8-6-2011

Enabling End-user-driven Data Interoperability – A Design Science Research Project

Oliver Gass
University of Mannheim, gass@eris.uni-mannheim.de

Alexander Maedche
University of Mannheim, maedche@eris.uni-mannheim.de

Enabling End-user-driven Data Interoperability – A Design Science Research Project

Oliver Gass
Chair of Information Systems IV (ERIS)
University of Mannheim,
68131 Mannheim, Germany
gass@eris.uni-mannheim.de

Alexander Maedche
Chair of Information Systems IV (ERIS) and
Institute for Enterprise Systems (InES)
University of Mannheim,
68131 Mannheim, Germany
maedche@eris.uni-mannheim.de

ABSTRACT

Today, end-users have many opportunities to consume Internet-based services relevant for their private and professional life. However, the easy consumption of new services is not reflected on the data side. The availability of new services potentially increases pre-existing data interoperability issues since the same data is scattered over even more locations. Looking at the simple example of basic personal contact data, users need to manually re-enter again and again this data to be able to register and consume services. The research described in this paper is the preliminary result of an ongoing design science research project addressing the topic of end-user-driven interoperability. We present a software artifact and the results of an experiment evaluating contact data interoperability within multiple Internet-based services.

Keywords (Required)

End-user computing, Data Interoperability, Design Science Research

INTRODUCTION

Interoperability has always been an issue when it comes to processing data. However, the dawn of the World Wide Web and the services available on it, have once again pushed the question of data interoperability into the spotlight: The density of Internet-based services has increased constantly over the last years. While at the beginning, the World Wide Web was not majorly delivery medium for HTML documents; more advanced techniques nowadays foster the participation of people and have lead to a variety of interactive services. These services provide easy access to functionalities which were either not available or too expensive one decade before. The competitive environment of the Internet has further accelerated the speed at which new innovations emerge and achieve acceptance. Also, companies have benefit from this development: Nowadays, they do not only integrate outside Web-services into their existing IT-landscape, but also start to consume complete enterprise application systems, for example, Customer Relationship Management solutions, from the cloud (McAfee, 2006, Dubey and Wagle, 2007). Unfortunately the easy consumption of these new services is not reflected on the data side. The more services are consumed, the wider data is spread raising the need to find a way to maintain a consolidated view. Certainly, scientific research (Wiederhold, 1993; Dong and Halevy, 2005; Halevy et al., 2006) and practice have already established numerous approaches and solutions to integrate data sources of different origins. However, usually such solutions pursue a rather heavy-weight integration approach involving IT experts. Their implementation requires advanced technical know-how and plenty of effort which extend the capabilities of single employees and small companies. Research and commercial solutions which explicitly address the end-user, regardless of if in his private or professional life, are still at an early stage.

The research described in this paper is the preliminary result of an ongoing design science research project. It contributes to the ongoing discussion of how to provide end-user-driven data interoperability of data scattered across several Web services. We describe a proof of concept prototype in the form of a software artifact which allows establishing end-user-driven interoperability of personal and professional contact data. Our experimental evaluation demonstrates that the software artifact is able to enhance productivity of end-users. The remainder of the paper is organized in x sections. First, we discuss existing related work and put it into context to our contribution. Second, we present our entire research framework including the research questions addressed in this paper and the research design we are following in our work. Third, we present our preliminary design principles, the conceptual model and the technical implementation of our software artifact. Fourth, we present the evaluation results of an experiment we have been carrying out with potential end-users using our software artifact to carry out a set of pre-defined tasks. Finally, we conclude the paper with a short summary and provide an outlook on future work.

RELATED WORK

Previous work most closely related to our research belongs to the personal information management discipline. Personal information management is part of the data management domain. It refers to both the practice and the study of the activities people perform in order to acquire, organize, maintain, retrieve and use information items such as documents, Web pages and email messages for everyday use to complete tasks and fulfill a person's various roles, e.g., as an employee. The ultimate goal of personal information management is that one always has the right information in the right place, in the right form, and of sufficient completeness and quality to meet the current need (Jones, 2007). While the purpose of personal information management is easy to describe and discuss because everyone has first-hand experience; it remains difficult to establish a clear distinct scientific definition: Lansdale (1988) defines personal information management as "the methods and procedures by which we handle, categorize and retrieve information on a day-to-day basis". Bellotti et. al. (2002, p. 182) understand it to be "the ordering of information through categorization, placement, or embellishment in a manner that makes it easier to retrieve when it is needed." An important concept in personal information management is a so called personal space of information. A personal space of information is defined as the information that is nominally under a person's control. A personal space of information can be considered an abstract collection of all of the various physical and digital information items. A fundamental challenge is maintaining an integrated view of the personal space of information. According to Jones (2007), the three major activities in personal information management are the activities necessary to enter, store and retrieve information from the personal space of information.

The proposed research activities follow a user-centered approach and can therefore benefit from earlier empirical research in personal information management. Such research tries to explore user behavior, in specific how users manage and organize their personal information. Henderson (2004), for example, has investigated how people organize their desktops and Jones et al. (2005) have reported on the extensive use of desktop folders. Other researchers have studied in detail the use, archiving and storage of emails (e.g. Whittaker and Sidner, 1996), of documents (e.g. Bondarenkoand Janssen, 2005), of time management tools (e.g. Blandford and Green, 2001) and personal information management software (e.g., Boardman and Sasse, 2004). Other researchers have been following the objective to improve or develop new methods to manage personal information. This has resulted in several projects which have studied new ways to organize or search personal information more effectively. They all attempt to go beyond the traditional hierarchical directory model of storing data and present a unified user interface for personal data. Examples are The LifeStreams Project (Freeman and Gelernter, 1996 or The Placeless Documents Project (Dourish et al. 2000). A recent representative of personal information management systems is Semex. Semex offers the user a flexible platform for personal information management. The current focus of Semex is on desktop search. In contrast to regular search options, as for example key word search, Semex offers a search-by-association functionality. Semex puts emphasis on extracting associations between personal information objects from multiple sources and on ensuring that they all mesh together seamlessly (Dong et. al, 2005; Dong and Halevy, 2005). Recent research that aims to examine especially the integration of Web services has been conducted by Geambasus et al. (2008). They have introduced an approach to organize, search, and archive data hosted on Web services, create heterogeneous object collections from it and share them in a protected way. Finally their approach allows the manipulation of data objects with standard applications or scripts. The concept has been piloted and successfully evaluated under the name Menagerie. Another approach that aims in the same direction is described by Jammalamadaka et al. (2005). They propose Pvault a personal data manager based on a client-server architecture. Pvault stores and retrieves data from a remote data server securely and also provides auto-fill capabilities. Auto-fill capabilities in particular have been addressed by Lee and Tsatsoulis (2005). They introduce SmartXAutofill, a software which provides intelligent data entry assistance for predicting and automating inputs for XML documents.

RESEARCH FRAMEWORK

The presented research has the goal to identify the design principles of a service which provides data interoperability for end-users following a design science approach. In contrast to previous research which usually aims to investigate one specific aspect of data interoperability, our research follows a multi-dimensional approach which considers findings of several research realms to develop a comprehensive data interoperability design. Our approach builds on three fields of justificatory knowledge: Existing data integration and interoperability concepts (e.g., Wiederhold, 1992). In addition, it includes major elements of the social computing idea, in particular, wisdom of the crowd and data sharing behavioral theories (e.g., Brabham, 2008). Finally, it emphasizes user-centered design aspects: With respect to user-experience our design makes use of established rich-Internet-application technologies (e.g. Farell and Nezlek, 2007). With respect to user-requirements we account for assumptions made by the lazy user theory. The lazy user theory which builds upon the technology acceptance model (compare Davis et al., 1989) tries to explain how an individual selects a solution to fulfill a specific need from a set of possible solutions. The lazy user model concludes that a user will always choose the solution which asks for the least amount of effort (Collan, M., 2007).

In this paper we address the following two sub-questions of our research project:

How to realize a software artifact for end-user-driven data interoperability of contact data?

Does such a software artifact increase efficiency of end-users when entering contact data into different forms?

Research Design

The presented research uses design science methods and follows the general design cycle described by Vaishnavi and Kuechler (2008). The general design cycle favors the inclusion of multiple methods to design, implement, and evaluate an artifact via a design science research approach (Vaishnavi and Kuechler, 2008). Figure 1 depicts the general design science cycle.

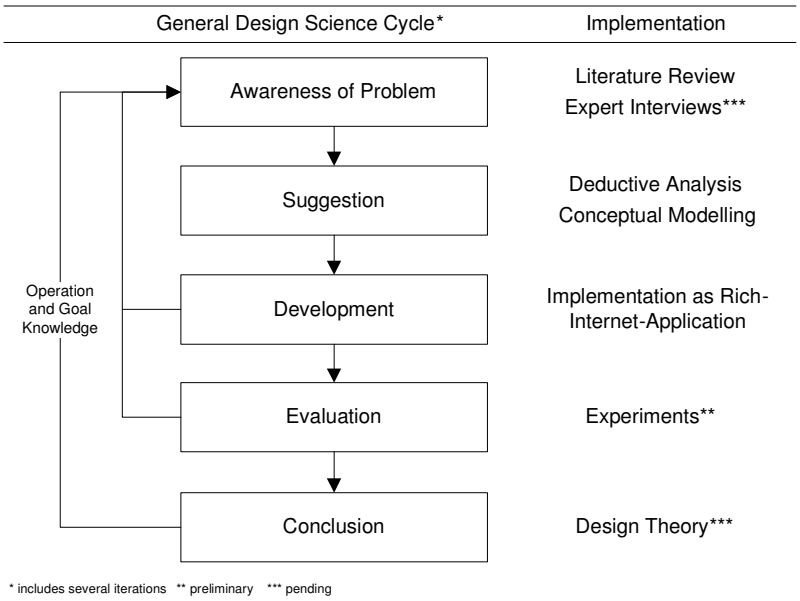

Figure 1: The General Design Science Cycle

The first step of the general design cycle framework is an awareness of a problem through problem identification and definition. After the identification of the problem, research is necessary to derive suggestions to address the problem. This includes the exploration of use-cases and the investigation of user-needs and user-requirements. After building an understanding of the problem domain, so far solely based on literature review, we have created a list of design principles that represent specific user requirements. We then used conceptual modeling techniques to transform these design principles into a comprehensive solution design. The conceptual model was explicitly chosen to be independent of implementation details. Further details will be added successively in the course of following iterations of the design science cycle. Our first design was instantiated as a Web service and was then subject to evaluation. The evaluation was conducted using a lab-environment with real-life users. Instead of relying only on literature, our research will be complemented by qualitative methods following iterations of the cycle. We intend to use deductive analytics to combine findings from interviews with findings from literature. In addition, we will increase the number of users in our experiment successively to make our findings generalized.

THE SOFTWARE ARTIFACT

This section explains in detail the software artifact we have implemented in the first iteration. First, we introduce our initial design principles and the conceptual framework underlying our implementation. Second, we lay out technological details of the software artifact.

Initial Design Principles

We have defined an initial set of design principles as a basis for our first prototype. These initial design principles will change and be extended in the cause of new findings achieved in future design science cycle iterations.

1. **Usability:** The user-interface has to be intuitive and comply with common standards known from desktop applications.

2. **Location independence:** The resulting solution has to allow the access to one's personal data from anywhere.
3. **Unlimited accessibility:** The solution must be based on widely-spread and open technologies to support as many end-user devices as possible, e.g., desktop, tablets and smart phones.
4. **Single Sign-On:** The solution should be able to seamlessly integrate with existing Web services, ideally without the necessity to login several times.
5. **Data Management functionalities:** The integration service must provide functionalities that help to maintain the quality of the integrated data, e.g., detect and merge doublets.
6. **Seamless integration with Web services:** The Web service must provide connectivity to major data sources, for example, popular Web services, to automatically detect changes. Further, it also needs to provide an easy way to use the consolidated data: This can be done by either allowing API access to the stored data or providing functionalities to semi-automatically paste consolidated data into online forms.
7. **Seamless integration with local services:** Often user data is still stored on local devices; therefore our solution must provide an import and export functionality.

Conceptual Framework

In order to comply with design principle 2 and 3 we pursue a Web-based solution. Our approach is based on the common client-server model. The server side comprises of an integration service and a database. The integration service provides the main data management and integration functionalities while the database stores the user-data. The client side includes the HTML front-end and a bookmarklet which enables the upload of data into open Web forms. The term bookmarklet refers to a JavaScript program which is stored as a browser bookmark. Figure 2 illustrates the current UML use-case of the prototype. The integration service is the center piece of our solution. It provides the functionalities to manage one's personal data and is also responsible for the integration of user data originating from local or remote sources. Data Management functionalities include the creation, modification and deletion of contact data within the integration service. They also include the consolidation of similar data objects, for example, two contacts referring to the same natural person.

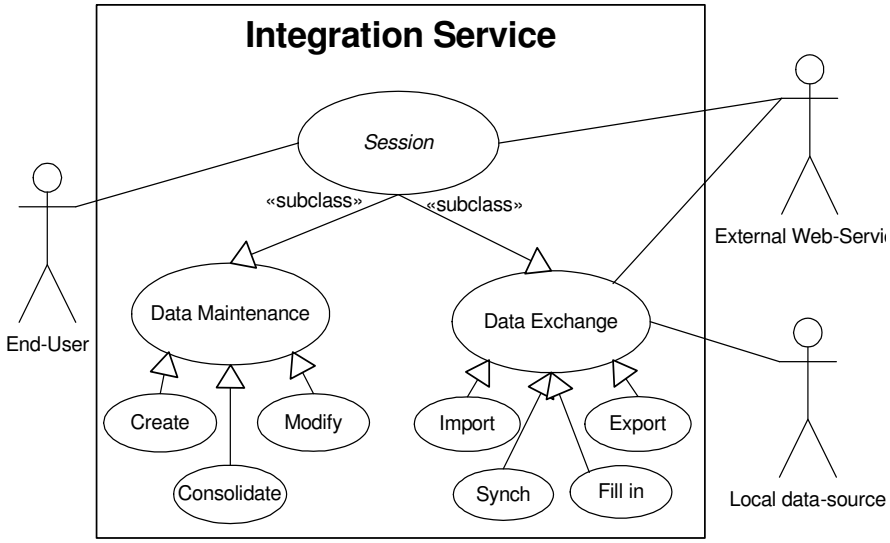

Figure 2: Use-Case

Another major functionality of the system is the connection to external data sources. The system supports Web based data sources as well as local data sources (design principle 6 and 7). The synchronization as of now is triggered manually. If possible the login to a remote data source is performed automatically (design principle 4).

From an architectural perspective we encapsulate the main functionalities of the integration service in three major layers: The communication layer is responsible for the connection to external Web services such as Googlemail, LinkedIn, Facebook, Xing (German career network) and StudiVZ (German social network). The transformation layer is responsible for converting data objects and handing them over to the presentation layer. In addition, it provides the data management functionalities demanded by requirement 5. The transformation layer also allows the import of data objects from local programs such as

Thunderbird and Microsoft Outlook (design principle 7). The presentation layer includes the HTML based user-interface to display and modify data.

From a data integration perspective, we established a mediator-wrapper-architecture as described by Wiederhold (1992). The mediator-wrapper architecture uses a wrapper schema to transform the local schemata of external data sources to the internal global schema of our Web-service. So far, we use predefined wrappers for each connected Web service.

A corner stone in our concept is the bookmarklet which provides an interface to external Websites. The concept of the bookmarklet offers some advantages compared to browser-plugins: first, bookmarklets are supported by most popular Web browsers. Second, the installation of a bookmarklet requires no restart of the browser which makes the integration service instantly available. Third, bookmarklets work also on machines where the user has no influence on the installed software. In contrast to the auto-fill functionality of some browsers our bookmarklet has direct access to a user's personal data stored in the integration service.

Implementation

The prototype uses available open-source tools, frameworks, libraries and protocols. Wherever possible we tried to reuse existing code, e.g., to provide a specific functionality. The center of the architecture is the Web application server Apache Tomcat. Besides the plain http implementation, Apache Tomcat provides a container for Java servlets and Java server pages. Furthermore, we use the DBMS MySQL for the persistent storage of user data. Internally data is stored in the format JSON, which is used by the presentation layer and also by the bookmarklet. Access to Linkedin and Gmail is established using the API of those services. If possible, the protocol OAuth is used to authenticate towards an external Web service. Where an API is not available our prototype uses a parser to extract information from the present document object model (DOM). Connection with local groupware programs is established through an import/ export functionality for CSV files. Furthermore, users can backup data using an XML export functionality.

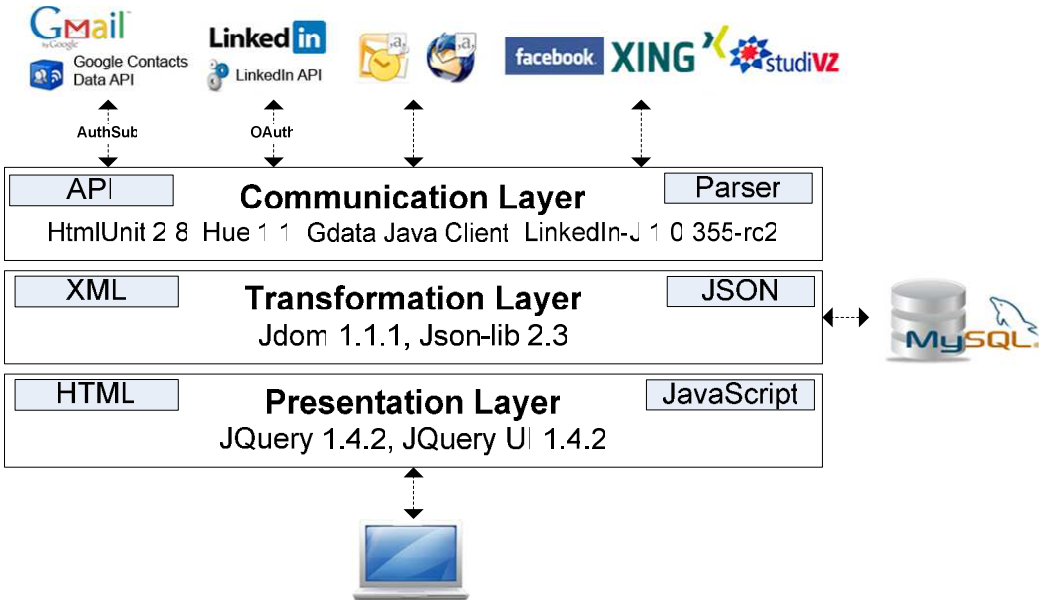

Figure 3: Technical Architecture

The presentation layer is made up of state-of-the art rich-Internet-application (RIA) technology. We use the JavaScript library JQuery to account for differing implementation of HTML and CSS among Web browsers. We picked JQuery UI to create an interactive user-interface providing drag and drop functionality. Data exchange between the presentation layer and the transformation layer is established using XMLHttpRequests. A major obstacle is the common "Same Origin Policy" of Web browsers which prohibits JavaScript access to objects outside the scope of a particular Website. We use an approach known as JSON with padding (JSONP) to bypass this problem. JSONP uses the possibility to include JavaScript files from a remote Web server thereby putting also the connected data objects in the context of the current Website. The transformation layer includes the libraries JDom to handle XML content and the Json-lib to handle JSON data. The connection layer uses the Java library LinkedIn-J to communicate with the LinkedIn API. The connection to Google is established using the Google Data Protocol; the authentication is performed via the proprietary AuthSub interface. Our parser is based on the open-source Web browser HtmlUnit. HtmlUnit is a "GUI-Less" browser for Java programs. It models html documents and provides an API that

allows browsing through a Web site. To retrieve information from a Website Doj is used; a selector engine for HtmlUnit that simplifies the extraction of elements from the DOM.

The bookmarklet is based on JavaScript technology. When the user clicks on the bookmarklet, a script is triggered that loads and injects a new JavaScript library into the DOM of the current Website. This library is responsible for scanning the current DOM for specific tags such as those of input fields and text boxes. Depending on the used name attributes, the script suggests a set of user data (e.g., name, email address and street) to fill into the form. The suggested information is presented in a JavaScript overlay on the current Website (see figure 4).

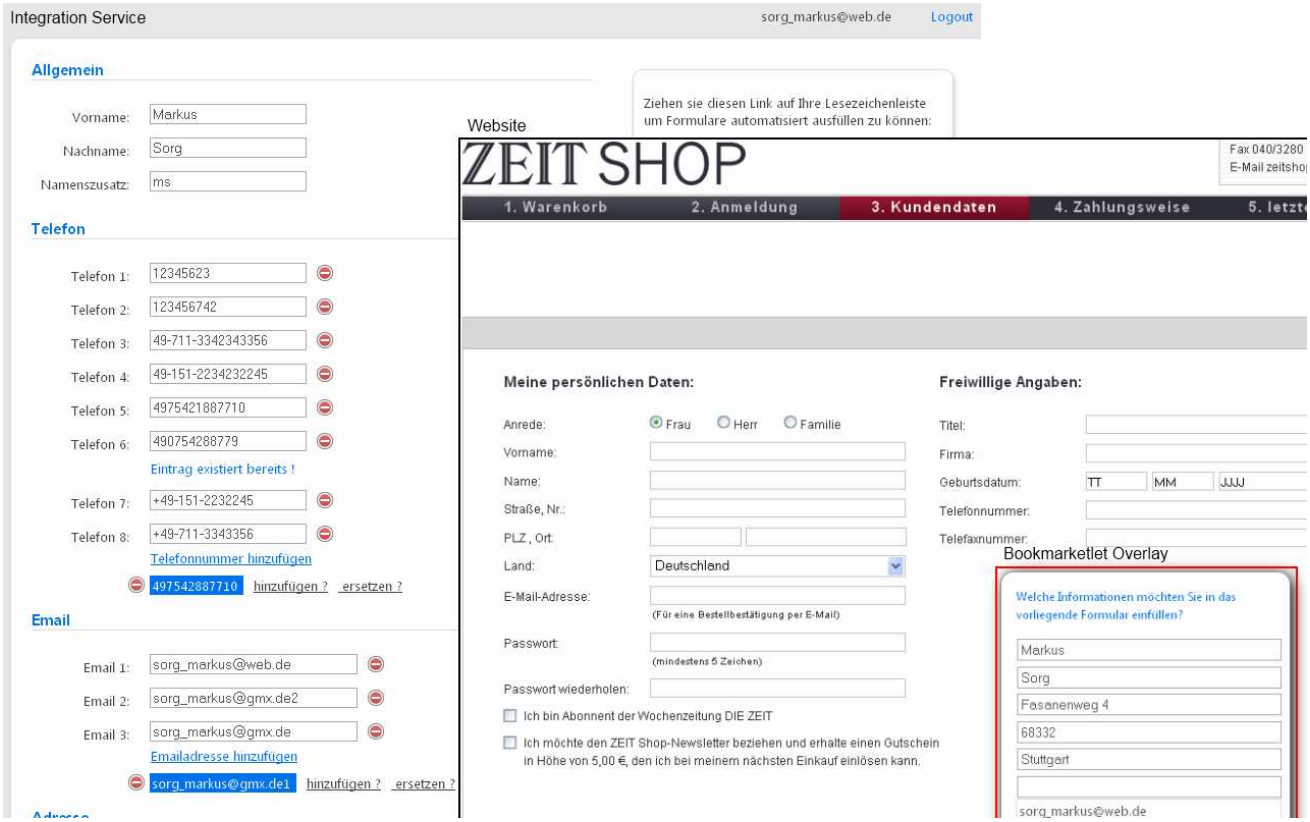

Figure 4: Integration Service Front-End and Bookmarklet Overlay

EVALUATION

The evaluation was conducted using an experiment setup (Jones, 1996). The experiment was designed to reveal the productivity increases created by the integrated storage of personal data in the Web service and the automatic exposure to external Website by the bookmarklet. We do not focus on measuring benefits created by other features of the prototype, such as the (semi-)automatic integration of external services or the data management functionalities.

Experimental Setup

Our experiment was executed with ten student subjects aged 23 to 30 who voluntarily signed up for the experiment. The experiment was conducted in a controlled lab environment with only the subject and the instructor present. The experiment ran on a standard desktop computer. The installed operating system was Ubuntu 10.10 and the provided Web browser Mozilla Firefox (3.6.13) with disabled browser cache. The integration service infrastructure (Tomcat and MySQL) was installed on the same machine. The independent variable of the experiment was the availability of the integration service. The dependent variable was the time a subject needed to successfully fill out and submit an online form. The decision if a form was correctly filled out was initially made by the subject itself, and then the form also had to pass the validity checks of the corresponding Website. Before the experiment every subject was allowed to store his/her personal contact information in the integration service. The personal information included the name, address, e-mail address, if available, credit- and debit card information and further information related to personal identification documents, such as the passport number. Besides the centrally stored information in the Web services, every subject had the original documents on hand. Subjects were not allowed to use other installed programs on the computer. Subjects were introduced into the concept of the integration service

and the usage of the bookmarklet with a basic instruction leaflet. The main activities the test subjects had to execute were also documented on a second leaflet. The subjects had to fill out six registration Web forms from different domains.

In detail, the following forms had to be completed:

Online Form	Personal information
Zeit onlineshop (newspaper)	salutation, name, address, telephone number, delivery address, e-mail address, password
Air Berlin (airline)	salutation, name, address, telephone number, e-mail address, birthday, user name, password, nationality
Europcar (car rental)	salutation, name, address, telephone number, delivery address, e-mail address, password, drivers license id, drivers license issuing country, places issued, drivers license expiration data, passport id, issuing country, place issued, issue date, expiration date
Virgin Airline (airline)	salutation, name, address, telephone number, e-mail address, birthday, user name, password, nationality, security question, answer
Hertz (car rental)	salutation, name, address, telephone number, e-mail address, birthday, user name, password, nationality
Spiegel Online (news magazine)	salutation, name, address, telephone number, e-mail address, birthday, user name, password, nationality

Table 1: Web Forms Content

In the first part of the experiment the subjects had to complete the six forms without the help of the integration service and the associated booklet. The six forms were presented in a random order. Time was taken in seconds using a custom developed stopwatch-bookmarklet. The test subjects had to click on it in order to start and stop the clock, the instructor made sure that a test subject did not stall time or stop the clock before the form was completely filled out. In the second part of the experiment, the subject had to complete the six forms again. This time he/she was allowed to use the support of the bookmarklet to fill out the forms, again in randomized order. The first round of the experiment was conducted with ten different subjects. After that we conducted a second round with the same ten subjects.

Results

Altogether, we collected 240 characteristics, including 120 characteristics for each run, 120 manual runs and 120 with the help of the bookmarklet, 40 for each form and 24 for each subject.

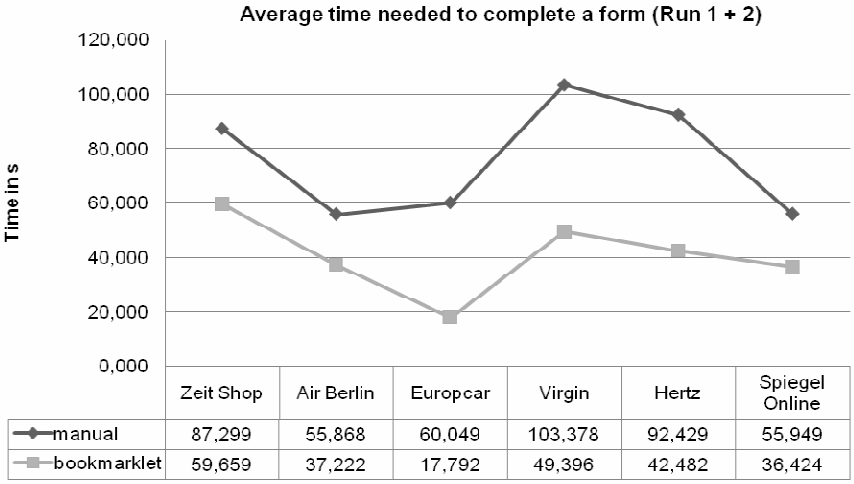

Figure 5: Average time needed to complete a form aggregated over all users and runs

Interpretation

The usage of the integration service and the bookmarklet has lead to an absolute decrease of the time needed to fill out each form. In relative numbers and aggregated over all users and all Web sites, the time necessary to complete the forms could be

reduced by 46,6% (Standard deviation 15,4%) indicating the potential of our approach. With respect to simple forms such as the Zeit form, it seems that the benefits of our approach are majorly created by the bookmarklet. The information asked there, for example, the email address, is usually memorized by users. Therefore, the measured time savings must go back to the fast auto-fill functionality of the bookmarklet and not to decreased search costs. However, when it comes to more complex forms, such as the Europcar form, which ask for personal information seldom used, the benefit of the integration service also seems to kick in. Here the user also benefits from decreased search costs resulting from the integrated view on his/her personal data. The outstanding high productivity increases measured for Virgin and Hertz, even though both forms ask only for simple information, seem to go back to the complicated layout of such Web forms that made it hard to enter information manually.

The chosen experiment design incorporates several weaknesses. First, the limited amount of test subjects does not allow for a generalized conclusion. Second, our population of subjects represents a rather homogeneous group. All subjects were of a similar age (23-30) and therefore rather familiar with the use of Web technology. In addition, they share a similar educational background since all of them were students. Hence, the results achieved are not transferable to end-user in general. Third, we have picked the Web forms based on the current capabilities of our bookmarklet. However, based on this biased selection we cannot convey our results on Web forms in general since our selection is not representative. Fourth we have encountered several threads to our internal validity: Due to our experiment setup we were not successful in avoiding learning effects. After filling out one form manually, users had gained some understanding about the general setup of a form which helped them to navigate faster. Also, some users developed an ambition to fill out a form faster than before. This effect leaves the degree of the measured efficiency increases debatable. Yet, it does refute the positive impact of our approach in general. Fifth, the chosen experiment setup only proves that an approach which makes use of an electronic tool, in this case our prototype is superior to an approach which does not. If our concept is actually superior to other manual but yet electronic based approaches, e.g., Excel macros or browser auto fill functionalities must be clarified in further experiments.

CONCLUSION

In this paper we have presented a holistic approach to provide end-user-driven data interoperability. A key element of the concept is a Web based service which integrates and stores data of different origin centrally. The Web service is complemented by a bookmarklet which allows the easy usage of data wherever needed. Regarding our research questions we have presented seven design principles that need to be considered in a concept for end-user-driven data interoperability. The evaluation also showed that our principle can be successfully transformed into a conceptual model which by all means increases the productivity of users working with personal contact data.

In future research we will continue this promising path and improve our approach. In specific, we plan the following future research activities: Qualitative methods to raise additional user-requirements which can be used to extend and refine our preliminary set of design principles. Additional design principles may ask for a flexible data model which allows a single user to add custom attributes to his/her personal contact data or the availability of an underlying social network which allows sharing and connecting of personal data objects between users. In terms of our conceptual model, we also want to address the question how to maintain and extend the connectivity with remote Web services. We have developed first thoughts to tackle this problem with a crowd sourcing approach. In order to proceed in our design science research and eventually refine our concept into a valid design theory it is also necessary to conduct more elaborated evaluation studies. Future evaluation will investigate not just the impact of our approach as a whole, but instead measure the impact of single design aspects.

ACKNOWLEDGEMENTS

We would like to thank our diploma student Markus Sorg for the implementation of the Web service. Furthermore, we want to thank the 10 students which voluntarily participated in our experiment.

REFERENCES

1. Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D. G., and Ducheneaut, N. (2004) What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.
2. Bellotti, V., Ducheneaut, N., Howard, M., Smith, I., and Neuwirth, C. (2002) Innovation in extremis: evolving an application for the critical work of email and information management. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*.
3. Blandford, A. E., and Green, T. R. G. (2001) Group and individual time management tools: what you get is not what you need. *Personal and Ubiquitous Computing*, 5(4), 213–230.

4. Boardman, R., and Sasse, M. A. (2004) Stuff goes into the computer and doesn't come out: a cross-tool study of personal information management. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.
5. Bondarenko, O., and Janssen, R. (2005) Documents at hand: Learning from paper to improve digital technologies. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 121–130.
6. Brabham, D. C. (2008) Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1), 75.
7. Collan, M., and Tetard, F. (2007) Lazy User Theory of Solution Selection. In *Proceedings or the CELDA 2007 Conference*, 7–9.
8. Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. (1989) User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8), 982–1003.
9. Dong, X., Halevy, A., Madhavan, J., and Nemes, E. (2005) Reference reconciliation in complex information spaces. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, June, 14–16.
10. Dong, X., and Halevy, A. Y. (2005) A Platform for Personal Information Management and Integration. In *CIDR*, 119-130.
11. Dourish, P., Edwards, W. K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D. B., et al. (2000) Extending document management systems with user-specific active properties. *ACM Transactions on Information Systems (TOIS)*, 18(2), 170.
12. Dubey, A., and Wagle, D. (2007) Delivering software as a service. *The McKinsey Quarterly*, 6, 2007.
13. Ducheneaut, N., and Bellotti, V. (2001) E-mail as habitat: an exploration of embedded personal information management. *interactions*, 8(5), 38.
14. Farrell, J., and Nezlek, G. S. (2007) Rich Internet Applications The Next Stage of Application Development. In *Information Technology Interfaces*, 2007. ITI 2007. 29th International Conference, 413 -418.
15. Freeman, E., and Gelernter, D. (1996) Lifestreams: a storage model for personal data. *ACM SIGMOD Record*, 25(1), 80–86.
16. Geambasu, R., Cheung, C., Moshchuk, A., Gribble, S. D., and Levy, H. M. (2008) Organizing and sharing distributed personal Web-service data. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, New York, NY, USA, 755–764,.
17. Halevy, A., Rajaraman, A., and Ordille, J. (2006) Data integration: The teenage years. In *Proceedings of the 32nd international conference on Very large data bases*.
18. Henderson, S. (2004) How do people organize their desktops? In *CHI'04 extended abstracts on Human factors in computing systems*, 1047–1048.
19. Jammalamadaka, R. C., Mehrotra, S., & Venkatasubramanian, N. (2005) Pvault: a client server system providing mobile access to personal data. *Proceedings of the 2005 ACM workshop on Storage security and survivability*, 123–129.
20. Jones, R. A. (1996) Research methods in the social and behavioral sciences. Sinauer Associates.
21. Jones, W. (2007) Personal information management. *Annual review of information science and technology*, 41, 453.
22. Jones, W., Phuwanartnurak, A. J., Gill, R., and Bruce, H. (2005) Don't take my folders away!: organizing personal information to get ghings done. In *CHI'05 extended abstracts on Human factors in computing systems*.
23. Kuechler, B., and Vaishnavi, V. (2008) On theory development in design science research: anatomy of a research project. *European Journal of Information Systems*, 17(5), 489–504.
24. Lansdale, M. W. (1988) The psychology of personal information management. *Applied Ergonomics*, 19(1), 55–66.
25. Lee, D., & Tsatsoulis, C. (2005) Intelligent data entry assistant for xml using ensemble learning. *Proceedings of the 10th international conference on Intelligent user interfaces*, 83–89.

26. McAfee, A. (2006) ENTERPRISE 2.0. *MIT Sloan management review*, 47(3), 21.
27. Whittaker, S., and Sidner, C. (1996) Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*.
28. Wiederhold, G. (1992) Mediators in the architecture of future information systems. *Computer*, 25(3), 38–49.
29. Wiederhold, G. (1993) Intelligent integration of information. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data,* 434–437.