Visualizing Past Personal Data Disclosures

Jan Kolter, Michael Netter and Günther Pernul
Department of Information Systems
University of Regensburg
93040 Regensburg, Germany
Email: {jan.kolter, michael.netter, guenther.pernul}@wiwi.uni-regensburg.de

***Abstract*—Today's rich service offer in the World Wide Web increasingly requires the disclosure of personal user data. Service providers' appetite for personal user data, however, is accompanied by growing privacy implications for Internet users. Addressing this rising threat, privacy-enhancing technologies aim at aiding users in protecting their personal data. Even though effective privacy laws facilitate users to edit and revoke already disclosed personal data, few PET solutions support users in exercising this right. Available tools lack intuitive interfaces and are built on powerful infrastructures on the provider side. In this paper we introduce the Data Disclosure Log component within a user-centric privacy architecture. Built on a browser-based logging extension, we present a visualization tool that displays past personal data disclosures from different perspectives. A graph-based view allows for the dynamic presentation of relations between selected entity types. Such an overview enables users to know the conditions of past personal data transactions at any time. This knowledge represents a prerequisite for an ex post revision or revocation of personal data. Usability and user acceptance of the developed prototype is evaluated in a conducted user test.**

***Keywords*-Privacy; Privacy-enhancing Technologies; Visualization; Usability**

I. Introduction

Many of today's services in the World Wide Web rely on the disclosure of personal user data. Interacting with more and more service providers, the user increasingly reveals his identity, which results in growing privacy implications. In the wake of the rising number of personal data misuses, users increasingly get concerned about personal data disclosures [1], [2].

Addressing these privacy concerns, the European Union enacted the Directive 95/46/EC on the protection of personal data [3]. In addition to data minimization, the principles of the Directive include the necessity of an explicit user consent before the collection of any personal data as well as the exclusive use for the purpose stated in a published privacy policy. A further principle of the Directive targets users' control of and access to personal data already transferred to a service provider. Similar legislations arose on national levels and in other areas of the world.

In order to take advantage of effective privacy rights, average Internet users rely on technical means that protect personal user information and facilitate a more informative decision about personal data disclosures. Targeting these user needs, privacy-enhancing technologies (PETs) emerged, a dedicated field of research in the area of IT-Security [4], [5]. A frequently discussed subject in this area is anonymity on network level. On application level, privacy-enhancing technologies aim for solutions that assist users in controlling and managing the disclosure of personal data.

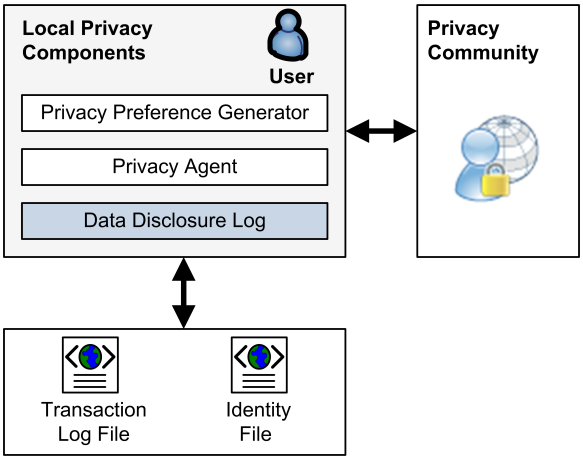


Figure 1. Privacy Architecture Overview

Unfortunately, available PET tools designed for the protection of personal data are used by only a minority of Internet users, while most users disclose personal data with no technical assistance and no information that would aid their decision-making.

In addition to apparent usability deficits of existing solutions, a main reason that prevents the widespread use of PETs is their dependency on service providers. The P3P specification, for instance, requires service providers to generate, publish and maintain a machine-readable P3P privacy policy, which is provided by only a small fraction of service providers [6]. Similarly, the approach of the European PRIME project assumes the installation of the PRIME middleware on the provider side. However, the wide adoption of the powerful PRIME infrastructure in proven back-end infrastructures of service providers seems considerably unlikely.

Acknowledging the conflicting interests of users and service providers as well as the need for usable tools,

we developed a user-centric, service provider-independent privacy architecture [7], which is sketched in Figure 1. A collaborative privacy community facilitates Internet users to share privacy-related information about service providers. This valuable information source is collaboratively edited by all participating users in a Wikipedia-like Web front-end that groups service provider information into articles. The privacy community was prototypically implemented and can be reviewed following this link[1].

Three local privacy components on the user side offer user-friendly tools that assist users in controlling potential, actual and past information flows, utilizing service provider information of the privacy community. In particular, the Privacy Preference Generator captures disclosure rules (so called privacy preferences) for up to twelve Internet service types [8], while the browser-based Privacy Agent matches these user preferences with machine-readable privacy policies of service providers. Contributing to a more informed disclosure decision of users, the Privacy Agent also depicts selected information of the privacy community, which facilitates a more informed disclosure decision of users. Moreover, the use of partial identities [9] is determined and stored in a local identity file.

The goal of this paper is the introduction of a usable Data Disclosure Log component that records and visualizes past personal data disclosures. Our solution provides a clear overview of logged personal data transactions, offering multiple views as well as comprehensible and intuitive user interfaces. Such an overview enables users to know the recipients of past personal data transactions at any time. This knowledge represents a prerequisite for an ex post revision and revocation of personal data, an essential privacy right in most countries.

The remainder of this paper is structured as follows. After describing related work in Section II, Section III sketches the characteristics and the output of a browser extension that logs personal data disclosures. Section IV introduces usability requirements and a conceptual overview of the proposed tool as well as an in-depth presentation of its visualization components. In Section VI we lay out the results of a conducted user test that proved the usability and the user acceptance of our implemented solution. Section VII summarizes the main contributions of this work.

II. Related Work

Developed within the PRIME project, the goal of the Data Track is to record personal data transactions and to offer users a transparent view of disclosed personal data [10]. In addition, the component provides supplemental functions that allow users to interact with service providers, facilitating an ex post adjustment of transferred personal data.

[1]http://www-ifs.uni-regensburg.de/Privacy/

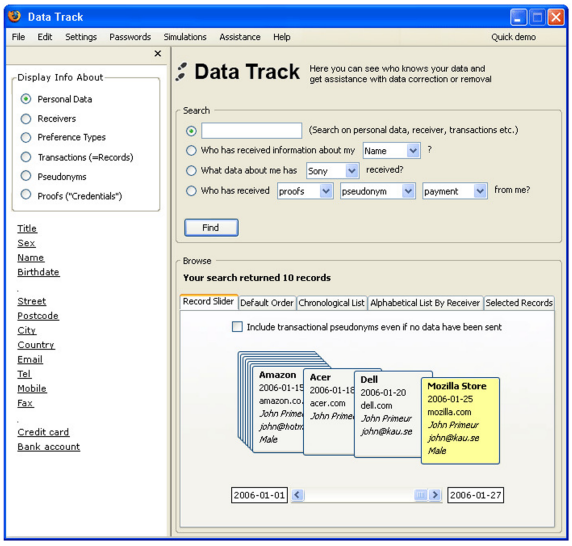


Figure 2. Data Track User Interface [11]

The design of the user interface addresses the intuitive capture of user input as well as the understandable representation of disclosed personal data. Figure 2 shows a screenshot of the user interface.

Data Track offers several representations of data transactions. Addressing the component's understandability, a user test showed that a chronological stack of data transactions earned the highest user acceptance [12]. The test also underscored the importance of intuitive data filters. Data Track selects transactions by offering questions that alternatively filter data transactions according to service providers and personal data types.

The presented Data Track provides widespread functionality for the management of already disclosed personal data. The component, however, relies on the powerful PRIME architecture, which builds on a complex privacy infrastructure on the client side and the provider-side.

In this work, we propose and prove the advantages of a provider-independent privacy infrastructure (see Section I) and show the benefits for the Data Disclosure Log component. Moreover, compared to Data Track this paper evaluates and employs the application of more sophisticated visualization schemes for the user-friendly presentation of past personal data transactions.

III. Logging Personal Data Disclosures in the Web Browser

An integral part of the targeted Data Disclosure Log component is a suitable logging tool. In the context of our proposed privacy architecture, we developed a Mozilla Firefox browser extension that detects and records personal data transfers.

In particular, the transaction logging tool identifies disclosed personal data types as well as the used process. The data type identification process follows a hybrid approach. First, the browser extension analyzes attributes of Web form fields using pre-defined lists of keywords. In an additional step, the user input is compared with regular expressions as well as a user profile. Apart from the identification of personal data disclosures, the proposed logging tool also detects the context of personal data submissions. Specifically, the tool identifies *Login*, *Registration* and *Purchase* processes.

After the submission of personal data, the developed browser extension writes the detected information in a transaction log file, which can be checked and edited at any time. The following listing presents an extract of a generated transaction log file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xml xmlns="http://www-ifs.uni-regensburg.de/Privacy">
  <hostname name="http://www.amazon.com" title="Amazon">
    <transaction process="Registration" id="1">
      <dynamic>
        <Uri>https://www.amazon.com/gp/css/homepage.html</Uri>
        <Timestamp>1250592961</Timestamp>
        <Cookies>true</Cookies>
      </dynamic>
      <data>
        <Given>Paul</Given>
        <Family>Revere</Family>
        <Email>prevere@hotmail.com</Email>
        <Password>a3cca2b2aale</Password>
        <Bdate>24.05.1971</Bdate>
        <Addresses>
          <Postal>
            <Street>Arlington Road</Street>
            <Housenumber>1606</Housenumber>
            <City>Boston</City>
            <Stateprov>MA</Stateprov>
            <Postalcode>02101</Postalcode>
          </Postal>
        </Addresses>
      </data>
    </transaction>
    ...
  </hostname>
  ...
</xml>
```

Listing 1. Extract of an Exemplary Transaction Log File

The XML file groups transactions by service providers. Our example shows a transaction with the service provider Amazon. The transaction was identified as a *Registration* process and included the personal data types given name, family name, e-mail address, password, date-of-birth and a postal address.

IV. Visualizing Logged Personal Data Disclosures

This section discusses the concept, design and implementation of an application for the user-friendly visualization of personal data transactions. The proposed solution along with its technical implementation employs basic concepts of graph theory and graph drawing. For more detailed theoretical foundations of these topics the interested reader is referred to [13], [14]. After collecting usability requirements, we present a conceptual overview of the designed visualization tool. We continue with a definition of entity types, before we introduce the design of four visualization components.

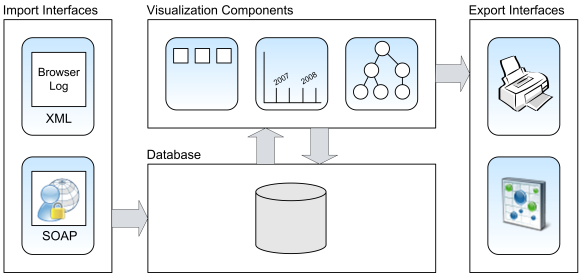


Figure 3. Architectural Overview of the Visualization Tool

A. Usability Requirements

The targeted visualization tool should allow users to analyze their disclosure behavior from different perspectives. Hence, the tool should provide understandable filtering functionality with regard to service providers and data types. Dedicated views should enable users to check, what personal data were transferred to a certain service provider as well as which service provider possesses a specific personal data type. In addition, a chronological analysis is required, which facilitates a temporal overview of data disclosures. Finally, the targeted tool should present selected relations between service providers, disclosed personal data types and partial identities, offering a clear, comprehensible overview of all involved information types.

In order to gain a high degree of user acceptance, special attention should be dedicated to the design of suitable user interfaces. An intuitive user interface allows users to derive a mental model of the visualization tool and to estimate its behavior, when unfamiliar functions are used [15]. A cohesive design will contribute to orientation and is of paramount importance.

In particular, Patrick et al. [16] suggest the use of intuitive control elements, like familiar menus, lists and buttons. Furthermore, the application of common interaction patterns like drag-and-drop is recommended. A user-friendly tool should also employ familiar functions like zooming, selecting and moving objects. Finally, the differing level of user experience should be considered, by adopting the user interface accordingly.

B. Conceptual Overview

The design of our proposed visualization tool follows the Information Visualization Data State Reference Model [17], [18], which defines the visualization process from raw data to visualization views.

The proposed tool is divided into several visualization components (see Figure 3). One of the two import interfaces parses local XML files containing logged personal data transactions (see Section III) as well as information about used partial identities provided by the Privacy Agent component (see Section I). A second import interface requests

(a) Identity (b) Partial Identity (c) Service Type (d) Service Provider (e) Process (f) Data Type

Figure 4. Icons of Defined Entity Types

service provider information utilizing Web services of the online privacy community. In particular, Web services of the privacy community are utilized to request static information about service providers, which is displayed as supplemental information in the offered visualizations. The application also accesses the service type of service providers from the pricacy community, which allows for a graphical categorization of service providers.

All captured input information are loaded into a local database that integrates imported data. Each provided visualization component requests and processes information from the local database and displays a specific view of personal data transactions to the user. Two export interfaces enable users to print and store generated visualizations.

C. Definition of Entity Types

The common goal of the corresponding visualization components is the understandable and intuitive presentation of relations between various types of data.

Ware classifies data into entities and their relations to each other [19]. An entity represents an object of interest, while relations define the structure that interconnects entities. Both entities and relations possess additional attributes.

Identifying objects of interest for our scenario, we define six entity types out of all information available in the generated database. These entity types represent aggregated information units and serve as foundation for all developed visualization components. The ability to relate entity types allows for a hierarchical display of dependencies. An entity describes a concrete instance of an entity type.

In the following we describe the identified entity types along with the relation to each other. The entity types were defined considering the relevance to the proposed tool and the resulting information value to the user. For this reason, the entity type Transaction – although a central entity type in the data model – was not defined, as by itself it provides no integral information to the user. Figure 4 displays icons of each defined entity type, guaranteeing a consistent and comprehensible user experience.

- **`Identity:`** The entity type Identity refers to identity classes that group partial identities with regard to their transitive linkabilities [9]. Partial Identity represents an inferior data type.
- **`Partial Identity:`** A Partial Identity is regarded as a data type subset of a user's complete identity [9]. Partial identities are used for interaction with service providers.
- **`Service Type:`** The entity type Service Type consists of twelve pre-defined service types and assigns entities of Service Provider to one of these twelve instances. For example, the service provider Amazon is subordinated to the service type *Shopping*. This categorization of Sevice Providers is maintained in the online privacy community.
- **`Service Provider:`** This entity type represents an abstract Web site a user interacts with and discloses personal data to. An instance of this entity type is, for example, *Amazon*. Service Provider marks a central entity type, as it maintains several relations to other entity types. Service Type can be interpreted as superior entity type, as it aggregates service providers. Inferior entity types are Process and Data Type.
- **`Process:`** Process groups transactions into processes like *Registration* or *Purchase* (see Section III). The goal of this entity type is to allocate data types to one of these categories and – as a consequence – to make data type disclosures to service providers more transparent. This allows, for example, to understand that credit card information has been transferred to Amazon in the context of a *Purchase* process.
- **`Data Type:`** This entity type contains all personal data types transferred to service providers. Entities of this entity type are, for example, *User Name*, *Date-of-Birth* or *E-mail Address*.

D. Application Design

In this section we introduce our proposed visualization components. Considering recommendations of the Visual Information Seeking Mantra [20], the given scenario requires the development of multiple views, which reduce the complexity of the presented visualization. Wang Baldonado et al. define guidelines for the use of multiple views [21]. Following these guidelines the presented application provides four individual views to the user. Each view presents disclosed personal data and identity information from a different perspective.

The goal of all visualization views is the abstraction of the textual representation. A clear graphical representation and its accompanying complexity reduction facilitate a quick comprehension of all essential information.

1) Basic Views: The presented visualization tool provides two basic views that focus on personal data disclosures to service providers, employing intuitive forms of presentation. Accordingly, these views concentrate on the entity types Service Provider and Data Type.

Figure 5 shows the service provider view, which displays all service providers the user disclosed personal data to. Service providers are visualized as tiles in the main window and represented by a standardized icon frame introduced

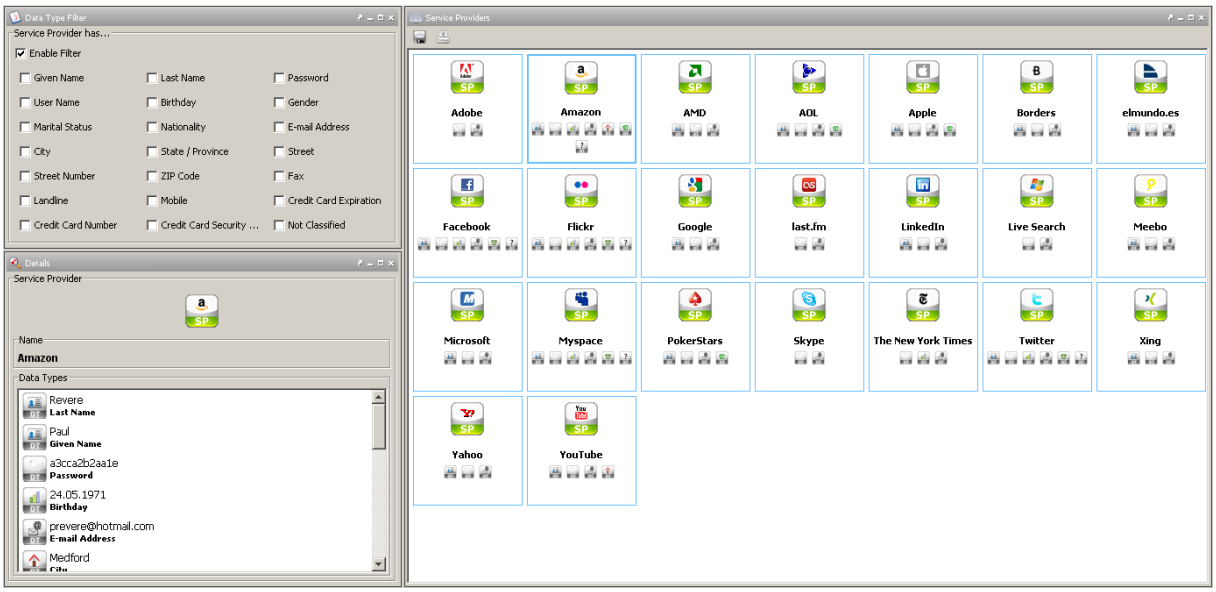


Figure 5. Service Provider Overview

in Section IV-C. If possible, an individual service provider icon is dynamically generated, downloading the respective Favicon of the service provider and dynamically setting it into the green icon frame. Such an individual service provider icon contributes to a quick comprehension of the shown information. Apart from the service provider, icon groups that represent certain data type groups are fit into the tile, allowing users to instantly estimate the kind of personal data disclosed to a provider.

A filter window enables users to limit the displayed service provider interactions with regard to the checked data types. This dynamic filter, for instance, allows users to specify the exclusive consideration of service providers an *E-mail Address* or a *Date-of-Birth* has been disclosed to, which is known as Dynamic Queries [22].

If the user selects a service provider tile, a list of disclosed individual data types are provided in a detail window.

Also highlighting service providers the user interacted with, a record slider view aligns service providers in a Cover Flow-like fashion, which – for the sake of brevity – is not depicted. The record slider view provides two spacious windows for the display of static service provider information from the privacy community and a list of disclosed data types.

2) Chronological View: The primary goal of the introduced application is the presentation of relations between entities. A comprehensive visualization tool, however, should also be capable of displaying a timely sequence of data disclosures. In the following, we describe the design of such a view.

The chronological view performs a temporal analysis of all communications with service providers, allowing users, for instance, to identify frequently interacted service providers. For the reason of clarity an adjustment of the observation period is available. As opposed to other views, the chronological view focuses solely on the entity type Service Provider and does not show relations to other types.

Accounting for the needs of a usable interface, we employ a horizontal timeline for the chronological presentation of service provider interactions. As users are already familiar with such an element, the application benefits from a short introduction phase and a higher user acceptance.

The structure and design of the chronological view is depicted in Figure 6. The view is divided into four parts. Most space is dedicated to the scrollable timeline, as proposed by the Center Stage pattern [22]. The horizontal axis is labeled with date values. The column above each date stacks all service providers the user transferred personal data to at that certain date.

The top side window provides an interface for the selection of a time interval, enabling users to define the observation period. Facilitating an intuitive user control, the active interval selection resembles a calendar, if the start or the end date field is clicked.

Below that window, a filter enables users to fade out service providers with regard to combinations of transferred personal data types, as already utilized in the service provider view.

If a service provider is selected by clicking on a respective icon in the timeline, a detail window provides supplemental information about that service provider using static information queried from the privacy community. The interplay of the main window and the detail window implements the concept of a Two-Panel Selector [22].

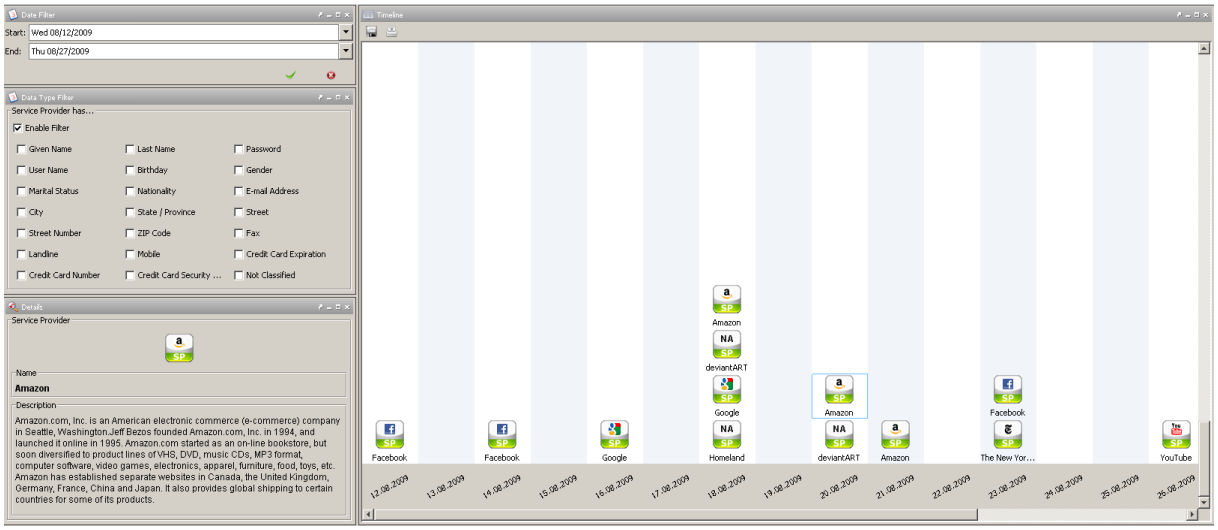


Figure 6. Chronological View

3) Graph View: The graph view aims for a generic presentation of diverse relations between entities, using a graph-based visualization scheme. As the proposed application integrates various data sources and defines entity types for the better abstraction of the textual representation, the comprehension of a large amount of information is possible, which facilitates the visualization of relations that are not evident to the user.

In the graph view the selection of entities and the display of their relations are individually controlled by the user, who is enabled to drag and drop selective entity nodes to the graph window. If an entity is added to the graph, relations to existing entities in the graph are visualized as edges, allowing users to dynamically build a tree of any available combination of entities.

The described procedure allows a maximum number of degrees of freedom. In order to lower the complexity for inexperienced users, this user group is limited to adding a whole entity type to the graph, which includes all available entity nodes of that type. Accounting for their low level of experience, this less complex graph generation simplifies user control. More experienced users can add and relate individual entities to the graph. The user experience is captured at the initial start of the application, offering two complexity modes – basic and advanced – to the user.

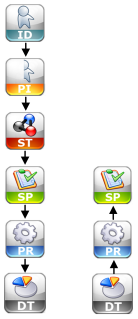
In order to facilitate a generic, implementable solution of the graph view, we specify relations that can be built and visualized from a particular entity type. This prevents the visualization of entity combinations that cannot be related. In particular, we define two directions of relations between entity types (see Figure 7). At the same time, pre-defined hierarchies do not limit users from selecting any available combination of entities.

The Top-Down hierarchy defines Identity as the most superior entity type, followed by Partial Identity and Service Type, which categorizes Service Providers. Furthermore, Service Provider is superior to the entity type Process, which is used to aggregate Data Types. Applying this hierarchy of entity types, a user could, for example, select a particular service provider and visualize its related processes (child nodes of the selected service provider) and their involved data types (child nodes of each process) in the graph.

The Bottom-Up hierarchy consists of three entity types and defines Data Type as top element and Process and Service Provider as child elements. This hierarchy allows, for instance, the selection of a particular data type and the presentation of service providers (child nodes) it was transferred to.

As mentioned above, the definition of entity type hierarchies limits the complexity of the view generation. Furthermore, a hierarchy facilitates the representation of relations between entities as a directional graph. The applicable hierarchy is implicitly chosen by the user, when the first element of the graph is added. If a Data Type or Process entity is initially selected, the Bottom-Up hierarchy applies. For all remaining entity types, the Top-Down hierarchy is chosen.

Figure 8 shows the design of the user interface, which is divided into four windows. Similar to the chronological view, the main window focuses on the display of entities. Due to the hierarchical limitations, we choose a hierarchical layout for the presentation of the directional graph [23], allowing a clear and intuitive representation of relations between entities. Alternatively, we provide a radial layout that aligns child nodes on outer circles originating from a root node in the center.



(a) Top-Down (b) Bottom-Up

Figure 7. Graph View Hierarchy Types

Available entity types the user can add to the graph are presented in a side window and are represented by the designed entity type icons (see Section IV-C). The first graph node is added by dragging and dropping an entity type to the main window. A pop-up window captures the selection of the single entity of that type, which serves as root element of the graph. The procedure for the addition of further entity types depends on the chosen complexity mode.

In the basic mode the user drags and drops whole entity types in the main window. An iterative algorithm scans all available instances of that entity type and adds entities to the graph that are related to the already displayed entities in the graph. Based on a single entity this procedure allows for a quick arrangement of the chosen elements and their relations to each other.

The advanced mode facilitates a more customizable visualization of relations between entities. Again, a single entity serves as root element of the graph. Subsequently, the user is able to add individual entities using menus that are offered by right-clicking a graph node.

By default, all graph nodes are represented by the designed entity type icons. Like the previous views, the graph view dynamically generates individual Service Provider icons downloading the repective Favicons. Furthermore we designed individual icons for each Service Type and for groups of Data Types, which support the intuitive presentation of selected entity nodes.

Below the entity selection window a tool tip window provides hints that guide users through the graph generation process. In order to keep an overview of larger graphs, a satellite window displays a tiny view of the graph. Similar to the chronological view, the graph view provides an additional window that presents detailed information about entities selected in the graph.

The usability of the interface is realized by a clear separation of information. The windows integrate elements used in other local privacy components, contributing to understandability and fostering user acceptance.

V. Implementation Details

The stand-alone tool is implemented as Java application and runs platform-independently with every Java 6 Runtime Environment[2]. For the design of the graphical user interfaces we utilize the Java Swing toolkit. The dynamic graph generation is enabled by Visual Library[3].

Taking these requirements into account, we employ a relational database to integrate and store available input data. In particular, we utilize the H2 database[4], which is generated at the start of the program and filled with data of the XML input files.

In addition to the local privacy files, the database is loaded with selected service provider information queried from the privacy community. The privacy community is also accessed for the download of data type and service type icons, which are used for the dynamic generation of graph node icons.

The implemented prototype of the visualization tool is available for download on the privacy community Web site[5].

VI. Evaluation

In order to evaluate the usability and user acceptance of the developed visualization tool, we conducted a user test with 26 test persons, acknowledging frequent recommendations that a single-digit sample is insufficient for meaningful test results [24], [25], [26]. Aiming at a heterogeneous test sample, the invited test persons showed a diverse academic and professional background. However, basic knowledge of Microsoft Windows as well as the occasional use of the World Wide Web were prerequisites for participating candidates. In order to avoid biased results, persons with close relationships to the interviewers were not considered.

In particular, the test sample included 17 university students, while nine test persons were graduated professionals. Hence, 15 out of the 26 test persons were 25 years old or younger, seven between 26 and 30, and four between 30 and 45. 22 of all test persons were male. Out of the 17 students nine were enrolled in a technical program and five in a business program. From the remaining students two were pursuing a teaching degree and one a diploma in mathematics.

Each test person was provided with a mock-up transaction log file, which was loaded into the tool. After the initial start of the tool, the test candidates were asked to read and understand a prepared tutorial, which took about five minutes on average.

The first task involved the service provider view. Test persons were asked about data type disclosures to a specific service provider. The solution of this question posed no further difficulties. All test persons unanimously associated

[2]http://java.sun.com/javase/6/
[3]http://graph.netbeans.org/
[4]http://www.h2database.com/
[5]http://www-ifs.uni-regensburg.de/Privacy/#/tools/

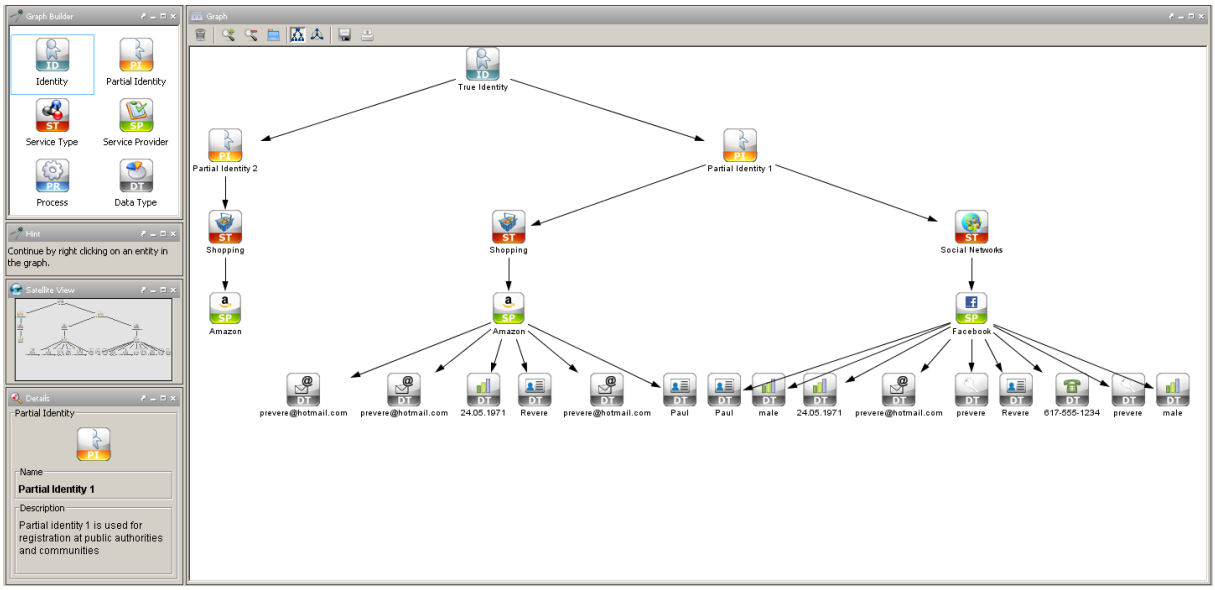


Figure 8. Graph View

the tiles with service providers, intuitively clicked the correct tile and read the list of disclosed data types from the detail window.

Affecting the same view, test persons were requested to identify service providers their date-of-birth was transferred to. 21 out of 26 test persons instantly activated the filter window and checked the correct box. The remaining five persons tried to solve the question by going through all service providers in the main window. Acknowledging the comments of these users, we slightly changed the wording of the filter window.

The next question asked about disclosures of the credit card number in a certain time period. 18 out of 26 test persons correctly identified and clicked the timeline button in the offered navigation bar, which led to the chronological view. Six test persons needed guidance from the interviewers how to find the button of the respective view. After reaching the chronological view, only two test persons were given hints how to use the time period filter.

The following questions targeted the graph view and represented the most challenging tasks. In particular, test persons were asked to display disclosed data types for each process executed at the service provider Amazon. 22 test persons correctly navigated to the correct graph view. Building the correct graph, most test persons had difficulties applying the offered entity types correctly. The majority of these persons also did not choose Amazon as root element, which was required for the proper solution of the task. The observed issues during the graph generation led us to create an animated built-in tutorial that describes the purpose and demonstrates the graph generation process of the graph view. Once the graph was generated, all users correctly interpreted the visualized relations and confirmed the intuitive presentation. Concerning the layout, 16 test persons preferred the hierarchical layout, while seven persons opted for the radial layout.

A similar task tested the graph view in the advanced mode. The change of the graph generation from a simple drag and drop in the basic mode to a menu-controlled node addition in the advanced mode caused further difficulties that served as additional input for the now available tutorial.

In the interview section all test persons attested a clear presentation of the presented information and relations. The test candidates also praised the intuitive user control. Asked about scenarios they would use the presented visualization tool for, test persons named the misuse of their personal data and the analysis of their disclosing behavior. The answers prove that test persons understood and valued the advantages of the developed visualization tool.

VII. Conclusions

In this paper we introduce a usable solution for the visualization of past personal data disclosures within a user-centric privacy architecture. Based on a browser extension that detects and stores personal data submitted to service providers, the presented visualization tool displays the resulting transaction log using intuitive visualization techniques. The tool offers multiple views that process and display entity types of personal data transactions from many perspectives. In addition to basic views that focus on recipients of transferred data, the chronological view allows for a temporal analysis of past personal data disclosures. A flexible graph view facilitates the visualization of various relations between entities, enabling users to dynamically

generate a self-defined graph. A conducted user test proved the usability and the user acceptance of our solution.

ACKNOWLEDGMENT

The authors would like to thank Alfred Kobsa, University of California, Irvine, for helpful comments and stimulating discussions.

REFERENCES

[1] C. Jensen, C. Potts, and C. Jensen, "Privacy Practices of Internet Users: Self-reports versus Observed Behavior," *International Journal of Human-Computer Studies*, vol. 63, no. 1-2, pp. 203–227, 2005.

[2] A. Kobsa, "Privacy-Enhanced Personalization," *Communications of the ACM*, vol. 50, no. 8, pp. 24–33, 2007.

[3] European Parliament, "EU-Directive 95/46/EC." Official Journal of the European Communities No L 281 31, October 1995.

[4] H. Burkert, "Privacy-enhancing Technologies: Typology, Critique, Vision," in *Technology and Privacy: The New Landscape* (P. Agre and M. Rotenberg, eds.), (Boston, MA, USA), pp. 126–143, MIT Press, 1997.

[5] I. Goldberg, D. Wagner, and E. Brewer, "Privacy-enhancing Technologies for the Internet," in *Proceedings of the 42nd IEEE International Computer Conference (COMPCON '97)*, (Washington, DC, USA), pp. 103–109, IEEE Computer Society, 1997.

[6] I. K. Reay, P. Beatty, S. Dick, and J. Miller, "A Survey and Analysis of the P3P Protocol's Agents, Adoption, Maintenance, and Future," *IEEE Transactions on Dependable Secure Computing*, vol. 4, no. 2, pp. 151–164, 2007.

[7] J. Kolter, T. Kernchen, and G. Pernul, "Collaborative Privacy - A Community-based Privacy Infrastructure," in *Proceedings of the 24th IFIP TC 11 International Information Security Conference, SEC 2009*, (Berlin), Springer, May 2009.

[8] J. Kolter and G. Pernul, "Generating User-understandable Privacy Preferences," in *Proceedings of the 4th International Conference on Availability, Reliability and Security (ARES 2009)*, (Fukuoka, Japan), IEEE Computer Society, March 2009.

[9] A. Pfitzmann and M. Hansen, "Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology." Technical report, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, February 2008.

[10] J. Pettersson, S. Fischer-Hübner, and M. Bergmann, "Outlining Data Track: Privacy-friendly Data Maintenance for End-users," in *Proceedings of the 15th International Conference on Informations Systems Development (ISD 2006)*, Springer Scientific Publishers, 2006.

[11] J. Pettersson, "HCI Guidelines." PRIME deliverable D6.1, February 2008.

[12] S. Fischer-Hübner, J. S. Pettersson, M. Bergmann, M. Hansen, S. Pearson, and M. Casassa Mont, "HCI Designs for Privacy-Enhancing Identity Management," in *Digital Privacy: Theory, Technologies and Practices*, ch. 11, pp. 230–249, Auerbach Publications (Taylor and Francis Group), 2007.

[13] J. Clark and D. A. Holton, *A First Look at Graph Theory*. World Scientific, 1991.

[14] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[15] C. M. Brown, *Human-Computer Interface Design Guidelines*. Intellect Books, 1998.

[16] A. S. Patrick, S. Kenny, C. Holmes, and M. van Breukelen, "Human Computer Interaction," in *Handbook for Privacy and Privacy-Enhancing Technologies* (G. W. van Blarkom, J. J. Borking, and J. G. E. Olk, eds.), ch. 12, pp. 249–290, 2003.

[17] E. H. Chi and J. Riedl, "An Operator Interaction Framework for Visualization Systems," in *Proceedings of the 1998 IEEE Symposium on Information Visualization (INFOVIS '98)*, (Washington, DC, USA), pp. 63–70, IEEE Computer Society, 1998.

[18] E. H. Chi, "A Taxonomy of Visualization Techniques Using the Data State Reference Model," in *Proceedings of the IEEE Symposium on Information Vizualization 2000 (INFOVIS '00)*, (Washington, DC, USA), p. 69, IEEE Computer Society, 2000.

[19] C. Ware, *Information Visualization: Perception for Design*. Morgan Kaufmann, 2nd ed., 2004.

[20] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," in *Proceedings of the 1996 IEEE Symposium on Visual Languages (VL '96)*, (Washington, DC, USA), p. 336, IEEE Computer Society, 1996.

[21] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for Using Multiple Views in Information Visualization," in *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '00)*, (New York, NY, USA), pp. 110–119, ACM, 2000.

[22] J. Tidwell, *Designing Interfaces*. Sebastopol: O'Reilly Media Inc., 2005.

[23] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for Visual Understanding of Hierarchical System Structures," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 11, pp. 109–125, February 1981.

[24] L. Faulkner, "Beyond the Five-user Assumption: Benefits of Increased Sample Sizes in Usability Testing," *Behavior Research Methods, Instruments and Computers*, vol. 35, no. 3, pp. 379–383, 2003.

[25] C. Perfetti and L. Landesmann, "Eight Is Not Enough," June 2001.

[26] J. Spool and W. Schroeder, "Testing Web Sites: Five Users Is Nowhere Near Enough," pp. 285–286, 2001.