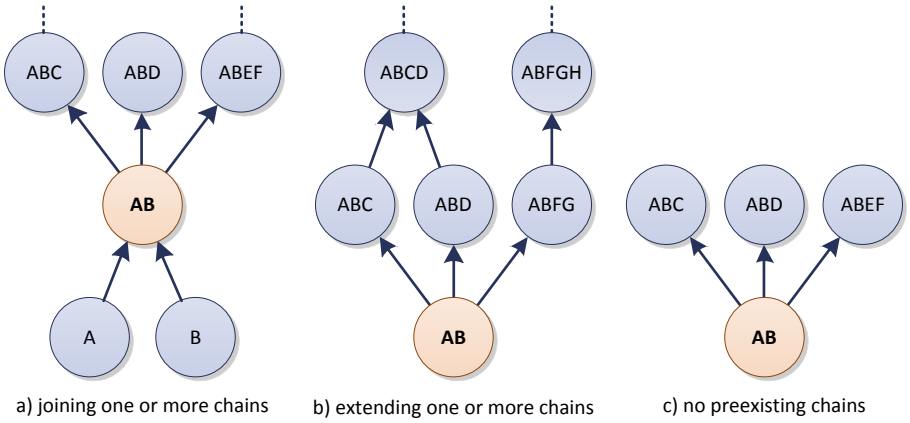

Figure 6.3: Person-group order schemes for order-mapping–based interpolation. Each circle represents the person-group feature of a disclosure situation while each letter represents a specific person. The arrows express an *is-subset-of* relation. The orange, bold-faced person group is that of a new disclosure situation still to decide, denoted as g^* within the text. The other groups are from previous disclosure situations in $\Phi(S, x)$.

functions provide situation pairs with a specific order mapping (*p*reserving, *r*eversing, *i*gnoring, and *l*oosing):

$$\begin{aligned}
\rho_p(S,x) &= \{\big((\ldots,l_1),(\ldots,l_2)\big) \in \rho(S) \mid l_1 \subset l_2\} \\
\rho_r(S,x) &= \{\big((\ldots,l_1),(\ldots,l_2)\big) \in \rho(S) \mid l_1 \supset l_2\} \\
\rho_i(S,x) &= \{\big((\ldots,l_1),(\ldots,l_2)\big) \in \rho(S) \mid l_1 = l_2\} \\
\rho_l(S,x) &= \{\big((\ldots,l_1),(\ldots,l_2)\big) \in \rho(S) \mid l_1 \parallel l_2\}
\end{aligned} \tag{6.2}$$

Note that $\rho_{p/r/i/l}(S,x) \subseteq \rho(S,x) \subset \Phi(S,x) \times \Phi(S,x) \subseteq S \times S$. Interpolation using these situation subsets benefits from greater subsets. In that it is important that the non-person-group features have been preprocessed accordingly to eliminate noise. For instance a *room* feature may have values “R1” and “R2” while both indicate the same social context, i.e. they should be joined to “R1/2”.

In real life there may be two situations with identical features but different disclosures, i.e. disclosure decisions may conflict with past decisions. However, the interpolation described subsequently assumes non-conflicting disclosures. In this regard previous disclosure situations could be sanitized by only considering the most recent one of conflicting situations.

6.3.2 Order Schemes

If and how an interpolation is possible depends on how g^* relates to the person groups features of the situations in $\Phi(S, x)$. One has to distinguish different order schemes, i.e. how g^* integrates into chains of person groups[3] from situations in $\Phi(S, x)$. The integration may be described with the notion of *upper* and *lower* neighbor situations of x in $\Phi(S, x)$ with regard to the person group feature. Formally these neighbors are given by the situation filter functions Ω (upper) and ω (lower)[4]:

$$\Omega(S,x) = \left\{ y \in \Phi(S,x) \mid \pi_1(x) \subset \pi_1(y) \wedge \nexists y' \in \Phi(S,x) : \pi_1(x) \subset \pi_1(y') \subset \pi_1(y) \right\}$$

$$\omega(S,x) = \left\{ y \in \Phi(S,x) \mid \pi_1(y) \subset \pi_1(x) \wedge \nexists y' \in \Phi(S,x) : \pi_1(y) \subset \pi_1(y') \subset \pi_1(x) \right\}$$

Subsequently, the notions of *upper* and *lower neighbors* refer to these definitions while the term *neighbors* in general denotes to the union of upper and lower neighbors. Based on the existence of upper and lower neighbors, there are four types of order schemes to consider (figure 6.3 illustrates the first three):

1. In the ideal case x has both upper and lower neighbors, i.e. g^* **joins existing chains** of person groups (see figure 6.3.a). The order mapping of situation pairs from lower and upper neighbors may then be used to interpolate a disclosure for situation x, i.e. l^*. The pairs are given by the Cartesian product of x' lower and upper neighbors:

   $$\omega(S,x) \times \Omega(S,x)$$

   Note that these pairs are a subset of $\rho(S, x)$.

2. In case x has either upper or lower neighbors but not both and in case these neighbors have upper respectively lower neighbors as well, i.e. x **extends existing chains** of person groups (see figure 6.3.b), the order mappings between the upper/lower neighbors and their upper/lower neighbors may be used for interpolation. The situation pairs in case of upper respectively lower neighbors are given by:

   $$\bigcup_{y \in \Omega(S,x)} \left\{ (y,z) \mid z \in \Omega(S,y) \right\} \quad \text{respectively} \quad \bigcup_{y \in \omega(S,x)} \left\{ (z,y) \mid z \in \omega(S,y) \right\}$$

   Again, these pairs are a subset of $\rho(S, x)$.

3. Otherwise, if x has neighbors but **no existing chains** for g^* (see figure 6.3.c), then there are no order mappings which could be used for interpolation.

[3] Chains of person groups are person groups whore are mutually comparable with regard to the subset relation.

[4] The expression $\pi_k(t)$ denotes a tuple projector, evaluating to the k-th element in tuple t.

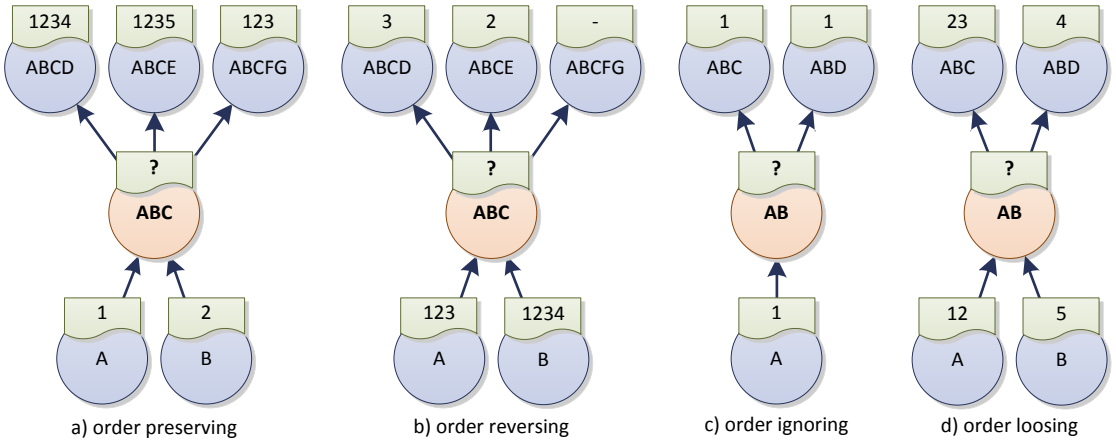

Figure 6.4: Joining existing chains with homogeneous order mappings. Each circle represents the person group feature of a disclosure situation while the orange, bold-faced circle refers to a new situation, referenced as x within the text. The letters represent person identifiers. The arrows display an *is-subset-of* relation with regard to the person group. The annotations at the top of the circles illustrate the information items disclosed in the relating situation. Here each digit represents one specific information item. The question mark indicates that the information items to disclose still need to be interpolated based on the upper and lower disclosures and corresponding order-mapping types.

4. The same is true for the case when x has **no neighbors** at all.

Disclosure interpolations may follow different strategies, concerning how to select order mappings when g^* joins or extends multiple chains and concerning fallback methods when no chains exist or when situation pairs from the chains in all or most cases have order-loosing mappings (which provide no hints how to interpolate a disclosure for a situation joining or extending such chains). The following sections describe possible interpolation strategies with reference to the four order-scheme types described above.

6.3.3 Interpolation When Joining Existing Chains

At first, it is assumed that all situation pairs given by the upper and lower situation neighbors of x have the same order-mapping type. This situation is illustrated in figure 6.4. For instance figure 6.4.b shows the case where a new situation joins existing chains of person groups which all have an order-reversing mapping, i.e. a greater information recipient group implicates a smaller set of disclosed information items. Figure 6.4.a shows the opposite case while figure 6.4.c illustrates order-ignoring mappings, i.e. the disclosed information items do not change and ignore the orders of person groups. Finally figure

6.4.d displays the constellation of order-loosing mappings, i.e. the information item sets have no order. Each constellation shown in figure 6.4 requires different interpolation strategies:

a) For **order-preserving** mappings, the disclosure l^* of the new situation with $g^* = \{A, B, C\}$ either may be the union of disclosures from lower situations or the intersection of disclosures from upper neighbors. In the displayed example both would result in different disclosures, either $l^* = \{1, 2\}$ or $l^* = \{1, 2, 3\}$. Both would follow the order mappings given by the upper and lower neighbors but which one should finally be used? While one could simply either always use the union of lower or the intersection of upper neighbors, a better approach from an interpolation perspective is to calculate the distance[5] of g^* to the intersection of person groups from upper neighbors and to the union of person groups from lower neighbors and then use upper or lower neighbors depending on the closest combined person group. If both distances are equal the choice could be made randomly or follow a default, e.g. using the union of disclosures from lower neighbors because this is more defensive in terms of the amount of information which is going to be disclosed. With regard to figure 6.4.a this would result in the interpolated disclosure $l^* = \{1, 2, 3\}$.

b) The case of **order-reversing** mappings generally is symmetrical to order-preserving mappings, i.e. one would use the union of disclosures from upper neighbors and the intersection of disclosures from lower neighbors. Then, based on the distance of g^* to the person group intersection from upper and union from lower neighbors, the nearest one is chosen. In the displayed example this would result in $l^* = \{2, 3\}$.

c) Interpolating between situations with **order-ignoring** mappings is straightforward in that l^* simply must be identical to the disclosure used in neighbor situations. Consequently, the illustrated example in figure 6.4.c would result in $l^* = \{1\}$.

d) Situation neighbors with **order-loosing** mappings do not provide any hints for interpolation as disclosure orderings do not correlate with person group orderings. As a fallback one could either choose the disclosure of one of the closest[5] neighbor situations or delegate the disclosure decision to a completely different mechanism (e.g. one of the learning methods mentioned in section 6.2).

[5] In context of sets the term distance refers to the Jaccard distance. For two sets A and B it is defined as $1 - (|A \cap B|/|A \cup B|)$ if $A \cup B$ is not empty and 0 otherwise. When using distance related terms for situations (whose features only differ in the person group), it actually means the distance of the situation's person group feature values.

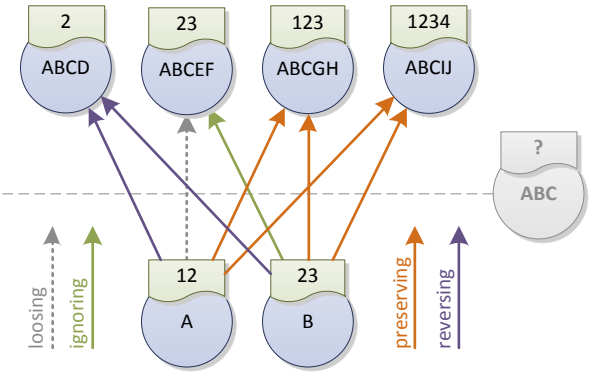

Figure 6.5: Heterogeneous order mappings. Similar to figure 6.4, the upper and lower neighbors of the new situation x with $g^* = \{A, B, C\}$ are displayed. Additionally, the order-mapping types of the upper and lower neighbor situation pairs are shown.

Handling heterogeneous order mappings

Until now it has been assumed that all situation pairs given by the upper and lower neighbors of the new situation x have the same order-mapping type. Interpolation gets more complicated if these pairs have different types. Figure 6.5 illustrates such a case. The most simple handling is to not interpolate and to delegate the prediction to another decision component. On the other side, heterogeneous mapping types can be handled by selectively dropping neighbor situation pairs until all remaining ones have similar order mappings. Situation pairs with order-loosing mappings do not contribute to an interpolation and thus may be discarded first. For the remaining situation pairs there are two general discard approaches to get a homogeneous set of types, based on order-mapping counts or on situation distances.

Discard based on order-mapping counts: First, among the situation pairs given by the lower and upper neighbors of x, discard these whose order-mapping type occurs less often than another one. If the remaining pairs still have multiple types, discard these whose order-mapping type occurs less often than another one within the situation pairs given by $\rho(S, x)$. For instance, if remaining pairs have 2 order-preserving and 2 order-ignoring mappings but $|\rho_p(S, x)| > |\rho_i(S, x)|$, the pairs with order-ignoring mappings are discarded. That is occurrence numbers are evaluated in a local to global manner.

Discard based on situation distances: Only keep situation pairs where both situations have a minimal distance[5] to x (compared to the other lower respectively upper neighbors).

Both approaches can be combined, i.e. if one strategy does not lead to homogeneous order mappings, the other one could be applied additionally. However, the order is important here. In some cases none of these strategies result in homogeneous order mappings. In that case the interpolation has to be refused. With reference to the example situations illustrated in figure 6.5, the approach using order-mapping counts would drop all but the order-preserving situation pairs, resulting in an interpolated disclosure of $l^* = \{1, 2, 3\}$. In contrast, the distance-based approach would filter all but the order-reversing pairs and thus interpolate $l^* = \{2\}$.

Eventually the following homogenization strategies (i.e. combination of approaches) may be applied:

1. do not homogenize (immediately refuse to interpolate);
2. drop order-loosing pairs;
3. drop pairs based on distance;
4. drop pairs based on order-mapping counts;

5-6. permutations of approaches *3* and *4*;

7-10. approach *2* in combination with either one of *3* to *6*

Next to homogenization by dropping pairs, another approach is to drop no pairs but assume all pairs have the same type, either order-preserving or order-reversing, whichever occurs more often among x' neighbor pairs or among pairs in $\rho(S, x)$. If none of both types occurs more often, interpolation is refused. However, this brute homogenization should not be applied when the majority of reference pairs (i.e. x' neighbor pairs or $\rho(S, x)$) have an order-loosing mapping. The possible homogenization strategies now extend to:

11. assume a uniform mapping type based on x' neighbors pairs;
12. assume a uniform mapping type based on $\rho(S, x)$;
13. apply *11* and apply *12* if needed

Which of these strategies performs best in practice is evaluated in section 6.6.

6.3.4 Interpolation When Extending Existing Chains

To some extent the interpolation of a disclosure when x extends existing situation chains (see figure 6.3.b) works similar to the case when x *joins* existing chains. However, some aspects need to be handled differently. Again, first it is assumed that all chains which x extends have the same order-mapping type, as illustrated in figure 6.6. Depending on the order mappings found in the chains extended by x, the label l may be interpolated as follows:

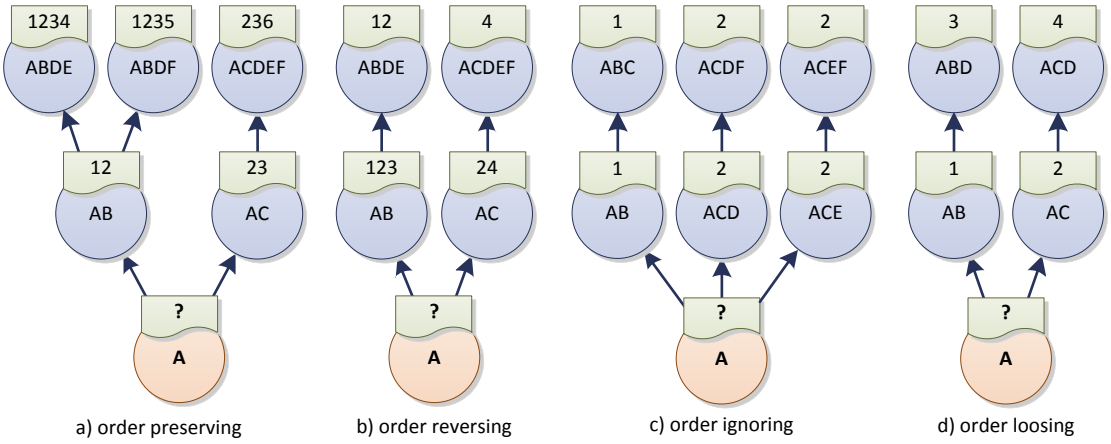

Figure 6.6: Extending existing chains with homogeneous order mappings. Each circle represents the person group feature of a disclosure situation while the orange, bold-faced circle refers to a new situation, referenced as x within the text. The letters represent person identifiers. The arrows display an *is-subset-of* relation with regard to the person group. The annotations at the top of the circles illustrate the information items disclosed in the relating situation. Here each digit represents one specific information item. The question mark indicates that the information items to disclose still need to be interpolated based on the situations whose person group chains x extends.

a) In case of **order-preserving** mappings, the new disclosure l^* is the intersection of the disclosures of x' neighbors if x extends the chains at the bottom. Otherwise, if x extends chains at the top, l^* is the union of the disclosures of x' neighbors. In the example displayed in figure 6.6.a this would result in $l^* = \{2\}$.

b) Interpolation works symmetrically when the extended chains are **order reversing**, i.e. l^* is interpolated by the union of upper neighbors respectively the intersection of lower neighbors. With regard to figure 6.6.b this would yield $l^* = \{1, 2, 3, 4\}$.

c) Interpolation in case of **order-ignoring** mappings differs from the case of joining chains in that the extended chains nevertheless may have different disclosures. In that constellation there is no interpolation which preserves order-ignoring mappings within the extended chains. Consider figure 6.6.c as an example. This case can be coped with by selecting the disclosure of a neighbor (which potentially results in non-ignoring order mappings with other neighbors) or by applying an order-preserving respectively order-reversing interpolation:

Select an existing disclosure: Chose the disclosure which occurs most often in x' closest[5] neighbors, in all of x' neighbors, or in $\Phi(S, x)$, depending on which reference set at first yields a winner (evaluated in the given order). If there is no winner, refuse to interpolate or try the next strategy.

Apply order-preserving or order-reversing interpolation: Assume an order-preserving or order-reversing mapping, depending on which type occurs most often in $\rho(S, x)$ and then interpolate as in *a)* respectively *b)*. Refuse to interpolate if both mapping types occur equally often or if order-loosing mappings occur more often in $\rho(S, x)$ than any other mapping type.

Both approaches may be used individually or combined in the order listed here. This results in 4 different strategies to interpolate when extending existing chains with order-ignoring mappings where two or more chains have different disclosures:

1. do not interpolate
2. select a neighbor disclosure
3. apply order-preserving or order-reversing interpolation
4. use approach *2*, if needed followed by approach *3*

Supposed that strategy *4* is used, the example displayed in figure 6.6.c would result in an interpolated disclosure of $l^* = \{1\}$. If all neighbors of x had an equal distance, the interpolated disclosure would be $l^* = \{2\}$. In case all neighbors have an equal distance and both disclosures would occur equally often, there would be an interpolation of $l^* = \emptyset$ if $|\rho_p(S, x)| > |\rho_r(S, x)|$, an interpolation of $l^* = \{1, 2\}$ in the opposite case, and a refused interpolation if order-preserving and order-reversing mappings occur equally often in $\rho(S, x)$.

d) In contrast to the case of joining chains, it is still possible to interpolate when x extends chains with **order-loosing** mappings since interpolation for an extending situation does not break the order-loosing mappings within the existing chains. This works similar to the case of extending chains with order-ignoring mappings where different chains have different disclosures. In particular strategies *1* and *3* may be used here too. When assuming an order-preserving mapping, the example in 6.6.d would result in $l^* = \emptyset$. The opposite case would yield $l^* = \{1, 2\}$.

Handling heterogeneous order mappings

In case x extends chains which have heterogeneous order mappings, the same strategies as used when *joining* chains apply here (see page 90), i.e. either refuse to interpolate or discard selected chains until all remaining chains have the same order-mapping type. The only difference is that distance-based discarding here only needs to consider the distances of x' neighbors, not both situations in the reference situation pairs.

6.3.5 Interpolation When Missing Existing Chains

If the new situation x has neighbors but these neighbors are not part of any chains, as illustrated in figure 6.3.c, then these neighbors do not provide interpolation hints themselves. However, similar strategies as used when extending chains with order-loosing mappings may be may be used, i.e. interpolation is based on occurrences of order-mapping types in $\rho(S, x)$ only – see case *d)* on page 93.

Always use global order-mapping counts

Actually this interpolation variant may also be used for all order schemes where x has at least one neighbor, i.e. in any case all pairs in $\rho(S, x)$ (not only the direct neighbor pairs) are used to select the order mapping used for interpolation. This is a comparatively simple interpolation variant which does not have to deal with heterogeneous order mappings. It only interpolates if there is exactly one most often occurring order-mapping type in $\rho(S, x)$ and if this type is either *preserving* or *reversing*. Otherwise the interpolation is refused.

6.3.6 Interpolation When Missing Any Neighbors

The disclosure of a new situation x where no existing situation in $\Phi(S, x)$ has a person group feature with a sub- or superset relation to x' person group feature g^*, i.e. x has no neighbor situations, cannot be interpolated based on order mappings, resulting in a refused interpolation respectively the delegation of the disclosure prediction to another decision component.

6.3.7 Summary

On a very high level an interpolation based on order mappings can be summarized as follows:

1. detect the order scheme given by a new situation x and previous situations S and correspondingly compile the set of reference situations from S to be used for interpolation;
2. discard certain reference situations in order to homogenize order-mapping types yielded by the reference situations;
3. interpolate disclosure for x based on detected order scheme, order-mapping types found in remaining reference situations, and possibly general order-mapping types in $\rho(S, x)$

If in any of these steps the interpolation has been refused, the disclosure prediction must be delegated to a fallback decision component, e.g. one of the general learning methods reviewed in section 6.2.

Obviously, the less situations pairs with an order-loosing mapping exist the better is the performance of an order-mapping–based interpolation. Additionally, the performance benefits from homogeneous order-mapping types within potential sets of reference situations. The evaluation of the presented interpolation-based disclosure prediction (see section 6.6) complements performance results with order-mapping statistics in order to illustrate this correlation.

The interpolation process may vary depending on several parameters. First, there are the different strategies used to homogenize heterogeneous order mappings and to handle order-ignoring mappings with multiple disclosures when extending situation chains. Other parameters control if to refuse an interpolation when extending chains or when missing existing chains. Further, the definition of lower and upper neighbors from section 6.3.1 may be altered to only consider neighbors with a minimal distance to the situation to interpolate (which in turn would make distance-based homogenization strategies obsolete). Finally, interpolations may be configured to require a minimum number of neighbors in order to get active. Section 6.6 describes the process of choosing a suitable parameter setup.

An implementation of this method can be found in the DILES software (see section 6.6 and appendix B), in particular in the class `OMILearner` within the package `diles.learn.wrappers.social`.

Implicit Self-Deactivation

A special characteristic of the interpolation-based learning method presented in this section is its implicit ability to estimate its suitability for an applied data set. This is the case when the parameters are set properly by a parameter optimization in face of a given training set. This method always works in conjunction with another fallback learning method which is activated when an interpolation is refused. Hence, a parameter optimization implicitly defines the cases when an interpolation should be used and when the fallback learner should be used. In particular any parameter specifying required neighbor situation constellations contributes to the implicit self-deactivation of the interpolation learner for situations where it performs worse than the fallback learner. A positive consequence of this characteristic is that the interpolation learner potentially performs at least as good as its fallback learner alone in all training set cases and in most new cases (the results presented in section 6.6.8 illustrate this aspect).

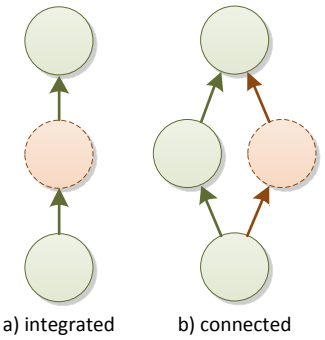

Figure 6.7: Path-integrated and path-connected disclosure predictions. The circles represent disclosures while the arrows display *is-subset-of* relations. A predicted disclosure (dashed circle) may integrate into or connect to paths given by disclosures of previous situations (solid circle). The trivial case when no existing paths are touched by a predicted disclosure is not illustrated here.

6.4 Validate Predictions Using Disclosure Patterns

The previous section described how to use disclosure patterns from section 4.1 to predict disclosures of new situations. Additionally, these patterns may be used to validate predictions in order to prevent wrong predictions. Even if disclosures cannot be predicted in a reasonable amount of cases, automating them with learning methods is still desirable if wrong predictions are detected in advance. Thus, the purpose of a validator is to decide if to *support* or *prevent* a prediction made by a learning method.

The patterns described in section 4.1 are based on social context information, in particular the set of persons receiving the information to disclose. Consequently, when validating the predicted disclosure of a new situation, only previous situations with identical non-social context should be considered. With reference to the formal groundwork in section 6.3.1, these situations are given by $\Phi(S, x)$ where S denotes all previous situations and x denotes the new situation.

6.4.1 Disclosure Paths

The pattern of *disclosure paths* describes the fact that the set of disclosures is a partially ordered set where consecutive subset relations between disclosures can be seen as disclosure paths (see figure 4.2 on page 46). More precise, each maximal chain in the poset of disclosures is considered as a disclosure path. Following are four approaches how to utilize these paths in order to validate predicted disclosures.

Integrated in existing disclosure paths: The most simple variant is to only accept disclosures which integrate into existing disclosure paths, i.e. which do not increase the number of disclosure paths (see figure 6.7.a).

Connected to existing disclosure paths: A looser validation is to also accept predicted disclosures which create new disclosure paths but which are connected to existing paths (see figure 6.7.b).

Major disclosure paths: Both previous validations may be varied in that not any existing paths but only major ones are considered. This raises the question when a path actually is considered as a major one, i.e. which length it must have at least. Straightforward candidates for this length threshold are the mean or a quartile of all path lengths.

6.4.2 Disclosure Complexity

Validation based on disclosure complexity follows the principle of not trying to automate an apparently complex disclosure behavior. The degree of complexity may be distinguished based on the poset width of previous disclosures and/or the number of unique disclosures in relation to the number of all disclosures (among the situations in $\Phi(S, x)$).

Poset width: The simplest approach is to set a maximum poset width and prevent disclosures for situations x where the poset width of disclosures from $\Phi(S, x)$ is greater than the specified maximum.

Ratio of poset width to number of unique disclosures: Alternatively, a maximum ratio of the poset width to the number of unique disclosures from previous situations $\Phi(S, x)$ could be used. For any situation set $\Phi(S, x)$ which yields a higher ratio, predictions would be prevented.

Ratio of unique disclosures to number of situations: Similar to the poset width a maximum number of unique disclosures may be used to prevent disclosure predictions for new situations. However, in practice this is likely to prevent many correct predictions if the cardinality of $\Phi(S, x)$ is significantly larger than than the maximum number of unique disclosures. A more robust validation also considers this cardinality and uses the ratio of the number of unique disclosures to the number of previous situations and only supports predictions when this ratio is below a certain maximum.

6.4.3 Disclosure Usage Counts

The idea of using disclosure usage counts for validation is to only support predictions of *established* disclosures. The notion of being established may by interpreted as follows.

Absolute usage counts: A simple interpretation is to only support predictions of disclosures which have been used at least a certain number of times within the situations in $\Phi(S, x)$.

Relative usage counts: A more sophisticated interpretation considers relative usage counts, i.e. it only supports predictions of major disclosures. Similar to major disclosure paths, straightforward count thresholds indicating major disclosures are the mean count or a quartile of all disclosure usage counts within $\Phi(S, x)$.

6.4.4 Order Mapping

In addition to interpolation (see section 6.3), order-mapping–based patterns may be used to validate disclosure predictions. The basic idea is to check if the predicted disclosure of a new situation *conforms* with the order mappings found among the neighbor situations of the new situation. Obviously this only works if there are neighbors, as illustrated in figure 6.3. In all other cases a default validation result (support or prevent) has to be used. Further, an order-mapping–based validation is not needed when the new situation and its prediction exactly match an existing situation and its corresponding disclosure. In that case the prediction would be supported right away. Depending on the new situation's neighbors, an order-mapping–based validation works as follows.

Validation When Joining Existing Chains

When the new situation joins existing chains (see figure 6.3.a), a predicted disclosure *conforms* with existing order mappings if the new situation conforms with, i.e. does not break, at least a certain percent of mappings. A predicted disclosure for a new situation x is defined to conform with the mapping of a situation pair (y, z), where y is a lower neighbor and z is an upper neighbor of x, if the order-mapping of the pair (y, z) is:

a) order loosing (because there is no relation which could be broken),
b) order reversing and the two new pairs (y, x) and (x, y) have an order-ignoring or order-reversing mapping,
c) order preserving and the two new pairs (y, x) and (x, y) have an order-ignoring or order-preserving mapping,
d) order ignoring and both new pairs also have an order-ignoring mapping.

Figure 6.8 provides two corresponding examples. In the left example ($l^* = \{1, 2, 3\}$) 3 out of 8 mappings are broken. In the right one ($l^* = \{2\}$) 5 out of 8 are broken. If only the closest neighbors had been considered (that is without the three rightmost upper neighbors), the left example would break all mappings while the right one would break

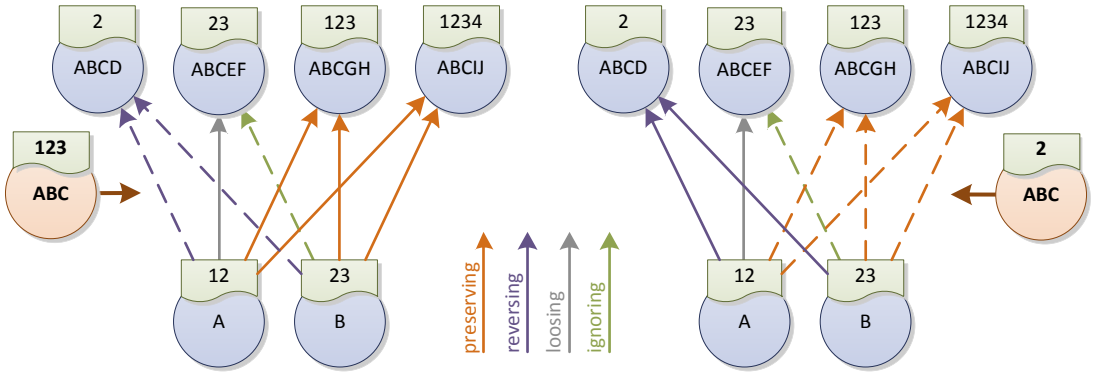

Figure 6.8: Two examples of order-mapping conformance when a situation (for which a prediction is to be validated) joins existing chains. Each circle represents the person group feature of a disclosure situation while the orange, bold-faced circles refer to a new situation. The letters represent person identifiers. The arrows display an *is-subset-of* relation with regard to the person group. The annotations at the top of the circles illustrate the information items disclosed respectively predicted in the corresponding situation. Here each digit represents one specific information item. Dashed arrows indicate order mappings which get broken by the predicted disclosure.

none. In the not illustrated example of a predicted disclosure of $l^* = \{2, 3\}$, the mappings of all pairs but those with an order-ignoring or an order-loosing mapping would be broken, i.e. 6 out of 8 broken mappings.

The cases *b)* and *c)* also accept order-ignoring mappings because it is not always possible to predict a disclosure which has a subset relation with the disclosure of its lower *and* upper neighbor. This is the case when their difference consists of only one item, as seen in the right example in figure 6.8: there is no disclosure which is a subset of $\{1, 2\}$ respectively $\{2, 3\}$ and at the same time a superset of $\{2\}$. Additionally, if one order mapping of the resulting two new pairs is order-ignoring, the other one implicitly equals the order-mapping of the existing pair.

Validation When Extending Existing Chains

Similar to the case of joining situation chains, when a new situation extends existing chains (see figure 6.3.b), a predicted disclosure *conforms* with existing order mappings if the new situation conforms with at least a certain percent of mappings. However, the definition of *conforms* differs. Let x be a new situation and let y be an upper neighbor situation which itself has an upper neighbor z. A predicted disclosure for x is defined to conform with the mapping of an upper situation pair (y, z), if this pair's mapping is:

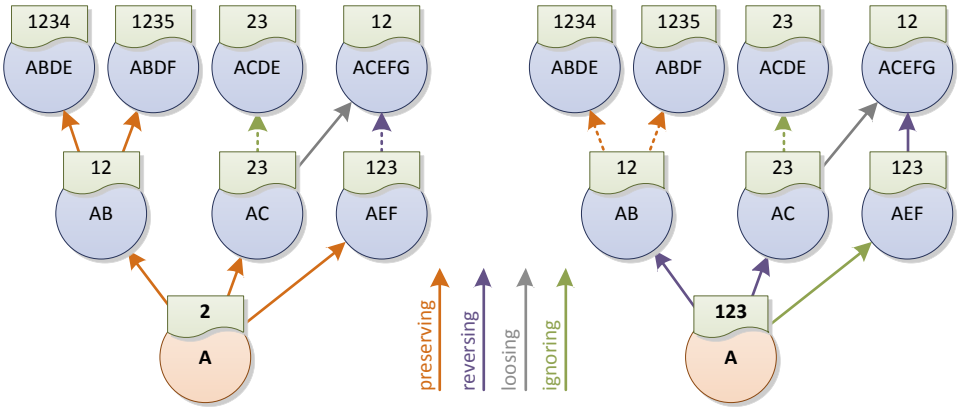

Figure 6.9: Two examples of order-mapping conformance when a situation, whose prediction is to be validated, extends existing chains. Each circle represents the person group feature of a disclosure situation while the orange, bold-faced circles refer to a new situations. The letters represent person identifiers. The arrows display an *is-subset-of* relation with regard to the person group. The annotations at the top of the circles illustrate the information items disclosed respectively predicted in the relating situation. Here each digit represents one specific information item. Dashed arrows indicate order mappings which get broken by the predicted disclosure.

a) order loosing (because there is no relation which could be broken),
b) order reversing and the new pair (x, y) has an order-ignoring or order-reversing mapping,
c) order preserving and the new pair (x, y) has an order-ignoring or order-preserving mapping, or
d) order ignoring and the new pair (x, y) also has an order-ignoring mapping.

The cases *b)* and *c)* also accept combinations of order-preserving respectively order-reversing with order-ignoring mappings because order-ignoring mappings in (x, y) still ensure that the pairs (x, z) and (y, z) have identical mappings. Order-mapping conformance is symmetrical when x extends chains at the top, i.e. when x has lower neighbors which also have lower neighbors. Here one would inspect the existing pairs (z, y) and the new pair (y, x).

Two examples of order-mappings conformance are illustrated in figure 6.9. In the left case the predicted disclosure of $l^* = \{2\}$ conforms with with the order mappings of 3 out of 5 existing upper neighbor situation pairs. In the right case, with $l^* = \{1, 2, 3\}$, there's a conformance with only 2 out of 5 order-mappings. An order-mapping–based validation with a minimum ratio of conforming order-mappings of 0.5 would support the prediction of the left case but prevent the right one.

Validation When Creating New Chains

In case the new situation has either upper or lower neighbors which do not have further upper respectively lower neighbors (see figure 6.3.c), the only way to validate disclosures based on order mappings is to use general order-mapping counts (among situation pairs in $\rho(S, x)$) as a reference. That is, only these disclosures are supported where one of the most often occurring order mappings among the resulting new situation pairs (given by x and its neighbors) also is one of the most often occurring order mappings in $\rho(S, x)$.

Parametrization

An order-mapping–based validation may be parametrized in several ways. First, validation when extending chains or when creating new chains may be skipped. The rationale here is that these two cases are weaker indications of order-mapping conformance. Further, when joining or extending chains, the required minimum ratio of conforming order mappings may be varied. Straightforward ratio thresholds are 0.5, 0.75, and 1.0.

6.4.5 Combinations

Some of the validation methods presented above can also used in combination while other impede each other or are redundant. For instance a validation based on disclosure usage counts makes a validation based on the creation of a new disclosure path obsolete (an already used disclosure cannot create new disclosure paths). Also, a complexity-based validator and an order-mapping–based validator impede each other in that the latter one requires a certain complexity of disclosures in order to come into action. However, reasonable combinations are:

Usage count and major disclosure paths: First, check if a minimum usage count (absolute or relative) of a disclosure is given. If this validation succeeds, check if the predicted disclosure is part of a major disclosure path.

Usage count and order mappings: First, check if a minimum usage count (absolute or relative) of a disclosure is given. If this validation succeeds, validate based on order-mappings.

6.5 Utilizing Scenario-specific Semantic Information

Next to disclosure patterns, scenario-specific semantic information may help learning methods in reasoning about disclosure predictions. For instance information types like *identifying* or *communicational* respectively *transient* or *long-living* (see section 3.2.2) could be used to define an ordering of information items similar to the set-based ordering utilized in sections 6.3 and 6.4. Then this ordering may also be used to interpolate and validate disclosures. In general, any knowledge about the relative sensitivity of personal information is a valuable input for disclosure assistance mechanisms. The same approach may also be applied to the medium used to communicate information (e.g. a display wall is more public than a personal device of an information recipient). However, here these ideas are only mentioned for the sake of conceptual completeness. Subsequent implementations and evaluations in this work do not investigate this further. An investigation of this approach seems to be more promising in face of a particular real-life scenario, i.e. for now it is pointed out as a candidate for future work.

6.6 Evaluation

The previous sections described several techniques for automating the disclosure of personal information. This section elaborates how these techniques may be evaluated. Basically an evaluation should be able to answer the following questions:

How do different learning methods compare to each other? This is the most obvious question. However, methods may be compared to each other in various ways. First, there are the different performance metrics to consider (see section 6.2). Additionally, these metrics may be compared globally, i.e. using all tests made with a learning method, or progressively, i.e. depending on the size of the training set used to learn a prediction model. For instance there might be methods which perform best when only a few previous situations exist while they are outperformed by other methods when a large situation set is available for training. Last but not least, each method should be compared to a plain or majority-based guessing of disclosures.

Which learner configurations perform best? Learners usually may be configured by several parameters. Especially the parametrization of a disclosure interpolation (see section 6.3) is very complex. Determining the best configuration for a learner is an import finding when evaluating learning methods.

How do different validation methods compare to each other? Section 6.4 described several validation methods for made predictions. The number of prevented negative (i.e.

wrong) and supported positive (i.e. correct) predictions is almost as important as general learner performance metrics. Validators face the opposing requirements to prevent as much negative predictions as possible while preventing as less positive predictions as possible. Thus, the actual question is which validator performs best depending on the relevance assigned to each requirement.

Which correlations exist between results and disclosure patterns? In context of the elaboration of disclosure patterns in section 4.1, it has been claimed that these patterns have an impact on disclosure assistance mechanisms. To investigate this impact, the evaluation system should provide means to inspect correlations between patterns and prediction performances, validation performances, as well as learner configurations.

Are there generally well-performing methods, configurations, or validators? In other words this question asks if some methods, configurations, or validators perform equally well for different users or if results are rather diversified across different users respectively disclosure behaviors. In the latter case disclosure assistance systems would have to dynamically choose appropriate settings which increases implementation complexity and required computing resources.

An evaluation system has been developed which is supposed to provide answers to these questions. The next sections up to 6.6.6 present this evaluation system. This is followed by a description of the data sets (i.e. scenarios) applied to the evaluation system in section 6.6.7. Finally, sections 6.6.8 and 6.6.9 present and discuss the corresponding evaluation results. Note that subsequently the developed evaluation system is referred to as DiLES[6] (disclosure learning evaluation system).

6.6.1 Evaluation System Overview

The general steps of the evaluation process are illustrated in figure 6.10: The system loads a scenario and sets up evaluation tasks for different combinations of scenario preprocessors, wrapping learners (e.g. hierarchical learners), and base learners. These parallelly executable evaluation tasks produce a set of performance results which finally are collected for analysis and visualization. Subsequently these steps are described in more detail.

[6]The software and data related to this system is available at the Open Science Repository of the Computer Science Department at Rostock University: `http://opsci.informatik.uni-rostock.de/index.php/DiLES`. Further information can be found in appendix B.

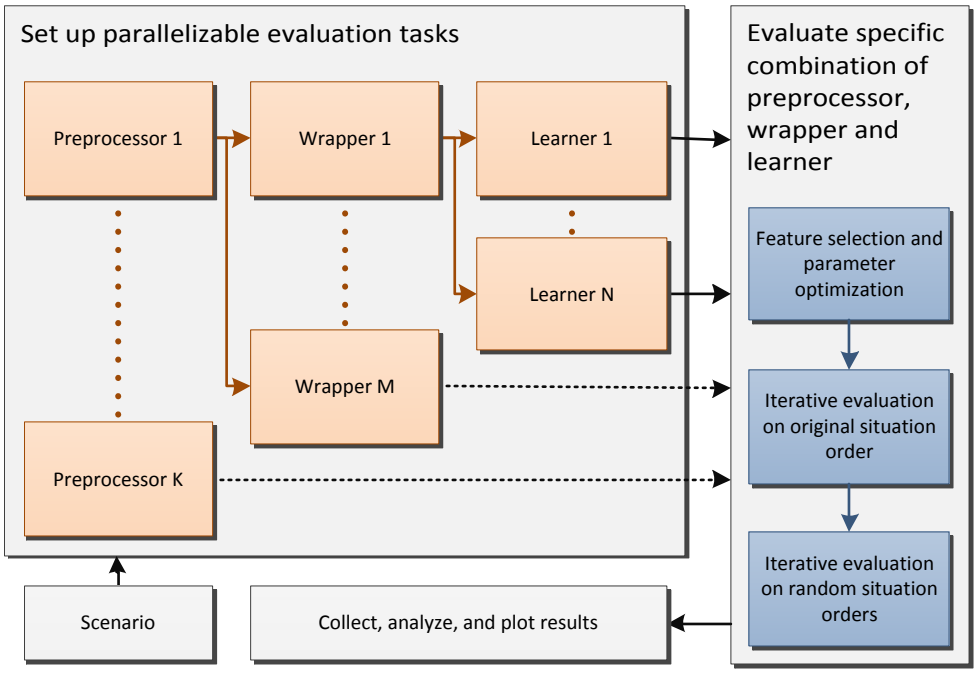

Figure 6.10: Overview of the evaluation process.

6.6.2 Scenario Representation

Scenarios to evaluate are described as a sequence of situations, where each situation describes a disclosure decision of a *subject* (the user) based on the disclosure modality and multiple context information items. This resembles the situation representation described in section 6.1.1 on page 74. A scenario may contain situations for multiple subjects. However, the evaluation system unfolds them to individual scenarios for each subject so that a finally evaluated scenario only contains situations for one specific subject. Table 6.1 shows an excerpt of a scenario, in particular its first four situations. The disclosure decisions have been omitted for readability reasons – examples like those given in section 6.1.2 on page 74 would fit here too. The only required items per situation are *subject*, *persons* (information receivers), and *disclosure*. All other items are domain-specific context items. Technically scenarios are written in YAML (Ben-Kiki *et al.*, 2009), which makes them easy to read and write for humans. Context information item values may be numeric, nominal (given as a string identifier), or sets (given as a sequence of nominal values). Disclosure decisions are given as sets of information items, each a string whose hierarchical structure is encoded using a dot-separator. Appendix B provides an example scenario definition and further details on how to define scenarios.

	S1	S2	S3	S4	..
Subject	Bob	Bob	Sue	Bob	
Trigger	start meeting	start meeting	start meeting	start meeting	
Medium	display wall	user devices	display wall	display wall	
Persons	Sue, Pete	Sue	Bob	Pete, Paul	
Tags	internal, meeting	external, meeting	internal, meeting	internal, meeting	
Disclosure	...	...	...	...	

Table 6.1: Illustrative excerpt of an exemplary scenario to evaluate. The columns describe situations from the perspective of a certain subject with corresponding context information items and the subject's disclosure decision.

ID	Description	Example
MR	Encode disclosure modalities as label root elements.	`canvas.documents.x`
ML	Encode disclosure modalities as label leaf elements.	`contacts.bob.mobile`

Table 6.2: DiLES built-in preprocessors.

6.6.3 Preprocessors

Prior to their evaluation scenarios may be passed through one or more preprocessors. Preprocessors may filter or edit features (situation context) as well as disclosures. For instance DiLES provides preprocessors for modality encoding: disclosure modalities may be encoded in disclosures either as root or leaf elements of individual disclosed items. Currently these two preprocessor types are the only ones evaluated but as described in section B.5 (Extending DiLES) new preprocessors and their combinations can be added easily.

Preprocessing scenarios for modality encoding is particularly important if a disclosure situation provides *multiple* modalities to communicate information. Here the chosen modality actually is part of the disclosure. Generally it is not possible to decide if a chosen modality dictates the information items to disclose with this modality or if it is the other way around (an information item dictates the modality to use for its disclosure). The first case motivates to encode modalities as top-level elements in the hierarchical structure of information items while the latter case recommends to encode them as leaf elements. In order to evaluate which encoding generally (or per subject) performs better, both corresponding preprocessors have to be considered. For later reference (especially in evaluation results) these two basic preprocessors are summarized in table 6.2.

Further preprocessors may be used to alter, eliminate or join features. However, as this is a more general machine learning issue not directly related to disclosure prediction, such

ID	Method	Implementation
KNN	k-Nearest Neighbors	Own (based on Segaran, 2007)
NB	Naive Bayes	Own (based on Segaran, 2007)
SVM	Support Vector Machine	LIBSVM (Chang & Lin, 2011)
RL	Rule Induction	Orange, CN2 (Demšar *et al.*, 2004; Clark & Niblett, 1989)
DT	Decision Tree	Own (based on Roach, 2006)
GS	Guess Majority	Own (trivial)

Table 6.3: Base learner implementation details.

ID	Method	Implementation
PS	Powerset	Own (based on Tsoumakas *et al.*, 2010)
BR	Binary Relevance	Own (based on Tsoumakas *et al.*, 2010)
HBR	Hierarchical Binary Relevance	Own (based on Wu *et al.*, 2005)
OMI	Order-Mapping Interpolation	According to section 6.3

Table 6.4: Wrapping learner implementation details.

preprocessors are not evaluated at this point. The specific scenarios considered in this work already have rather high-level features which can be used directly. Additionally, basic feature selection is also part of a later step in the evaluation process – then with regard to a specific subject and learning algorithm. Anyway, as already mentioned new preprocessors can be added easily once the need for them arises.

6.6.4 Wrapping and Base Learners

Evaluating disclosure prediction performances of learning algorithms is the primary purpose of DiLES. Section 6.2 described several base and wrapping learning methods. Additionally section 6.3 presented a new pattern-based wrapping learner. Evaluating these methods means to evaluate all combinations of base and wrapping learners. The tables 6.3 and 6.4 provide implementation details for the specific learning methods used in DiLES. Similar to preprocessors DiLES can be extended to consider further algorithms. Details on this can be found in the appendix section B.5.

Parametrization

Some of the learning methods accept parameters which influence their behavior. Especially the Order-Mapping Interpolation (OMI) method has many parameters. Next to performance results, DiLES also shows the particular learner configurations used for spe-

cific evaluations. Table 6.5 summarizes the parameters of configurable learning method implementations.

6.6.5 Evaluation Tasks

An evaluation task is given by a specific combination of a preprocessor, base learner, and wrapping learner. As illustrated in figure 6.10 it involves feature selection and parameter optimization, followed by one iterative validation using the original situation sequence order of a scenario and a given number of iterative validations using randomized situation orders. Each iterative validation step is followed by an evaluation of a disclosure-pattern–based validator (as described in section 6.4).

Feature Selection and Parameter Optimization

DiLES uses an evolutionary approach to select features (in context of the task's specific combination of preprocessor, base, and wrapping learner) and to parametrize the used base and wrapping learner. In particular it uses a genetic algorithm with ranking, mutation, and crossover to find an optimal solution, i.e. a selection of features and parameter values. Obviously this requires that base and wrapping learners define a reasonable parameter space to search in. The fitness of a solution is evaluated by performing a cross-validation (type and parameters are configurable) on the available preprocessed scenario situations. However, when the size of the parameter and feature space is below a certain (configurable) limit, a complete grid search is used instead of the evolutionary approach. Details on the implementation can be found in the source of DiLES, specifically in the file `src/diles/learn/util.py` (see appendix B).

Iterative Validation

Typically learning methods are evaluated using cross validation. However, in face of the targeted application use case, an iterative validation is more expressive to evaluate the performance of a learning method. The rationale is that an iterative validation resembles the way in which learners are applied when actually used. That is, the number of situations which may be used to train a learner grows iteratively. A scenario with n situations results in $n-2$ iteration steps, where step i ($1 \leq i \leq n-1$) uses the first i situations for training and situation $i+1$ for testing. This setting may be varied in that i starts at a greater value than 1 and that multiple situations are used for testing, e.g. also $i+2$, $i+3$ etc. In contrast, a cross validation only estimates the performance when a certain number of situations to learn from is already available. Such an evaluation could bias performance

Learner & Parameters		*Description*
DT	`fitness`	Heuristic function used to decide which attribute to split on when building the tree. Example: gain
KNN	`k`	Number of neighbors to consider when voting for a label (i.e. disclosure). Example: 3
	`dwsigma`	The *sigma* value to use for a Gaussian function which weights votes based on a neighbor's distance. A greater sigma increases the weights of greater distances (vice versa, a smaller sigma reduces the influence of farther neighbors). Example: 2.0
NB	`iprob, iweight`	The initial probability of a feature-value-label combination and its weight. These parameters describe a default conditional probability based on *iweight* hypothetical previous samples (*Laplace* correction used for feature values in new situations which did not occur in training situations). Example: 0.5, 1.0
	`lpcm`	Laplace correction mode, either on a per-label basis or globally. Generally, per-label correction performs better when labels are balanced (uniform class distribution). In contrast, if label occurrences are unbalanced, a global Laplace correction tends to yield better predictions. Example: global
RL	The parameters for the Rule Induction (RL) learner directly correspond to those accepted by the underlying Orange system (Demšar *et al.*, 2004). They are used to adjust the process of finding and evaluating rules and to limit the size of a generated rule set. More detailed information can be found in the Orange documentation.	
SVM	The parameters for the SVM learner directly correspond to those accepted by the underlying LibSVM system (Chang & Lin, 2011). They are used to select and configure kernel and error calculation functions. More detailed information can be found in the LibSVM documentation.	
PS	The Powerset wrapper does not accept any parameters.	
BR	The Binary Relevance wrapper does not accept any parameters.	
HBR	The Hierarchical Binary Relevance wrapper does not accept any parameters.	
OMI	The OMI wrapper parameters are described in detail in section 6.3. Following are only short descriptions of the particular terms used in DILES evaluation results.	
	`variants`	Interpolation variants. Specifies if to only consider cases of joining situation chains, or also extending or creating situation chains, or if to stick to a simple global order-mapping-count–based interpolation (as described in section 6.3.5). Example: [ipolextend, ipolcreate]
	`minneighbors`	The minimum number of neighbors required for interpolation. Note that a high value of this parameter practically deactivates any interpolation. Example: 2
	`nearestonly`	Only consider the nearest neighbors for interpolation. Example: False
	`homogenize`	Homogenization strategies. Strategies and their combinations are listed in section 6.3.3 on page 91. Example: [droploosing, distance]

Table 6.5: Learning method configuration parameters. For specific possible values besides the given examples, please inspect the `paramspace` attribute in learning method implementations (more details in appendix B) respectively the evaluation results described in section 6.6.8.

results in that it favors methods which perform well on larger training sets but perform significantly worse on small training sets.

In any case, an iterative validation is performed using the scenario situations in their original order. Depending on the used scenario the order may not be relevant. In that case, a more expressive evaluation is to do multiple iterative validations, each with a randomized order. The number of of randomized orders can be specified as a command line argument when running DiLES's scenario evaluation command (see appendix B for details).

Finally, in each iteration step, a predicted disclosure is validated using the prediction validators described in section 6.6.5. The outcome of each validator is a simple binary value whether the prediction is supported or not.

Predicted disclosures, true disclosures (according to the scenario), prediction confidences, and validator results are collected for each iteration step. These values are wrapped by so called performance objects which provide various metrics on the values, in particular the metrics described in section 6.2.1 and 6.2.2. Performance objects may provide metrics for all predictions made for a specific situation order or for predictions made in a certain iteration among all situation orders. This allows to investigate the progress of learner performances when the training set grows as well as general performance results. The example evaluation results presented in section 6.6.8 illustrate these different performance perspectives.

Parallelization

The number of evaluation tasks is given by product of the number of subjects, preprocessor chains, base learners, and wrapping learners. Considering one subject only, the methods currently available in DiLES result in 48 evaluation tasks. Each additional subject multiplies this number. Depending on the number of situations in a scenario and the number of used randomized situation orders, a single task may require a few seconds up to several minutes. The latter case motivates to parallelize tasks in order to reduce the evaluation run-time. DiLES automatically runs evaluation tasks parallelized, one task for each available CPU core.

6.6.6 Analysis and Visualization

Using the result of an evaluation, DiLES renders various plots which provide different views on these results. Specifically, the following plots are available.

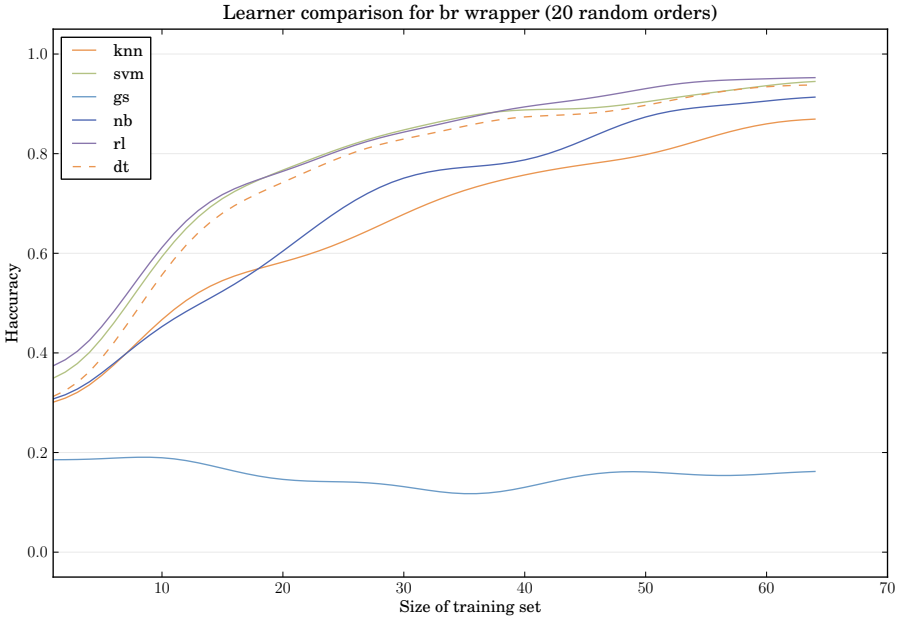

Figure 6.11: Example plot of DILES' evaluation result analysis. Progress of the metric *h-accuracy* for all base learners in combination with a Binary Relevance (BR) wrapper.

Progress Analysis

Several plots show how learning methods evolve over time. DILES provides comparisons of base and wrapping learners for specific metrics as well as a comparison of all metrics (accuracy, match, etc.) for a specific combination of preprocessor, base and wrapping learner. Figures 6.11 to 6.13 are examples of such progress view plots. Note that these plots show smoothed results (using a moving average) since not single iteration steps but general trends for increasing training set sizes are supposed to be visualized. In the given examples the y-values for each training size set (i.e. iteration step) are means from 20 randomized situation orders. If original single metric values are binary, e.g. in case of the *match* metric, the y-values represent ratios – this is indicated by an asterisk in the legend.

Global Analysis

Next to progress-oriented analysis DILES renders plots which summarize performance metrics globally by averaging metrics across all iteration steps (except the first 10, where each method is likely to fail similarly) and randomized situation orders. In particular

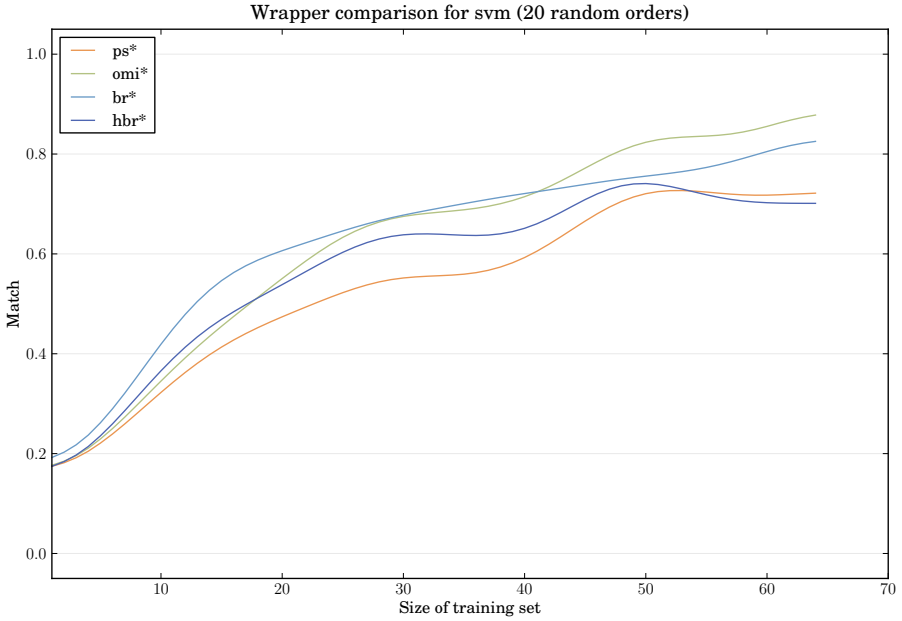

Figure 6.12: Example plot of DiLES' evaluation result analysis. Progress of the metric *match* for all wrapper methods in combination with a Support Vector Machine (SVM) learner.

there is one plot for each metric (see section 6.2) which globally compares combinations of preprocessors, base and wrapping learners. Further, there are global plots which compare prediction validators. Examples for such plots are given by figures 6.14 to 6.16.

Global metric comparison Figures 6.14 and 6.15 are examples for plots comparing combinations of preprocessor, base learner and wrapper with reference to specific performance metrics and subjects of the evaluated scenario. Results are illustrated using box and whisker plots. The shown values are based on predictions made in each step of the iterative evaluation process (i.e. it also includes predictions made when only a small training set has been used).

Validator performance Figure 6.16 is an example of a plot comparing the performance of prediction validators. Tables 6.6 and 6.7 list the prediction validators provided by DiLES and which are referenced in plots like figure 6.16. The green bars represent the ratio of positive (matching) predictions among all predictions made during the iterative evaluation process. In contrast the orange bars represent negative (mismatching) predictions. The lighter fraction of the orange bars represent the ratio of negative predictions prevented by a corresponding validator (combination) given in the x-axis. The lighter fractions of

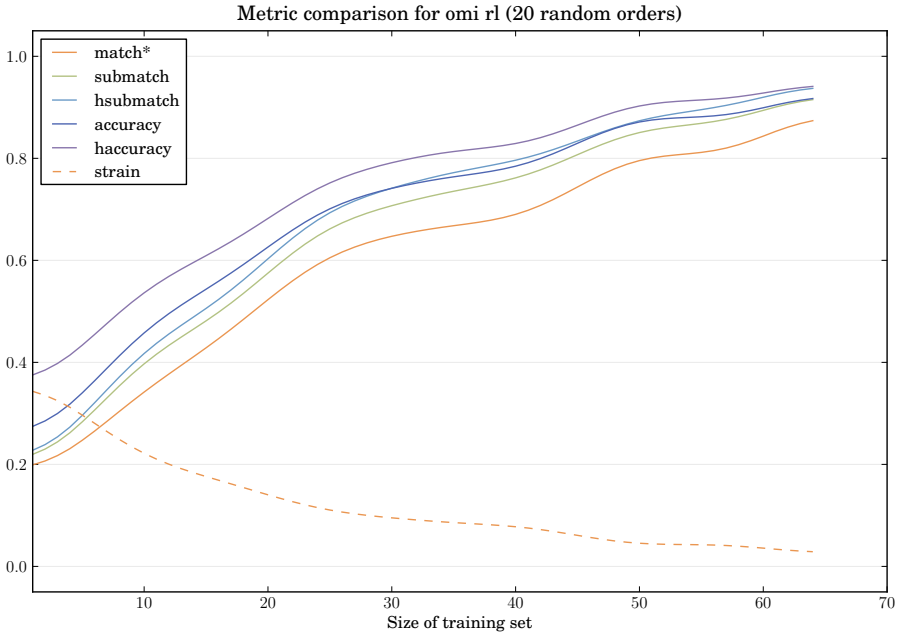

Figure 6.13: Example plot of DILES' evaluation result analysis. Progress of different metrics for an Order-Mapping Interpolation (OMI) wrapper in combination with a Rule Induction (RL) learner.

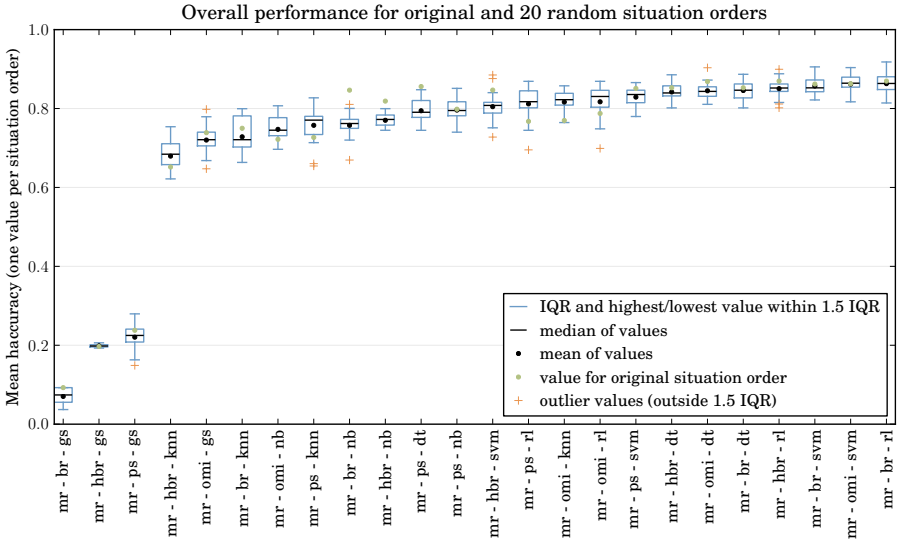

Figure 6.14: Example plot of DILES' evaluation result analysis. Global metric comparison (*h-accuracy*).

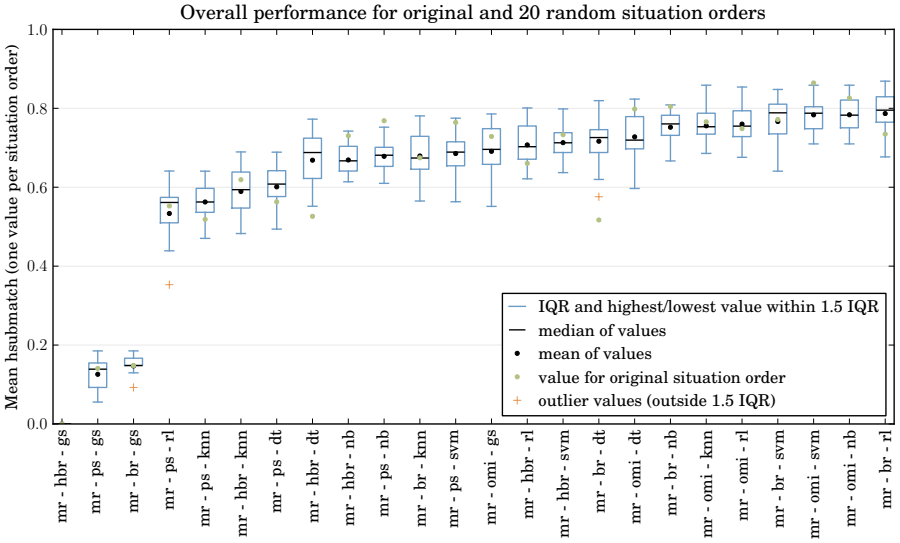

Figure 6.15: Example plot of DiLES' evaluation result analysis. Global metric comparison (*h-submatch*).

the green bars provide the ratios of positive predictions prevented by a validator. Ideally the green bar would be completely dark green (no prevented positive prediction) and the orange bar would be completely light orange (all negative predictions prevented). In the shown example, the validator *pwr25* prevents almost all negative predictions. However, it also prevents almost all positive predictions, i.e. it does not provide any useful contribution. On the other side, the validator combination *uc1 + cmdp* prevents almost every second negative prediction while only a few positive predictions are prevented.

Runtime Performance

Though runtime statistics of a learning mechanism are not a major issue it is still interesting to investigate how different mechanisms compare to each other in terms of computational resources for parameter optimization, training, and prediction. Similar to progress and global analysis, DiLES illustrates this information on a per subject basis. Figure 6.17 is a corresponding example. Here the training times from all iteration steps are shown for each combination of a preprocessor, wrapper and base learner. Note that the absolute values are not very meaningful since they mostly depend on implementation details. Instead, these plots are supposed to show relative differences among wrapper and learner combinations. The displayed example shows that the Binary Relevance (BR) and Hierarchical Binary Relevance (HBR) wrappers in combination with a Decision Tree (DT),

Validators based on disclosure complexity	
`pw1`	Poset width of previous disclosures is at most 1.
`pw2`	Poset width of previous disclosures is at most 2.
`pwr25`	Ratio of poset width of previous disclosures to the number of unique previous disclosures is at most 0.25.
`pwr50`	Same as `pwr25` but with a maximum ratio of 0.5.
`pwr75`	Same as `pwr25` but with a maximum ratio of 0.75.
`udr25`	Ratio of the number of unique previous disclosures to the number of previous situations is at most 0.25.
`udr50`	Same as `udr25` but with a maximum ratio of 0.5.
`udr75`	Same as `udr25` but with a maximum ratio of 0.75.
Validators based on disclosure paths	
`cmdp`	Predicted disclosure is connected to a major disclosure path (length is at least the mean of all disclosure paths).
`csdp`	Predicted disclosure is connected to a super disclosure path (length is at least the 3rd quartile of all disclosure paths).
`cxdp`	Predicted disclosure is connected to an existing disclosure path.
`imdp`	Predicted disclosure is integrated in a major disclosure path (length is at least the mean of all disclosure paths).
`isdp`	Predicted disclosure is integrated in a super disclosure path (length is at least the 3rd quartile of all disclosure paths).
`ixdp`	Predicted disclosure is integrated in an existing disclosure path.
Validators based on order mapping	
`om50`	Situation to predict joins existing situation chains and the predicted disclosure harmonizes with at least 50% of the order mappings among the neighbor situations.
`om75`	Same as `om50` but requires 75% harmonizing order mappings.
`om100`	Same as `om50` but requires 100% harmonizing order mappings.
`omx50`	Same as `om50` but also consider the case of extending situation chains.
`omx75`	Same as `omx50` but requires 75% harmonizing order mappings.
`omx100`	Same as `omx50` but requires 100% harmonizing order mappings.
`omxc50`	Same as `omx50` but also consider the case of creating new situation chains and then check if the new order mapping is one that globally occurs most often.
`omxc75`	Same as `omxc50` but requires 75% harmonizing order mappings.
`omxc100`	Same as `omxc75` but requires 100% harmonizing order mappings.
Validators based on usage counts	
`uc1`	Absolute usage count of predicted disclosure is at least 1.
`uc2`	Absolute usage count of predicted disclosure is at least 2.
`ucm`	Predicted disclosure is a major disclosure (usage count is at least the mean of all disclosure usage counts).
`ucs`	Predicted disclosure is a super disclosure (usage count is at least the 3rd quartile of all disclosure usage counts).

Table 6.6: Validators and their support conditions. The validators listed in this table are specific implementations of the concepts described in section 6.4.

Validators based on prediction confidences or hierarchical information	
`minconf`	Confidence is above a static minimum confidence.
`dynconf`	Confidence is above a dynamic (evolving) confidence threshold.
`hierarchy`	All predicted information elements are leafs within the hierarchically structured set of information items (only applies when using the Hierarchical Binary Relevance (HBR) wrapper).

Table 6.7: Validators and their support conditions. The validators listed in this table are specific implementations of the concepts described in section 6.2.2 and 6.2.3.

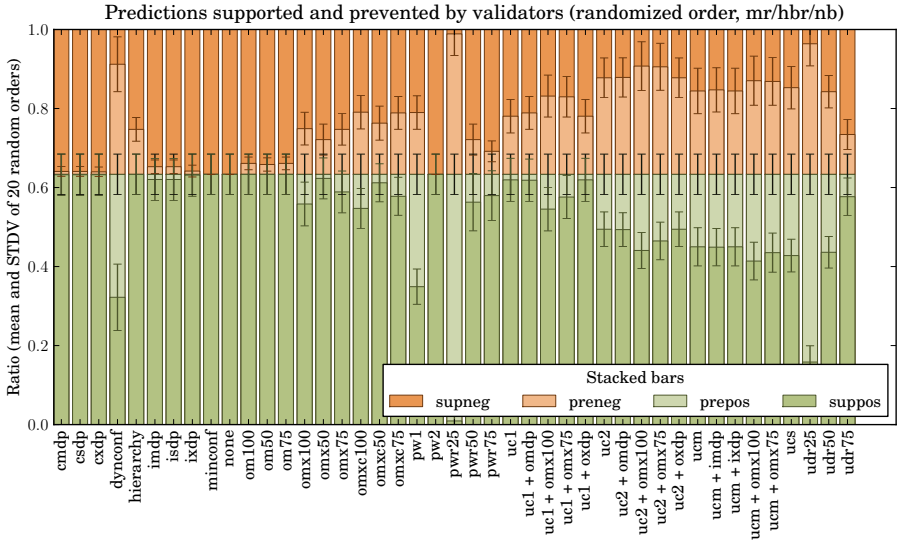

Figure 6.16: Example plot of DiLES' evaluation result analysis. This plot shows the performance of prediction validators for a specific combination of preprocessor, wrapper and learner.

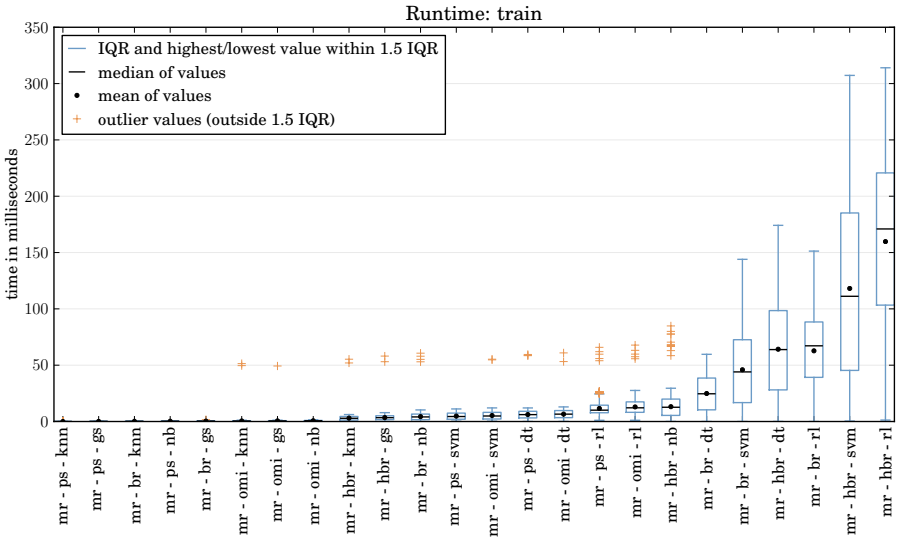

Figure 6.17: Example plot of DiLES' evaluation result analysis. Training time performance for different combinations of wrappers and base learners.

Rule Induction (RL) or Support Vector Machine (SVM) learner require significantly more training time than other combinations.

Scenario Analysis

In order to correlate evaluation results with patterns of information disclosure as described in section 4.1, scenarios are analyzed according to these patterns and then illustrated in summarizing plots (one per subject occurring in a scenario). Figure 6.18 shows two examples of scenario analysis plots. The subject related to the left plot appears to have a more complex disclosure behavior than the subject related to the right plot. On the left side, the number of unique disclosures is greater. Additionally, the right subject often disclosed the same information. Also, order mappings are rather homogeneous on the right side – in almost all cases they were either order-reversing or order-ignoring. Such information helps in reasoning about different performance results among subjects of a scenario.

Plot Compositions

Since there may be a high number of plots resulting from an analyzed scenario, viewing plots individually is cumbersome and not very helpful. For this purpose DiLES arranges plots in interactive HTML-based combined views. These views allow to selectively compare

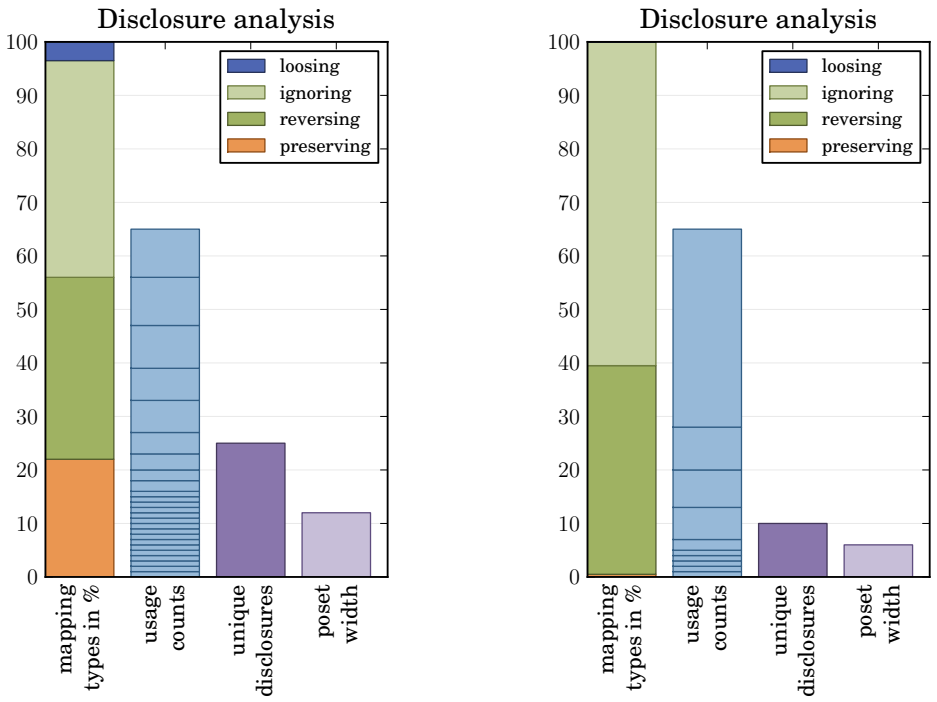

Figure 6.18: Example plot of DiLES' evaluation result analysis. Scenario analysis.

different plots by filtering by subject, preprocessor, base learner, wrapping learner, and metric. Next to result plots the composition view also shows (a) the scenario used for the evaluation, (b) the particular learner and wrapper configurations used (as a result of the afore mentioned parameter optimization), and (c) noteworthy statistics of a learner across the whole iterative evaluation (e.g. which interpolation method has been used how often by the OMI wrapper). Figures 6.19 to 6.20 illustrate these plot compositions.

6.6.7 Scenarios

Getting data, i.e. scenarios, which may be used to evaluate disclosure prediction is a problematic issue. The targeted application area of smart environments and corresponding use cases including versatile information exchange are still in development and not yet used on a daily basis in productive settings. To alleviate this lack of real world data the evaluation has been done with a manually composed scenario and a scenarios derived from the data gathered in the survey described in section 4.3. Another promising approach is to automatically generate scenarios based on specific patterns of disclosure behavior.

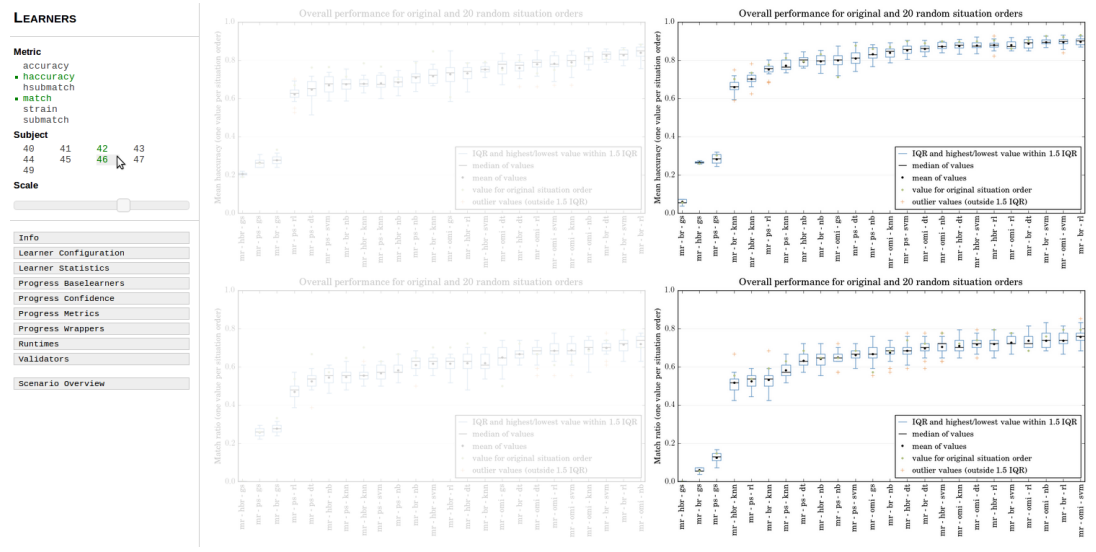

Figure 6.19: Screenshot of DiLES plot compositions. This example shows rankings of preprocessor, wrapper and learner combinations for selected metrics and subjects. Plots to show are selected in the parameter sidebar on the left. Green parameter values indicate that a corresponding plot is shown. To identify the plots for specific parameter values, the mouse may be moved over a corresponding value to fade out all non-matching plots (as shown in this example screenshot).

Manually Composed Scenario

The manually composed scenario is about Bob, a web developer working as a freelancer. Sue contacts Bob and requests him to develop and set up a customized WCMS for her company. The scenario describes several meetings which take placing during the lifetime of this project. A detailed description of this scenario including all situations can be found in appendix C.

Scenarios Derived from DiHabs Survey

The data of the conducted DiHabs survey (see section 4.3) has been converted to be used as a DiLES scenario. Here the type of information has been used to set up a hierarchy on disclosed information as well as a label-based context information besides the persons which are supposed to retrieve disclosed information. The following excerpt of the scenario file illustrates the conversion result:

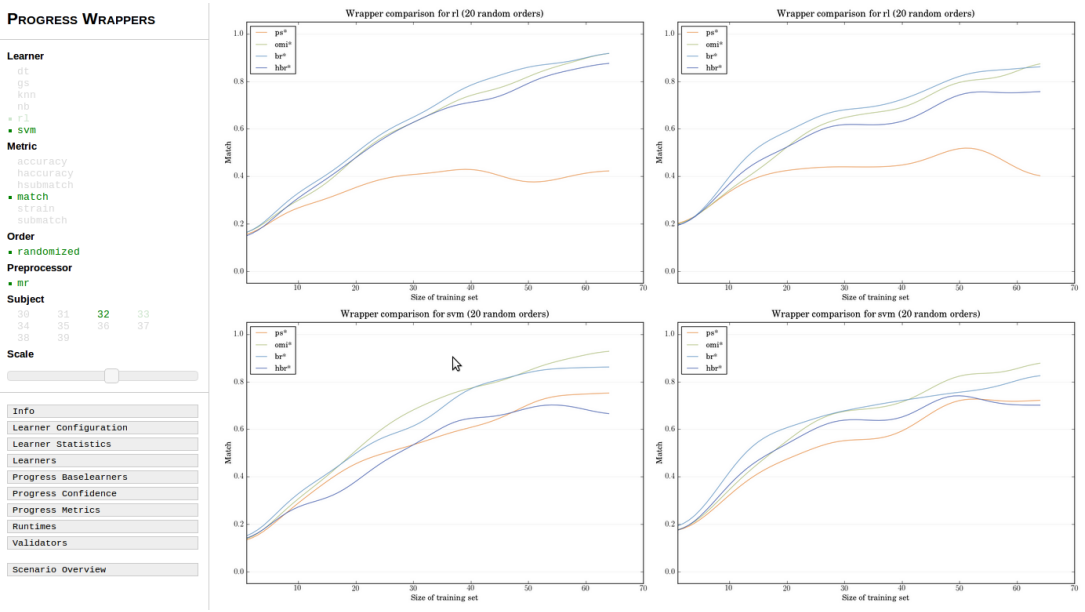

Figure 6.20: Screenshot of DILES plot compositions. This example shows plots which compare the progress of different performance metrics (during the iterative evaluation process) for the used wrapper methods. If the mouse pointer is moved over a specific plot, the corresponding parameters in the sidebar are highlighted.

```
- disclosure: ['x:type-2.item-0', 'x:type-2.item-1']
  labels: ['2']
  persons: ['1', '2']
  subject: '3'
- disclosure: ['x:type-3.item-5', 'x:type-3.item-6']
  labels: ['3']
  persons: ['2', '3', '4', '6']
  subject: '3'
```

This shows two situations which correspond to two answers made by the survey participant number 3. In the first situations persons 1 and 2 where part of the situation and thus information receiver. The situation is annotated with a label `'2'` which indicates the situation type, i.e. information type asked for in the interview. The information type also contributes the hierarchical structure of disclosed information items. Since the conducted DIHABS survey did not consider different disclosure modalities for the same types of information, all disclosures have the same modality (here encoded as an `'x'`). Because the interview data has been anonymized, this scenario has no meaningful values but plain numbers. Nevertheless it expresses survey-based disclosure behavior. See appendix C for more details on this scenario.

Generated Scenarios

Automatically generated scenarios bear the risk of being not representative concerning real world situations (even more than manually composed scenarios). On the other side, they allow to express scenarios where subjects follow specific patterns of disclosure behavior. This makes it possible to evaluate learning mechanisms with regard to explicit disclosure patterns. Another advantage of generated scenarios is the comparatively high amount of evaluation data provided by them. In this work this approach is not investigated further but seen as a potential future work. However, a starting point for work in this area is provided in Bünnig (2009a).

6.6.8 Results

As anticipated, evaluation results of the investigated scenarios differ widely among the subjects of a scenario. For instance, there is no base learner and no wrapper method which performs best for most subjects. The individuality of results per subject are illustrated in the following subsection. Nevertheless, overall rankings of base learners, wrappers and their combinations indicate which methods to focus on in future research or when working on disclosure assistance implementations. Such rankings are provided by the next but one sub section.

Results per Subject

Presenting the evaluation result for the disclosure decisions by each subject involved in the evaluated scenarios would exceed the space available in this work. Instead the complete results are available in an open science repository at `http://opsci.informatik.uni-rostock.de/index.php/DiLES`– arranged within the afore mentioned HTML-based interactive plot compositions. As a reference, here only the results for selected subjects of the scenario generated from the DiHabs survey results are shown. In particular the results are investigated with reference to the questions listed in the introduction of section 6.6:

- How do different learning methods compare to each other?
- How do different validation methods compare to each other?
- Which correlations exist between results and disclosure patterns?
- Which learner configurations perform best?

Performance of wrappers and base learners Figures 6.21 to 6.24 show the iterative progress of the *match* metric for all learners, each in combination with a specific wrapper.

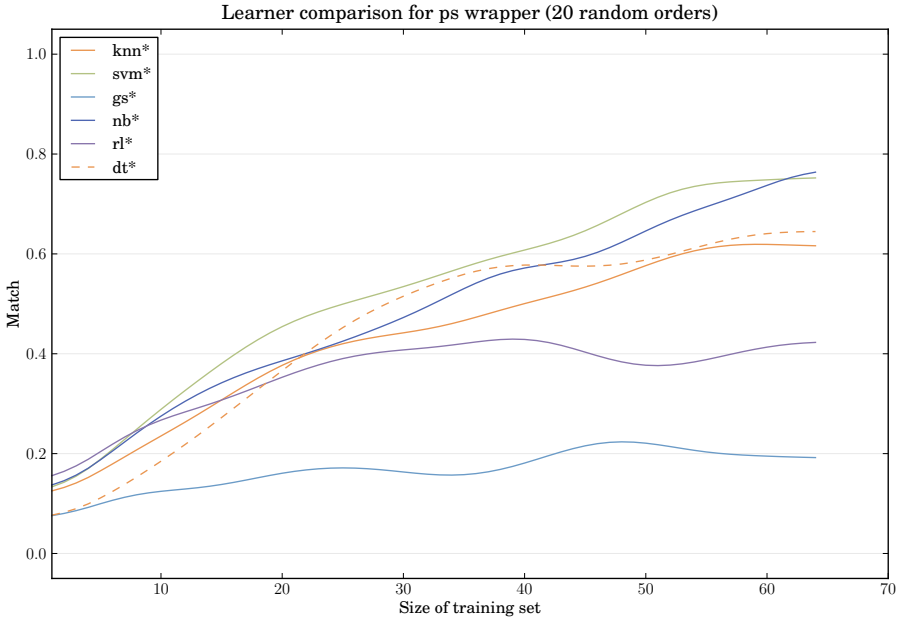

Figure 6.21: Progress evaluation results of base learners with a PS wrapper (scenario: *dihabs*, subject: *32*).

That is each single plot allows to compare base learners, given a wrapper method. All plots together allow to compare impacts of wrapper methods on base learners. The given examples illustrate several interesting effects. First, the performance of a base leaner may vary significantly depending on the used wrapper. For instance the RL learner performs best when used with a BR or HBR wrapper but drops to second worst when used with Powerset (PS) wrapper. The OMI wrapper on the other side diminishes the differences of base learners – even the Guess Majority (GS) learner performs quite well.

In fact the OMI-wrapper is the best performing in this case. This is backed up by figure 6.25 which shows a global (i.e. non-iterative) comparison of wrapper and base learner combinations. However, other metrics may yield slightly different rankings as shown in figure 6.26.

Figures 6.27 to 6.30 show the progress evaluation results for subject 27. Here the wrapper methods have less impact on the performance of base learners. Also, the OMI wrapper performs worse than for subject 32. However, it is not significantly worse than the other wrappers. This is illustrated in figure 6.31.

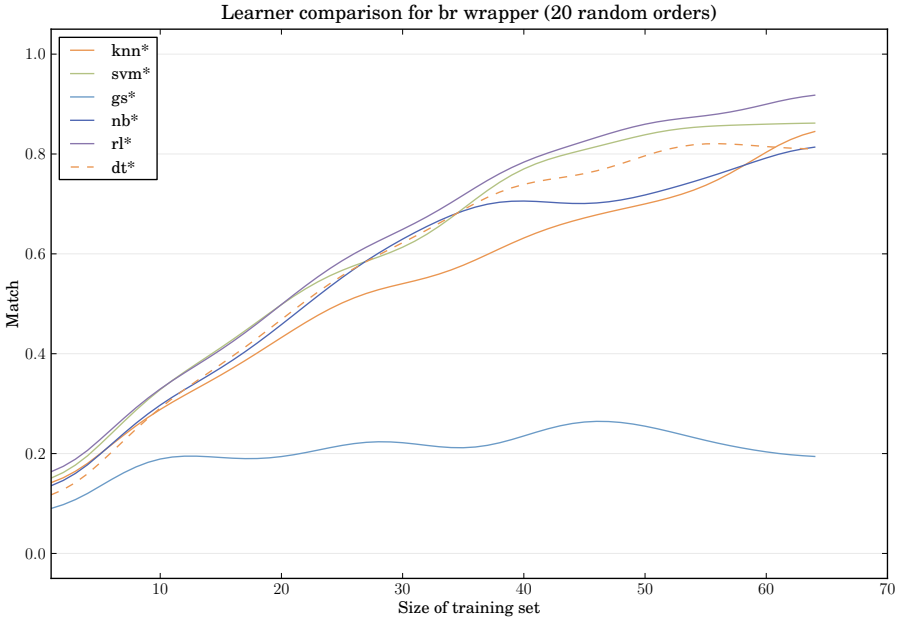

Figure 6.22: Progress evaluation results of base learners with a BR wrapper (scenario: *dihabs*, subject: *32*).

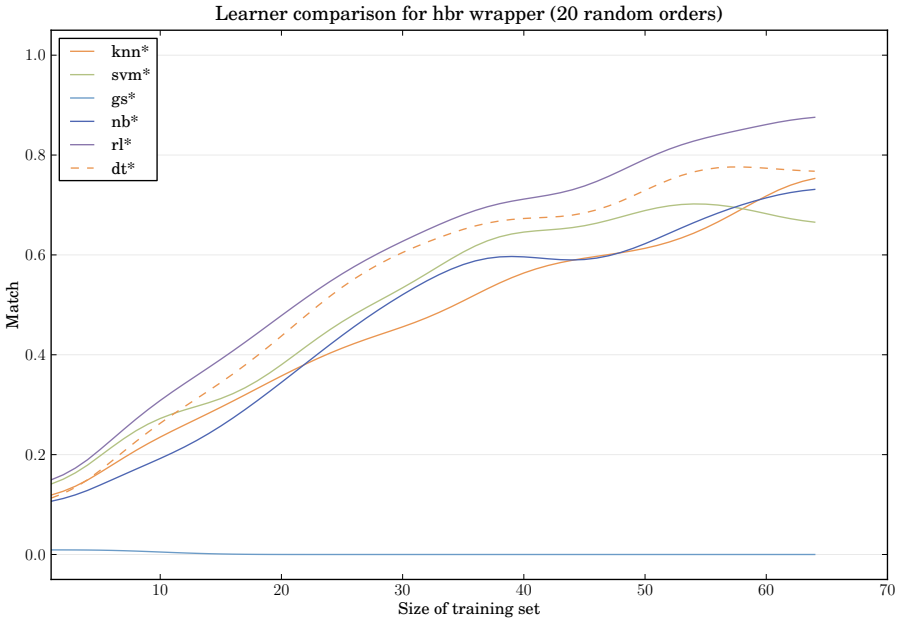

Figure 6.23: Progress evaluation results of base learners with an HBR wrapper (scenario: *dihabs*, subject: *32*).

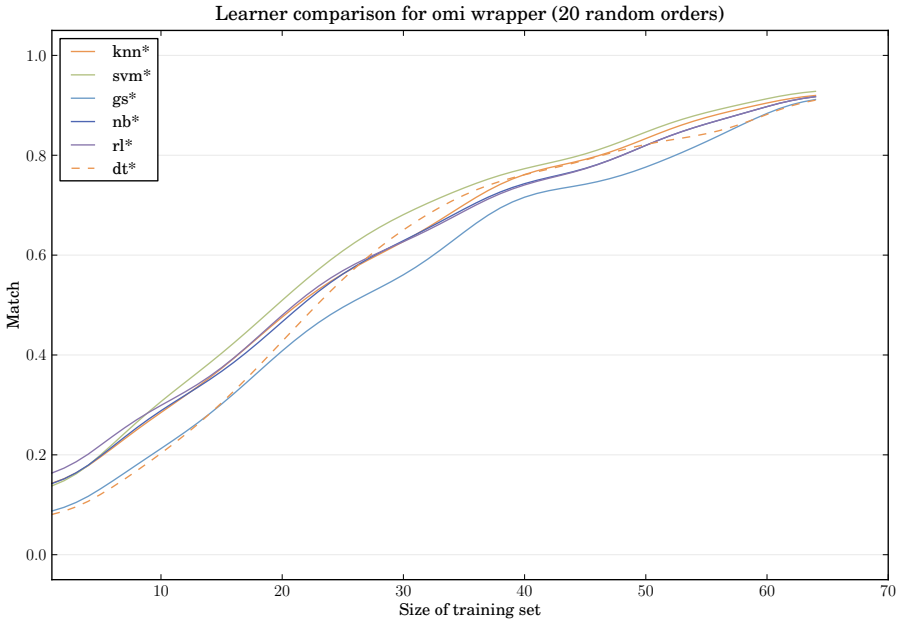

Figure 6.24: Progress evaluation results of base learners with an OMI wrapper (scenario: *dihabs*, subject: *32*).

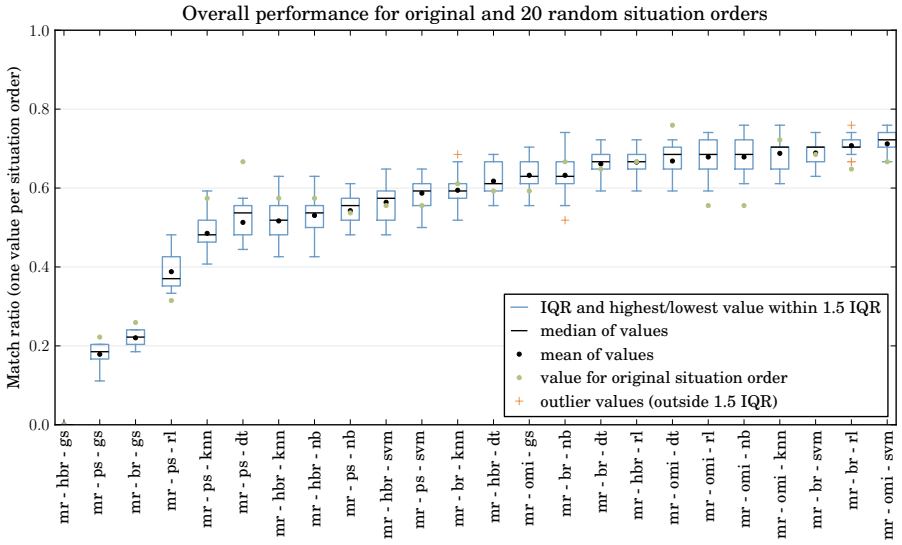

Figure 6.25: Global evaluation results of all learner and wrapper combinations according to metric *match* (scenario: *dihabs*, subject: *32*).

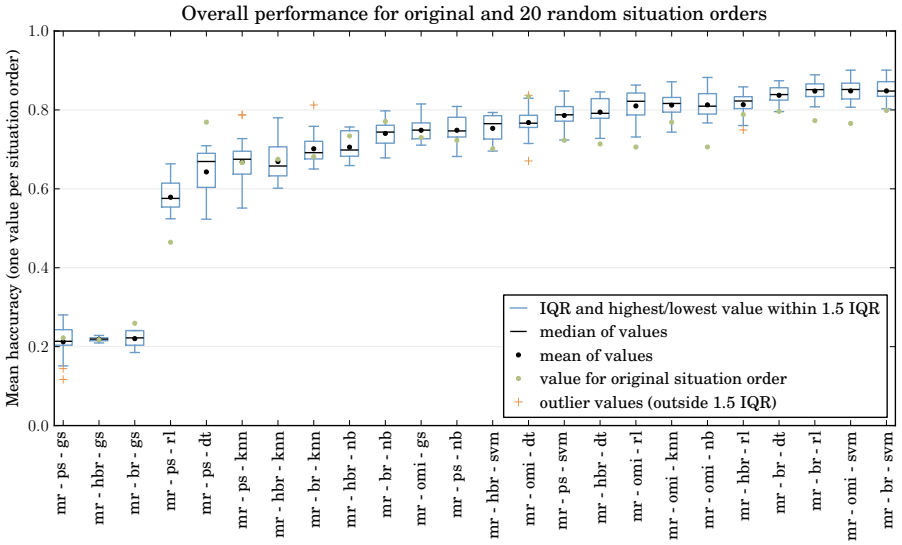

Figure 6.26: Global evaluation results of all learner and wrapper combinations according to metric *h-accuracy* (scenario: *dihabs*, subject: *32*).

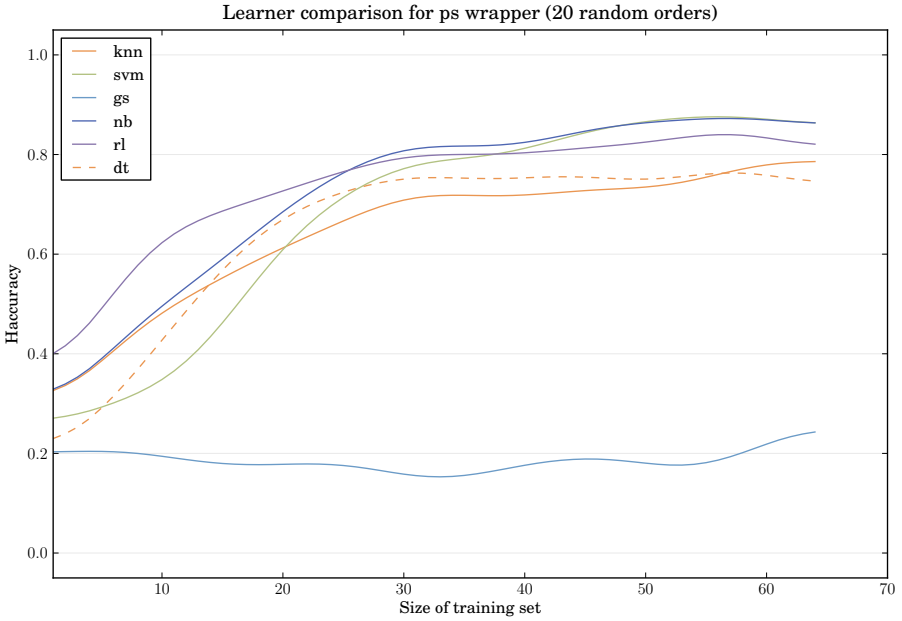

Figure 6.27: Progress evaluation results of base learners with a PS wrapper (scenario: *dihabs*, subject: *27*).

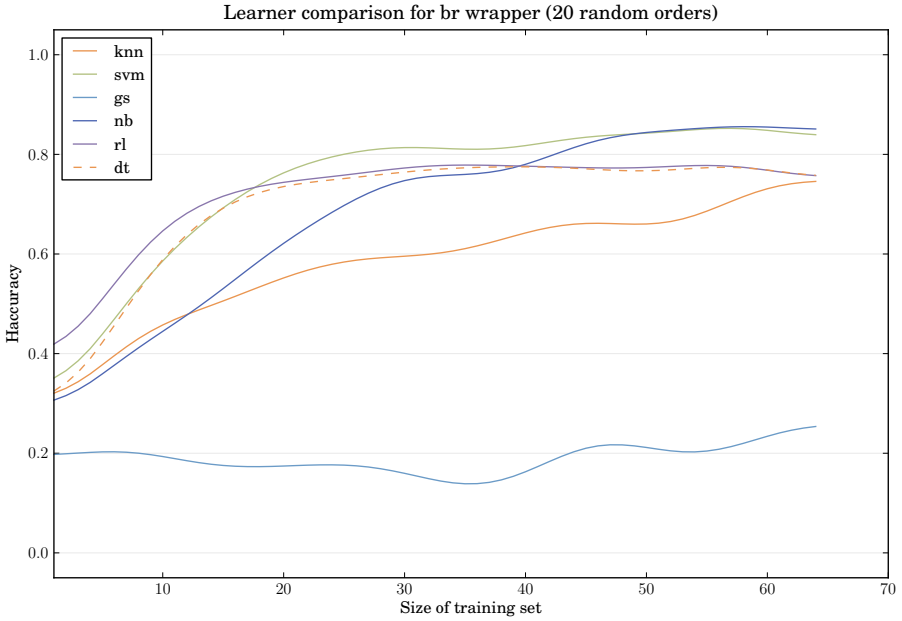

Figure 6.28: Progress evaluation results of base learners with a BR wrapper (scenario: *dihabs*, subject: *27*).

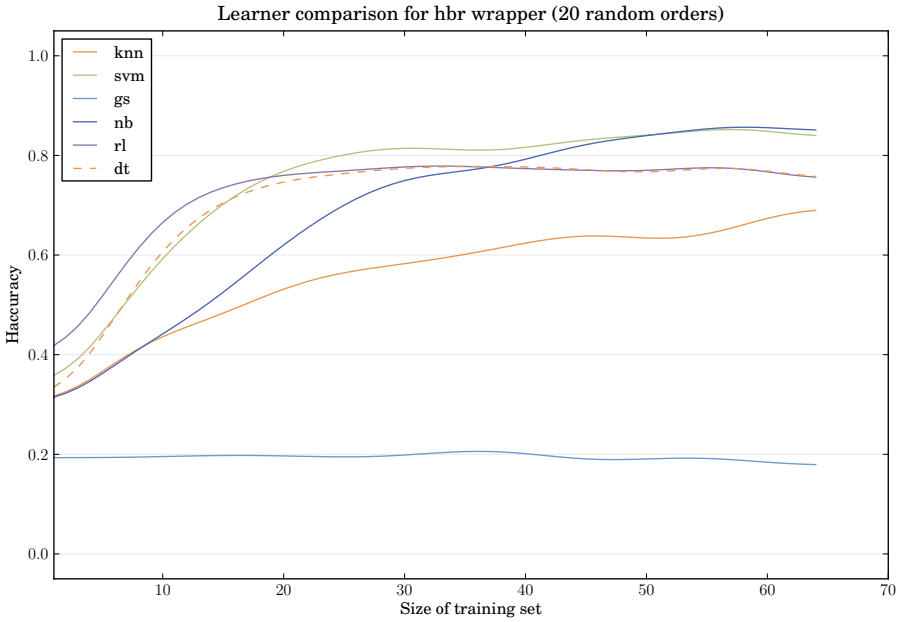

Figure 6.29: Progress evaluation results of base learners with an HBR wrapper (scenario: *dihabs*, subject: *27*).

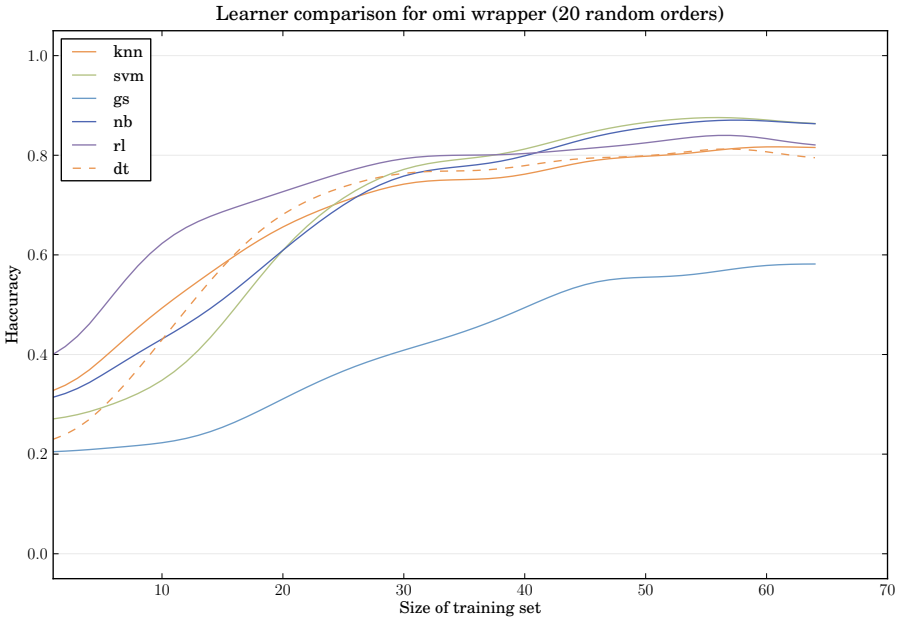

Figure 6.30: Progress evaluation results of base learners with an OMI wrapper (scenario: *dihabs*, subject: *27*).

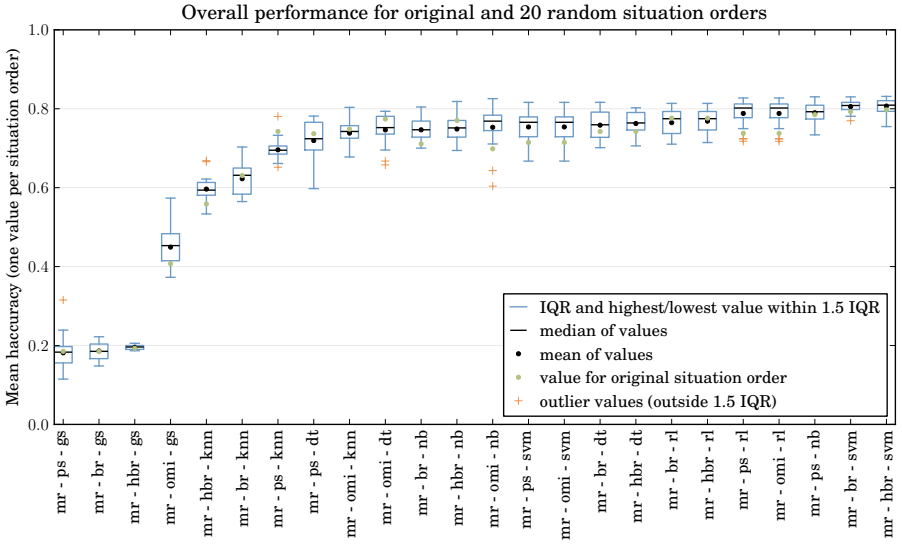

Figure 6.31: Global evaluation results of all learner and wrapper combinations according to metric *h-accuracy* (scenario: *dihabs*, subject: *27*).

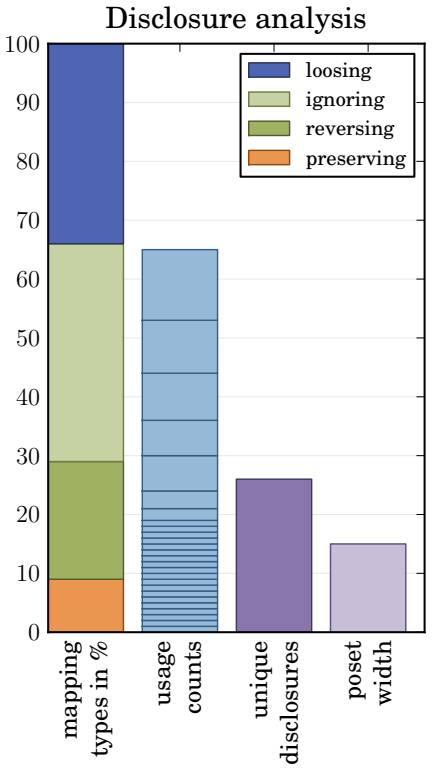

Figure 6.32: Subject 27.

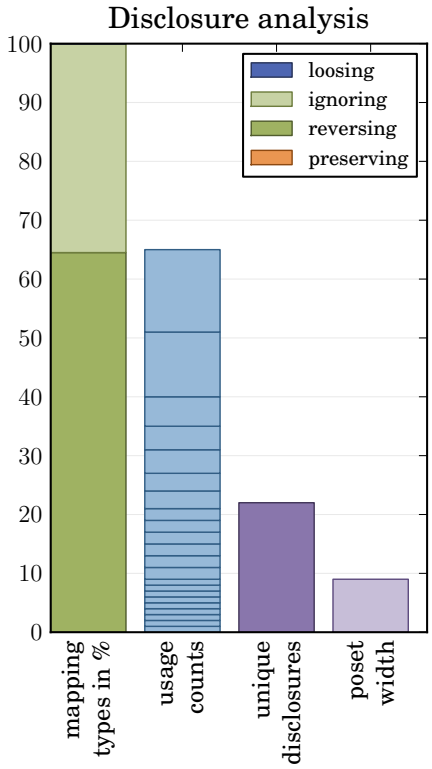

Figure 6.33: Subject 32.

Correlation with disclosure patterns The scenario (respectively disclosure) analysis plots provided by DiLES help in reasoning about the differences of performance metrics among subjects. Figures 6.32 and 6.33 show the scenario analysis plots for subject 32 and 27. The disclosure behavior of subject 32 shows that order mappings either are order-ignoring or order-reversing. In contrast, the order mappings for subject 27 are more diversified. Additionally the poset width is greater. These characteristics explain why the OMI wrapper does not provide much contribution in this case. Also, the greater number of unique disclosures from subject 27 explains the general worse performance of wrappers and base learners compared to subject 32.

Validators Next to learners DiLES also evaluates prediction validators. Figures 6.34 and 6.35 show the validator evaluation results for both subjects (for the OMI and PS wrapper, which had the best *match* performance for these subjects). The general performance metrics for subject 27 are worse than for subject 32, but validators still are able to prevent a significant portion of negative predictions. Usage-count–based validators in combination with order-mapping–based validators perform best in this case, i.e. they prevent

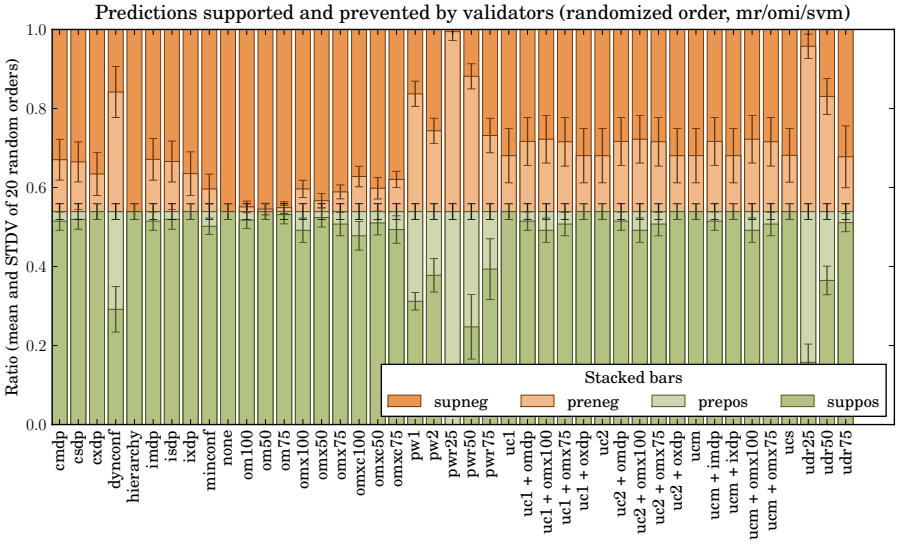

Figure 6.34: Prediction validator evaluation results in context of an SVM learner used with an OMI wrapper (scenario: *dihabs*, subject: *27*)

more negative predictions while supporting more positive predictions than others. The situation is different for subject 32. Here the OMI wrapper already contributed to positive predictions, which is the reason why order-mapping–based validators do not provide additional contributions. Better candidates are the confidence based validators `minconf` and `dynconf`. Which one is better depends on whether supported positive or prevented negative predictions are more important for the corresponding subject.

Learner and wrapper configurations The configurations used for wrappers and base learners during DiLES' evaluation process are determined using cross-validation based evolutionary parameter optimization. Inspecting these configurations is especially interesting for the OMI wrapper (because it has a reasonable amount of parameters and because it is a new wrapper method introduced by this work). Figure 6.36 shows the configuration summaries provided by DiLES. These configurations also explains why the OMI wrapper does not provide much contribution in case of subject 27. On subject 27 the OMI wrapper has been configured to only consider the neighbors with a minimal distance. In contrast, the OMI wrapper used for subject 32 has been configured to involve all neighbor situations when interpolating disclosures. Hence, in case of subject 27 there a less situations when the minimal required number of neighbor situations is available, which deactivates the OMI wrapper more often (see section 6.3.7 for details on this parameter optimization based behavior of the OMI wrapper).

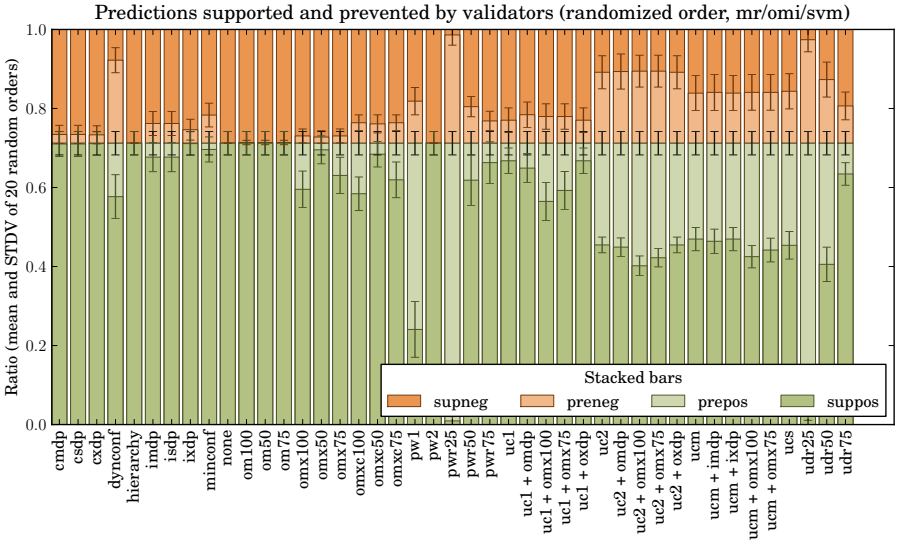

Figure 6.35: Prediction validator evaluation results in context of a SVM learner used with an OMI wrapper (scenario: *dihabs*, subject: *32*)

mr - omi - svm - 27
c: 512
g: 3.0517578125e-05
homogenize: ('droploosing',)
minneighbors: 2
nearestonly: True
t: 2
variants: ('globalom',)

mr - omi - svm - 32
c: 512
g: 0.001953125
homogenize: ('distance',)
minneighbors: 2
nearestonly: False
t: 2
variants: ('globalom',)

Figure 6.36: Combined learner and wrapper configuration used for the OMI wrapped SVM learner for subject 27 and subject 32 (scenario: *dihabs*).

Overall Rankings

The results above presented relate to specific subjects of a scenario. The reason for this is that disclosure behavior is rather individual, i.e. an automatic disclosure mechanism primarily needs to be evaluated in context of one subject. However, it is still interesting if certain learners, wrappers, validators and configurations generally tend to perform better than others. This references the introductorily stated question: "Are there generally well-performing methods, configurations, or validators?". Tables 6.8 to 6.12 list such general rankings for an evaluation using the scenario generated from the DiHabs survey.

Learner and wrapper rankings The learner and wrapper rankings are based on the means of corresponding metrics: the displayed values are the means of all per-subject-means (e.g. the values in table 6.10 are based on mean values as displayed in figure 6.14). Note that the metrics are based on an iterative validation, i.e. it also includes performance results from predictions when only a small set of disclosure situations have been used to train a learner respectively wrapper.

match	
SVM	0.59 0.02
RL	0.57 0.02
DT	0.56 0.02
NB	0.56 0.02
KNN	0.50 0.02
GS	0.20 0.03

h-accuracy	
SVM	0.81 0.01
RL	0.80 0.01
DT	0.79 0.01
NB	0.77 0.01
KNN	0.70 0.01
GS	0.32 0.03

h-submatch	
SVM	0.69 0.01
RL	0.67 0.02
NB	0.66 0.01
DT	0.65 0.02
KNN	0.60 0.01
GS	0.22 0.03

strain	
SVM	0.11 0.01
RL	0.12 0.01
DT	0.13 0.01
NB	0.13 0.01
KNN	0.17 0.01
GS	0.39 0.02

Table 6.8: Overall ranking of base learners for the DiHabs scenario evaluation. Each table corresponds to a certain performance metric and lists the learner ID as well as the metric's mean value and error margin for a 0.95 confidence interval.

match	
OMI	0.57 0.01
PS	0.49 0.02
BR	0.48 0.02
HBR	0.46 0.02

h-accuracy	
OMI	0.77 0.01
HBR	0.68 0.02
PS	0.68 0.02
BR	0.66 0.03

h-submatch	
OMI	0.66 0.01
PS	0.56 0.02
BR	0.56 0.02
HBR	0.54 0.03

strain	
OMI	0.15 0.01
HBR	0.16 0.01
BR	0.18 0.01
PS	0.22 0.02

Table 6.9: Overall ranking of wrappers for the DiHabs scenario evaluation. Each table corresponds to a certain performance metric and lists the wrapper ID as well as the metric's mean value and error margin for a 0.95 confidence interval.

Scoring of validators: The validators are ranked using weighted sums of scores. For each combination of a base learner, a wrapper and a subject, validators first get scored by a weighted sum of prevented negative and supported positive predictions. The weights simply express if prevented negative or supported positive predictions are more important. The initial score for a validator v is calculated as follows:

$$w_{neg}v_{preneg} + w_{pos}v_{suppos}$$

where v_{preneg} is the mean of prevented positive predictions and v_{suppos} is the mean of supported positive predictions (the mean of all iteration steps from the iterative validations on the original and x randomized situation orders). Then all validators are assigned a relative score within $[0, 1]$ which is given by:

$$s_v / s_{best}$$

where s_v is the initial absolute score of a validator and s_{best} is the best initial absolute score of all validators. For each combination of subject, base learner and wrapper these scores are summed up and finally normalized to be again within $[0, 1]$. A validator which

match		h-accuracy		h-submatch		strain	
OMI-SVM	0.61 0.03	BR-RL	0.82 0.02	OMI-SVM	0.70 0.03	HBR-SVM	0.10 0.01
OMI-RL	0.59 0.04	HBR-RL	0.82 0.02	HBR-SVM	0.69 0.03	HBR-RL	0.10 0.01
PS-SVM	0.59 0.03	HBR-SVM	0.82 0.02	BR-SVM	0.69 0.03	BR-RL	0.10 0.01
OMI-NB	0.59 0.03	HBR-DT	0.82 0.02	OMI-RL	0.69 0.03	HBR-DT	0.10 0.01
HBR-SVM	0.59 0.03	BR-DT	0.81 0.02	PS-SVM	0.68 0.03	BR-DT	0.11 0.01
BR-SVM	0.58 0.03	OMI-SVM	0.81 0.02	OMI-NB	0.68 0.03	OMI-SVM	0.11 0.01
BR-RL	0.58 0.04	OMI-RL	0.80 0.02	BR-RL	0.68 0.03	BR-SVM	0.11 0.01
HBR-RL	0.58 0.04	PS-SVM	0.80 0.02	HBR-RL	0.68 0.03	PS-SVM	0.12 0.01
PS-NB	0.57 0.03	BR-SVM	0.79 0.03	PS-NB	0.67 0.03	HBR-NB	0.12 0.01
OMI-DT	0.57 0.04	OMI-NB	0.78 0.02	HBR-DT	0.66 0.03	OMI-RL	0.12 0.02
HBR-DT	0.57 0.03	OMI-DT	0.78 0.02	BR-DT	0.66 0.03	OMI-NB	0.13 0.01
BR-DT	0.56 0.04	PS-NB	0.77 0.02	OMI-DT	0.66 0.03	BR-NB	0.13 0.01
OMI-KNN	0.56 0.03	OMI-KNN	0.77 0.02	BR-NB	0.65 0.03	PS-NB	0.14 0.01
BR-NB	0.55 0.03	PS-DT	0.76 0.03	OMI-KNN	0.65 0.03	OMI-DT	0.14 0.02
PS-DT	0.54 0.04	PS-RL	0.76 0.03	PS-RL	0.65 0.03	PS-RL	0.15 0.02
PS-RL	0.54 0.04	HBR-NB	0.76 0.02	HBR-NB	0.63 0.03	OMI-KNN	0.15 0.02
HBR-NB	0.53 0.03	BR-NB	0.75 0.02	PS-DT	0.63 0.03	PS-DT	0.15 0.02
OMI-GS	0.51 0.04	PS-KNN	0.71 0.02	PS-KNN	0.59 0.03	HBR-KNN	0.17 0.01
PS-KNN	0.50 0.03	OMI-GS	0.69 0.03	OMI-GS	0.59 0.03	BR-KNN	0.18 0.01
BR-KNN	0.48 0.03	HBR-KNN	0.68 0.02	BR-KNN	0.58 0.02	PS-KNN	0.19 0.01
HBR-KNN	0.46 0.03	BR-KNN	0.66 0.02	HBR-KNN	0.56 0.02	OMI-GS	0.21 0.02
PS-GS	0.15 0.03	PS-GS	0.25 0.02	PS-GS	0.17 0.03	HBR-GS	0.36 0.03
BR-GS	0.11 0.04	HBR-GS	0.22 0.02	BR-GS	0.11 0.04	BR-GS	0.44 0.03
HBR-GS	0.02 0.02	BR-GS	0.11 0.04	HBR-GS	0.02 0.02	PS-GS	0.54 0.05

Table 6.10: Overall ranking of wrapper and base learner combinations for the DiHabs scenario evaluation. Each table corresponds to a certain performance metric and lists the wrapper and learner ID as well as the metric's mean value and error margin for a 0.95 confidence interval.

is always the best one, for each combination of subject, base learner and wrapper, would then have a score of 1. Individual score values provide not much insights, however, they are suitable to compare different validators.

Runtime rankings The runtime rankings shown in table 6.12 are supposed to illustrate runtime differences among the base learners and wrappers evaluated here. Note that individual values do not provide much insights because they vary when the evaluation is run on different platforms and because they are implementation-dependent. However, they allow to compare runtimes among wrappers or learners. Comparing runtimes of wrappers for specific base learners is more robust than vice versa. This is because all wrappers are own implementations in Python while the base learner implementations partly are implemented in C and also come from different sources.

$w_{neg} = 2$, $w_{pos} = 1$	
uc2 + omx100	0.94
uc2 + omx75	0.94
uc2 + omdp	0.90
ucm + omx100	0.90
ucm + omx75	0.90
uc2	0.90
uc2 + oxdp	0.90
udr25	0.90
dynconf	0.88
ucm + imdp	0.87
ucm	0.86
ucm + ixdp	0.86
ucs	0.84
uc1 + omx100	0.84
uc1 + omx75	0.84
pwr25	0.79
udr50	0.77
uc1 + omdp	0.77
uc1 + oxdp	0.76
uc1	0.76
omxc100	0.75
omxc75	0.75
pw1	0.70
omx100	0.69
omxc50	0.68
omx75	0.68
udr75	0.63
omx50	0.62
isdp	0.61
imdp	0.61
ixdp	0.59
cmdp	0.57
csdp	0.57
cxdp	0.56
pw2	0.55
om100	0.55
hierarchy	0.55
om75	0.54
pwr50	0.54
minconf	0.53
om50	0.52
none	0.47
pwr75	0.47

$w_{neg} = 1$, $w_{pos} = 1$	
uc2 + omx75	0.97
uc2 + omx100	0.96
uc2 + omdp	0.96
uc2	0.95
uc2 + oxdp	0.95
ucm + omx75	0.94
ucm + omx100	0.93
ucm + imdp	0.92
ucm	0.92
ucm + ixdp	0.92
uc1 + omx75	0.92
uc1 + omx100	0.91
ucs	0.90
uc1 + omdp	0.89
uc1 + oxdp	0.88
uc1	0.88
omxc75	0.84
omxc100	0.84
dynconf	0.83
udr50	0.82
omxc50	0.81
omx75	0.80
omx100	0.80
udr25	0.79
omx50	0.77
pw1	0.76
isdp	0.75
imdp	0.75
udr75	0.75
ixdp	0.74
cmdp	0.72
csdp	0.72
cxdp	0.72
hierarchy	0.71
om100	0.71
om75	0.70
pw2	0.70
om50	0.69
minconf	0.68
none	0.65
pwr25	0.62
pwr50	0.60
pwr75	0.56

$w_{neg} = 1$, $w_{pos} = 2$	
uc2	0.96
uc2 + oxdp	0.96
uc2 + omdp	0.96
uc1 + omdp	0.95
uc1 + oxdp	0.95
uc1	0.95
uc2 + omx75	0.95
uc1 + omx75	0.95
uc2 + omx100	0.94
uc1 + omx100	0.94
ucm + imdp	0.94
ucm	0.94
ucm + ixdp	0.94
ucm + omx75	0.93
ucm + omx100	0.92
ucs	0.91
omxc75	0.89
omxc50	0.88
omxc100	0.88
omx75	0.87
omx50	0.86
omx100	0.86
isdp	0.84
imdp	0.84
ixdp	0.84
hierarchy	0.83
cmdp	0.83
csdp	0.83
udr50	0.83
cxdp	0.82
udr75	0.82
om75	0.82
om100	0.81
om50	0.81
pw2	0.80
minconf	0.79
pw1	0.78
none	0.77
dynconf	0.73
udr25	0.66
pwr50	0.62
pwr75	0.61
pwr25	0.45

Table 6.11: Overall ranking of validators for the DiHabs scenario evaluation. Each table corresponds to a weight combination for prevented negative and supported positive predictions. See tables 6.6 and 6.7 for descriptions of validator IDs.

Optimization (s)	
HBR-RL	225.35 18.91
HBR-SVM	200.09 10.43
BR-RL	120.07 9.10
BR-SVM	106.87 8.12
BR-KNN	50.42 3.41
OMI-RL	35.42 2.38
HBR-KNN	28.26 1.67
OMI-SVM	21.28 1.53
OMI-KNN	21.20 1.55
BR-NB	18.73 1.29
HBR-NB	17.67 1.13
OMI-NB	17.26 1.22
PS-RL	14.86 0.96
OMI-DT	13.97 0.36
PS-SVM	8.57 0.55
OMI-GS	5.12 0.07
PS-NB	2.93 0.13
PS-KNN	2.29 0.01
HBR-DT	1.21 0.08
BR-DT	0.52 0.04
HBR-GS	0.10 0.01
PS-DT	0.10 0.00
BR-GS	0.01 0.00
PS-GS	0.00 0.00

Training (ms)	
HBR-RL	194.17 12.17
HBR-SVM	146.25 10.18
BR-RL	81.50 6.50
HBR-DT	77.08 5.01
BR-SVM	58.99 4.19
BR-DT	31.00 2.23
HBR-NB	17.69 1.50
OMI-RL	13.08 1.08
PS-RL	12.43 0.82
OMI-DT	6.47 0.22
PS-DT	6.02 0.21
HBR-GS	5.80 0.71
HBR-KNN	5.34 0.70
BR-NB	5.33 0.38
OMI-SVM	4.65 0.20
PS-SVM	4.05 0.15
OMI-NB	0.66 0.00
OMI-GS	0.52 0.01
BR-GS	0.45 0.03
OMI-KNN	0.42 0.00
PS-NB	0.23 0.00
BR-KNN	0.22 0.02
PS-GS	0.09 0.00
PS-KNN	0.00 0.00

Prediction (ms)	
BR-KNN	13.53 0.95
HBR-KNN	7.30 0.43
BR-NB	2.58 0.18
BR-SVM	2.44 0.18
BR-RL	2.39 0.18
HBR-RL	1.78 0.09
HBR-SVM	1.67 0.09
HBR-NB	1.58 0.09
PS-KNN	0.65 0.00
OMI-NB	0.57 0.03
OMI-KNN	0.49 0.05
PS-NB	0.45 0.02
OMI-SVM	0.41 0.02
OMI-RL	0.31 0.01
BR-DT	0.27 0.02
PS-SVM	0.27 0.01
HBR-DT	0.26 0.01
OMI-GS	0.22 0.01
OMI-DT	0.21 0.01
PS-RL	0.15 0.00
HBR-GS	0.05 0.00
BR-GS	0.03 0.00
PS-DT	0.02 0.00
PS-GS	0.00 0.00

Table 6.12: Overall ranking of runtimes for the DiHabs scenario evaluation. Each table lists the mean runtime and error margin for a 0.95 confidence interval (among all per-subject-means) for all base learner and wrapper combinations. Optimization time is given in seconds, while training and prediction time is given in milliseconds. Similar to validator rankings, not individual values but the comparison of values are the insight provided by these rankings.

6.6.9 Discussion

The evaluation results presented in the previous section illustrate that machine learning methods are a suitable approach for automating respectively assisting disclosure decisions. However, they also show that the quality of predictions not only depend on particular learning methods but also the subjects disclosures are predicted for.

Learning Methods

Especially the results presented by the overall rankings show that – regarding performance metrics – the most promising base learning methods are SVM, DT, and RL. Concerning wrappers, generally best performance is given by the OMI, HBR and BR wrappers. How-

ever, the differences here are less significant. In fact in several cases the plain PS wrapper performs almost as good as the other wrappers. This raises the question if the overhead introduced by more sophisticated wrappers than the PS wrapper is worth their improvements. Potential overhead issues are (a) increased runtime, (b) increased implementation complexity, and (c) reduced human readability and adjustability.

Runtime Regarding runtime performance (see table 6.12) especially the HBR and BR wrapper significantly increase the runtime of parameter optimization, training and prediction (up to a factor of 35). While time may not be a crucial factor in the practical application of a machine learning based disclosure assistance system, the related difference in usage of processing resources is an important point to consider – particularly in context of assistance systems running on power-constrained mobile devices.

Implementation complexity All three sophisticated wrappers significantly increase the implementation complexity and are less likely to be available in standard machine learning libraries on whatever platform an actually implemented disclosure assistance system is deployed.

Human readability and adjustability Anticipating the findings of the upcoming section 6.7, both the HBR and BR wrapper practically undo any transparency given by a used base learner. The OMI wrapper still provides a reasonable level of transparency and in some cases even does not influence the transparency of its base learner at all. Obviously the PS wrapper does not obfuscate the behavior of its base learner in any case.

Besides implementation complexity, the OMI wrapper clearly entails less overhead-based negative implications than the HBR and BR wrapper. At the same time the OMI wrapper's performance metrics keep up with those of the HBR and BR wrapper – in some cases it even outperforms them. Comparing the OMI and PS wrapper, the former's predictions mostly are more accurate than those of the PS wrapper. In fact the OMI wrapper has the potential to only perform worse than the PS wrapper in a very few cases – given the wrapper parameters are constantly optimized (i.e. whenever the set of previous situations to learn from grows). Note that this is not the case for the results presented in the previous section. There learner and wrapper parameters have been optimized only once per subject, base learner and wrapper – otherwise the evaluation system's runtime would be too long. Summarized, in the majority of cases OMI appears to be the best choice as a wrapper while SVM and RL are the generally best performing base learners.

Learning Method Configurations

There is no overall ranking of base learner and wrapper configurations because they are rather diversified and lack a general trend. Hence there are no generally recommendable configuration settings. Instead parameters should be optimized dynamically. Especially the OMI wrapper parameters have a strong link to the individual disclosure behavior of a subject.

Preprocessors

The scenarios evaluated here indicate no trend if disclosure modalities should be encoded as root or leaf elements in the hierarchical structure of disclosed information. One reason is that the DiHabs scenario has no varying modality at all. The manually composed scenario has varying modalities but these modalities have a distinct correlation with disclosed information items. Hence, performance metrics for the evaluated preprocessors Root Modality (MR) and Leaf Modality (ML) are almost similar. Concerning the question how to encode disclosure modalities further scenarios have to be composed and evaluated.

Validators

Prediction validators are a contributing extension to a machine learning driven disclosure assistance mechanism. Especially in the case when learners often fail to correctly predict disclosures, the more valuable are prediction validators. In fact, the number of detected incorrect predictions mostly behaves proportionally to the number of incorrect predictions.

As shown in table 6.11 the best validators for the DiHabs scenario are *uc2* (a predicted disclosure must have been disclosed at least two times before) in combination with a validator which checks the order mappings and disclosure path relations of a prediction (e.g. *omx75* and *omdp*). The *uc2* validator alone appears to be a good generic choice. However, if one wants to make sure to use the best validator for a particular subject, it has to be determined individually (also because the decision of whether to focus on prevented negative or on supported positive predictions is rather subjective). Finally the validator evaluation for the manually composed scenario yields different results – here *uc1* performs significantly better than the *uc2* validator[7]. The consensus is that a disclosure assistance mechanism which utilizes validators has to incorporate multiple validators and dynamically decide which one to choose for a particular subject.

[7]Results for the manually composed scenario are not shown here for the lack of space. Please inspect the online available result illustrations at `http://opsci.informatik.uni-rostock.de/index.php/DiLES`.

Correlations with Disclosure Patterns

As illustrated by the result plots for the two example subjects 27 and 32 in the previous section, there is an obvious correlation between the distribution of order-mapping types and the performance of the OMI wrapper: when there are primarily order-ignoring and either order-reversing or order-preserving mappings (i.e. no or only a few order-loosing mappings), then the OMI wrapper shows its strengths. Rather obvious correlations are the number of unique disclosures, the poset width and disclosure usage counts (the first two generally reduce prediction quality when growing, the latter one when falling).

Generalizability

The evaluation results for the two investigated scenarios provide common insights on base learners and wrappers. The OMI wrapper and the SVM and RL learner appear to perform best in most cases. In contrast, results for validators, preprocessors and learner respectively wrapper configurations cannot be generalized. Here disclosure assistance mechanisms have to keep ready multiple methods and choose the best one on a per-subject basis.

User Metrics

The performance metrics used to assess the learning methods not necessarily align with subjective metrics of users. The reason is that the negative impact of a misclassification not only depends on "syntactical" errors (e.g. number of falsely predicted information items) but also on the sensitivity a user associates with a particular information item. In fact in context of an objective evaluation of learning methods this methodological problem cannot be avoided. However, in practice the potential mismatch of technical and subjective metrics should not be that relevant. The composite disclosure control system presented in chapter 5 also provides other mechanisms than machine learning based disclosures, e.g. manual rules and explicit user decisions. These mechanisms should be preferred to manage rather sensitive information. In that case the sensitivity of information whose disclosure is managed by machine learning methods is less diversified, i.e. the difference of objective and subjective metrics is less significant.

6.7 Model Interaction

In chapter 5 (Composite Disclosure Control) it has been reasoned that a comprehensive disclosure assistance has to utilize different mechanisms to control information disclosure. Automating disclosures by learning a user's disclosure behavior is just one mechanism.

Section 5.2.2 already described general possibilities to integrate a learning-based disclosure assistance within a composite disclosure control system. This section elaborates learner-specific integration possibilities. In particular, the learning methods considered so far are analyzed with respect to their human readability and manual tunability (Bünnig, 2008).

6.7.1 Naive Bayes Classifier

For each independent variable (e.g. situation, modality and person attributes) a naive Bayes classifier calculates the disclosure probabilities for each set of information items disclosed before. Subsequently, per information set these probabilities are combined (assuming they are independent from each other) and ranked. This calculation process isn't very intuitive for humans in terms of explaining disclosure decisions. However, to communicate decisions made by the classifier more comprehensible, the classifier could state the n most influential context attributes of the predicted disclosure decision. In response to wrong predictions users could adjust the influence of individual attributes. For instance if a location attribute significantly contributed to a wrong disclosure decision, the user could degrade the relevance of the location attribute until a correct prediction would have been made. Of course, this is only applicable when the correct information set to disclose is already known to the classifier.

6.7.2 Decision Trees

Decision trees describe information disclosure preferences as a graph with a hierarchical structure of decisions. The actual classification (disclosure decision) is given by the leafs. The nodes test features for particular values and branches denote which node to process next depending on the result of a value test. In contrast to a naive Bayesian classifier conditional dependencies between attributes are regarded due to the top-down concatenation of feature evaluations. In principle a decision tree is capable of classifying correctly at least the training set but the accuracy is limited if there are any size restrictions (e.g. if a tree is supposed to conceptualize privacy preferences in a human readable way). While a disclosure behavior expressed by a decision tree enables users to review and tune automatic disclosure decisions they do not really match human decision processes as they always start from a single root. Further they are hard to adjust since changing decisions on particular nodes may need a rather extensive rearrangement of the tree in order to still catch the decisions made within the situations that set up the training set. The same is true when the training set gets extended by new manual disclosure decisions that have been misclassified by the current tree. In that case a user has to review again the whole tree in order to understand and evaluate it. In other words one can say that decision trees (re-

garding their output representation) are suitable for expressing static privacy preferences but inappropriate for iterative adjustments of privacy preferences.

6.7.3 Rule Learner

In principle any decision tree also can be expressed as a set of rules. In that case a rule describes a path from the tree's root to a leaf. There is one rule for each leaf which combines the decisions made within the path from the root to that leaf. The leaf constitutes the consequent of a rule. However, such a set of rules is far more complex than necessary. The purpose of rule learning is to find a smaller set of rules than the one directly read from a decision tree. The less rules the easier it is for users to review them in order to compare them to their intuitive privacy preferences. This is a clear advantage compared to decision trees.

In its basic form rules are logically linked Boolean expressions (material conditionals) testing features for specific values. Rules can either be read in order or arbitrary. In-order rules can be seen as one deeply nested if-then-else construct. Such a rule set is not very intuitive since it must be read completely top-down to understand the last rule. A set of rules where each rule alone is a valid classification is more comprehensible to users. However, they bear the risk that situations outside the training set that created the rules could be classified differently by different rules. Further they tend to be larger than in-order rules. Nevertheless this can be seen as an acceptable drawback since it is easier for users to adjust or extend rules. When rules are ambiguous the user can be consulted for an exact disclosure decision. Subsequently the rules can be adapted by relearning them including the situation that caused the case when there was no unique classification.

Rules learned by inductive logic programming are of first-order logic and therefore more expressive than rules based on propositional logic (Muggleton, 1994). They are able to describe relations between features, optionally in a recursive manner. Rule sets created by inductive logic programming may be shorter than propositional rules and they *may* be closer to a human's idea of handling private data. However, at the same time rules expressed by first-order logic can be hard to verify intuitively. Especially in case of recursive expressions users may fail to realize all consequences of an information disclosure rule.

6.7.4 Instance-Based Learning

Instance-based learning (sometimes also called case-based reasoning or analogy learning) memorizes past situations with corresponding disclosure decisions and tries to decide new situations based on analogy to previous situations. It is a lazy learning method as there

actually is no learning at all until a new situation has to be classified. Any generalization out of the training set is done whenever an unknown situation has to be classified. The most popular example are nearest neighbor classifiers which decide new situations based on its distance to previous situations. A typical approach is to regard the k nearest situations and to adopt the disclosure decision made in the majority of this k situations, optionally voted by their distance value. This is called a k-nearest neighbor classifier.

Concerning transparency instance-based learning is able to give reasons for its decision by exposing the situations used for deriving a decision for a new situation. Users can tune the decisions made by adjusting the relevance of past situations and individual features (and thus their resulting distance values), similar as mentioned for a naive Bayes classifier.

6.7.5 Support Vector Machines

SVMs practically are block boxes. It's internal mechanism of using a kernel function based feature space transformation and calculating distances to support vectors is beyond a normal human decision process. With regard to tuning there are only generic possibilities like dropping situations and weighting features, though SVMs do not provide indications how individual situations and features influence its predictions.

6.7.6 Multi-Label and Hierarchical Wrappers

Wrappers like Binary Relevance (BR) and Hierarchical Binary Relevance (HBR) in fact obliterate any readability and tunability of wrapped base learners since a potentially high number of base learners are used whose labels (disclosures) do not correspond to finally disclosed information sets. The same limited generic tunability options exist as for SVMs.

6.7.7 Order-Mapping Interpolation Wrapper

In contrast to BR and HBR wrappers the OMI wrapper does not obliterate readability and tunability characteristics of a used base learner. It does not combine multiple base learners but acts as an additional decision instance in front of a single base learner, i.e. either the OMI wrapper predicts a disclosure or its base learner. Similar to instance-based learners there is no model of the OMI wrapper one could investigate. Instead made predictions are human readable in that the past situations used for interpolation may be displayed to users. Then users are provided with hints which situations to drop in order to tune the interpolation process. Additionally, when investigating made predictions, users may mark persons which should not be considered for interpolation.

6.7.8 Summary

In conclusion good candidates for readable and manually adjustable learning methods are k-Nearest Neighbors (KNN) and Rule Induction (RL) learners in combination with a Powerset (PS) or OMI wrapper. Still, specific representations of learner models and made predictions as well as interfaces to interact with them are a field of research on its own. However, the analysis provided here is a good starting point for future work in this area.

6.8 Conclusion

This chapter dealt with machine learning based automation of information disclosure decisions. First, it defined the corresponding learning problem and reviewed existent learning methods matching the defined problem. New metrics to asses learning methods in context of information disclosure have been defined (e.g. *submatch* and *strain*). Further, a new learning method which acts as a wrapper around standard learning methods has been developed. This new method investigates order mappings with regard to hierarchical information sets and information receiver sets. Additionally the disclosure patterns elaborated in chapter 4 have been used to develop prediction validators which are supposed to prevent negative predictions. In order to evaluate the reviewed and developed learning related mechanisms, an evaluation system (DiLES) has been developed. This system is not only capable of evaluating the mechanisms presented here but is extensible for further investigations. Based on two scenarios describing a sequence of disclosure situations, the evaluation system has been used to compare the presented learning methods and validators. Finally, learning methods have been reviewed with regard to their suitability for an integration into a composite disclosure control system. In particular the focus has been on human readability and adjustability of learned models.

One main finding of this chapter is that the developed OMI wrapper is a competitive alternative to other sophisticated wrappers like HBR and BR. Not only that its performance metrics are similar and sometimes better, it also introduces less overhead in terms of computation resources and does not obfuscate the decision model of its base learner. Additionally its ability to decide when to delegate predictions to the base learner makes it a safe alternative to a simple PS wrapper. Another main finding is that the presented validator mechanisms are a valuable extension to a learning method in order to reduce the negative impact of misclassifications. Generally the evaluation results support a machine learning based disclosure control assistance. Though the *match* performance is below 0.5 in some cases, the *strain* metric mostly is in acceptable bounds (less than 0.2) which means that disclosure predictions at least are a good suggestion template for final manual disclo-

sures. Finally the OMI wrapper and the generally well performing RL learner also meet transparency requirements for human readability and adjustability.

In conclusion, the mechanisms presented in this chapter are able to handle issue III from the introductory problem statement in section 1.2 (*automating disclosure control in a user-adaptive but easy to manage fashion*).

7 Guidelines for Interpersonal Privacy

This chapter consolidates the findings made in this work with the general guidelines presented in section 3.1 – within the specific context of managing *interpersonal privacy in smart environments*. It serves as a starting point for engineers designing and implementing smart environments with implicit as well as explicit interpersonal information exchange.

Section 7.1 lists potential pitfalls which may violate the concept of interpersonal privacy. Having these issues in mind when engineering smart environments helps to avoid disruptions in social interactions mediated by the environment. The subsequent section 7.2 extends principles presented in section 3.1 by additionally focusing interpersonal privacy. These two sections are not direct consequences of technical findings made in this work but represent the author's insights on conveying interpersonal privacy to smart environments, based on sociological research (see section 2.1.3) and explorative studies (see sections 3.1 and 3.2.2). In contrast, the final section 7.3 compiles specific recommendations based on technical results of this work.

In order to distinguish the principles and guidelines presented in this chapter from general privacy principles, first a short recap of the concept of interpersonal privacy (as described in section 2.1.3) is given. Interpersonal privacy is a process of dialectic and dynamic boundary regulation where subjects regulate interaction with other persons via several behavioral mechanisms. It adjusts desired level of inputs and outputs by providing information and receiving corresponding feedback. The goal is to reach a certain state of participation in a social environment. Practicing interpersonal privacy involves factors like social norms, allegiance, self-representation, and expectations in social interactions. Rather than managing information and associated rights, it is about composing a set of hidden and disclosed information for a particular social interaction. Privacy preferences depend on roles individuals may act as and often lack general patterns. Smart environments impact interpersonal privacy in that they introduce new privacy regulation mechanisms by providing multiple modalities to communicate with others.

7.1 Pitfalls of Violating Interpersonal Privacy

These pitfalls extend the ones compiled by Lederer *et al.* (2004) (see section 3.1) by emphasizing potential disruptions of social interactions.

7.1.1 Employ Manifold Communication Modalities

While many modalities to communicate information (e.g. shared displays, audio channels, device-to-device) allows to tailor information exchange to individual requirements, it also complicates communication in that users easily choose improper modalities which distribute information to an unintended audience or which mediate information in an unexpected manner. Manifold communication modalities also make it hard to predict to whom and how information flows in case of automatically triggered information exchange. Instead users should be provided with a *manageable* set of tools supporting social interactions.

7.1.2 Hide Communication Modalities

This pitfall partly arises from the previous one, i.e. in case of too many modalities some of them easily do not get recognized. The idea of ubiquitous computing generally motivates to let technology vanish into the background. However, care has to be taken to not hide functionality, e.g. information exchange modalities. Users should be able to recognize and understand how information may be distributed in an environment.

Hidden modalities also exist when non-communicational services "leak" information by altering the environment in a way perceivable by other persons within the environment. For instance any user-preference–based environment-adaption indirectly communicates personal preferences to other inhabitants. In case of assistive technologies this may even reveal medical information. Services adjusting the environment in a publicly perceivable manner should highlight this fact to their users.

7.1.3 Amplify Automation

Smart environments aim to assist users by automating tedious and repetitive tasks. Exaggerating this theme may lead to unexpected or even unrecognized automatic distribution of personal information to other environment inhabitants. When automating processes, special care has to be taken to not only provide *correct* but also to avoid *unexpected* au-

tomated communication. The degree of automation should develop with the "familiarity" inhabitants of smart environments have with deployed technologies.

7.1.4 Distort Mediated Information

The modalities available to exchange information with other persons in an environment present information in different forms, i.e. mediated information usually is transformed to some extent. These transformations should not involve significant *semantic* changes. For instance information distributed to mobile devices may be scaled down and thus hide content of originally communicated information or even change its meaning. As a result intended boundaries between self and others may get altered or intended personae are not perceived as such by others. When the environment offers to disclose information items it should provide a preview how the information finally arrives at the information receiver.

7.1.5 Discard Interaction Contexts

The meaning of an information item always is linked to the context in which it is shared. A sentence said at a specific time in a specific social setting might yield different interpretations when repeated in another situation. Hence, asynchronous communication and a corresponding loss of context bears the risk of changing the information one originally disclosed. Smart environments which provide modalities for asynchronous communication, e.g. by recording disclosed information items, may disrupt social interaction posthumously. In practice the possibilities to prevent such disruptions are limited as not all context linked to disclosed information can be captured. Even worse, the linked context might again be composed of sensitive information items. The general guideline to prevent such pitfalls is to avoid the temporarily decoupled communication of information or to limit it to information items which are not vulnerable to related semantic transformations.

7.2 Enhanced Existing Principles

This section enhances the privacy principles and guidelines presented in sections 3.1.1 to 3.1.3, in particular the notions of feedback and control by Bellotti & Sellen, the privacy principles by Langheinrich, and the genres of disclosure by Palen & Dourish. These principles are extended by putting a special focus on information disclosure during social interactions.

7.2.1 Feedback

In context of the concept of privacy in public the guideline of providing feedback expresses the need to show which information is communicated to whom under which circumstances. In context of interpersonal privacy, the fact that and which information is exchanged usually implicitly is known by the owner as he fired up the process. Instead, a proper feedback here means that the person disclosing information should know *how it is perceived* by his communication partners. This could be accomplished by providing previews of how information is presented to others.

7.2.2 Control

Obviously, managing interpersonal privacy requires appropriate control mechanisms to regulate information exchange. Especially in social interactions it is important to provide coarse grained control mechanisms because a social setting might change spontaneously and users may not be able or willing to deflect their focus from their current interaction. In practice this means users should be empowered to easily switch used modalities and adjust which persons receive information items. Additionally, users should be able to veto automatic disclosure processes.

7.2.3 Flexibility

The guideline that privacy control mechanisms should be flexible is especially important for interpersonal privacy management. More than for other privacy concepts, preferences are highly individual and often not generalizable to limited abstract types (e.g. mostly private versus mostly public). In practice this means that universal coarse-grained control should be accompanied by optional fine-grained control. Additionally, one should not assume too simple patterns of privacy management (e.g. disclose less to strangers and more to friends).

7.2.4 Effort

While powerful control mechanisms are important, initially the overall effort required for managing interpersonal privacy should be as low as possible in order to prevent disruptions in social interactions because of privacy control. Next to coarse-grained controls this means that information exchange mechanisms and services should ship with a sensitive default behavior (e.g. communicate no information at all, or only follow sufficiently established disclosure patterns).

7.2.5 Proximity and Locality

Langheinrich's principle of proximity and locality proposes to limit access to information by spatial and temporal constraints. In context of interpersonal privacy this principle's rationale is to prevent unintended transformations of information due to a loss of its original context. Social interactions mediated by smart environments should be strongly coupled to the place and time of their occurrence.

7.2.6 Genres of Disclosure

The notion *genres of disclosures* refers to known and established practices of information disclosures, i.e. common patterns used to negotiate the boundary between self and others. Respecting existing genres simplifies the realization of proper feedback and control mechanisms because privacy related implications usually are already known and accepted. Of course new interaction possibilities as given in smart environments also introduce new genres which yet have to get established. For instance a meeting of persons from different companies might usually end with some exchange of contact information, e.g. using business cards. Smart environments may shift this genre of disclosure in that contact information is exchanged automatically using mobile devices of participants or using a shared screen. The bottom line is that automatic disclosure mechanisms should respect current expectations of users. New disclosure practices which conflict with known genres should be avoided. However, it is always possible to let new genres emerge from existing ones by carefully deploying corresponding information exchange practices. A conflict with a known genre would be if the slides of a presentation given in a smart environment are accessible for every attending person. While this is a great feature in terms of collaboration, it simply does not align with current common practices. The introduction and communication of such features should be well thought out and not happen silently.

The guideline to align mechanisms in smart environments with existing genres of disclosure is the least tangible one because genres are constantly evolving and because different user groups may have different expectations.

7.3 Additional Recommendations

As stressed multiple times in this work, users may follow different patterns of privacy management, ranging from simple binary to sophisticated ones. This means that multiple control mechanisms, where each supports certain patterns, should be available (see chapter 5). For instance it shouldn't be taken for granted that disclosed information inversely

scales with the number of persons receiving information or that information receivers can be allocated into a small set of distinct groups (see chapter 4). This section summarizes the findings about disclosure patterns and control methods made in this work to general recommendations for designers and engineers working on privacy sensitive smart environments. The next section 7.3.1 builds on chapter 4 (Information Disclosure Patterns). Conclusions from chapter 5 (Composite Disclosure Control) are recapitulated in section 7.3.2. Finally section 7.3.3 derives recommendations from the results in chapter 6 (Learning Disclosure Decisions).

7.3.1 Privacy Patterns

Abstraction of persons to roles or allocating them in groups in order to simplify privacy management is possible for certain information types but should be avoided as a general theme. As shown in the survey results in section 4.2.2, identifying and communicational information like phone numbers and location generally is disclosed using only a few unique values. This indicates that values may be mapped to abstract information recipient groups like *friends*, *colleagues*, or *public*. In contrast, information which entails potentially manifold derived information, e.g. *music (taste)*, is disclosed as various values which cannot be grouped reasonably. This possibility of abstraction is also influenced by the ordering of disclosures when modeled as sets of information items. Some information types arrange in an almost total order (again location and contact data) while others are only partially ordered or not ordered at all. This characteristic determines to which extent hierarchical relations between abstract roles may be utilized (e.g. friends see more information than colleagues). The main message here is that one should not generally assume simple disclosure patterns which motivate to organize information recipients in groups – such patterns may only be applicable to specific information types.

Order mappings describing relations between sets of disclosed information items and sets of persons receiving these items often are expected to be *reversing* or *ignoring*. While this is true for some users, it is definitely not a general pattern. Besides individual users, the most often occurring order mapping type again depends on the type of shared information. The more self-contained a disclosure, the more often order-reversing and order-ignoring mappings occur. In contrast, the more derivable information a disclosure entails, the more often order-loosing mappings occur. The diversity based on information type and users inhibits any general recommendation on patterns to expect in terms of order mappings. If control methods utilize order mappings, they should be able to adapt to individual distributions of mapping types.

7.3.2 Disclosure Control Methods

Several disclosure control methods and their composition have been presented in chapter 5. In order to fully support the various needs users may have in managing their information during activities in smart environments several methods have to be deployed and staged in a certain manner. The essence here is that it is important to provide methods which cover *preliminary*, *situational*, as well as *retrospective* approaches to privacy control. This ensures privacy may be practiced in a defensive, pragmatic, and optimistic way.

When deciding which control methods to concentrate on initially, predefined rules and implicit actions are recommended. The first one because it aligns with currently existing mechanisms to manage information. Though rules are impractical in many cases, most users feel familiar with them at first. On the other side, implicit actions (disclosure decisions mapped to physical actions) are comparatively novel control methods which, when properly designed and implemented, may easily be adopted by users. This basic setting can then be extended to a complete composite disclosure control system which also involves generic recommendations and an individual learning-based disclosure assistance. In any case, users should be able to veto any automatic processes and to review processes at a later time in order to understand and possibly adjust automated disclosures. This helps to establish new practices of privacy management, i.e. to constitute new genres of disclosures.

7.3.3 Automated Disclosure Assistance

The problem of automating disclosure decisions using machine learning methods have been elaborated in chapter 6. It presented several learning techniques, including a novel one, and evaluated them with example disclosure situations. The general outcome is that learning methods are a recommendable technique to automate disclosure decisions. Though they do not perform well in all situations, they are at least suitable to suggest disclosures and in that reduce the workload users face when managing information exchange.

Recommended methods to initially concentrate on are k-Nearest Neighbors (KNN) and Rule Induction (RL) in combination with a Powerset (PS) or Order-Mapping Interpolation (OMI) wrapper (to handle the hierarchical multi-label characteristics of the underlying learning problem). Both the KNN and RL learner mostly perform at least as good as the others while being human readable and adjustable. The RL learner has the additional advantage of being both readable and adjustable by humans (unless very low-level context information is used). Similarly, the performance of the PS and OMI wrappers competes with the Binary Relevance (BR) and Hierarchical Binary Relevance (HBR) wrappers, while

being less costly in terms of resources and while not obfuscating the model of the wrapped basic learner. Since a well optimized OMI wrapper in principal is at least as good as a PS wrapper, it may safely used as the default wrapper. Its only downside compared to the PS wrapper is its more complex implementation.

The performance of learners should not only be measured according to the number of correct predictions but also according to their *strain* metric, which assesses the workload of manual corrections of predictions.

Finally, validators as presented in section 6.4 should be used additionally as they prevent a significant number of wrong predictions and thus indicate when disclosure decisions by a learning mechanism should not be used but delegated to another decision component (e.g. a user herself).

The basic message is that although learning mechanisms do not perform well in all cases, they are still helpful when wrong predictions are detected and instead used as a suggestion or a template for a final manual disclosure decision.

7.4 Conclusion

This chapter dealt with issue (IV) from the introductory problem statement. It compiled general privacy guidelines for designers and engineers of smart environments targeting social interactions.

The pitfalls help engineers to evaluate existing solutions with regard to potential violations of interpersonal privacy. The enhanced existing principles complement the general understanding of how to develop privacy sensitive environments. Finally, the recommendations based on the findings made in this work support engineers in making specific technical decisions when working on mechanisms to manage interpersonal privacy within smart environments.

8 Conclusion and Future Work

This thesis addressed the issue of managing privacy in smart environments, while emphasizing problems and solutions in context of interpersonal privacy. At first, it elaborated different concepts of privacy and how ubiquitous computing in general and smart environments in particular may interfere with these concepts. Subsequently already existing work aiming to solve these issues have been reviewed. The following chapters developed solutions to some of the open issues, namely to understand patterns of interpersonal privacy management, to orchestrate different disclosure control methods to a composite disclosure control system, and to automate disclosure decisions using machine learning techniques. These solutions have been rounded out to guidelines for developing smart environments aiming to support interpersonal privacy.

Following is an analysis to which extent the proposed solutions resolve the issues stated in section 1.2: (I) consideration of social aspects in interpersonal privacy management and resulting patterns in information disclosure decisions, (II) suitable disclosure control mechanisms that match these patterns, as well as their orchestration, (III) automating disclosure control in a user-adaptive but easy to manage fashion, and (IV) general guidelines and principles to develop interpersonal privacy sensitive smart environments and to evaluate corresponding solutions. Each subsequent section summarizes the contributions to clear a particular issue and the unsolved problems to be handled in future work.

8.1 Patterns of Interpersonal Privacy Management

Chapter 4 tackled the issue of understanding social aspects in privacy management. It developed a model to express disclosure decisions in social interaction in smart environments and elaborated patterns of privacy management based on relations between information receiver groups and disclosed information items. These patterns build upon set-based structural information and thus introduce objective characteristics to the generally vague concept of interpersonal privacy. These objective characteristics provide novel parameters to programmatically address the management of personal information in social interactions.

Still, there is room for future research in this area. Patterns based on the decision parameters situation and modality only have been touched briefly. Candidates for objective characteristics are sensitivity, visibility and persistency of modalities as well as social attributes of a situation.

8.2 Disclosure Control Mechanisms and Their Orchestration

The orchestration of disclosure control methods to a composite disclosure control system has been handled in chapter 5. It presents a conceptual framework to solve the issue that different users and different information types demand for different practices of managing information exchange. The proposed system covers multiple approaches to privacy management, including preliminary, situational, and retrospective control. Its methods range from simple and optimistic to sophisticated and pessimistic control. The system not only sums up different control methods but integrates them with each other in order to balance shortcomings and leverage strengths of individual methods. In that it serves as a comprehensive blueprint for engineering privacy-sensitive smart environments.

Still open issues in this area primarily exist on the proper realization of individual components, especially with regard to user interfaces and data exchange between different components. This is also the reason why there is no prototype implementation of the proposed system – there is still enough research to be done on individual components (chapter 6 is an example for this).

8.3 Automating Disclosure Control

The issue of automating disclosure control to let users manage personal information individually but with a reduced workload was the concern of chapter 6. It developed a formal groundwork to transform disclosure behavior to a learning problem. Next to suitable existing machine learning methods this chapter presented a novel method which predicts disclosures by interpolating past disclosures based on order-mappings of information receivers and items. Besides explicitly incorporating patterns elaborated in chapter 4, this new method has the salient property of only becoming active when there is a high probability of correct predictions (given a preliminary parameter optimization). Additionally various validators have been developed which utilize patterns from chapter 4 and which significantly reduce the negative impact of incorrect predictions. A toolkit has been developed which allows to evaluate learning methods and validators for arbitrary scenarios. The presented results of two example scenarios show that disclosures may be automated using machine learning methods. For some users of the example scenarios a *match* ratio

of 0.9 could be reached. Still, for other users this ratio was below 0.2. However, in these badly performing cases, validators where able to detect up to 85% of wrong predictions. Additionally, the *strain* metric was around 0.2 and the *h-accuracy* metric settled at 0.6. Hence, even if machine learning methods are not suitable to automate disclosures in all cases, they still reduce the workload by generating templates for manual disclosures. In the end each user has to decide a personal performance threshold to separate automated and suggested disclosures. In any case, the mechanisms presented in chapter 6 solve issue (III) by either automating disclosures or by assisting in manual disclosures.

Of course the presented results are valid for the two example scenarios only, i.e. there cannot be made a final decision if issue (III) is clearly solved. Nevertheless, the developed evaluation toolkit makes it easy to perform additional evaluations on other scenarios and learning methods in the event of future requirements.

It still has to be assessed if the metrics evaluated by the toolkit align with subjective user metrics. This, however, requires the capturing of scenarios in productively used smart environments. Another still unsolved part of issue (III) is the specific design of modalities for human interaction with learned prediction models. Starting points for this work have been provided at the end of chapter 6.

8.4 General Guidelines and Principles

Chapter 7 was supposed to provide missing guidelines and principles for developing environments with explicit support for interpersonal privacy. It picked up existing principles and extended them by additionally focusing privacy issues in smart environments during social interactions. These rather abstract guidelines then have been complemented by more technical recommendations for engineering privacy solutions, based on the findings made in chapter 4 and 6. In that chapter 7 contributes to solving issue (IV).

On the other side, this issue hardly can be solved completely since guidelines may change with the evolution of smart environments and their integration into day-to-day life. Additionally, the technical recommendations also have to be adjusted with the technical progress of mechanisms used to manage personal information. Especially more precise guidelines and recommendations related to the design of user interfaces are an ongoing open issue.

Appendices

A DiHabs

This appendix chapter provides methodological details about the survey conducted in chapter 4 (Information Disclosure Patterns) for the sake of transperency and reproducibility. It briefly describes how the survey system DiHabs may be used to define interviews and to analyze corresponding survey results. Finally it shows the interview definition used for the survey presented in section 4.3 as well as a screenshot-based walk through the resulting interview.

A.1 Introduction

DiHabs is a Django[1] application for conducting online interviews about information disclosure behavior within social interactions. Software and data related to DiHabs is hosted at the Open Science Repository[2] of the Computer Science Department at Rostock University. Please consult the software's *README* file for deployment and configuration instructions.

A.2 Interviews

Interviews are defined as plain Python dictionaries which must have the the following items:

`name`**:**
: Name of the interview.

`npersons_min` **and** `npersons_max`**:**
: Minimum and maximum number of persons a participant has to name when asked for her social environment.

`npersons_max_sample`**:**
: Maximum number of persons to actually use for generating person groups.

[1] Django: `https://www.djangoproject.com/`
[2] Open Science Repository: `http://opsci.informatik.uni-rostock.de/index.php/DiHabs`

`ngdups`:
: Number of person group duplicates (to check for conflicting disclosures).

`groups`:
: Either a precompiled list of groups (where a group is a tuple of person index numbers) or a list specifying how much groups to generate for different group sizes (i.e. the value `groups[i]` specifies the number of groups of size `i+1`).

`messages`:
: A dictionary of messages used in page templates. Each message is interpreted as MARKDOWN[3] formatted text. Inspect DIHABS's *README* or the example interview definition in section A.4 for more detailed information.

`itypes`:
: A list of dictionaries, each specifying a particular information type respectively situation where participants are supposed to decide disclosures for.

  `name` **and** `plotname`:
  : Name of the information type (`plotname` should be a short name to be used in plots).

  `nvalues_min` **and** `nvalues_max`:
  : Minimum and maximum number of information items participants must provide.

  `desc_input`:
  : Description of the information items participants are supposed to provide.

  `desc_select`:
  : Description of the information type and situation where the disclosure of information items have to be decided. This description must include the social context and modalities of information disclosure.

A.3 Analysis

Survey results are stored in a PYTHON SHELVE[4] database (mapping participant session IDs to corresponding interview data). The DiHabs application provides a script to analyze results in such a database. Supposed the DiHabs application's Python module is available in `PYTHONPATH`, it can be run as follows:

```
$ python -m dihabs.analyze
```

The `--help` option shows a summary of this script's functionality.

[3] MARKDOWN: `http://daringfireball.net/projects/markdown/`
[4] PYTHON SHELVE: `http://docs.python.org/library/shelve.html`

List Results

A list of all results in a survey results database is shown when using the `--list` option:

```
$ python -m dihabs.analyze --results=results.db --list
  0: 69c48713... (2010-09-07 09:39:31.107497, 7ea0be9a..., Demo)
  1: 1dd50bff... (2010-09-07 09:47:48.395140, 7ea0be9a..., Demo)
  2: 3e269763... (2010-09-09 12:49:35.249546, 7ea0be9a..., Demo)
  3: 6246deab... (2010-09-09 12:49:48.900607, 7ea0be9a..., Demo)
  4: fb1776cf... (2010-09-09 12:49:51.995939, 7ea0be9a..., Demo)
...
```

For each participant, this shows the participant's session ID, the interview start-time, and the interview ID and name. Especially the session ID is needed to select individual results for further analysis.

Inspect a Specific Result

The data related to a specific participant can be shown using the `--dump` option:

```
$ python -m dihabs.analyze --results=results.db --usid 69c48713... --dump
usid: 69c487130a3a4869b97725247b19761f:wq
interview: {'templates': {}, 'npersons_max': 20, 'name': 'September', 'messages ..
start: 2010-09-07 09:39:31.107497
end: 2010-09-07 10:14:34.895914
npersons: 12
groups: [92, 8, 12, 6, 14, 46, 158, 152, 2, 20, 28, 16, 4]
nvalues: [10, 5, 3, 9, 8]
test: False
disclosures: [{'group': 92, 'values-per-type': [262, 10, 1, 328, 7], 'values': 10 ..
```

A textual summary of detected disclosure patterns can be retreived similarly using the `--analyze` option (instead of `--dump`).

Render Order-Mapping Graphs for a Specific Result

In order to inspect order-mappings in the disclosure behavior of a specific participant, the analyzer script may render graphs which illustrate the order-mappings of situation pairs where the information receiver person groups are comparable with each other:

```
$ python -m dihabs.analyze --results=results.db --usid 69c48713... --graph \
        --dest exampleplots/
```

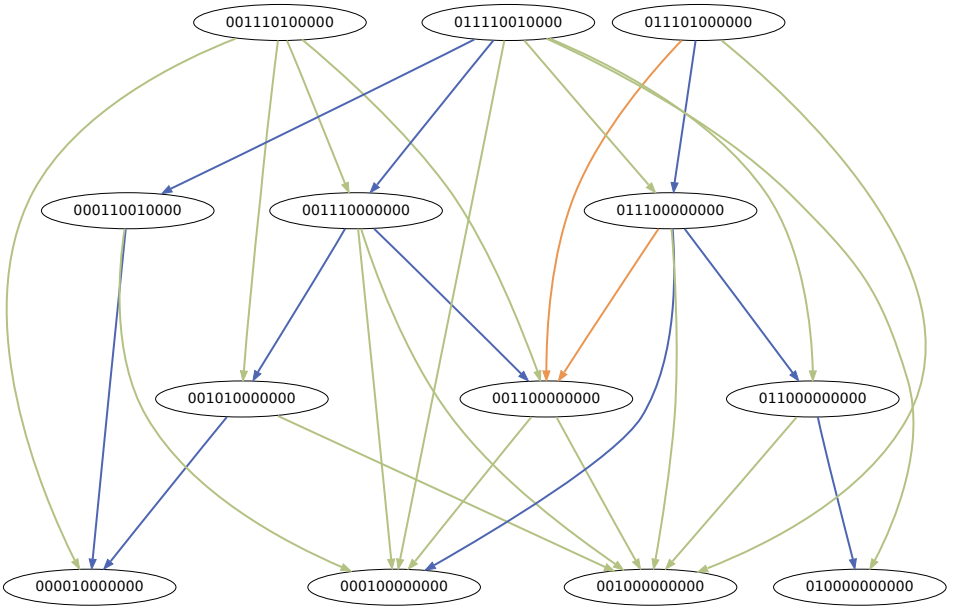

Figure A.1: DiHabs example plot.

This renders graphs for each information type found in a result. Figure A.1 shows an example graph. Here the circles represent disclosure situations while the binary numbers represent the presence of persons (information receiver) in a situation, i.e. the number *000011* indicates that persons *5* and *6* were present (i.e. part of information receiver group) whole persons *1* to *4* were not present. The arrows show *is-superset-of* relations concerning the person groups. The colors indicate the type of order mapping. Green arrows reflect *order-reserving*, orange *order-preserving*, blue *order-ignoring*, and dashed purple arrows indicate *order-loosing* mappings. Order-mapping relations are transitive except when two nodes are connected directly.

Render Pattern Usage Plots for All Results

The options `--plot` and `--plot2` may be used to render plots which illustrate the occurrence of certain information disclosure patterns, including order-mapping types.

Examples for these plots can be found in section 4.3.

A.4 Example

The site project used for the survey in chapter 4 may be referenced as an example for the generic documentation above. Especially the files `sipro/settings.py` and `sipro/urls.py` should be consulted. The corresponding source code can be found on the CD shipped with this thesis as well as at the Open Science repository at `http://opsci.informatik.uni-rostock.de/index.php/DiHabs`. Details about this Django site project can be found in its `README` file. The project also contains the used interview specification and the captured results.

The raw interview specification is shown in the next subsection. This is followed by a screenshot-based walk through the resulting online interview. Since most of the survey participants used German as their first language, the interview is written in German too.

Interview Specification

As already mentioned, interviews are specified as ordinary Python dictionaries. Text parts are written in MARKDOWN[5] to keep them compact while still supporting an appropriate formatting. The different specification attributes have been described in section 4.2.2.

```
spec = {

# ------------------------------------------------------------------------
# general attributes
# ------------------------------------------------------------------------

'name': "Stud",
'npersons_min': 10,
'npersons_max': 20,
'npersons_max_sample': 12,
'ngdups': 0,
'groups': [(1,), (2,), (3,), (4,),
           (1,2), (2,3), (2,4),
           (1,2,3,), (2,3,4), (3,4,7),
           (1,2,3,5), (2,3,4,6), (1,2,3,4,7),
          ],

# ------------------------------------------------------------------------
# messages to use in templates
# ------------------------------------------------------------------------

'messages': { # generic messages
    'title': "Umfrage zur Freigabe persönlicher Informationen",
    'start': """
        # Umfrage zur Freigabe persönlicher Informationen
```

[5] MARKDOWN: `http://daringfireball.net/projects/markdown/`

```
    Durch die Teilnahme an dieser Umfrage unterstützt du ein
    Dissertations-Projekt zum Thema Datenschutz. Vielen Dank schon mal!
    Ziel der Umfrage ist es, strukturelle Informationen in
    Datenschutzpräferenzen zu erfassen. Einen Eindruck davon
    bekommst du am Ende der Umfrage durch eine erste Auswertung deiner
    Antworten.

    ## Überblick

    Die Umfrage besteht aus drei Teilen. Da
    Datenschutz eine individuelle Angelegenheit ist, müssen auch die
    Fragen dazu individuell formuliert werden, z.B. sollten sie sich
    auf Personen beziehen die du kennst. Die ersten beiden Teile der
    Umfrage dienen dazu diese individuellen Informationen zu erfassen.

    Im ersten Teil wirst du gebeten, einige Personen aus deinem Umfeld
    anzugeben, z.B. Freunde, Verwandte oder Kollegen. Im zweiten Teil
    bitten wir dich für bestimmte Informationstypen (z.B.
    Kontaktinformationen) einige Beispielwerte anzugeben. Diese Eingaben
    werden im dritten Teil verwendet um die eigentlichen
    Datenschutzpräferenzen zu ermitteln, z.B. welchen Personen du
    welche Kontaktinformationen weitergeben würdest.

    Abschließend liefert dir eine erste Auswertung deiner Antworten am
    Ende der Umfrage einige Information über dein Vorgehen bei der
    Freigabe persönlicher Informationen.

    ## Datenschutzhinweis

    Die von dir eingegebenen Personen und
    Beispiele für persönliche Informationen werden **nicht
    gespeichert**, sondern ausschließlich für die individuelle
    Formulierung der Fragen im dritten Teil der Umfrage verwendet.
    Tatsächlich spielt es für uns keine Rolle, ob du bei den Personen
    volle Namen, Spitznamen oder kryptische Pseudonyme eingibst.
    Hauptsache *du* kannst mit den Personennamen etwas anfangen,
    wenn sie im dritten Umfrageteil wieder auftauchen. Simultan verhält
    es sich mit den Angaben persönlicher Informationen.
    Gespeichert werden lediglich die Antworten im dritten Teil. Dabei
    werden alle Eingaben durch Nummern ersetzt, so dass
    wir keine Rückschlüsse auf tatsächliche Eingaben oder deine Person
    machen können. Alle gespeicherten Daten sind komplett anonymisiert.

    ## Kontakt

    Bei Fragen schreibe einfach eine Mail an
    <christian.buennig@uni-rostock.de>.

    """,
'start_ack': "Weiter",
'finish': """
    # Vielen Dank für die Teilnahme!
    """,
'privacy': """
    Die hier eingegebenen Namen oder Bezeichnungen werden ausschließlich
    für die Formulierung nachfolgender Fragen verwendet und **nicht
    gespeichert**!
    """,
```

```
    'setnames_title': "Personen aus deinem Umfeld",
    'setnames_intro': """
        Die nachfolgenden Fragen beziehen sich auf konkrete, dir bekannte
        Personen. Dafür müssen nun 10-20 Personen aus deinem Bekanntenkreis
        (privat, familiär, beruflich u.s.w.) benannt werden.

        Wichtig ist, dass nicht *nur* Freunde oder *nur* Verwandte
        angegeben werden. Es können z.B. auch Personen angegeben werden,
        mit denen man eher selten zu tun hat.
        """,
    'setnames_need_more': """
        Das hat leider nicht gepasst.
        Es sind %d oder mehr Personen erforderlich.
        """,
    'setvalues_title': "Persönliche Informationen",
    'setvalues_need_more': """
        Das hat leider nicht gepasst.
        Es sind %d oder mehr Einträge notwendig.
        """,
    'disclose_title': "Weitergabe persönlicher Informationen",
    'disclose_persons': """
        Die Antworten auf dieser Seite beziehen sich auf die folgende
        Personengruppe:
        """,
    'badstep': """# Wiederholte Eingabe
        <div class="inputerror">

        **Bereits getätigte Eingaben können leider nicht wiederholt bzw.
        geändert werden.**

        </div>
        """,
    'badstep_ack': "Okay",
},

# ------------------------------------------------------------------------
# information types
# ------------------------------------------------------------------------

'itypes': [
    { 'name': 'Aufenthaltsorte', 'plotname': "Location",
      'desc_input': """
          Nenne bitte eine Reihe von Orten an, an denen du dich
          regelmäßig oder gelegentlich aufhältst (das kann täglich oder
          auch einmal im Jahr bedeuten). Beispieleingaben wären *zu Hause*,
          *Büro*, *Strand*, *Lieblingskneipe*, *im Auto*, ...

          Es sind mindestens %d Eingaben erforderlich.
      """,
      'desc_select': """
          Stell dir vor, die oben genannten Personen sind an deinem
          Aufenthaltsort interessiert. Bei welchen der folgenden Orte wäre
          es okay, wenn *alle* oben genannten Personen den Ort bei Nachfrage
          automatisch erfahren, d.h. ohne dass du die Weitergabe deines
          Ortes noch einmal explizit bestätigst.
      """,
      'nvalues_min': 3,
      'nvalues_max': 10,
```

```
},
{ 'name': 'Telefon & IM', 'plotname': "Phone",
  'desc_input': """
      Nenne bitte die Telefonnummern oder Instant-Messaging-Kontakte,
      unter denen du zu erreichen bist. Hier sind nicht tatsächliche
      Nummern von Interesse, sondern deren Bezeichnungen, z.B. *Privat*,
      *Mobil*, *Skype*, *ICQ*, ...

      Es sind mindestens %d Eingaben erforderlich.
  """,
  'desc_select': """
      Welche der folgenden Telefonnummern oder IM-Kontakte würdest du
      *allen* oben genannten Personen vorbehaltlos weitergeben?
  """,
  'nvalues_min': 2,
  'nvalues_max': 6,
},
{ 'name': 'E-Mail',
  'desc_input': """
      Nenne bitte deine verschiedenen E-Mail-Konten, die du verwendest.
      Wie bei den Telefonnummern sind hier keine tatsächlichen
      Nummern erforderlich, sondern frei wählbare Bezeichnungen wie
      *Privat*, *Uni*, *Sportverein*, ...

      Es sind mindestens %d Eingaben erforderlich.
  """,
  'desc_select': """
      Welche der folgenden E-Mail-Adressen würdest du *allen* oben
      genannten Personen vorbehaltlos weitergeben?
  """,
  'nvalues_min': 2,
  'nvalues_max': 6,
},
{ 'name': 'Musikgeschmack', 'plotname': "Music",
  'desc_input': """
      Nenne bitte einige Künstler oder Alben, die deinen Musikgeschmack
      möglichst breit wiedergeben (also nicht nur die 5 Lieblingsbands).

      Wichtig ist, dass die Auswahl deinen tatsächlichen Geschmack
      beschriebt und nicht etwa einem *öffentlichen* Profil deines
      Musikgeschmacks entspricht. Da es in dieser Umfrage um
      Datenschutz-Präferenzen geht, sind auch eher *private* Angaben
      sehr wichtig.

      Es sind mindestens %d Eingaben erforderlich.
  """,
  'desc_select': """
      Stell dir vor du verbringst einen geselligen Abend mit den oben
      genannten Personen. Für die musikalische Untermalung soll jeder
      etwas Musik beisteuern. Welche Auswahl würdest du bei dieser
      Gruppe treffen?
  """,
  'nvalues_min': 5,
  'nvalues_max': 10,
},
{ 'name': 'Filmgeschmack', 'plotname': "Movies",
  'desc_input': """
      Nenne bitte ein paar Filme, die deinen Filmgeschmack möglichst
```

```
            breit wiedergeben. Ob es nun alte, neue, bereits gesehene oder
            noch anzuschauende Filme sind, spielt keine Rolle.

            Wichtig ist, dass die Auswahl deinen tatsächlichen Geschmack
            beschriebt und nicht etwa einem *öffentlichen* Profil deines
            Filmgeschmacks entspricht. Da es in dieser Umfrage um
            Datenschutz-Präferenzen geht, sind auch eher *private* Angaben
            sehr wichtig.

            Es sind mindestens %d Eingaben erforderlich.
        """,
        'desc_select': """
            Stell dir vor du veranstaltest einen Filmabend mit den oben
            genannten Personen und du bist für das Filmprogramm verantwortlich.
            Welche der nachfolgenden Filme könnten Teil dieses Filmprogramms
            sein?
        """,
        'nvalues_min': 5,
        'nvalues_max': 10,
    },
],
}
```

Interview Screenshots

The interview specification shown in the previous section results in the following steps of the online interview.

Umfrage zur Freigabe persönlicher Informationen

Durch die Teilnahme an dieser Umfrage unterstützt du ein Dissertations-Projekt zum Thema Datenschutz. Vielen Dank schon mal! Ziel der Umfrage ist es, strukturelle Informationen in Datenschutzpräferenzen zu erfassen. Einen Eindruck davon bekommst du am Ende der Umfrage durch eine erste Auswertung deiner Antworten.

Überblick

Die Umfrage besteht aus drei Teilen. Da Datenschutz eine individuelle Angelegenheit ist, müssen auch die Fragen dazu individuell formuliert werden, z.B. sollten sie sich auf Personen beziehen die du kennst. Die ersten beiden Teile der Umfrage dienen dazu diese individuellen Informationen zu erfassen.

Im ersten Teil wirst du gebeten, einige Personen aus deinem Umfeld anzugeben, z.B. Freunde, Verwandte oder Kollegen. Im zweiten Teil bitten wir dich für bestimmte Informationstypen (z.B. Kontaktinformationen) einige Beispielwerte anzugeben. Diese Eingaben werden im dritten Teil verwendet um die eigentlichen Datenschutzpräferenzen zu ermitteln, z.B. welchen Personen du welche Kontaktinformationen weitergeben würdest.

Abschließend liefert dir eine erste Auswertung deiner Antworten am Ende der Umfrage einige Information über dein Vorgehen bei der Freigabe persönlicher Informationen.

Datenschutzhinweis

Die von dir eingegebenen Personen und Beispiele für persönliche Informationen werden **nicht gespeichert**, sondern ausschließlich für die individuelle Formulierung der Fragen im dritten Teil der Umfrage verwendet. Tatsächlich spielt es für uns keine Rolle, ob du bei den Personen volle Namen, Spitznamen oder kryptische Pseudonyme eingibst. Hauptsache *du* kannst mit den Personennamen etwas anfangen, wenn sie im dritten Umfrageteil wieder auftauchen. Simultan verhält es sich mit den Angaben persönlicher Informationen. Gespeichert werden lediglich die Antworten im dritten Teil. Dabei werden alle Eingaben durch Nummern ersetzt, so dass wir keine Rückschlüsse auf tatsächliche Eingaben oder deine Person machen können. Alle gespeicherten Daten sind komplett anonymisiert.

Kontakt

Bei Fragen schreibe einfach eine Mail an christian.buennig@uni-rostock.de.

Weiter

Figure A.2: Interview welcome page.

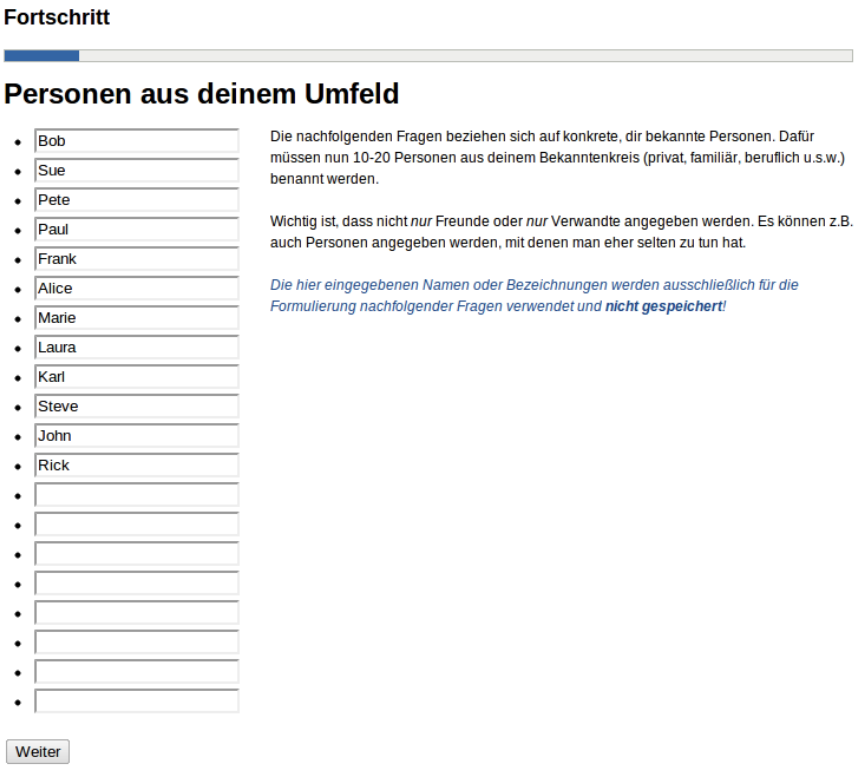

Figure A.3: In the first interview step participants are asked to name persons from their social environment.

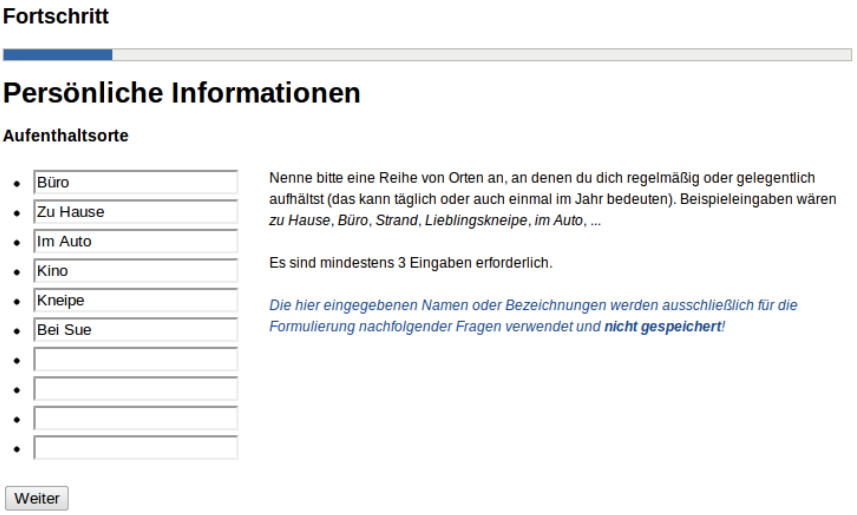

Figure A.4: The interview steps 2 to 6 request participants to list personal information items they potentially may share with other persons. This page ask for some places the participant frequently or occasionally is located at.

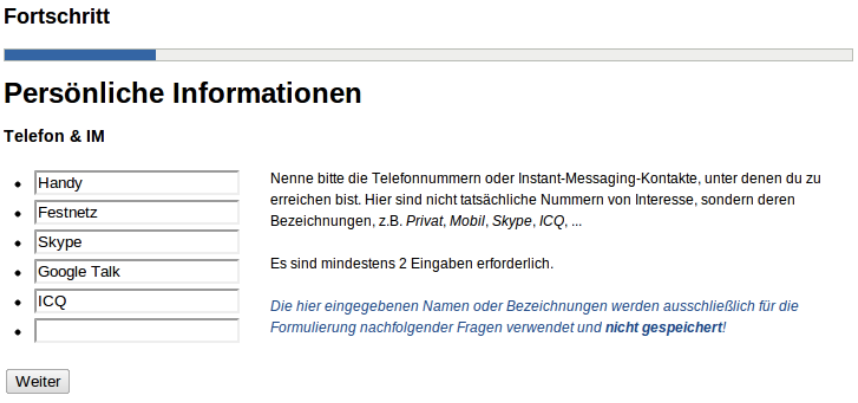

Figure A.5: Participants are asked to list real-time contact modalities they are using.

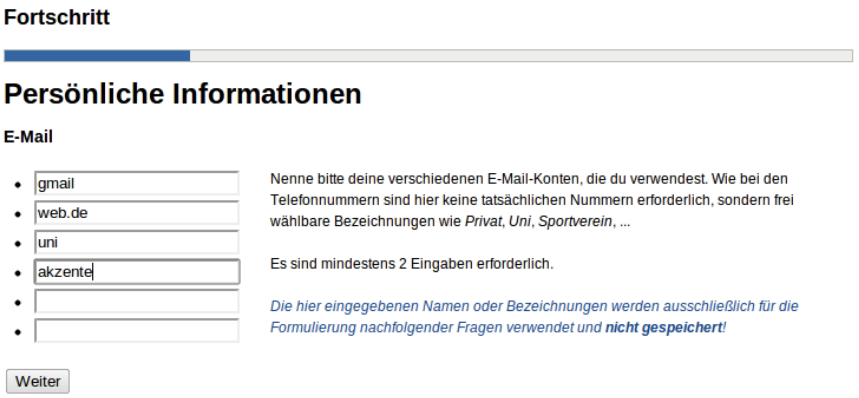

Figure A.6: Step 5 of the interview asks for the different e-mail accounts a participant uses.

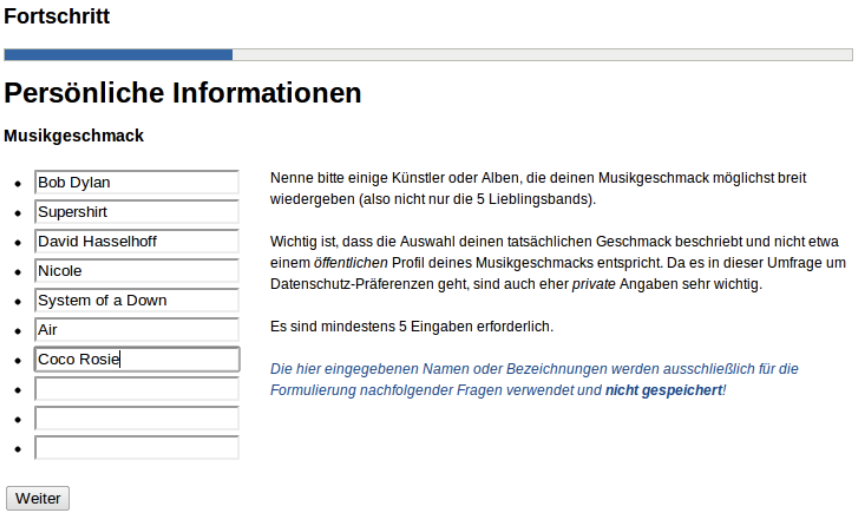

Figure A.7: In this step participants are supposed to list music artists or albums they are listening too from time to time.

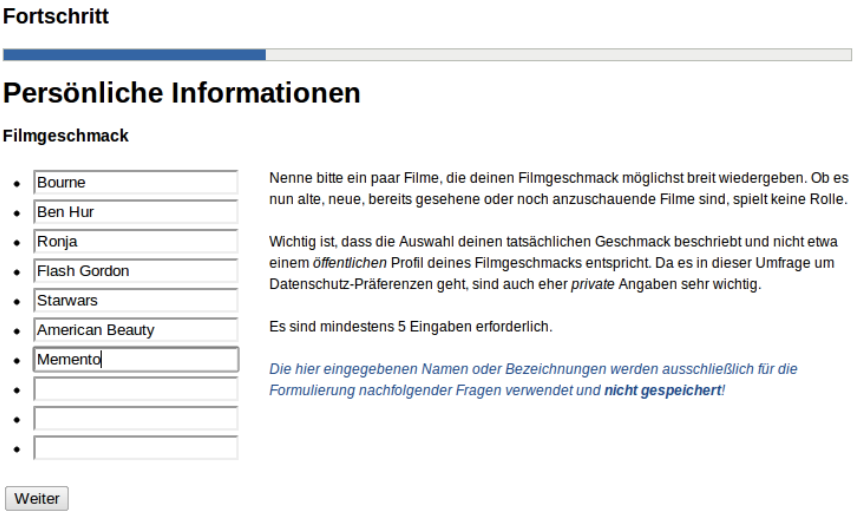

Figure A.8: The interview step 7 asks for movies participants have watched or plan to watch.

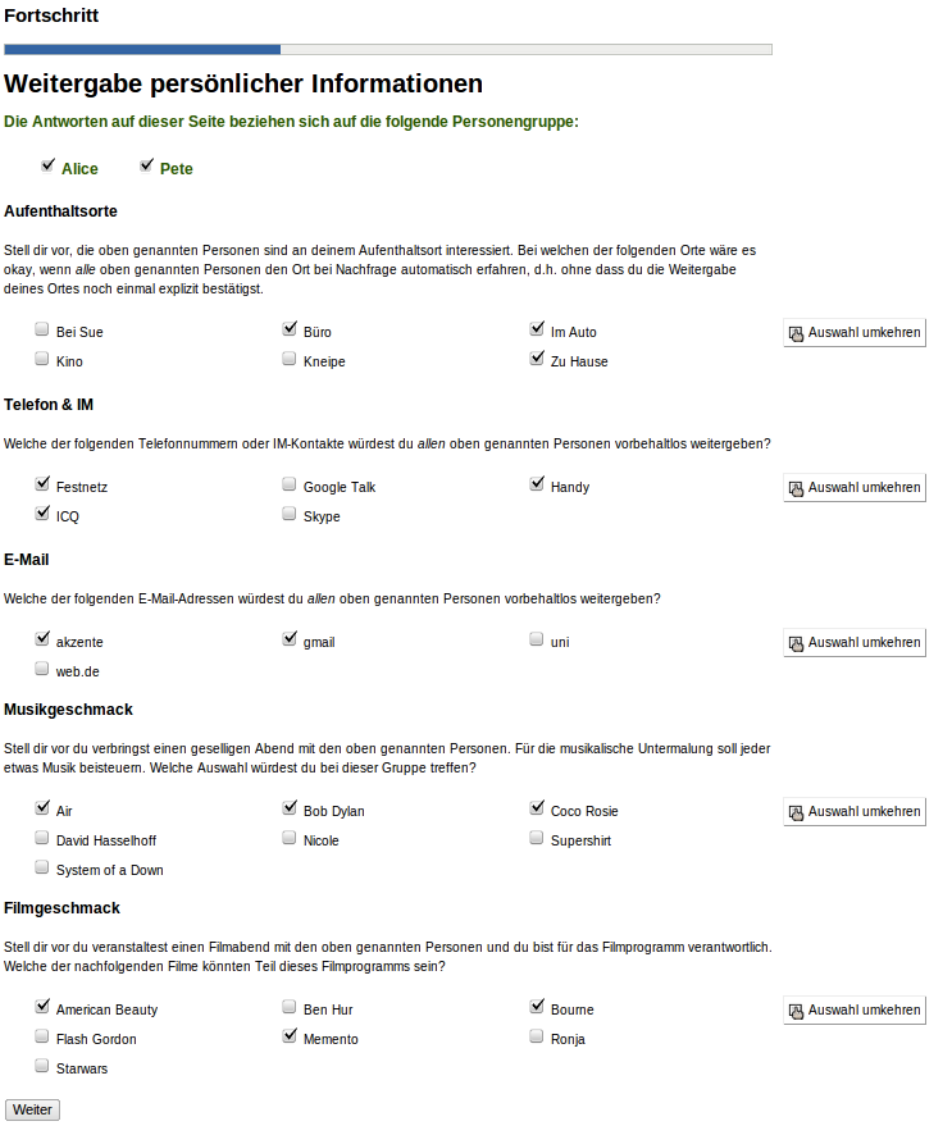

Figure A.9: This and the next 12 steps of the interview capture the actual disclosure behavior (the previous questions only collected data to personalize questions in this part.) For 13 different group constellations of the persons given in the first step, participants must state which of the previously specified personal information items they are willing to share.

B DiLES

This appendix chapter provides methodological details about the evaluations made in chapter 6 (Learning Disclosure Decisions). For the sake of transperency and reproducibility, it briefly describes the setup, usage, and extension of the used evaluation system DiLES.

B.1 Introduction

DiLES is an evaluation system for learning methods supposed to conceptualize information disclosure preferences in social interaction within smart environments. It is a Python application which uses own learning algorithms as well as algorithms provided by external libraries like LibSVM[1], Orange[2] and SciPy[3]. Evaluation results are rendered using matplotlib[4] and presented within an interactive HTML-based interface.

Software and data related to DiLES is hosted at the Open Science Repository[5] of the Computer Science Department at Rostock University.

B.2 Setup

Buildout[6] is used to set up the development and usage environment (DiLES is not meant to get installed but used directly within the source tree).

To create a buildout, run:

```
$ python bootstrap.py
$ bin/buildout # this one fails (see buildout.cfg for details)
$ bin/buildout # this one and subsequent calls should succeed
```

[1] LibSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[2] Orange: http://orange.biolab.si/
[3] SciPy: http://www.scipy.org/
[4] matplotlib: http://matplotlib.sourceforge.net/
[5] Open Science Repository: http://opsci.informatik.uni-rostock.de/index.php/DiLES
[6] Buildout: http://www.buildout.org/

As a result, there are some ready to use scripts in the `bin` directory. Ready to use means, all dependencies are installed and used properly. These scripts provide some help when run with the `--help` option.

B.3 Usage

Defining Scenarios

Scenarios are described in YAML[7]. They contain of a list of dictionaries of context information and disclosures, each describing a single disclosure situation. However, the first dictionary is not handled as a situation but as a default dictionary providing default values for information items not given in subsequent dictionaries. The only required items in a situation dictionary are *subject* (the one who has to decide a disclosure), *persons* (a list of potential information receiver), and *disclosure* (a list of information items to disclose). Disclosure information items must be strings which optionally encodes a hierarchical structure using a dot-separator. An simple example scenario might look like this:

```
# first, the defaults
- medium: display-wall
- room: 123
- tags: []

# situation 1
- subject: Bob
- persons: Sue
- trigger: start-meeting
- tags: meeting
- disclosure:
    - workspace.project-x
    - workspace.project-y
    - contact

# situation 2
- subject: Bob
- persons: Sue, Paul
- trigger: start-meeting
- tags: meeting
- disclosure:
    - workspace.project-x.overview
    - contact.business
```

This scenario describes two situations in which Bob decides the disclosure of information for collaboration purposes at the beginning of a meeting. The first situation is a meeting of

[7] YAML: `http://yaml.org/`

Bob and Sue only. Here Bob shares all working documents related to projects x and y, as well as contact information. In the second situation, this time with Paul as an additional participant, Bob shares only an overview document of project x and business related contact information. The context items *tags* and *trigger* describe the circumstances of information disclosures and are just examples of how situations may be described besides the requires items *subject*, *persons*, and *disclosure*.

In case a scenario includes situations where subjects may disclose information using one of multiple possible modalities, the chosen modality may be encoded within the disclosure a an itme's prefix, separated by a colon:

```
- subject: Bob
- persons: Sue
- medium: wall, mobile
- disclosure:
    - wall:workspace.project-x.overview
```

Scenario preprocessors handle such colon-separated modality information in different ways.

Using the Scripts

Once a buildout of DiLES has been created, several scripts in the `bin` directory are provided. Each scripts provides usage instructions when run with the `--help` option. Following is a short overview about the scripts.

`diles-evaluate-scenario`:
: Evaluates preprocessors and learners with regard to a given scenario.

`diles-join-evaluation-results`:
: Joins multiple result files generated by *diles-evaluate-scenario*. This is useful when a scenario has been evaluated in multiple steps, each only considering certain subjects, preprocessors, and learners (which may be necessary for memory-usage reasons).

`diles-plot-evaluation-results`:
: Renders plots for an evaluation results file as produced by the script *diles-evaluate-scenario*.

`diles-rank-evaluation-results`:
: Generates various *global* rankings of evaluation results, e.g. which validators performed best in general (i.e. not in context of a specific subject, preprocessor or learning method).

`diles-render-plot-summary`:
: Renders HTML-based interactive interface to the plots rendered by the script *diles-plot-evaluation-results*.

`diles-analyze-scenario`:
: Analyzes a scenario and renders corresponding plots. These plots are recognized by `diles-render-plot-summary`. These scenario analysis plots help in correlating evaluation results with disclosure patterns.

`diles-convert-dihabs-results-to-scenario`:
: Converts the results of a DiHabs survey to a DiLES scenario file.

`tests`:
: Runs all or selected tests (written as DOCTESTS[8]).

`python`:
: A Python interpreter with access to the `diles` package (for interactive testing and debugging).

B.4 Packages

The package `diles` consists of several sub-packages and modules. Learning-related functionality is grouped in the sub-package `diles.learn`. Functionality related to scenario evaluation is grouped in the sub-package `diles.scenario`. Modules contained in the main package `diles` provide generally used utilities. Inspect the software's README for more detailed information.

B.5 Extensions

Adding new scenario preprocessors, base learners, wrapping learners, or validators is easy, as shown in the following sub-sections.

Next to the instructions below, it pays off to inspect the source code directly which contains comprehensive and detailed documentation. Especially the DOCTESTS[9] are very helpful in understanding how different units of DiLES work and interact.

Adding New Base Learners

To add a new base learner, create a new module in the package `diles.learn.learners` which contains a class deriving from `diles.learn.Learner`. To get automatically recognized, this class' name must end with *Learner*. The class must define a static attribute named `paramspace` which holds a dictionary mapping the learner's constructor parameters

[8] DOCTESTS: `http://docs.python.org/library/doctest.html`
[9] DOCTESTS: `http://docs.python.org/library/doctest.html`

to lists of possible values. This *paramspace* is used for finding optimal learner configurations. The constructor must accept these parameters as keywords.

Following is an example, providing a simple guessing learner:

```
class GuessLearner(diles.learn.Learner):

    paramspace = {'foo': [1,2,3,4]}

    def __init__(self, foo=1):
        self.foo = 1 # unused, just for the paramspace example

    def train(self, samples, labels, fwl=None, fwt=0):
        label = labels[0]
        confidence = 0.2
        ranking = None
        self.prediction = label, confidence, ranking

    def predict(self, sample):
        # we always predict the same
        return self.prediction
```

As seen in this example, the prediction method of a learner is expected to return a 3-tuple: the predicted label, a confidence value between 0 and 1 and optionally a ranking of all known labels (i.e. a list of rank-value/label pairs). This ranking is mainly used for debugging purposes and may also be *None* as in this example.

For more sophisticated examples inspect the package `diles.learn.learners`.

Adding New Wrapping Learners

To add a new wrapping learner, create a new module in the package `diles.learn.wrapppers` which contains a class deriving from `diles.learn.Learner`. To get automatically recognized as a wrapper, this class' name must end with *Wrapper*. Similar to base learners, the class must define a static attribute named `paramspace` which holds a dictionary mapping the wrappers's constructor parameter names to lists of possible values.

In contrast to base learners, wrapping learner constructors must accept as first argument the class of a base learner. Additionally, next to parameters used by the wrapper itself, the constructor must accept arbitrary parameters for the base learner. The following example explains these requirements more illustratively:

```
class DummyWrapper(diles.learn.Learner):
```

```
paramspace = {'bar': ['a', 'b']}

def __init__(self, basecls, bar='a', **baseparams):
    super(DummyWrapper, self).__init__()
    self.bar = bar # unused, just for illustration purposes
    self.model = basecls(**baseparams)

def train(self, samples, labels, fwl=None, fwt=0):
    self.model.train(samples, labels, fwl=fwl, fwt=fwt)

def predict(self, sample):
    return self.model.predict(sample)
```

Of course, this is a rather useless wrapper which directly forwards any call to an instance of its base learner class. A dummy wrapper for the above mentioned guessing learner could be instantiated like this:

```
dw = DummyWrapper(GuessLearner, bar='b', foo=3)
```

Adding New Validators

Validators are methods of the class `Validator` in the module `diles.learn.validator`, following a certain naming pattern. In particular any method whose name starts with `_vld_` is considered as a validator method. A validator must return *True* to support a prediction or *False* to prevent it. The existing validator methods in the referenced module shall serve as examples for implementing additional validators.

Adding New Preprocessors and Combinations

Preprocessors are defined in the module `dile.scenario.preprocessors`. A preprocessor function expects a list of situations from a scenario and must return a new list of situations. Preprocessors may be chained. Adding a new preprocessor means to add a new function to the referenced module and list this function in at least one chain given by the module's `chains` attribute. Again, inspect the referenced module's source for explanatory examples.

C Scenarios

This chapter provides additional information on the scenarios which lead to the DiLES results presented in section 6.6.8. The next section is a detailed description of the manually composed scenario around the subject *Bob*. For each disclosure situation it lists the situation parameters available as context information to learning methods as well as the actually disclosed information sets. The subsequent section refers to the scenario generated from the answers participants made in the DiHabs survey presented in section 4.3.

C.1 Manually Composed Scenario

This manually composed scenario is about Bob, a web developer working as a freelancer. Sue contacts Bob and requests him to develop and set up a customized WCMS for her company. The scenario describes several meetings which take placing during the lifetime of this project.

The communication modalities used in this scenario are a display wall and direct transmissions of information to mobile devices of meeting participants. Each disclosure situation has a trigger — the triggers used here are the start and the end of a meeting (technically this is just another context information). At the end of meetings, usually contact information is shared. In contrast to exchange of contact information as it happens today, where an initial but temporarily unlimited exchange is sufficient, this scenario envisions a temporarily limited access-token—based information exchange where tokens have to be renewed regularly.

Specific characteristics of this scenario are a distinct hierarchical structure of disclosed personal information and a general order-reversing mapping among disclosure situations.

The main purpose of this scenario is to illustrate the composition of scenarios to evaluate with DiHabs. For this reason it is kept rather simple. More complex scenarios could involve additional context information, more diversified order mapping types, and a greater set of participating persons.

First Meeting

Bob and Sue meet to get to know each other.

Bob contributes a high quality portfolio of references of selected previous work, using the impressive display wall.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.x.reference wall:documents.projects.y.reference wall:documents.projects.z.reference
Repeated	1 times	
Modalities	distribute, wall	
Persons	Sue	

At the end of the meeting Bob hands out a low-resolution and watermarked portfolio, contact information and access to his calendar (Sue can see possible times for a next meeting and schedule one). All information items are distributed directly to Sue's mobile device.

Subject	Bob	Disclosure
Trigger	end-of-meeting	distribute:documents.projects.x.reference.lowres distribute:documents.projects.y.reference.lowres distribute:documents.projects.z.reference.lowres distribute:contact.phone.mobile distribute:contact.phone.work distribute:calendar.view.free-meeting-times distribute:calendar.edit.schedule-meeting
Repeated	1 times	
Modalities	distribute, wall	
Persons	Sue	

Second Meeting

Second meeting of Bob and Sue, know with Paul, a partner developer of Bob. This meeting mainly is about technical details.

Bob shares some first drafts as a starting point for discussions.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.drafts
Repeated	1 times	
Modalities	distribute, wall	
Persons	Sue, Paul	

Contact information exchange at the end of the meeting. All participants receive grants to utilize several standard communication channels.

Subject	Bob	Disclosure
Trigger	end-of-meeting	distribute:contact.mail.work
Repeated	1 times	distribute:contact.phone.mobile
Modalities	distribute, wall	distribute:contact.phone.work
Persons	Sue, Paul	distribute:calendar.view.free-meeting-times
		distribute:calendar.edit.schedule-meeting

In the same meeting, Paul is granted more general access to contact information.

Subject	Bob	Disclosure
Trigger	end-of-meeting	
Repeated	1 times	distribute:contact distribute:calendar.view
Modalities	distribute, wall	distribute:calendar.edit.schedule-meeting
Persons	Paul	

Working Meetings

Bob has several meetings with Paul to work out specific elements of the site.

In addition to the documents contained in the project's workspace, Bob shares - as usual - some generic website projects related working documents.

Subject	Bob	Disclosure
Trigger	start-of-meeting	
Repeated	5 times	wall:documents.projects.generic
Modalities	distribute, wall	wall:documents.projects.w
Persons	Paul	

As usual, Bob grants Paul a complete view and limited editing rights to his calendar as well as complete contact information at the end of the meeting.

Subject	Bob	Disclosure
Trigger	end-of-meeting	
Repeated	5 times	distribute:contact distribute:calendar.view
Modalities	distribute, wall	distribute:calendar.edit.schedule-meeting
Persons	Paul	

Bob meets with Dent, a painter/artist, to work on some graphics. He shows Dent the drafts of the site.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.drafts
Repeated	4 times	
Modalities	distribute, wall	
Persons	Dent	

For future collaboration, Bob prefers plain phone and mail contact.

Subject	Bob	Disclosure
Trigger	start-of-meeting	distribute:contact.phone.work distribute:contact.mail.work
Repeated	4 times	
Modalities	distribute, wall	
Persons	Dent	

Bob meets with Sue, to discuss the progress.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.drafts
Repeated	2 times	
Modalities	distribute, wall	
Persons	Sue	

Bob meets with Sue, to discuss the progress.

Subject	Bob	Disclosure
Trigger	end-of-meeting	distribute:contact.phone.mobile distribute:contact.phone.work distribute:calendar.view.free-meeting-times distribute:calendar.edit.schedule-meeting
Repeated	2 times	
Modalities	distribute, wall	
Persons	Sue	

Another technical meeting with Sue and Paul.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.drafts
Repeated	1 times	
Modalities	distribute, wall	
Persons	Sue, Paul	

Another technical meeting with Sue and Paul.

Subject	Bob	Disclosure
Trigger	end-of-meeting	distribute:contact.phone.mobile distribute:contact.phone.work distribute:calendar.view.free-meeting-times distribute:calendar.edit.schedule-meeting
Repeated	1 times	
Modalities	distribute, wall	
Persons	Sue, Paul	

Again, the contact information exchange with Paul is more open.

Subject	Bob	Disclosure
Trigger	end-of-meeting	distribute:contact distribute:calendar.view distribute:calendar.edit.schedule-meeting
Repeated	1 times	
Modalities	distribute, wall	
Persons	Paul	

Final meetings

Bob meets with Dent, to check how the graphics integrate with the current site.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.release distribute:contact.phone.work distribute:contact.mail.work
Repeated	2 times	
Modalities	distribute, wall	
Persons	Dent	

In the final meetings with Sue, Bob presents release candidates of his work.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.release
Repeated	2 times	distribute:contact.phone.mobile
Modalities	distribute, wall	distribute:contact.phone.work
Persons	Sue	distribute:calendar.view.free-meeting-times
		distribute:calendar.edit.schedule-meeting

Since future collaboration is less likely now, Bob shares only limited contact information with Sue.

Subject	Bob	Disclosure
Trigger	start-of-meeting	wall:documents.projects.w.release
Repeated	2 times	distribute:contact.phone.work
Modalities	distribute, wall	
Persons	Sue	

C.2 Scenario Generated from DiHabs Survey

The scenario generated from the DiHabs survey is too large to be shown here completely (59 subjects, each with 65 disclosure situations). Instead the reader is referred to the online available evaluation results mentioned in section 6.6.8[1]. The linked result view includes a complete dump of the scenario data.

[1] DiLES evaluation results: `http://opsci.informatik.uni-rostock.de/index.php/DiLES`.

Abbreviations and Symbols

General Abbreviations

AmI	Ambient Intelligence
APPEL	A P3P Preference Exchange Language
CBR	Case-based reasoning
IoT	Internet of Things
IQR	Interquartile range
k-NN	k-Nearest Neighbors
P3P	Platform for Privacy Preferences
RL	Rule Learner (learner method ID)
UDHR	Universal Declaration of Human Rights

Learner, Wrapper and Preprocessor IDs

BR	Binary Relevance Wrapper
DT	Decision Tree Learner
GS	Guess Majority Learner
HBR	Hierarchical Binary Relevance Wrapper
KNN	k-Nearest Neighbors Learner
MR	Root Modality Preprocessor
ML	Leaf Modality Preprocessor
NB	Naive Bayes Learner
OMI	Order-Mapping Interpolation Wrapper
PS	Powerset Wrapper
RL	Rule Induction Learner
SVM	Support Vector Machine Learner

Mathematical Symbols

$\mathcal{P}$	Powerset operator: $\mathcal{P}(X)$ evaluates to the powerset of X
π	Tuple projector: $\pi_n(t)$ evaluates to the n-th element in tuple t, e.g. $\pi_2((a,b,c,d)) = b$
$\parallel$	In context of sets, this denotes the relation of unequal sets where neither one is a subset of the other, i.e. $A \parallel B \Leftrightarrow A \not\subseteq B \wedge A \not\supseteq B$

Bibliography

Aarts, E., & Encarnação, J. 2006. Into Ambient Intelligence. *Pages 1–17 of:* Aarts, Emile, & Encarnação, José (eds), *True Visions*. Springer.

Ackerman, Mark S. 2000. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Hum.-Comput. Interact.*, **15**(2), 179–203.

Ackerman, Mark S., Cranor, Lorrie Faith, & Reagle, Joseph. 1999. Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences. *Pages 1–8 of: Proc. of the 1st ACM Conf. on Electronic Commerce*. EC '99. New York, NY, USA: ACM.

Adams, Anne. 2000. Multimedia Information Changes the Whole Privacy Ballgame. *Pages 25–32 of: CFP '00: Proc. of the 10th Conf. on Computers, Freedom and Privacy*. New York, NY, USA: ACM.

Al-Muhtadi, Jalal, Ranganathan, Anand, Campbell, Roy, & Mickunas, M. Dennis. 2002a. A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments. *Pages 771–776 of: ICDCSW '02: Proc. of the 22nd Intl. Conf. on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society.

Al-Muhtadi, Jalal, Campbell, Roy, Kapadia, Apu, Mickunas, M. Dennis, & Yi, Seung. 2002b. Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. *Page 74 of: ICDCS '02: Proc. of the 22nd Intl. Conf. on Distributed Computing Systems*. Washington, DC, USA: IEEE Computer Society.

Al-Muhtadi, Jalal, Hill, Raquel, Campbell, Roy, & Mickunas, M. Dennis. 2006. Context and Location-Aware Encryption for Pervasive Computing Environments. *PerComW'06: Fourth Annual IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, **00**, 283–289.

Altman, Irwin. 1975. *The Environment and Social Behavior: Privacy, Personal Space, Territory, and Crowding.* Monterey, CA, USA: Brooks/Cole Publishing Company.

Augusto, Juan Carlos. 2007. Ambient Intelligence: The Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence. *Pages 213–234 of:* Schuster, Alfons J. (ed), *Intelligent Computing Everywhere*. Springer.

Barrett, Rob, & Maglio, Paul P. 1998. Informative Things: How to Attach Information to the Real World. *Pages 81–88 of: UIST '98: Proc. of the 11th Annual ACM Symposium on User Interface Software and Technology.* New York, NY, USA: ACM.

Beckwith, R. 2003. Designing for Ubiquity: The Perception of Privacy. *Pervasive Computing, IEEE*, **2**(2), 40–46.

Bellotti, Victoria, & Sellen, Abigail. 1993. Design for Privacy in Ubiquitous Computing Environments. *Pages 77–92 of: ECSCW'93: European Conf. on Computer-Supported Cooperative Work.* Norwell, MA, USA: Kluwer Academic Publishers.

Ben-Kiki, Oren, Evans, Clark, & döt Net, Ingy. 2009. *YAML Specification.* Available at `http://yaml.org/spec` (accessed April 12, 2011).

Beresford, Alastair R., & Stajano, Frank. 2003. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, **02**(1), 46–55.

Besmer, Andrew, Watson, Jason, & Lipford, Heather Richter. 2010. The Impact of Social Navigation on Privacy Policy Configuration. *Pages 7:1–7:10 of: Proc. of the 6th Symposium on Usable Privacy and Security.* SOUPS '10. New York, NY, USA: ACM.

Bessler, Sandford, & Jorns, Oliver. 2005. A Privacy Enhanced Service Architecture for Mobile Users. *PerComW'05: Third Annual IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, **00**, 125–129.

Bohn, Jürgen, Coroama, Vlad, Langheinrich, Marc, Mattern, Friedemann, & Rohs, Michael. 2003. Disappearing Computers Everywhere – Living in a World of Smart Everyday Objects. *In: Proc. of New Media, Technology and Everyday Life in Europe Conf.*

Bünnig, Christian. 2008. Learning Context Based Disclosure of Private Information. *In: Proc. of European Research Workshops (Internet of Things & Services) at Smart Event 2008.* Strategies Telecoms & Multimedia.

Bünnig, Christian. 2009a. Simulation and Analysis of Ad Hoc Privacy Control in Smart Environments. *Pages 307–318 of: Proc. of Intelligent Interactive Assistance and Mobile Multimedia Computing, IMC 2009.* CCIS, vol. 53. Springer.

Bünnig, Christian. 2009b. Smart Privacy Management in Ubiquitous Computing Environments. *Pages 131–139 of: HCII 2009 - Human Interface and the Management of Information.* LNCS, vol. 5618. Springer.

Bünnig, Christian. 2011a. *DiHabs - A Disclosure Habits Survey System.* Software and data available at `http://opsci.informatik.uni-rostock.de/index.php/DiHabs` (Open Science Repository of the Computer Science Department at Rostock University).

Bünnig, Christian. 2011b. *DiLES - A Disclosure Learning Evaluation System.* Software and data available at `http://opsci.informatik.uni-rostock.de/index.php/DiLES` (Open Science Repository of the Computer Science Department at Rostock University).

Bünnig, Christian, & Cap, Clemens H. 2007. Five Objectives to Diminish User Concerns About Privacy in Smart Environments. *Pages 179–188 of: Proc. of MoMM2007 & iiWAS2007 Workshops*. books@ocg.at, vol. 231. Austrian Computer Society.

Bünnig, Christian, & Cap, Clemens H. 2009. Ad hoc Privacy Management in Ubiquitous Computing Environments. *Pages 85–90 of: Proc. of 2nd Intl. Conf. on Advances in Human-Oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2009)*. IEEE Computer Society.

Burgelman, J.-C., & Punie, Y. 2006. Information, Society and Technology: Close Encounters of a Different Kind. *Pages 19–35 of:* Aarts, Emile, & Encarnação, José (eds), *True Visions*. Springer.

Burghardt, Christoph, & Kirste, Thomas. 2008. Synthesizing Probabilistic Models for Team-assistance in Smart Meetings Rooms. *In: Adjunct Proc. of the 2008 ACM Conf. on Computer Supported Cooperative Work, CSCW 2008*.

Burghardt, Christoph, Wurdel, Maik, Bader, Sebastian, Ruscher, Gernot, & Kirste, Thomas. 2011. Synthesising Generative Probabilistic Models for High-Level Activity Recognition. *Pages 209–236 of:* Chen, Liming, Nugent, Chris D., Biswas, Jit, & Hoey, Jesse (eds), *Activity Recognition in Pervasive Intelligent Environments*. Atlantis Ambient and Pervasive Intelligence, vol. 4. Atlantis Press.

Cadiz, J. J., & Gupta, Anoop. 2001. *Privacy Interfaces for Collaboration*. Tech. rept. MSR-TR-2001-82. Microsoft Research, Redmond, WA, USA.

Cas, J. 2005. Privacy in Pervasive Computing Environments - A Contradiction in Terms? *Technology and Society Magazine, IEEE*, **24**(1), 24–33.

Cesa-Bianchi, Nicolò, Gentile, Claudio, Tironi, Andrea, & Zaniboni, Luca. 2004. Incremental Algorithms for Hierarchical Classification. *In: Advances in Neural Information Processing Systems (NIPS) 2004*.

Chang, Chih-Chung, & Lin, Chih-Jen. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (accessed August 20, 2012).

Chaum, David L. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, **24**(2), 84–90.

Clark, Peter, & Niblett, Tim. 1989. The CN2 Induction Algorithm. *Machine Learning*, **3**, 261–283.

Clauss, Sebastian, Pfitzmann, Andreas, & Hansen, Marit. 2002. Privacy-Enhancing Identity Management. *The IPTS Report*, **67**(Sept.), 8–16.

Cochrane, Peter. 2000. Head to Head. *Sovereign Magazine*, Spring, 56–57. Available at `http://archive.cochrane.org.uk/opinion/archive/articles/prof.php` (accessed August 20, 2012).

Cook, Dian J., & Das, Sajal K. 2005. *Smart Environments - Technology, Protocols, and Applications*. Wiley.

Cranor, Lorrie Faith, Guduru, Praveen, & Arjula, Manjula. 2006. User Interfaces for Privacy Agents. *ACM Trans. Comput.-Hum. Interact.*, **13**(2), 135–178.

Dekker, Marnix, Etalle, Sandro, & Hartog, Jerry. 2007. Privacy Policies. *Pages 383–397 of:* Petković, Milan, & Jonker, Willem (eds), *Security, Privacy, and Trust in Modern Data Management*. DCSA. Springer.

Demšar, Janez, Zupan, Blaž, Leban, Gregor, & Curk, Tomaz. 2004. Orange: From Experimental Machine Learning to Interactive Data Mining. *Pages 537–539 of: Knowledge Discovery in Databases: PKDD 2004*. LNCS, vol. 3202. Springer. Project Website at `http://orange.biolab.si/` (accessed August 20, 2012).

Deutscher Bundestag. 1949 (May). *Grundgesetz der Bundesrepublik Deutschland*. Available at `http://www.bundestag.de/dokumente/rechtsgrundlagen/grundgesetz/` (accessed August 20, 2012).

Dey, Anind K., Salber, Daniel, & Abowd, Gregory D. 1999. *A Context-based Infrastructure for Smart Environments*. Tech. rept. Georgia Institute of Technology.

Diffie, W., & Landau, S.E. 2007. *Privacy on the Line: The Politics of Wiretapping and Encryption*. MIT Press.

DiGioia, Paul, & Dourish, Paul. 2005. Social Navigation as a Model for Usable Security. *Pages 101–108 of: Proc. of the 2005 Symposium on Usable Privacy and Security*. SOUPS '05. New York, NY, USA: ACM.

Ducatel, K., Bogdanowicz, M., F.Scapalo, Leitjen, J., & Burgelman, J-C. 2001 (Feb.). *Scenarios for Ambient Intelligence in 2010*. ISTAG Report, IPTS, European Commision. Available at `ftp://ftp.cordis.europa.eu/pub/ist/docs/istagscenarios2010.pdf` (accessed August 20, 2012).

Esquivel, Abraham, Haya, Pablo A., García-Herranz, Manuel, & Alamán, Xavier. 2007. Managing Pervasive Environment Privacy Using the "fair trade" Metaphor. *Pages 804–813 of: OTM Workshops*. LNCS, vol. 4806. Springer.

Etzioni, Amitai. 1999. *The Limits Of Privacy*. New York: Basic Books.

European Parliament and Council of the European Union. 1995 (Oct.). *Directive 95/46/EC*. Available at `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML` (accessed August 20, 2012).

European Parliament and Council of the European Union. 2002 (July). *Directive 2002/58/EC*. Available at `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32002L0058:EN:HTML` (accessed August 20, 2012).

Evans, David, Beresford, Alastair R., Burbridge, Trevor, & Soppera, Andrea. 2007. Context-Derived Pseudonyms for Protection of Privacy in Transport Mid-

dleware and Applications. *PerComW'07: Fifth Annual IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, **0**, 395–400.

Flaherty, David H. 1992. *Protecting Privacy in Surveillance Societies: The Federal Republic of Germany, Sweden, France, Canada, and the United States.* reprint edn. University of North Carolina Press.

Garfinkel, Simson. 2000. *Database Nation.* O'Reilly.

Gaver, William, Moran, Thomas, MacLean, Allan, Lövstrand, Lennart, Dourish, Paul, Carter, Kathleen, & Buxton, William. 1992. Realizing a Video Environment: EuroPARC's RAVE System. *Pages 27–35 of: Proceedings of the SIGCHI conference on Human factors in computing systems.* CHI '92. New York, NY, USA: ACM.

Grudin, Jonathan, & Horvitz, Eric. 2003. Presenting Choices in Context: Approaches to Information Sharing. *In: UBICOMP 2003 - Workshop on Ubicomp Communities - Privacy as Boundary Negotiation.*

Hansmann, Uwe, Merk, Lothar, Nicklous, Martin S., & Stober, Thomas. 2003. *Pervasivce Computing - The Mobile World.* 2 edn. Springer Perofessional Computing. Springer.

Henery, R. J. 1994. Classification. *In:* Michie, D., Spiegelhalter, D.J., & Taylor, C.C. (eds), *Machine Learning, Neural and Statistical Classification.* Prentice Hall.

Hong, Jason I., & Landay, James A. 2004. An Architecture for Privacy-Sensitive Ubiquitous Computing. *Pages 177–189 of: MobiSys '04: Proc. of the 2nd Intl. Conf. on Mobile Systems, Applications and Services.* New York, NY, USA: ACM.

Huberman, Bernardo A., Adar, Eytan, & Fine, Leslie R. 2005. Valuating Privacy. *IEEE Security and Privacy*, **3**, 22–25.

Jendricke, Uwe, Kreutzer, Michael, & Zugenmaier, Alf. 2002 (Oct.). *Pervasive Privacy with Identity Management.* Tech. rept. 178. Institut für Informatik, Universität Freiburg.

Jiang, Xiaodong, Hong, Jason I., & Landay, James A. 2002. Approximate Information Flows: Socially-Based Modeling of Privacy in Ubiquitous Computing. *Pages 176–193 of: Proc. of UbiComp 2002.* LNCS, vol. 2498/2002. London, UK: Springer.

Joinson, Adam N., Paine, Carina, Reips, Ulf-Dietrich, & Buchanan, Tom. 2006. Privacy and Trust: The role of situational and dispositional variables in online disclosure. *In: Proc. of Workshop "Privacy, Trust and Identity Issues for Ambient Intelligence" at 4th Intl. Conf. on Pervasive Computing (Pervasive).*

Kapadia, Apu, Henderson, Tristan, Fielding, Jeffrey, & Kotz, David. 2007. Virtual Walls: Protecting Digital Privacy in Pervasive Environments. *Pages 162–179 of: Proc. of the Fifth Intl. Conf. on Pervasive Computing (Pervasive).* LNCS, vol. 4480. Springer.

Kirste, T. 2006. Smart Environments. *Pages 323–339 of:* Aarts, Emile, & Encarnação, José (eds), *True Visions.* Springer.

Knowles, Elizabeth M. 1999. *The Oxford dictionary of quotations.* 5 edn. Oxford University Press, USA.

Kobsa, Alfred. 2007. Privacy-enhanced Personalization. *Commun. ACM*, **50**(8), 24–33.

Kobsa, Alfred, & Schreck, Jörg. 2003. Privacy Through Pseudonymity in User-Adaptive Systems. *ACM Trans. Inter. Tech.*, **3**(2), 149–183.

Konidala, Divyan M., Duc, Dang N., Lee, Dongman, & Kim, Kwangjo. 2005. A Capability-Based Privacy-Preserving Scheme for Pervasive Computing Environments. *PerComW'05: Third Annual IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, **00**, 136–140.

Kortuem, G., Kawsar, F., Fitton, D., & Sundramoorthy, V. 2010. Smart Objects as Building Blocks for the Internet of Things. *Internet Computing, IEEE*, **14**(1), 44 –51.

Langheinrich, Marc. 2001. Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems. *Pages 273–291 of: Proc. of UbiComp 2001.* LNCS, vol. 2201. Springer.

Langheinrich, Marc. 2002. A Privacy Awareness System for Ubiquitous Computing Environments. *Pages 315–320 of: Proc. of UbiComp 2002.* LNCS, vol. 2498/2002. Springer.

Lederer, Scott, Mankoff, Jennifer, Dey, Anind K., & Beckmann, Christopher P. 2003a (July). *Managing Personal Information Disclosure in Ubiquitous Computing Environments.* Tech. rept. UCB/CSD-03-1257. EECS Department, University of California, Berkeley.

Lederer, Scott, Mankoff, Jennifer, & Dey, Anind K. 2003b. Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. *Pages 724–725 of: CHI '03: Extended Abstracts on Human Factors in Computing Systems.* New York, NY, USA: ACM.

Lederer, Scott, Hong, Jason I., Dey, Anind K., & Landay, James A. 2004. Personal Privacy Through Understanding and Action: Five Pitfalls for Designers. *Personal Ubiquitous Computing*, **8**(6), 440–454.

Lessig, Lawrence. 2006. *Code - And Other Laws of Cyberspace - Version 2.0.* New York: Basic Books.

Lucky, Robert W. 2008. Reflections - Zero Privacy. *Spectrum, IEEE*, **45**(7), 20 –20.

Maibaum, Nico, Sedov, Igor, & Cap, Clemens H. 2002. A Citizen Digital Assistant for e-Government. *Pages 284–287 of: EGOV '02: Proc. of the First Intl. Conf. on Electronic Government.* London, UK: Springer.

Mattern, Friedemann. 2003 (May). From Smart Devices to Smart Everyday Objects. *Pages 15–16 of: Proc. of sOc'2003 (Smart Objects Conference).*

Muggleton, Stephen. 1994. Inductive Logic Programming: Derivations, Successes and Shortcomings. *SIGART Bull.*, **5**(1), 5–11.

Myles, Ginger, Friday, Adrian, & Davies, Nigel. 2003. Preserving Privacy in Environments with Location-Based Applications. *IEEE Pervasive Computing*, **02**(1), 56–64.

Nohara, Yasunobu, Inoue, Sozo, & Yasuura, Hiroto. 2005. Toward Unlinkable ID Management for Multi-Service Environments. *PerComW'05: Third Annual IEEE Intl. Conf. on Pervasive Computing and Communications Workshops*, **00**, 115–119.

Palen, Leysia, & Dourish, Paul. 2003. Unpacking "Privacy" for a Networked World. *Pages 129–136 of: CHI '03: Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM.

Pollach, Irene. 2007. What's wrong with online privacy policies? *Commun. ACM*, **50**(9), 103–108.

Prabaker, M., Rao, J., Fette, I., Kelley, P., Cranor, L., Hong, J., & Sadeh, N. 2007 (Sept.). *Understanding and Capturing People's Privacy Policies in a People Finder Application*. UBICOMP'07: Workshop on UBICOMP Privacy.

Pratt, Walter F. 1979. *Privacy in Britain*. Bucknell University Press.

Roach, Christopher. 2006. *Building Decision Trees in Python*. O'Reilly Python DevCenter Article. Available at `http://onlamp.com/pub/a/python/2006/02/09/ai_decision_trees.html` (accessed August 20, 2012).

Ronzani, Daniel. 2009. The Battle of Concepts: Ubiquitous Computing, Pervasive Computing and Ambient Intelligence in Mass Media. *Ubiquitous Computing and Communication Journal*, **4**(2).

Saha, D., & Mukherjee, A. 2003. Pervasive Computing: A Paradigm for the 21st Century. *Computer*, **36**(3), 25 – 31.

Saleh, R., Jutla, D., & Bodorik, P. 2007 (Aug.). Management of Users' Privacy Preferences in Context. *Pages 91–97 of: IEEE Intl. Conf. on Information Reuse and Integration. IRI 2007*.

Satyanarayanan, M. 2001. Pervasive Computing: Vision and Challenges. *Personal Communications, IEEE*, **8**(4), 10 –17.

Schneier, Bruce. 2010. Schneier on Security: Privacy and Control. *Journal of Privacy and Confidentiality*, **2**(1), 3–4.

Schoeman, Ferdinand D. 1984. Introduction. *In: Philosophical Dimensions of Privacy: An Anthology*. Cambridge University Press.

Segaran, Toby. 2007. *Programming Collective Intelligence*. 1 edn. O'Reilly.

Shadbolt, Nigel. 2003. Ambient Intelligence. *IEEE Intelligent Systems*, **18**(July), 2–3.

Siegemund, Frank. 2004. A Context-Aware Communication Platform for Smart Objects. *Pages 69–86 of: Pervasive Computing*. LNCS, vol. 3001. Springer.

Solove, Daniel J. 2007. *The Future of Reputation: Gossip, Rumor, and Privacy on the Internet*. Yale University Press.

Solove, Daniel J., Rotenberg, Marc, & Schwartz, Paul M. 2008. *Privacy, information, and technology*. 2 edn. Aspen Publishers.

The Guardian Online. 2010. *Privacy no longer a social norm, says Facebook founder*. Available at `http://www.guardian.co.uk/technology/2010/jan/11/facebook-privacy` (accessed August 20, 2012).

Toninelli, Alessandra, Montanari, Rebecca, Lassila, Ora, & Khushraj, Deepali. 2009. What's on Users' Minds? Toward a Usable Smart Phone Security Model. *IEEE Pervasive Computing*, **8**(2), 32–39.

Tsoumakas, G., & Katakis, I. 2007. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, **3**(3), 1–13.

Tsoumakas, Grigorios, Katakis, Ioannis, & Vlahavas, Ioannis. 2010. Mining Multi-label Data. *Pages 667–685 of:* O. Maimon, L. Rokach (ed), *Data Mining and Knowledge Discovery Handbook*, 2 edn. Springer.

UN General Assembly. 1948. *Universal Declaration of Human Rights*. Available at `http://www.unhcr.org/refworld/docid/3ae6b3712c.html` (accessed August 20, 2012).

United States Code. 1974. *Privacy Act of 1974, 5 U.S.C. § 552a*.

W3C. 2002a (Apr.). *A P3P Preference Exchange Language 1.0 (APPEL1.0)*. Available at `http://www.w3.org/TR/P3P-preferences/` (accessed August 20, 2012).

W3C. 2002b (Apr.). *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*. Available at `http://www.w3.org/TR/P3P/` (accessed August 20, 2012).

Warren, Samuel D., & Brandeis, Louis D. 1890. The Right to Privacy. *Harvard Law Review*, **4**(5). Available at `http://groups.csail.mit.edu/mac/classes/6.805/articles/privacy/Privacy_brand_warr2.html` (accessed August 20, 2012).

Weiser, Mark. 1991. The Computer for the 21st Century. *Scientific American*, **265**(3), 66–75.

West, Ryan. 2008. The Psychology of Security. *Commun. ACM*, **51**(4), 34–40.

Westin, Alan F. 1967. *Privacy and Freedom*. New York: Atheneum.

Wright, David, Gutwirth, Serge, Friedewald, Michael, Vildjiounaite, Elena, & Punie, Yves. 2008. *Safeguards in a World of Ambient Intelligence*. The International Library of Ethics, Law and Technology, vol. 1. Springer.

Wu, Feihong, Zhang, Jun, & Honavar, Vasant. 2005. Learning Classifiers Using Hierarchically Structured Class Taxonomies. *Pages 313–320 of:* Zucker, Jean-Daniel, & Saitta, Lorenza (eds), *Abstraction, Reformulation and Approximation*. LNCS, vol. 3607. Springer.

Zelkha, E., & Epstein, B. 1998. *From Devices to Ambient Intelligence*. Talk given at the Digital Living Room Conference. Available at `http://www.epstein.org/brian/ambient_intelligence` (accessed August 20, 2012).

Zhang, Qingsheng, Qi, Yong, Zhao, Jizhong, Hou, Di, Zhao, Tianhai, & Zhang, Jihong. 2007. Context-Aware Learning Privacy Disclosure Policy from Interaction History. *Pages 3–7 of: Proc. of 3rd Intl. Conf. on Natural Computation, ICNC 2007*, vol. 5. Washington, DC, USA: IEEE Computer Society.

Publications and Talks

Publications

1. Christian Bünnig: *Simulation and Analysis of Ad hoc Privacy Control in Smart Environments.* In *Proc. of Intelligent Interactive Assistance and Mobile Multimedia Computing (IMC) 2009*, Rostock, Germany, November 2009. (DOI 10.1007/978-3-642-10263-9_27)

2. Christian Bünnig und Clemens H. Cap: *Ad hoc Privacy Management in Ubiquitous Computing Environments.* In *Proc. of 2nd Inl. Conf. on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services (CENTRIC 2009*, Porto, Portugal, September 2009. (DOI 10.1109/CENTRIC.2009.20)

3. Christian Bünnig: *Smart Privacy Management in Ubiquitous Computing Environments.* In *Proc. of 13th Inl. Conf. on Human-Computer Interaction (HCI-I) 2009*, San Diego, USA, July 2009. (DOI 10.1007/978-3-642-02559-4_15))

4. Christian Bünnig: *Learning Context Based Disclosure of Private Information.* In *Proc. of European Research Workshops (IoTS) at Smart Event'08*, Sophia-Antipolis, France, September 2008.

5. Christian Bünnig: *A Bayesian Approach to Context Based Information Disclosure.* In *Proc. of Baltic Conference on Advanced Topics in Telecommunication (BaSoTi'08)*, Tartu, Estonia, August 2008. (ISBN 978-3-86009-052-7)

6. Christian Bünnig und Clemens H. Cap: *Five Objectives to Diminish User Concerns About Privacy in Smart Environments.* In *Proc. of MoMM2007 & iiWAS2007 Workshops*, Jakarta, Indonesia, December 2007. (ISBN 978-3-85403-231-1)

7. Sebastian Speicher und Christian Bünnig: *Fast MAC-Layer Scanning in IEEE 802.11 Fixed Relay Radio Access Networks.* In *Proc. of Intl. Conf. on Networking, Systems and Mobile Communications and Learning Technologies (ICN / ICONS / MCL 2006)*, Mauritius, April 2006. (DOI 10.1109/ICNICONSMCL.2006.98)

Talks

1. *Simulation and Analysis of Ad hoc Privacy Control in Smart Environments*, IMC 2009, Rostock, Germany, 11 November 2009
2. *Ad hoc Privacy Management in Ubiquitous Computing Environments*, CENTRIC 2009, Porto, Portugal, 24 September 2009
3. *Smart Privacy Management in Ubiquitous Computing Environments*, HCI-I 2009, San Diego, USA, 23 July 2009
4. *Learning Context Based Disclosure of Private Information*, IoTS, Smart Event 2008, Sophia-Antipolis, France, 18 September 2008
5. *A Bayesian Approach to Context Based Information Disclosure*, Baltic Conference on Advanced Topics in Telecommunication, BaSoTI 2008, Tartu, Estonia, 22 August 2008
6. *Five Objectives to Diminish User Concerns About Privacy in Smart Environments*, TwUC workshop at iiWAS 2007, Jakarta, Indonesia, 04 December 2007

Theses

1. Personalization, user-adapted behavior, and information exchange are central themes of smart environments. Next to improving the general user experience they also bear various privacy issues. Interpersonal privacy is a specific conceptualization of privacy which addresses the protection of social norms and boundaries as well as the self-representation of a subject within a social environment. Smart environments may interfere with this privacy concept because they mediate the communication between its inhabitants using various and partly hidden modalities.
2. In order to cope with the complex communication modalities and in order to prevent privacy control from displacing actual activities in smart environments, users need an assistance which automates as much information management tasks as possible. Especially an assistance for interpersonal privacy management should enable users to manage their information individually (e.g. defensively, unconcerned, or pragmatically) and with a strong link to a current situation (instead of primarily considering access rights specified for information items).
3. Existing assistance solutions for privacy management in smart environments do not sufficiently handle the specific challenges of interpersonal privacy, where information disclosure decisions are highly individual as well as more intuitive and dynamic rather than rational and static. One reason for the minor consideration of privacy control in context of social interactions is the lack of objective parameters which influence corresponding information disclosures.
4. This issue can be handled by defining a proper model of disclosure decisions, which integrates both social and technical disclosure parameters and thus also regards interpersonal privacy management. The model proposed by this thesis represents information receivers and disclosed information items as sets, which allows to extract several set-based patterns of information disclosure. These patterns provide novel objective parameters to programmatically address the management of interpersonal privacy.
5. Usage and relevance of different patterns vary significantly among different users, types of information, and situations – as shown by the analysis of a conducted survey about interpersonal disclosure behavior. No sole existing disclosure control method is able to assist in privacy management for all observed patterns.

6. Individual practices to manage information disclosure in smart environments can be supported by orchestrating multiple control methods. The composite disclosure control system proposed in this thesis covers all user-side patterns and practices of privacy management identified by current research (e.g. different temporal management schemes and different granularities of disclosure control).
7. One component of the proposed orchestration of disclosure control methods uses machine learning techniques. As shown by an evaluation of several existing and a novel learning method, to a large extent privacy preferences can be conceptualized using machine learning methods. This allows an automated enforcement of privacy preferences which significantly reduces the privacy-management–related workload compared to manual disclosures.
8. For subjects where learning methods fail to precisely conceptualize privacy preferences, the predicted information items to disclose still represent workload-reducing templates for manual disclosures.
9. The novel learning method, a wrapper method to handle hierarchical multi-label learning problems, interpolates new disclosures by using the previously developed patterns of interpersonal privacy. Compared to other hierarchical wrapper methods, its runtime performance overhead is up to 15 times smaller, while the prediction performance is similar or better in most cases. Additionally, the interpolation wrapper does not obfuscate the human readability of the decision model of the wrapped base learner.
10. A significant portion of incorrect predictions made by a learning method can be detected by verifying that predicted disclosures align with privacy patterns found in past disclosures. With regard to the evaluations made in this work, usually at least one out of four incorrect predictions are detected (i.e. prevented) by validators which utilize the previously developed patterns of interpersonal privacy management.
11. The findings and solutions of this thesis reduce a smart environment's potential disruptions of interpersonal privacy. Concluding guidelines and recommendations presented at the end of this thesis assist engineers in developing environments which meet the specific requirements for privacy management in social interactions.

Selbstständigkeitserkärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung fremder Hilfe angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Christian Bünnig

Rostock, 31. August 2012