Paloma Díaz · Tim Hussein
Steffen Lohmann · Jürgen Ziegler (Eds.)

Proceedings of the International Workshop on

Data-Centric Interactions on the Web (DCI 2011)

Co-located with the 13th IFIP TC13 Conference on

Human-Computer Interaction (INTERACT 2011)

Lisbon, Portugal, September 6, 2011

Title: Proceedings of the International Workshop on Data-Centric Interactions on the Web (DCI 2011)

Editors: Paloma Díaz, Tim Hussein, Steffen Lohmann, Jürgen Ziegler

ISSN: 1613-0073

CEUR Workshop Proceedings
(CEUR-WS.org)

Editors' addresses:

Universidad Carlos III de Madrid
Departamento de Informática
Avenida de la Universidad 30
28911 Leganés, Madrid (Spain)

{pdp | slohmann}@inf.uc3m.es

University of Duisburg-Essen
Department of Computer Science and Applied Cognitive Science
Lotharstr. 65, 47057 Duisburg, Germany

{tim.hussein | juergen.ziegler}@uni-duisburg-essen.de

Preface

While the original World Wide Web was mainly a Web of documents, today's Web is characterized by an ever-growing amount of data that is published and linked in structured formats. Data about products or services is increasingly made available for public access by companies (e.g., Amazon A2S API, Google Maps API, etc.). Governments and organizations publish more and more often statistics and other data on the Web, usually with the aim to increase transparency and empowering the public to utilize this data (e.g., data.gov, data.gov.uk, etc.). In other attempts, structured data is extracted from Wikipedia (e.g., DBpedia) or from multiple Web sources (e.g., Freebase). Last but not least, huge amounts of data and metadata are created by the Web users themselves, either indirectly (e.g., via social tagging) or in directed community efforts (e.g., Open Directory Project, OpenStreetMap, etc.). These are just a few of the many examples where data is published on the Web nowadays.

Several standards and best practices for the description, publication, linking, and exchange of Web data have been developed in the past. Popular examples are W3C specifications such as XML, SOAP, RSS, and RDF, the recommended best practices of the Linked Data initiative, and various vocabularies that emerged from these approaches (e.g., Dublin Core, SKOS, etc.). A similar line is taken in less formal attempts to structure data, such as microformats or advanced tagging approaches (e.g., geotagging, hashtags, etc.).

However, the full potential of the data can only be exploited with well-designed user interfaces and powerful interaction techniques that allow an efficient exploration and utilization of the data. Although some early attempts to investigate the interaction with this data have been made in the past, there is still a large number of research questions and practical challenges that are not yet sufficiently addressed. Reoccurring interaction problems are differently solved and more general design recommendations and guidelines are just beginning to emerge. A need for reusable design patterns and interaction techniques as well as novel ideas, tools, and methods to present Web data to the users is clearly recognizable. These and related issues of data-centric interactions on the Web were addressed by the workshop. This volume contains revised versions of the peer-reviewed papers that were presented at the workshop.

Novak and Preusse discuss the importance of providing different views on the same data to support sensemaking and social interaction. They present an approach that aims to support collaborative interaction and sensemaking between citizens and municipal administrations through visual tools. *Ravendran et al.* discuss possibilities for online banking customization through tag-based user interfaces. They introduce a customization framework derived from a literature study and present a prototype that illustrates their idea for some key resources of online banking. *Brunk and Heim* present tFacet, a tool that uses well-known interaction concepts to facilitate faceted exploration of semantic data. It implements features for hierarchical-faceted navigation and easy customization of the presented results by a combination of table and tree components. Another tool that accesses Web data is presented by *Parra et al.*. More! can be used on mobile devices to gather and display information about speakers at scientific events, such as conferences or workshops. The approach is based on the SWRC and FOAF ontologies and follows REST and Linked

Data principles to structure and access the data. Finally, *Stegemann et al.* describe X3S, a component-based approach to filter and present semantic data from the Web. They demonstrate the applicability of their approach by an editor that is able to create, edit, and preview X3S stylesheets, and report a comparative user study that evaluates their approach.

The workshop was co-located with the 13th IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2011) which took place in Lisbon from September 6 to 9. We thank all authors for their contributions and the organizers of INTERACT 2011 for providing us with the opportunity to organize this event.

November 2011

Paloma Díaz, Tim Hussein
Steffen Lohmann, Jürgen Ziegler

Contents

Designing Visual Systems for Social Data Analysis in Open Government Applications
Jasminko Novak, Sebastian Preuße **7**

Online Banking Customization via Tag-based Interaction
Rajinesh Ravendran, Ian MacColl, Michael Docherty **19**

tFacet: Hierarchical Faceted Exploration of Semantic Data Using Well-Known Interaction Concepts
Sören Brunk, Philipp Heim **31**

More!: Mobile Interaction with Linked Data
Gonzalo Parra, Joris Klerkx, Erik Duval **37**

Interacting with Semantic Data by Using X3S
Timo Stegemann, Tim Hussein, Werner Gaulke, Jürgen Ziegler **49**

Designing Visual Systems for Social Data Analysis in Open Government Applications

Jasminko Novak[1,2], Sebastian Preuße[3]

[1] University of Applied Sciences Stralsund, Zur Schwedenschanze 15
18435 Stralsund, Germany
[2] European Institute for Participatory Media, Wilhelmstr. 67
10117 Berlin, Germany
[3] Humboldt-Viadrina School of Governance, Social Innovation Lab, Wilhelmstr. 67
10117 Berlin, Germany
j.novak@eipcm.org, sebastian.preusse@silab.humboldt-viadrina.org

Abstract. In this paper we discuss our preliminary experiences in designing a visual system for social data analysis in collaboration between citizens and municipal authorities through a web-based open government system. Based on the integration of theoretical models with findings from formative evaluations and stakeholder workshops with a prototype system, we present a multi-perspective visualization model and user interface designs addressing specific requirements of social data analysis in heterogeneous stakeholder settings.

Keywords: Social Data Analysis, Information Visualization, Multi-Perspective Visualization, Collaborative Sensemaking, Open Data, Open Government

1 Introduction: From Visualization to Collaborative Sensemaking

The use of information visualization for facilitating the discovery of insights in large or complex data collections has been extensively researched. Common approaches consider the use of visualization techniques for augmenting cognitive capabilities of individual users in recognizing contexts, patterns or relationships in a data collection as a means of constructing new knowledge [2]. More recently, there has been a growing interest in supporting the interpretation of large or complex information sets through collaborative use of shared visualizations in asynchronous distributed scenarios. Systems such as Many Eyes [9] or SenseUs [4] provide web-based platforms allowing users to create interactive visualizations of own or shared data collections and to share them with others. Besides creating customized visualizations from predefined templates (e.g. pie charts, stacked graphs) such systems allow the saving and recall of specific visualization states (e.g. zoom, filter or time scale parameters) which can then be accessed by others (application state bookmarking [10]). Collaborative analysis is supported by textual comments and graphical annotations that link user contributions to the related views and vice-versa (double-lined discussions [4]). Most recent solutions allow existing visualization systems to be extended with user comments and discussions [11].

The basic idea of such „social data analysis“ [10] is that by coupling visualization with asynchronous social interaction, the process of individual sensemaking in which people create new knowledge by collecting, organizing and interpreting information [6] can be made more effective [4]. The associated notion of collaborative sensemaking emphasises that sensemaking is often a social process in which the meaning of data and information is negotiated against specific social contexts involving shared backgrounds, frames of reference, goals and perspectives [1],[5]. While these questions may not play a critical role in the above examples of general-purpose platforms, they become of central importance when designing for purposeful collaboration with shared goals, such as in collaborative decision-making in organizations [4] or in heterogeneous stakeholder networks with conflicting perspectives on the meaning of information [5].

Previous research suggests that such contexts require considering specific requirements in order to support effective sensemaking through exploratory information access and analysis [5]. In particular, this refers to the need and potential of providing visual views of heterogeneous information collections from clearly defined multiple perspectives reflecting personal points of view of individual users or shared perspectives of specific user groups. This raises both questions of how such personal and shared perspectives can be defined as well as how they can be effectively visualised and made useful for social discovery transfer in heterogeneous data collections [5]. Aforementioned approaches do not consider these requirements and offer only incidental support corresponding to this need (e.g. the possibility of bookmarking and sharing specific views of the *same* visualization [10][11]

In this paper we discuss our preliminary experiences in designing a visual system for social data analysis in collaboration between citizens and municipal authorities through a web-based open government system. In particular, we discuss the requirements and design principles for designing visual systems for this specific class of applications and present one possible solution for turning them into practice. This is based on the integration of theoretical underpinnings with findings from formative evaluations and stakeholder workshops with a prototype system. The developed solution presents a multi-perspective visualization model and user interface designs incorporating well-known visualization techniques (tree maps, geo-visualisation) applied in a novel way to satisfy the requirements of social data analysis in heterogeneous stakeholder settings. In this way, the paper provides an application-oriented contribution to this emerging area of research.

2 Application Setting: Social Data Analysis for Open Government

The underlying idea of open government approaches is that through open publishing and public analysis of data collections concerning public services better government and satisfaction of citizen needs can be achieved. Different web portals make various forms of data collections related to citizen life (e.g. city budgets, transportation statistics) available online. While such open data collections are being provided by an increasing number of local and national authorities or citizen initiatives (e.g. Germany’s Open Data Network), the provision of systems and tools for easy collection, organization and analysis of this data is still at its beginnings. The few existing visual tools in this domain typically provide straightforward applications of well-known visualization techniques (e.g. tree maps of country budgets [15]) without considering specific use cases or application settings and their requirements. The more sophisticated current visual systems for social data analysis (such as [4], [9] or [11]) are designed as general-purpose tools and could in principle be applied to such contexts, but this has not yet been the case (to the best of our knowledge). Similarly, none of these approaches consider the existing body of knowledge on supporting collaboration in heterogeneous settings or the use of visual information interfaces in such settings [5].

We argue that the effective application of such approaches to the application domain of open government requires considering specific aspects and requirements of this particular application context and developing domain-specific solutions. On one hand, this concerns the inherently problematic nature of the underlying collaborative scenario. Use cases of social data analysis specifically discussed in previous work point either to relatively homogeneous user groups with implicit shared interests (e.g. a group of analysts analyzing a shared data set in [9]), or encompass a heterogeneous population of general users sharing and commenting on personal visualizations [9], or they refer to a general audience who may become aware of a shared interest defined by using the visualization (e.g. baby names or job statistics [4]).

In contrast, the open government applications are characterized by a collaboration setting formed around specific goals and conflicting interests of different user groups (e.g. citizens, local authorities). Previous research has shown that using visual information tools to support such contexts requires considering very specific requirements. This includes the need for providing visual overviews of information collections from clearly defined perspectives reflecting personal points of view of individual users or shared perspectives of specific groups of users [5]. This raises both questions of how such personal and shared perspectives can be effectively elicited, visualised and made useful for social discovery transfer in open data collections [5]. Such concerns are also grounded in theoretical frameworks of collaboration in heterogeneous settings. The notion of "boundary objects" emphasises the importance of artefacts which allow perspectives of different user groups to be used independently while at the same put in relation to each other [8][7]. Similar requirements form the basis of the well-known "perspective making – perspective taking" model of knowledge transfer [1]. Finally, a number of current practical initiatives aiming at supporting citizen participation in local government through easy-to-use tools for collaborative elicitation of citizen needs [13][14][16], reflect an existing practical need for capturing a shared perspective of a specific stakeholder group (citizens) and communicating and visualizing it for others (municipal officials).

This raises the following questions: How can we effectively design visual tools for social data analysis in such heterogeneous application contexts? Which design principles from literature on collaboration support and from recent general-purpose visualization platforms can be successfully transferred? What specific requirements characterize this application context? And can we identify conceptual and design elements of a specific visualization model that could satisfy such classes of applications and thus further current practice in this emerging field?

3 E-Local: A Web Platform for Participatory City Management

We have been investigating these questions within the E-Local project for participatory city management [12]. The E-Local project develops a Web2.0 platform for stimulating active citizen participation in local government through collaborative elicitation of citizen needs, their interactive visualization and online dialogue between the citizens and the local administration. A central element of its design is the data-centric interaction around a shared visualization facilitating online discussion.

3.1 Use Case Scenario

The basic use case of the E-Local project is depicted in Fig. 1. Citizens enter need requests (e.g. "Please fix the road holes in Wins street") through the E-Local web interface or a mobile app. The requests are stored in the E-Local database and displayed in a geographic visualization based on Google Maps (GPS-data is captured by the phone app or the user can pinpoint the location). Once the request enters the E-Local system it can be edited online and forwarded to one's social network to collect supporters (via Email, the E-Local network or one's Facebook account). During this "mobilization phase" the request can be discussed and commented upon by all users. After a determined time this phase ends and the request (with corresponding supporter votes and comments) is forwarded to the municipal administration. This starts the online dialogue ("feedback and discussion phase") in which municipal officials can accept the request and propose a solution or refute it (with an explanation).

Figure 1. Typical Use Case and Visual Design of the E-Local Prototype

In both cases the citizens can vote for accepting or refuting the administration's decision. In case of refutal, they can start a citizen initiative for solving the request and raising the required funding on their own[1]. All requests accepted by the local authorities enter the "implementation phase" in which their progress is monitored and communicated by the local administration. In this way, a model for pro-active citizen participation in the management of local municipal affairs in a collaborative process with local authorities is realized (which distinguishes E-Local from other citizen platforms, such as [13], [14], [16]). Through the described workflow a data set of citizen needs and corresponding discussions is created, visualized to different parties and made available for collaborative analysis and social interaction around the shared visualization. The basis for the visualization is provided by collaborative user-generated data sets (citizen needs in the first-iteration prototype), which are then integrated with external open data sets (municipal data in the second system design).

3.2 Visual System Design and Protoype Implementation

The main hypothesis of the system concept is that centering the interaction around a shared visualization can further collaboration between the two stakeholder groups. For this reason a central element of the user interface both for citizens and municipal officials is a visual map of the citizen needs dataset contextualized by geographic location (Google Maps). The map is practically permanently present and accompanies the vast part of information access, interaction and discussion (e.g. see request input, overview and detail pages in Fig 1). While citizens and the municipal officials are likely to use the map for different purposes (e.g. entering requests, finding out what is going in one's neighbourhood for citizens vs. monitoring areas with intense citizen activity or comparing needs with planned municipal actions), the map represents a shared context of reference around which the communication between their different perspectives can occur. Thus, the map implements a kind of a "boundary object", providing an important mechanism for collaboration in heterogeneous settings [8][7].

At the same time, to cater for the distinctive needs of the two very different groups of users the E-Local system has been designed as a multi-perspective information system from its outset. The system design involves two different views on the user data and the corresponding user interfaces: one for the citizens and one for the municipal officials. While the basic functionalities and the GoogleMaps visualization are available to both groups, the municipal officials have access to an additional instrument, the CityCockpit. The primary means of visual exploration and information access for the citizens is provided by the Google Maps visualization that allows them to quickly spot and get information about relevant citizen requests in their neighborhood. The main design assumption here has been that "locality" is the primary measure of relevance for the citizen's perspective. In contrast, based on requirements interviews with municipal representatives, the primary means of access for the municipal officials has been conceived in terms of a monitoring and control center with an event ticker, statistical reports and visual charts for aggregated analysis of citizen requests in addition to the GoogleMaps visualization. The described

[1] To this end the E-Local platform interfaces with the online donation platform betterplace.org.

delineation of perspectives in the user interface implements the second important and theoretically grounded requirement for supporting collaboration between heterogeneous groups of users: the availability of means for the users to work within their own distinctive perspective corresponding to their own terms, interests and frames of reference (perspective making) [1]. Such a combination of separate user interfaces with distinctive perspectives and a shared visualization (Google Maps) used by both groups implements the model of perspective making and perspective taking: while working in own interfaces allows the expression of distinctive perspective of each group (perspective making), the map as a boundary object connects the two perspectives thus supporting the discovery of relationships between them (perspective taking), without forcing any group to abandon their own perspective.

To implement the described model in practice, the system architecture allows easy implementation and linking between different views on available data sets through a MVC-like architecture and considers the conceptual multi-perspective visualization model proposed in [5]. The prototype system implements the described use case and functionalities based on the Django web application framework with a mysql database. The frontend application is realized with Ajax and the mobile application is an Android App. The system architecture is designed in a modular way with APIs allowing easy integration of new data sets. In order to quickly realise a functioning version of the system that can be tested and iteratively extended in practice, the first-iteration protoype system implements a basic version with the communication and data collection module alongside with the basic visualization mode (geo-localization view) and basic social analysis functions (comments and discussions). The preliminary CityCockpit has been provided at the level of a visual mockup, illustrating basic reporting and analysis functionalities for municipal officials (Google Maps visualization, new requests ticker, filtering of requests by topic, requests pending answers, chart of citizen requests distribution by topic). Such implementation allowed us to elicit feedback from target users in early formative tests and stakeholder workshops with a cooperating municipality in order to verify the described design decisions and assumptions before proceeding to final realization.

3.3 Lessons from Formative Evaluation and Stakeholder Workshops

A formative evaluation of a first-iteration E-Local prototype has been undertaken with 5 participants representing the citizen users. The participants were roughly equally spread by age and sex within two demographic groups (two aged 24-26, three aged 50-56; two male and three female participants) and within a range of professional backgrounds (fashion design, public administration, consulting, electric technician). Most of them were regular Internet users (four participants between 0,5-7 h/day) and highly proficient in computer use (three out of five) with three of them using one of the well-known social networks. The participants were provided with access to the web application with the basic functionalities implementing the typical E-Local use case as depicted in Fig. 1 (submitting a request, extending the request, displaying requests on a map, searching for requests, setting up user profiles, submitting comments, going through the 3-phase cycle of the online discussion). They were asked to complete a set of typical tasks corresponding to this use case and received no prior training with the prototype. User feedback was collected through a Likert-scale questionnaire. Table 1 depicts the main results of relevance for this paper.

Table 1. Results of the formative user evaluation of the E-Local Prototype (citizen group).

E-Local structure and functionalities		---	--	-/+	++	+++	
Resolving a problem with the municipal authorities through Elocal...	is time consuming	0	0	2	2	1	is uncomplicted and direct
	is untransparent	0	0	3	1	1	is transparent
The 3-phases model...	is not clear	0	0	3	1	1	is logical and understandable
	is too cumbersome	0	2	3	0	0	is expeditious enough
The permanent presence of the map...	is unnecessary	0	0	0	1	4	is meaningful
	distracts me from the task	0	0	0	2	3	provides me continuously with an overview
The navigation with the map...	is not helpful for searching	0	0	0	0	5	helpful for searching
The overal construction of the Website...	is confusing	0	0	2	2	1	is clear
	is inapropriate for the given contents	0	0	1	3	1	is apropriate for the given contents
Creating a citizen request...	was complicated	0	0	0	2	3	was simple
	was not personalizable enough	0	0	0	4	1	was sufficiently personalizable

Statement of attitude Elocal...	---	--	-/+	++	+++
...is fun to use	0	1	2	2	0
...interaction is clear and understandable	0	0	3	2	0
...would strengthen the feeling of community within a neighbourhood	0	1	1	3	0
...would help to improve the relationship between citizens and the municipality	0	0	0	3	2
...would try to bring as many of my friends and acquaintances as possible to Elocal	1	2	1	1	0

Expectations How would you primarily use E-Local?	---	--	-/+	++	+++
...would primarily keep looking for what is happening in my area and participate accordingly	0	0	1	1	3
...would try to publish all problems that I become aware of as citizens requests	0	1	2	2	0
...would use Elocal to let the town authorities know what i really think of them	2	0	1	2	0

On one hand, the results suggest the suitability of the overall concept and the E-Local prototype with largely positive to very positive responses to the individual functionalities[2]. They are likely to be partially biased due to the fact that most participants were highly proficient in using computers and the Internet on a daily basis. However, this bias is partly offset by the fact that such users tend to represent a typical core group of users in community-based portals. On the other hand, positive to very positive responses to the overall construction of the web prototype, the clarity of interaction design and the central role of visualization suggest that the described principles of the system design and the central role of visualization resonate positively with the target users. In particular this relates to the use and permanent presence of the map as a central element for structuring navigation, interaction and collaboration. All test participants rated the usefulness and importance of the map as a central interaction element high or very high (Table 1, upper center, rows three and four). In addition, four out of five users confirmed that their primary mode of use would be to continuously monitor the activities in their local area and participate accordingly (Table 1, bottom right, row three). This supports the design decisions of choosing “locality” as the primary concept for modeling the citizen’s perspective (citizen requests dataset) and choosing the map as a central element for mediating interaction.

[2] The only exception is the 3-phases workflow model that hasn’t quite convinced and was perceived neutrally (three users) or as cumbersome (two users; Table 1, row two). This requires further investigation into the exact reasons for such perception.

The results thus indicate that the described requirements and design assumptions for a multi-perspective visualization interface and data-centred visual interaction model could be well suited for such a user group.

As none of these test users provided free comments on the questionnaire, we undertook an additional informal focus group with five additional participants to get more contextual feedback. These participants were recruited among trainees and employees of the Humboldt-Viadrina School of Governance who were not directly associated with the project. They undertook the same tasks as the previous group but provided only informal feedback. While the general feedback on the overall concept and usability of the prototype largely corresponded to the above results, several remarks provided a more differentiated picture. In particular, this includes a more critical stance towards the explicit 3-phase model with automatic transition constraints ("What if I would want to finish a phase earlier, because I feel to have enough supporters for my issue?") and the use of the number of supporters as representative for the importance of an issue (incidentally, the same issue was raised later on by the municipal officials). With respect to the visualization model, the most interesting observation was the request to provide the same kind of analysis and reporting tools foreseen for the municipal officials also to the citizens ("Why is the CityCockpit intended only for the municipal administration and not for citizens? I would like to be able to see the statistics and visual charts too. Especially, if they tell me how the money spent by the administration is related to my needs.") Another participant inquired about the possibility to provide information about already planned actions of the municipal administration for a given area so as to avoid duplicate requests ("I would like to know what the municipal administration has already planned to do in my area so I don't need to waste time on entering requests which will be resolved anyway".). This feedback suggests that the visualization of the municipal perspective should not be considered only in its own right but in relation to the citizen's perspective as well. Incidentally, this is a nice illustration of how the theoretical requirements of the "perspective making – perspective taking" model [1], are confirmed in practice: while the voicing of the citizen perspective (perspective making) is important in its own right, there is also a need to put this in relation to a visible municipality perspective, which needs to be understood by the citizens as an orientation for their action (perspective taking).

The feedback of the municipal administration was elicited in a stakeholder workshop with municipal officials of the cooperating municipality (a town of approx. 60,000 inhabitants). The workshop involved 10 participants: the heads of the administrative departments ("Dezernenten") and their senior staff, the coordinator for interdepartmental projects, the communications officer (also in charge of the internet strategy) and the head of IT. As the basis for the discussion, after an introductory presentation the prototype system was presented in a live-demo by replaying a scenario for a typical use flow on the citizen-side and on the administration-side. This included the CityCockpit in form of a visual mockup (features as stated in Section 3.2). The main findings of this workshop are summarized in Table 2.

Table 2. Main results from the stakeholder workshop with municipal officials.

Feedback	Requirement
Increase the system usefulness by making visible what the administration already does for the citizens.	Provide a clear visualization of the administration perspective.
Support the discovery of relationships between citizen requests and current municipal actions (e.g. repairing street holes in winter; planning construction measures) – for the officials *and* for the citizens (!)	Support discovery of relations between visualizations of different perspectives.
Provide mechanisms for assessing the relevance of citizen needs by criteria other than number of supporters.	Support interactive parameterization of visualizations
Tools for monitoring and analysis must make existing work easier, not adding additional workload ("Who will analyse all this data?", "We have enough to do").	Integrate visualization and analysis closes with existing tasks and processes.

Overall the results can be summarized as follows: while gathering and visualizing citizen needs was considered useful by some participants (e.g. the department of construction), the majority of municipal officials felt that in order to become useful for them, the system needs to more actively support the communication of actions that the administration performs for its citizens (see Table 2). The CityCockpit should also provide an easy analysis of the relative importance of citizen needs in comparison to each other and to existing (or planned) budgetary actions as well as to the available budget for specific areas or topics. Moreover, it should make clear these relations to the citizens, not only the administration (with the expected effect of reducing the number of generated citizen requests). Finally, all participants shared the concern that functions for activity monitoring and data analysis must make existing work easier instead of adding additional workload (reflecting the basic worry regarding the introduction of the system in practice). In terms of requirements, this points to the need to support the visualization of their own perspective for each group of users (citizens vs. officials) while at the same time making it perceivable for the other and providing easy-to-use ways for discovering relationships between them. Such results (alongside with previously discussed citizen evaluation) led to a complete redesign of the CityCockpit mockup and the associated multi-perspective visualization model for social data analysis that are presented in the next section.

4 The Open City Cockpit

The results of formative user evaluation and workshops led to a complete redesign of the City Cockpit. The main focus has been the development of a multi-perspective visualization model, which would satisfy both the needs of the municipal officials and the citizens while supporting collaborative data analysis between the two groups. This resulted in a visualization design that extends existing approaches to social data analysis and is grounded both in theory and in requirements elicited from practice.

4.1 Visualising Multiple Perspectives

The new design of the Open City Cockpit is depicted in Fig. 3. One shared user interface provides a unique point of access for all users, regardless of affiliation (citizens vs. officials). Within this interface any user can choose any of the available perspectives to be displayed in one of the available visualization types. The current design includes two main visualization types: a geographic map based on Google Maps (Fig. 3, left) and a tree map visualization (Fig. 3 right). A perspective is now represented by the corresponding data set (citizen needs dataset vs. municipal budget dataset) and a selected visualization type. To satisfy the need of both supporting working within one's own perspective as well as facilitating the discovery of relationships between the perspectives, the user can choose between three different modalities of display: a one monitor view, a two monitor view and a mixed monitor view. One monitor view displays only one perspective of the selected visualization type. In the two monitors view (Fig. 3, right) the selected perspectives are displayed next to each other (each in a selected visualization type) and are independently controlled (e.g. zoom, filter, select). In the mixed monitor view, the two selected perspectives are superimposed upon each other, whereby the portion of the space occupied by each and the transparency of the superposition can be interactively adjusted through movable sliders (Fig. 3, left). The visualization of both perspectives is manipulated with a single control (e.g. zooming in on one, is automatically followed in the other). For all visualizations, the data sets corresponding to a given perspective are colour coded: green for citizen data, blue for the municipal data set.

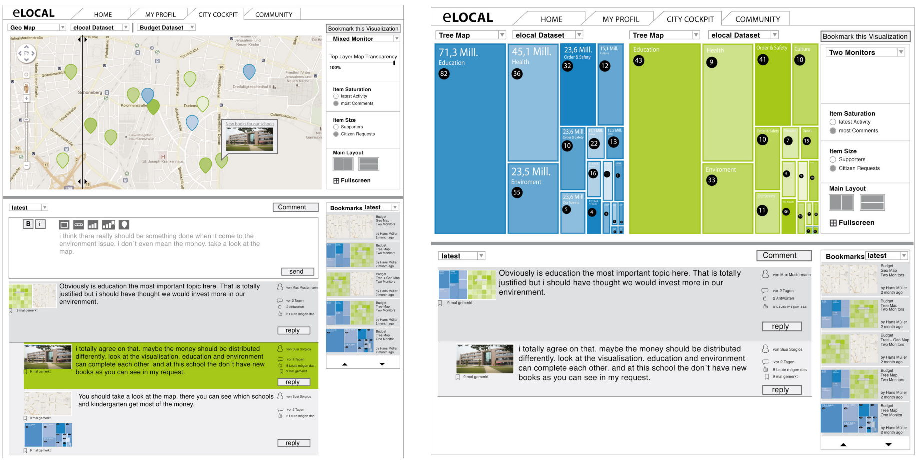


Figure 3. Multi-Perspective Visualization and Interface Design of the Open City Cockpit

By choosing a geographic visualization of the citizen data and of the municipal budget in a mixed monitor view, both the citizens and the officials can easily determine which budgetary actions are planned in their area (blue icons) and compare this to related citizen requests (green icons). Similarly, by selecting a tree map visualization of the municipal budget and of the citizen needs in a two monitor view it is easy to compare the relations between expenditures in the budget (blue squares) to relations between the number of citizen requests in related topics (green squares).

4.2 Interfacing visualization and discussion

The social analysis of the visualizations follows the principle of doubly-linked discussions proposed in recent work [4][11]. A currently displayed visualization state can be saved in form of a visual bookmark that can be recalled from the bookmarks panel at any given time (Fig 3., rightmost column in both interface designs). User comments are linked to the current state of the displayed visualization, unless the user manually overrides this and specifies it more precisely: by assigning a specific item (citizen request, budgetary action) to the comment or selecting an already saved visual bookmark as its point of reference. In this way, a more precise and natural referencing between comments and visualizations can be realized (e.g. a user may inspect several visualizations or different zoom levels before reaching a conclusion which may relate to one of the previous visualizations or a specific citizen request which spurred his interest in the first place). Selecting a comment then displays a pop-up with the corresponding visualization state and upon user confirmation switches the current view to this particular visualization. In case a single item has been referenced, then the corresponding visualization state highlighting this specific item is displayed. The same holds in reverse, selecting a visualization state displays a list of comments referencing it, while selecting an item highlights the referencing comments (Fig. 3, comments panel).

4.3 Implementation

We are currently working on implementing the main elements of the described visualization model and interface design in form of interactive prototypes that can be used for the next user evaluation cycle. For the implementation of the tree map visualization we are using an open source library from the javascript information visualization toolkit [17]. The data sets for the municipal budget at this stage are based on open data sources adhering to the JSON standard. The implementation of the application state bookmarking (visual bookmarks) will follow the model of URL-based state vectors proposed in [11]. In a further stage the data from the cooperating municipality will be integrated into the prototype.

5 Conclusions

The presented analysis and first experiences in designing a visual system for social data analysis in open government applications suggest that the heterogeneity of this setting requires considering and integrating specific requirements and design principles from collaboration theory and empirical practice. The first experiences in the development and formative evaluation of a possible solution based on multi-perspective visualization point to the need for further investigation of application-oriented design principles in this emerging field of research.

Acknowledgements

Stephan Breidenbach has initiated the E-Local project and contributed important ideas to the early concept of a participatory citizen platform. Martin Schneider implemented the basic version of the first prototype and Oliver Frueauff and Annekathrin Wilke provided valuable support in technical and organizational project coordination.

References

1. Boland J.R., Tenkasi R.V.: Perspective Making and Perspective Taking in Communities of Knowing, Organization Science, 6,4 (1995)
2. Card, S.K., Mackinlay, J.D., and Shneiderman, B. Readings in Information Visualization: Using Vision to Think. Morgan-Kaufmann (1999)
3. Fitzgerald, B., Wood, S.A., Social Data Analysis at Swivel: Lessons Learned & Next Steps, In Proc. ACM CHI 2008, Extended Abstracts and Applications, Florence (2008)
4. Heer, J., Viégas, F.B., Wattenberg, M.,Voyagers and Voyeurs: Supporting Asynchronous Collaborative Visualization, Communications of the ACM, Vol. 52, No. 1 (2009)
5. Novak, J., Helping Knowledge Cross Boundaries: Using Knowledge Visualization to Support Cross-Community Sensemaking, Proc. of HICSS-40, Hawaii International Conference on System Sciences, Hawaii (2007)
6. Russell, D. M., Stefik, M. J., Pirolli, P., and Card, S. K. The Cost Structure of Sensemaking. Proc. of ACM INTERCHI'93, 269-276 (1993)
7. Schmidt, K., Simone, C.: Coordination mechanisms: towards a conceptual foundation of CSCW systems design. Journal of Computer Supported Coop. Work, 5, 2-3 (1996)
8. Star, S.L.: The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving. In Readings in Distributed Artificial Intelligence, ed. M.Huhn, L.Gasser, Morgan Kaufman (1989)
9. Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J., & McKeon, M. Many Eyes: A Site for Visualization at Internet Scale. In Proc. IEEE Information Visualization (InfoVis) (2007)
10. Wattenberg, M. and Kriss, J. Designing for social data analysis. IEEE Transactions on Visualization and Computer Graphics 12, 4, 549–557 (2006)
11. Willett, W., Heer, J., Hellerstein, J.M., Agrawala, M.: CommentSpace: structured support for collaborative visual analysis. In CHI pp. 3131-3140 (2011)
12. http://www.silab.humboldt-viadrina.org/elocal
13. http://fixmystreet.com
14. http://www.frankfurt-gestalten.de/
15. http://www.openspending.org/
16. http://maerker.brandenburg.de
17. http://thejit.org/

Online Banking Customization via Tag-based Interaction

Rajinesh Ravendran, Ian MacColl, Michael Docherty

Queensland University of Technology (QUT)
2 George Street, Brisbane, QLD 4000
{r.ravendran, i.maccoll, m.docherty}@qut.edu.au

Abstract. In this paper, we describe ongoing work on online banking customization with a particular focus on interaction. The scope of the study is confined to the Australian banking context where the lack of customization is apparent. This paper puts forward the notion of using tags to facilitate personalized interactions in online banking. We argue that tags can afford simple and intuitive interactions unique to every individual in both online and mobile environments. Firstly, through a review of related literature, we frame our work in the customization domain. Secondly, we define our main idea and identify a range of taggable resources in online banking. Thirdly, we describe our preliminary prototype implementation with respect to interaction customization types. Lastly, we conclude with a discussion on future work.

Keywords: website customization, online banking, tags, interaction

1 Introduction

The workshop theme is centered on data-centric interaction on the Web. The ever-increasing amount of data available online has lead to a growing interest in exploring its potential use to enhance Web-based interaction. This paper discusses the use of user-contributed data in particular user tags in the online banking space with the aim of delivering a personalized interaction.

This paper is specifically aimed at customization in the online banking context because it is an imperative dimension of user satisfaction, particularly among the younger generation in Australia [1]. This is particularly true as customization allows online banking to be more responsive to the individual needs of each user. According to a survey conducted by Nielsen Australia[1], online banking is the preferred channel of banking over ATM, phone and branch. However, customization of online banking is still poorly addressed [1]. This paper aims to address this gap.

The rest of this paper is structured as follows. In section 2, the related work on customization is presented and background on the proposed technology is offered. In section 3, the proposed approach is outlined along with results of a background study to identify user taggable resources in online banking. In section 4, the preliminary prototype implementation is detailed with reference to interaction customization types. In section 5, the paper is concluded with discussion on future work.

[1] http://au.nielsen.com/news/20070426.shtml

2 Related work

2.1 Customization overview

Based on the literature, a customization framework has been derived, which offers an overview of the customization domain, and position of the work described in this paper (highlighted). For the purpose of literature review, a broad understanding of customization was adopted defined as "the ability for a website to be shaped in a way that better fulfils the wants of individual users" [1].

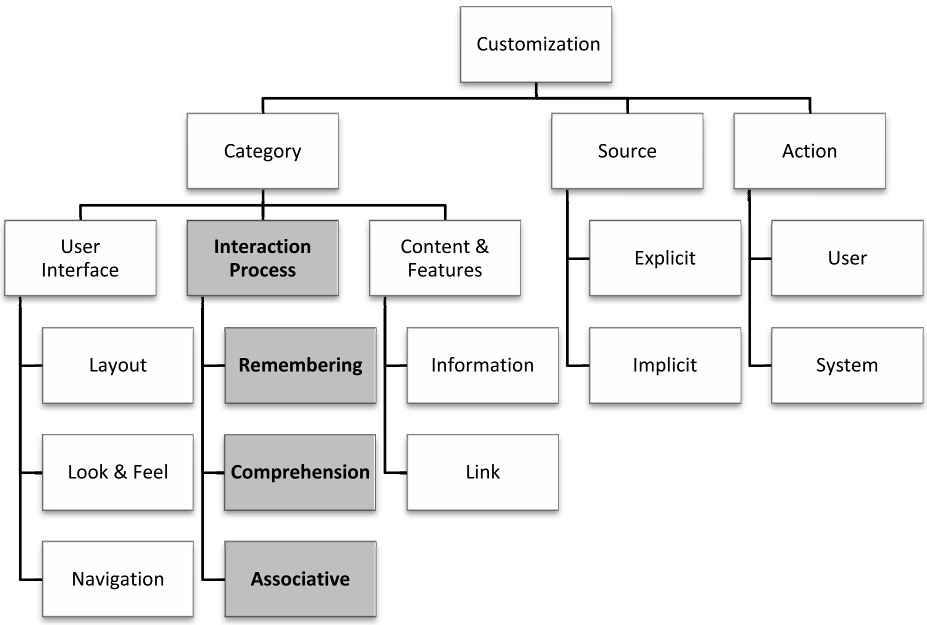


Fig. 1. Customization Framework

Figure 1 is derived from the literature based on previous work on performance personalization system [2], user interface customization [3], interaction process customization [4] and personalization framework [5].

The diagram illustrates three key dimensions of customization: category, source and action. Three *categories* of customization are *user interface*, *interaction process* and *content & features*. These categories can be further expanded into finer attributes or levels. *Sources* used to facilitate these different customization categories can be divided into two: *explicit* and *implicit*. The former includes data sources which are personal and overt such as cookies, user profiles and personal tools. The latter covers data sources that are complex which provide latent information about user's website usage or behavior such as usage logs or purchase history. The *action* of initiating and carrying out customization is the responsibility of the *user* or the *system*. Customization initiated and carried out by the user is generally known as static application of customization, while system initiated and executed customization is commonly described as dynamic application of customization. However, in some instances customization may be initiated by the user and carried out by the system or vice-versa. Arguably, these entities are inseparable and equally important hence their inclusion in some stage of the customization process is advantageous for an inclusive result.

The three types of customization outlined by Fung [4] as part of interaction process customization are used as a basis to explore potential customizations via tags. This customization category will be referred to as interaction customization hereon.

2.2 Tags / Tagging

Tags, also known as user-defined metadata are a popular Web 2.0 technology, enabling users to assign keywords to Web resources (e.g., photo, video, people, etc) primarily for the purpose of personal information management (PIM). Tags are largely personal and contextual [6], and considered as a potential source of knowledge [7]. Recognized as an easy-to-use, dynamic and engaging technology, tags aid users to recall and retrieve information content and when represented as tag clouds they facilitate visual information retrieval [8]. Also, the underlying meanings of tags may be discovered through semantic analysis to form associations between like-minded individuals [7].

In the financial services space, tags are widely used to assist personal financial management via third party tools such as Mint[2] and Yodlee[3], where a user can assign tags to annotate transactional data for purposes such as budgeting, expense tracking, etc. However, these tools, allow tags to be assigned to financial transactions at a high level as category or description, but not at a lower level for details such as bank account, biller, description, etc. There may be compelling advantages in doing so in the online banking environment, opening doors to tag-based interactions alongside personal financial management. Although the ability to tag financial data has existed for a while now primarily through third party tools, this trend is likely to change with banks considering the inclusion of personal financial management features. This view is particularly evident among Australian banks who have began to do so as part of their offerings with Australia and New Zealand Bank (ANZ) pioneering the initiative through its ANZ-MoneyManager[4] service. The inclusion of tags as part of online banking adds to the relevance and practicality of the proposed approach to customize online banking interaction.

[2] http://www.mint.com

[3] http://www.yodlee.com

[4] http://www.anz.com/ANZ-moneymanager/default.asp

3 Proposed approach

3.1 Main idea

The main idea of our approach is to use user-defined tags and tag-based visualization (tag cloud) to facilitate customization in online banking. Tags and tag clouds can offer a more intuitive and interactive online banking website. Since tags represent resources in a personal manner, tags can be used to facilitate comprehension of user intentions. Possible actions may be inferred from tags based on the resources they represent and the relationships between those resources. Also, relevant tags can be recommended to users and based on semantic relations of tags across the network, related services can be aggregated and recommended, which banking users may find useful.

3.2 Taggable resources

In order to define the range of taggable resources in online banking, a piece of information not readily available from the literature, a background case study was conducted. The study involved manual examination of personal banking websites of two leading banks in Australia: Commonwealth Bank[5] and Suncorp Bank[6]. Personal banking was preferred over other types of banking because it appeals to a wider base of users. Both online and mobile banking environments were considered in this study.

To identify taggable resources we observed both aforementioned online banking websites, focusing on services offered and the different information required for the services including fund transfer, bill payment, product application and internal messaging. The results of the study are presented in Table 1. A total of seven resources were identified, grouped into five categories: account, description, biller, application and message.

Table 1. Taggable resources

Resource Id	Category	Type / Description	Environment
R1	Account	Personal account which are user owned accounts such as every day, savings, cheque, credit card, etc	Online and Mobile
R2		Payee account which are either linked accounts such as personal accounts or other third party accounts that are internal, external or overseas	Online and Mobile
R3	Description	Personal description of a transaction as self-reference. Transaction types include offline such as EFTPOS, direct debit, etc; and online such as bill pay, fund transfer, shopping, etc	Online and Mobile

[5] http://www.commbank.com.au/
[6] http://www.suncorp.com.au/

R4		Payee description of transaction for recipient's reference	Online only
R5	Biller	All types of registered and unregistered billers	Online and Mobile
R6	Application	All types of financial products such as account, credit card, loans, etc	Online and Mobile
R7	Message	Personal communication between customer and bank	Online only

The findings of the study show that the resources identified are typically available in both environments which suggest that tag-based interaction is applicable to different contexts. However, it is noteworthy that mobile banking to a large extent only supports the use of pre-existing resources rather than new resources. For example, a fund transfer can only be carried out to a saved payee but not to a new payee in which case the payee information has to be added through the online environment. This is possibly due to the inconvenience of entering information through a mobile device or to security and privacy concerns related to the use of mobile banking.

4 Prototype implementation

An early prototype implementation with tag integration for a few key resources namely account, description and biller has been developed. The prototype is software-based, intended to demonstrate the mechanics of each customization particularly in the desktop environment for two main activities: bill payment and fund transfer. Although the prototype is still in its early days, it extends support to the proposed approach especially in terms of feasibility and practicality.

The following sub-sections elaborate the different types of interaction customization and the proposed use of tags to address them. The examples largely depict scenarios of day to day online banking activities.

4.1 Remembrance-based customization

This customization type is defined as the ability to provide customization through simple remembering of a user's information based on the recurrence rate of a particular action on a website [4].

Remembering-type customization can be fulfilled through tags assigned to resources that are presented as tag clouds. This provides a visual retrieval interface that can simplify and ease the execution of past or recurring transactions. Simply by clicking on a tag, related information about a transaction that the tag is associated with can be retrieved and displayed. If a selected tag is associated with two or more tags then the tag cloud can be filtered to show tags which are co-occurring with the selected tag. This removes the need to navigate to a different page or perform a manual search query. This also means for carrying out a past or recurring transaction, users will only need to update necessary information such as amount (if different) and possibly retain other details such as bank accounts and description.

Based on user's tagged resources namely transaction description, remembrance-based customization is introduced. The following example assumes a user pays a monthly mobile bill and tagged the transaction as "mobile" in the first month, and the following month the user returns to carry out the same activity.

Scenario 1: Mobile bill payment. User clicks on "mobile" tag (1) from the tag cloud. As a result, the bill payment form is automatically completed.

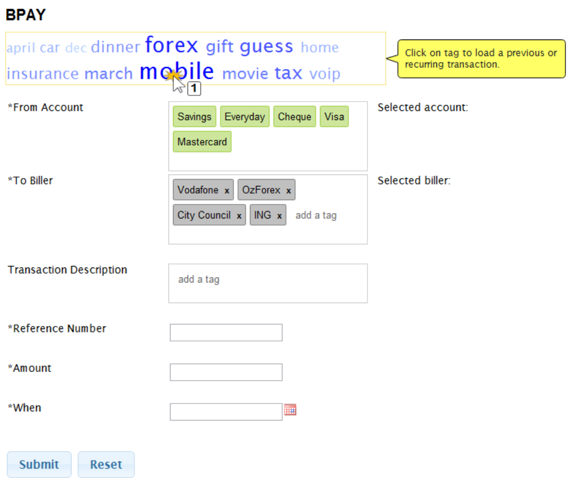


Fig. 2. Bill payment (initial form + user click (1))

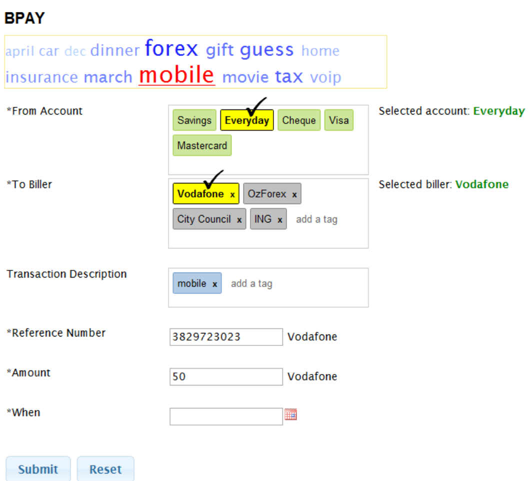


Fig. 3. Bill payment* (completed form)

*The selected tag ("*mobile*") from the tag cloud is highlighted (underline) and the transaction details are loaded and the account and biller tags ("*Everyday*" & "*Vodafone*") are automatically selected (tick icon).

4.2 Comprehension-based customization

This customization type is defined as the ability to recognize user's behaviors and provide assistance towards fulfilling the user's needs [4].

Comprehension-type customization can be fulfilled by inferring banking actions like fund transfer based only on tags selected by the user. Such inference is possible particularly for tags with certain types of relations such as two bank accounts. These relations when combined with simple pre-defined rules can aid in populating relevant actions. A sample pre-defined rule is the ability to transfer funds from savings account to credit card account but not the other way around due to nature of the accounts. However, it would be possible to transfer two-ways between a savings account and a current account. The default choice for the suggested actions can be made based on past user actions to closely reflect user's needs. However, as a key HCI design rule, it is important not to automatically carry out an action to ensure the locus of control remains with the user [9].

Based on user's tagged resources namely accounts and billers, the comprehension-based customization is detailed. This customization is realized through user selection of tags, where a set of actions are inferred by analyzing the relations between tags selected. This allows users to carry out their banking activities with minimal effort. This is achieved by examining the tripartite structure of tags comprising of user, tag and resource and subsequently, applying pre-defined rules to the underlying resources. Rules are defined for a particular resource owned by a user such as bank account based on account type. For example, a savings account would have three rules: transfer_from, transfer_to and view, which denotes that the account can be used to send or receive money, and be viewed. Meanwhile a credit card account (e.g., "Visa") would only have two rules: transfer_to and view, which denotes that the account can only receive money and be viewed. Similarly, a payee account (e.g., "Dad") would also have the same set of rules as the credit card account. The account with 'transfer_from' action will always act as the sender/primary account while the account with 'transfer_to' action will act as the recipient/secondary account. In the event tags of two accounts with 'transfer_from' action are selected, it would be possible to have identical actions of transferring and receiving funds between both accounts. In order to reduce the overall complexity involving multiple accounts, a limit for the number of selectable tags particularly for personal accounts is desirable (refer to Table 1 for account types).

The following examples show the ability of conducting an internal and external fund transfers just by selecting tags. Although the examples only illustrate fund transfers to a single account, it is possible to carry out fund transfers to multiple accounts at once.

Scenario 2: Internal fund transfer from Savings to Everyday account. User clicks on "Savings" (1) and "Everyday" (2) tags, a set of possible actions for these accounts are populated. The suggested actions are 1) 'Transfer from Savings to Everyday', 2)

'Transfer from Everyday to Savings' and 3) 'View transaction history of Savings and Everyday'.

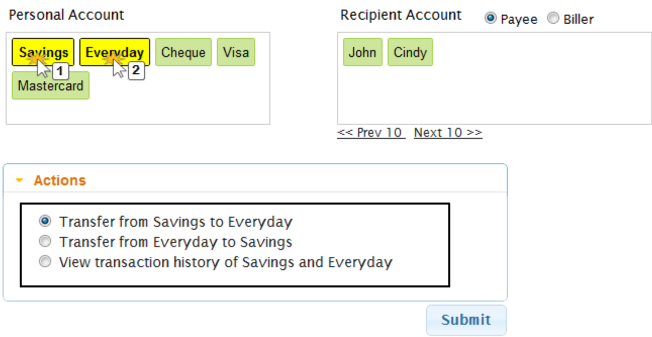


Fig. 4. Internal fund transfer

Scenario 3: External fund transfer from Everyday to John's account. User clicks on "Everyday" (1) and "John" (2) tags, a set of possible actions for these accounts are populated. The suggested actions are 1) 'Transfer from Everyday to John' and 2) 'View transaction history of Everyday and John'

Fig.5. External fund transfer

4.3 Association-based customization

This customization type is defined as the ability to provide customization through association of user's behavior with other individuals who share similar interests or needs[4].

Associative-type customization can be fulfilled through tag recommendation to users primarily as suggestions in the form of dropdown box or visually through the use of tag clouds to display related tags. The semantics of tags can be used to find closer association between tags across the network and to select/rank the most relevant sets of tags based on similarity score [7]. Based on the derived tags, information about related services may also be aggregated. This information can possibly aid the discovery of services which otherwise may not been known to users. This is particularly appropriate in light of an integrated online banking bill payment service such as BPay[7] in Australia, participated by merchants and service providers throughout the country. However, aggregation of services based on tag-relatedness is more likely to be useful for services that can be easily abandoned.

Based on user's tagged resources namely biller and description, association-based customization is explicated. The associations are divided into tag suggestions and tag-based service aggregation. Both these associations are based on two sets of tags: personal and public (cross-network). For tag suggestions, personal tags take precedence over public tags, and the most relevant public tags can be suggested based on number of associations to a resource. To improve the relevance of tags suggested to an individual, analysis of semantic relatedness is useful [7]. Additionally, by analyzing the semantics relatedness of tags, similar services may be discovered and aggregated. This is based on the notion that tags with high semantic relatedness are likely to represent a similar type of biller. In order to ascertain the similarity of discovered billers, further validation can be carried out based on attributes such as industry type, nature of business, etc. For this purpose, semantic databases like Freebase[8] and local business directories can be used. Although semantic analysis of tags can aid in personalizing tag recommendation [7], it is subject to good levels of semantics in tag sets. The outcome of semantic analysis can be undermined by the presence of idiosyncratic tags that carry strong personal connotations. One possible solution is tag reuse through tag suggestions [10], which can potentially reduce the use of idiosyncratic tags over time. Also, the utility of tag-based service aggregation may entail reciprocal action from users to tag with a reduced personal sense.

The following examples illustrate this customization by assuming a user carries out two select bill payments to two billers: Vodafone (mobile) and OzForex (foreign currency exchange).

Scenario 4: Tag recommendation for multiple bill payment (mobile and money transfer). User clicks on "Vodafone" (1) and then "OzForex" (2) biller tags, and clicks to enter a description tag (3). As a result, a set of related tags are recommended that are used in the context of the selected billers.

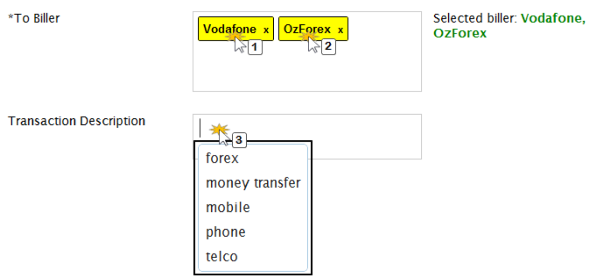
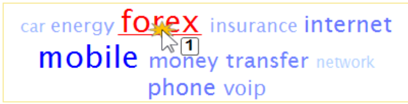


Fig. 6. Tag recommendation (suggestion)

[7] http://www.bpay.com.au/
[8] http://www.freebase.com/

Scenario 5: Tag-based services recommendation. User clicks on "forex" (1) tag, related services are populated with aggregated information on service usage.



Service Name	Website	Total Users	Average per month
OzForex	http://www.ozforex.com.au	50,000	$ 1500.00
Western Union	http://www.westernunion.com.au	30,000	$ 1200.00
Travelex	http://www.travelex.com.au	10,000	$ 1100.00
Monegram	http://www.moneygram.com.au	5,000	$ 1000.00

Fig. 7. Tag recommendation (service aggregation)

Figure 7 shows a table of foreign currency exchange services with aggregated information such as total users and average per month. Such information can allow users to discover related services with the aggregated details serving as a practical rating for services. In the above context, the service with the highest users and average may be perceived to offer a more competitive exchange rate than the rest. Even in cases where the aggregated information is not very useful, users may still benefit solely from discovery of services.

5 Conclusion & Future Work

This paper firstly provides a conceptual understanding of interaction-focused customization in the larger customization domain. Secondly, it defines the range of user taggable resources in both online and mobile banking environments. Thirdly, a practical view of the proposed tag-based interaction is offered via a preliminary prototype implementation. The main point of interest of the research lies in the potential use of tags to facilitate personalized banking interactions. The preliminary prototype highlights this specifically in the desktop environment. The ability to employ a light-weight user-driven technology such as tags to facilitate various interaction customization types is advantageous and useful, and may be achieved with minimal effort for implementation and adoption. However, a drawback to using tags is the lack of detailed information about resources which they represent that may be needed to ensure correct use of online banking. Even though the ability to recall information as a form of self-communication is significant with tags [11], users may still want to view detailed information about a resource. One possible solution is to provide detailed information about tagged resources on demand in the form of dynamic tooltip as a simple hover effect on desktop and tap and hold action on mobile devices. The use of dynamic tooltips also allows the most up-to-date information such as balance amount to be shown on the fly, which is important for carrying out a transaction.

From here, we envisage further exploration of tags and tag clouds in the mobile environment, particularly addressing the spatial constraint present. This can highlight other form of interaction techniques which may help to better achieve the different customization types. Also, we will focus on evaluating the prototype with the aim of assessing its utility and usability in both online and mobile environments as they are key factors of user performance, satisfaction and acceptance [12]. This will allow us to explore ways in which the prototype can be improved. The proposed method of evaluation is experimentation using a pretest-posttest control group design. This design would provide strong internal validity, suitable to measure cause-effect relationships [13]. Each customization type will be evaluated separately in a similar fashion to Fung's study [4] and experiential feedback from participants will be gathered using a posttest questionnaire. The feedback will be used to improve the prototype incrementally for the upcoming experiments. The results of the experiments are expected to indicate the suitability of the customizations in the online banking context and also inform a set of guidelines for the design and implementation of tag-based customizations.

6 Acknowledgement

This research work is sponsored by the Smart Services Cooperative Research Centre (CRC) of Australia (http://www.smartservicescrc.com.au) as part of the financial services project. We would like to thank our fellow colleague Ms Claudia Murillo for her feedback and also the reviewers of this paper for all their comments and suggestions.

7 References

1. Rahim, M.M., JieYing, L.: An empirical assessment of customer satisfaction with Internet Banking applications: An Australian experience. 12th International Conference on Computers and Information Technology (ICCIT), pp. 314-320 (2009)
2. Riemer, K., Totz, C.: The many faces of personalization–An integrative economic overview of mass customization and personalization. The customer centric enterprises: Advances in mass customization and personalization, New York, Berlin 35-30 (2003)
3. Bunt, A., Conati, C., McGrenere, J.: Supporting interface customization using a mixed-initiative approach. 12th International Conference on Intelligent User Interfaces (IUI), pp. 92-101. ACM (2007)
4. Fung, T.: Banking with a personalized touch: Examining the impact of website customization on commitment. Electronic Commerce Research 9, 296-309 (2008)
5. Wu, D., Im, I., Tremaine, M., Instone, K., Turoff, M.: A framework for classifying personalization scheme used on e-commerce websites. 36th International Conference on System Sciences, pp. 12. IEEE, Hawaii (2003)

6. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, tagging paper, taxonomy, Flickr, academic article, to read. 17th Conference on Hypertext and Hypermedia, pp. 31-40. ACM, Odense, Denmark (2006)
7. Durao, F., Dolog, P.: A personalized tag-based recommendation in social web systems. International Workshop on Adaptation and Personalization for Web 2.0, pp. 40. Citeseer (2009)
8. Hassan-Montero, Y., Herrero-Solana, V.: Improving tag-clouds as visual information retrieval interfaces. International Conference on Multidisciplinary Information Sciences and Technologies, InSciT2006, pp. 25-28. Citeseer, Mérida, Spain (2006)
9. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Pearson Addison Wesley (2004)
10. Sood, S., Owsley, S., Hammond, K., Birnbaum, L.: Tagassist: Automatic tag suggestion for blog posts. In: International Conference on Weblogs and Social Media. Citeseer, (Year)
11. Panke, S., Gaiser, B.: With my head up in the clouds: using social tagging to organize knowledge. Journal of Business and Technical Communication 23, 318-349 (2009)
12. Höök, K.: Evaluating the utility and usability of an adaptive hypermedia system. Knowledge-Based Systems 10, 311-319 (1998)
13. Marathe, S., Sundar, S.S.: What drives customization?: control or identity? 28th International Conference on Human Factors in Computing Systems (CHI'11), pp. 781-790. ACM, Vancouver, BC, Canada (2011)

tFacet: Hierarchical Faceted Exploration of Semantic Data Using Well-Known Interaction Concepts

Sören Brunk[1] and Philipp Heim[2]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway
soren.brunk@deri.org
[2] Institute for Visualization and Interactive Systems (VIS),
University of Stuttgart, Germany
philipp.heim@vis.uni-stuttgart.de

Abstract. Information stored in the Semantic Web is becoming more and more interesting for average Web users. Due to the complexity of existing tools, however, accessing it is difficult. In this paper we therefore introduce tFacet, a tool that uses well-known interaction concepts to enable hierarchical faceted exploration of semantic data for non-experts. The aim is to facilitate the formulation of semantically unambiguous queries to allow a faster and more precise access to information in the Semantic Web for the broader public.

Keywords: Faceted exploration, faceted search, Semantic Web, known interaction concepts, hierarchical facets.

1 Introduction

The amount of information available as semantic data is growing rapidly. As of June 2011, the *Linking Open Data (LOD)* cloud [1], for instance, contained more than 200 different datasets. The most popular dataset within the LOD cloud is the *DBpedia* dataset [2]. It contains structured information that is extracted from Wikipedia articles and converted into a semantic representation. It is publicly accessible via the *SPARQL* [3] query language, allowing for semantically unambiguous access. In comparison to the often ambiguous text search, it allows the formulation of more complicated requests accurately and thus supports targeted and quick access to information.

A disadvantage of using SPARQL is, however, that users have to learn the language first in order to access information through SPARQL queries; making it rather a language for experts. In order to meet the needs of all users, a simpler user interface is required to create semantically unambiguous and complex queries without the need to learn the complex syntax of a query language. One approach to solve this problem is based on the concept of *faceted exploration* [4]. In faceted exploration, the user always sees all remaining search options within the search process and can select them step-by-step in order to refine the query.

Several applications exist that implement the concept of faceted exploration in order to allow users to access semantic data. Examples are *mSpace* [5], *Parallax* [6], *Faceted Wikipedia Search* [7] or *gFacet* [8]. Most existing applications, however, don't use the full potential of faceted exploration in combination with semantic data and thus lack the ability to create powerful queries. Others allow the creation of more complex queries but are often difficult to use. mSpace and Faceted Wikipedia Search, for example, are easy to use, but they don't support the creation of hierarchical facets; that is, facets that are connected to the results through indirect attributes. In contrast, Parallax and gFacet do support hierarchical facets. However, the creation of those can be difficult for inexperienced users due to the use of new and unfamiliar interaction concepts within these tools.

For that reason we introduce tFacet, a tool that uses well-known interaction concepts in order to make the power of faceted exploration available also for inexperienced users.

2 tFacet

tFacet, like the other tools, uses the concept of faceted exploration to access semantic data. However, it thereby applies interaction concepts that are well-known from other applications and thus allows the widespread use of already existing knowledge in the formulation of semantically unique queries. It is implemented using the *Adobe Flex Framework* [9] and therefore runs in any Web Browser with the Flash plugin. Using SPARQL as the query language allows for faceted exploration of any RDF dataset accessible via a SPARQL endpoint. By default tFacet uses the DBpedia dataset but can be easily configured to access other datasets as well.

2.1 Initial Search Space Limitation

Each exploration within tFacet starts with an initial limitation of the search space. This step is necessary to reduce the number of possible results as well as the number of facets to a displayable amount. The user selects a base result class from a tree representation of all classes contained in the dataset. The better the user knows what he is looking for, the more precisely he can restrict the search space. For example in the DBpedia Ontology he could select "Eurovision song contest entry" as a very specific base class limiting the search space to 1054 objects. On the other hand, if he is not able to specify his search goal in detail in advance, he could choose a more abstract base class such as "film"; containing 53619 objects (see Figure 1).

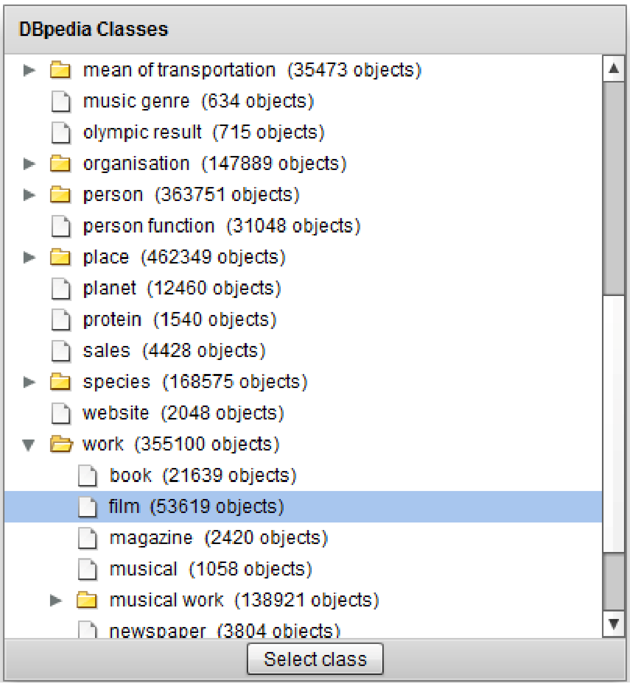


Fig. 1. Selection of base class "film" from the DBpedia ontology.

2.2 Hierarchical Faceted Exploration

By pre-selecting a base class, the exploration is limited to objects of that class. Thus in Figure 2 only objects of the class "film" are displayed in the result set (A). On the left side a directory tree representation of facets is shown. It displays all facets that can be used to filter objects in the result set. At first, only the top hierarchy of the tree is visible, containing all facets related to direct attributes of the result set. In case of films, for instance, it contains facets such as directors or actors of the corresponding movies. In addition, some elements of the tree can be expanded in order to show lower hierarchy levels with additional facets. Those facets refer to indirect attributes of the result set. For movies these could be, for example, the birthplace of the director of the movie.

The user now can select individual facets from the tree to show the facet's detail view in the right area (Figure 2, C). Facets in the detail view are arranged vertically and always contain all remaining search options as selectable attributes to explore or filter the result set. By selecting individual attribute values, the result set is reduced to objects having that value. For instance, by selecting a director, the result set is filtered to show only movies from that director. Attributes in different facets are combined using the AND-operator while attributes within one facet are combined using the OR-operator. In this manner, the result set can be refined iteratively until the information wanted has been found.

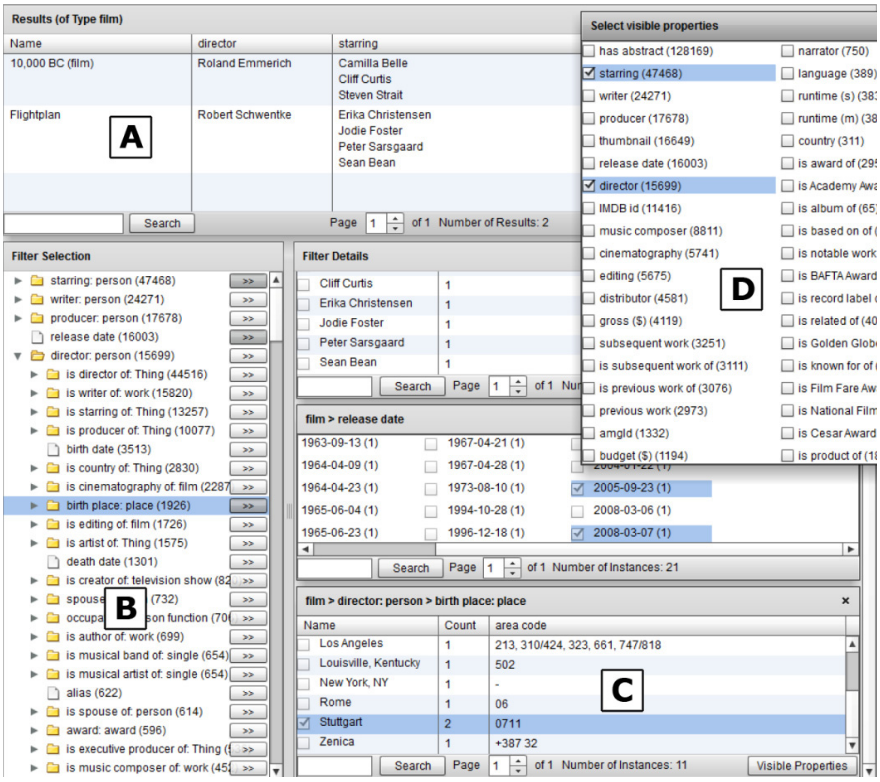


Fig. 2. The main view of tFacet is divided into three parts: Result set (A), facet tree (B) and facet details (C). Columns can be added interactively (D).

In addition to the result set, a filter operation also updates all other facets that are shown in the detail view. All attributes that would lead to an empty result set when selected are hidden. Furthermore, the expected number of results when selecting a certain attribute is shown in brackets besides this attribute. This feature helps to avoid dead-ends and allows a user to estimate the outcome of a specific filter operation.

All facets available in the directory tree can be extracted automatically from semantic data by using SPARQL queries. All properties of the selected base class are collected and displayed in the tree. If a property leads to a literal this is represented by a document symbol in the tree and cannot be further explored (see “release date” in Figure 2, B). If a property leads to objects of another ontological class it is represented by a directory symbol and can be further explored (see “director: Person” in Figure 2, B). The name of such a sub-directory is composed of two components: (1) the type of the property (here “director”) and (2) the class of the objects (here “Person”). Representing object properties as sub-trees allows also deeply nested information to be used for faceted exploration. In this way, the birthplaces of the directors of the movies in the result set can be used as hierarchical facet to show, for example, only movies that were directed by directors born in Stuttgart (Figure 2, C).

2.3 Additional Columns

Until now we have used the linked structure of the Semantic Web only for faceted exploration. It is also possible to use this structure to enhance the presentation of the result set by additional details. To demonstrate this, tFacet implements a functionality that allows users to add additional columns to both the result set as well as the facets and also to use those columns for sorting.

In order to add an additional column, the user can click the button "Visible Properties" to see all properties and choose the ones that should be visible in the list (Fig. 2, D). For example, in this way it is possible to use information about the actors of a movie ("Starring") both to filter and sort the result set. Furthermore, having the properties available as additional columns allows the relations between information to be directly visible for the user (e.g. the relations between movies and actors) in contrast to the separated representation via facets.

But often there is more than one value for a property (e.g. usually more than one actor plays in a movie). In that case, the current implementation of tFacet uses a simple list to display multiple values. This can cause problems however, if many values exist for a property, for example, if a movie has many actors. To avoid this problem an abbreviated list could be shown initially or just the number of entries, with the possibility to expand the view if necessary. With numerical data it would also be possible to show an aggregated view.

In general, the idea of additional columns is not only limited to direct properties, but can be implemented also for indirectly related properties. For that, one could imagine displaying a directory tree like in Fig. 2 D to select information only connected indirectly. However, one problem with such an implementation would be how to maintain the clarity of the presentation as many trees and their hierarchical directory structures could confuse the user more than help him create search requests.

2.4 Well-known interaction concepts

tFacet uses several interaction concepts that are known from common applications in order to ease the use of faceted exploration. The most important ones are:

- **Directory tree**: In many applications, directory trees are used for the navigation in and the management of hierarchical data. A logical conclusion was therefore the representation of hierarchical facets in a directory tree within tFacet. Like in popular file managers, nodes shown as folder symbol can be explored further while nodes shown as file symbol represent a leaf node.
- **Subdivision of the user interface into three parts**: Also, a partitioning of the user interface into an overview (directory tree), a detailed view (right pane) and a result view is frequently used. The possibility to show multiple detail views (in this case facets) at the same time is not widespread, for the definition of filters in more than one facet, however, necessary.
- **Organization into columns:** In many grid-based applications (for example, in Windows or Mac OS applications) the user can determine individually, which columns are of interest for him and have these displayed. In a similar way in tFacet

he can display a menu of all properties he might be interested in and add or remove individual columns. The same is true for sorting by individual columns by clicking on the column header.

3 Summary and Future Work

In this paper, we proposed tFacet, a new application to enable all users to create semantically unambiguous search requests using the concept of faceted exploration. The aim was to enable the full potential of this concept by keeping its usage as simple as possible through the use of well-known interaction concepts. Through the use of a directory tree even remote information can be included in the query as hierarchical facets and the possibility to show connected properties in additional columns facilitates control over the level of detail of the information displayed.

Thus, tFacet offers interesting approaches to make the potential of semantic data accessible for everyone. A key part of this is the strategy to transfer already known concepts into the specific interaction environment of the Semantic Web.

In the future we plan further development of tFacet to make its usage even simpler. For example, data-specific visualizations, such as sliders or maps could be used to ease the creation of queries and improve the presentation of results. An integration of the approaches discussed in Section 2.3 to extend the columns functionality to use hierarchical properties could be useful for a consistent and powerful implementation of this concept. Furthermore, an evaluation of the user interface would be helpful, especially to see if the use of known interaction concepts can help users to reach their goal faster. Including a comparison to similar tasks and tools in order to measure empirically, whether the intended goal of tFacet, a simplified usage, has been achieved.

References

1. LOD cloud (2010): http://richard.cyganiak.de/2007/10/lod/lod-datasets_2010-09-22.html
2. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Proc. of the 6th ISWC 2008, 2008, pp. 722-735.
3. SPARQL (2008): http://www.w3.org/TR/rdf-sparql-query/
4. Yee, K.-P.; Swearingen, K.; Li, K.; Hearst, M.: Faceted metadata for image search and browsing. In: Proc. of the CHI 2003, ACM Press, 2003; pp. 401-408.
5. Schraefel, m.c.; Smith, D.; Owens, A.; Russell, A.; Harris, C.; Wilson, M.: The evolving mSpace platform: leveraging the semantic web on the trail of the memex. In: Proc. of Hypertext 2005, ACM Press, 2005; pp. 174-183.
6. Huynh, D. and Karger, D.: Parallax and companion: Set-based browsing for the Data Web (2009). http://davidhuynh.net/media/papers/2009/www2009-parallax.pdf
7. Hahn, R.; Bizer, C.; Sahnwaldt, C.; Herta, C.; Robinson, S.; Bürgle, M.; Düwiger, H.; Scheel, U.: Faceted Wikipedia Search. In: Proc. of BIS 2010, pp. 1-11.
8. Heim, P.; Ertl, T.; Ziegler, J.: Facet Graphs: Complex Semantic Querying Made Easy. In: Proc. of the 7th Extended Semantic Web Conference (ESWC2010), Part I, Springer, 2010, pp. 288-302.
9. Adobe Flex (2011): http://www.adobe.com/de/products/flex/

More!: Mobile Interaction with Linked Data

Gonzalo Parra[1], Joris Klerkx[1], and Erik Duval[1]

[1] Computer Science Department, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, B-3001 Leuven, Belgium
{Gonzalo.Parra, Joris.Klerkx, Erik.Duval}@cs.kuleuven.be

Abstract. Science2.0 can radically change how researchers connect and collaborate. In this paper, we focus on researchers who attend academic events, such as workshops or conferences. We have developed a mobile social discovery tool, called More!, that presents academic and Web2.0 information sources and enables researchers to explore information about speakers and their work. More! also supports follow-up of future work. The application is powered by information sources that expose Linked Data through a RESTful API. More! has been extensively evaluated, showing promising results.

Keywords: Social discovery, Science2.0, Research2.0, Web2.0, mobile applications, mobile devices, Linked Data

1 Introduction

Science2.0 is the result of applying Web2.0 tools and approaches to regular research processes in order to increase participation and collaboration [1]. Our Science2.0 work focuses on openness and sharing, mashing up data and services, and using Web2.0 tools for communication. Although efforts to encourage research collaboration are quite dispersed and many challenges remain, first experiences suggest that Science2.0 is considerably more productive than the traditional way of doing research [2]. One of the key goals behind the Science2.0 concept is to support the connection of researchers in order to nurture fruitful cooperation. To this end, research support systems are beginning to apply social networking approaches [2][3] by supporting discovery, connection, sharing and discussion among researchers.

In this paper, we focus on the scenario where researchers are presenting their work while attendees in a conference or seminar may be interested to find more information about the topic and the speaker. The attendee may want to (i) find information about previous and current work of the presenter, (ii) stay up to date on new work from the presenter and (iii) share the work of the speaker with colleagues, for instance the members of a research group, who may or may not be attending the same event. In more conventional settings, the attendee may use a search engine, talk with the presenter afterwards, consult the proceedings, etc. Our paper describes how the use of mobile technology can improve this process.

Generally speaking, our work deals with awareness for researchers of relevant Web2.0 information sources and communication channels, such as social networks, micro-blogs, blogs, etc. “More!” is our mobile application that groups relevant information about a speaker and presents it in a way that can be easily exposed and integrated in the normal workflow of an academic event. This work is becoming more relevant as, over the last years, Web2.0 tools are being increasingly used to support every phase of the research lifecycle [4][5]. Furthermore, our tool is both relevant for junior researchers (who may not recognize the person on stage or the name in the conference schedule), and for established researchers who move into a new domain [6][7].

The structure of this paper is as follows: in section 2, we present related tools and approaches to support face-to-face event enhancements, enriched profiles, and social discovery. Section 3 describes the inspiration and design of More!. Section 4 explains the back-end infrastructure that supports the application. We conducted a user evaluation to gather feedback regarding the usability and functionality of the tool. The results of this evaluation are presented and discussed in section 5. Finally, we conclude and present challenges for further work in section 6.

2 Related Work

The main objective of More! is to deliver instant information from different sources to a researcher attending a face-to-face event, such as a conference, a seminar or a workshop. While there are related tools that focus on enhancing research events, or providing rich user profiles or social discovery capabilities, none of them cover the full scope of our application. Our work focuses on all three areas in order to enable social discovery through rich user profiles at scientific events.

Face-to-Face Event Enhancements. There has been some work on enhancing the experience of researchers attending face-to-face events. Conference planning and navigation tools like the IBM Event Maps [8] or the Conference Navigator [9] focus on helping conference attendees to browse and organize their conference schedule. Moreover, the Conference Navigator supports community-based personalization and recommendations [9]. Like these desktop tools, mobile applications like Conference Guide [10], Conference Compass [11] and CHI 2011 Mobile Application [12] give attendees of the event the possibility to explore and manipulate their personal conference schedule on-site.

These applications provide an overview of a specific conference based on a pre-defined schedule and help the user to plan their attendance at such events beforehand. In contrast, our application focuses on helping users to get more information about the speaker and the presentation topic while at the event, through links to academic and Web2.0 information sources. Thus, we focus more on discovery and exploration than on planning or navigation. In addition, More! addresses research activities beyond the presentation at hand, such as becoming aware of additional or later work of the presenter, or informing colleagues about it, etc.

Enriched Profiles. Tools like Gist [13] and Rapportive [14] provide enriched contact profiles on mobile and desktop clients. The idea behind these tools is to make available, when needed, extra information about a person through a mobile application or a browser extension, so that it can be integrated in for instance web based email applications. It is important to mention that enriched profiles provide information about people that the user already knows or had contact with before, but they do not provide information about a person that is not part of the current social network of users. In contrast, More! offers this information to researchers who are interested in the work, ideas or resources of a person that they possibly do not know yet.

Social Discovery. Location-aware devices are used to connect geographically close people. Mobile applications like Shhmooze [15], Banjo [16] and Sonar [17] focus on helping users to discover interesting people based on proximity, social network links and profiles. While these applications could be used at a specific research event, they are not tied to the purpose of the event. More! aims to be nicely integrated in the workflow of a face-to-face research event, providing extra information about the current speaker.

JumpScan is an application that mimics the functionality of More!, but that does not focus on the Science2.0 context [18]. The main difference with our application is that JumpScan does not support sharing of profiles and does not include any academic information sources - two characteristics that are crucial in the context of More!. However, we expect that the findings from our evaluations and experience will also be valid, and probably useful, for JumpScan.

3 The Application: More!

In order to create not only a useful, but also a usable application, we developed More! following a rapid prototyping approach [19] with frequent user feedback cycles. The application is inspired by the mobile music discovery service Shazam [20], which enables users to identify a song by recording a small fragment of the music. Through a fingerprinting technique, the song is identified and different kinds of information about it are retrieved; such as artist, title, album, and a YouTube and iTunes link. Shazam enables the user to share their discovery with other users through various channels, such as Twitter, Facebook and e-mail.

In the context of a research event, we want to make the discovery and sharing process as smooth as Shazam does for music. The core elements we target with our application are: (i) a fingerprint that enables a frictionless exploration process, (ii) automatic linking to different information sources, and (iii) the ability to share the discovery. These core elements found in the music discovery application guide our design. A more detailed explanation of the design process is presented in [21].

In the current prototype, we use Quick Response (QR) codes as the fingerprint to identify researchers. These are matrix barcodes that can encode any kind of data, such as numeric, alphanumeric and binary characters [22]. Currently, there are several mobile applications available that resolve these codes and return the encoded value to the application to take further action. For our fingerprinting purposes, we encode the URL of the researcher's More! page.

The user interface of our application includes four clusters of information: general, academic, social network identities, and communication & sharing (see Figure 1). These clusters expose the following information:

- general information: full name, photo, e-mail and affiliation;
- academic information: paper and slides being presented, and publications list;
- social networks identities from Twitter, SlideShare, blog, Delicious, LinkedIn, and Facebook.

With these information sources, the attendee can explore the research paper and slides of the current presentation, and the publications list of the speaker. Moreover, participants can 'identify' and 'follow' the speaker on some of the mainstream Web2.0 social tools. As a result, the attendee will get access to previous, current and future work of the speaker.

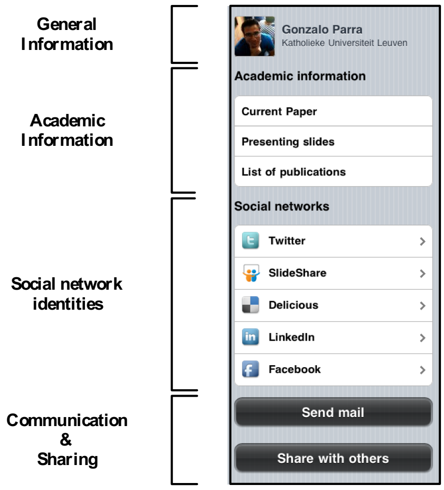


Fig. 1. The More! application.

More! enables the attendee to establish direct contact with the researcher via e-mail or social networks. Moreover, the tool allows sharing the discovery with other possibly interested researchers that attendees have more direct relations with (for instance members of their research group).

The workflow of the application in a conference scenario is as follows:

1. The speaker or conference organizer exposes a QR code (resolvable to an URL link) to the audience. We have experimented with different means to do so, for instance by exposing the code on the presentation slides, participant tags and including it in the event schedule.
2. Conference participants capture and decode the QR code, using any code reader application available on their handheld device (such as ScanLife [23]). After decoding, they are automatically redirected to the More! web application. As an alternative, the attendee can also use a URL and a regular web browser to load the application directly.
3. More! presents the data on the client tool.

4 Back-end Service: Research.fm

The More! application depends on the availability of information about the research and Web2.0 presence of the speaker. This section presents our work on how to model, obtain, store, manage and share these data.

4.1 Model

We follow a domain driven design methodology [24] where a single model, the Semantic Web for Research Communities ontology (SWRC) [25] is used to enable manipulation and linking of resources. In order to capture all relevant data, we extended SWRC with the FOAF [26], SIOC [27] and vCard [28] ontologies.

The SWRC model has a representation of the most relevant research entities, such as Person, Publication, Organization, and their relationships. The main concepts we use from the ontology are presented in Figure 2. The Publication entity has one or more authors, with zero or more online accounts; and these are affiliated to an organization.

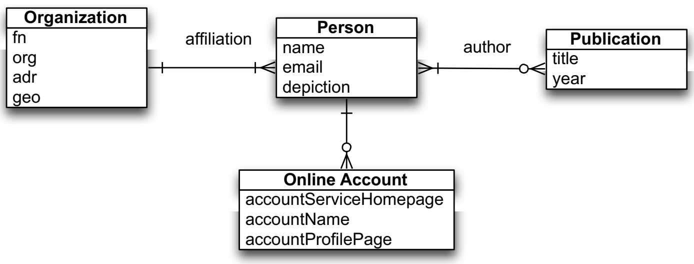


Fig. 2. Main Concepts of the SWRC Ontology.

The FOAF and SIOC ontologies have been used to extend the description of the Person entity in order to be able to capture online accounts. For example, by making use of the foaf:OnlineAccount class together with the foaf:hasOnlineAccount property, we can model the different web identities (Online Account) of a swrc:Person, including its homepage and profile page of the user. Furthermore, the vCard ontology is used to extend the Organization entity, for example, for giving a better structure to addresses.

4.2 API

Our goal is to enable open access to large amounts of structured data on research, with our current focus on publications and authors. These data can power a variety of tools that can help researchers to better understand their community [7]. We want to reduce access barriers, provide multiple communication options and expose data for easy integration [29]. For these reasons, we deployed a back-end infrastructure that stores the information about researchers and exposes it via a RESTful API that we refer to as "Research.fm", in order to connect and leverage the Linked Open Data technology [30]. This infrastructure is ongoing work in the context of the European FP7 STELLAR Network-of-excellence that aims to provide access to social network presence and publication data of researchers in a standardized way [29] [31].

The Research.fm API implements the Cool URI [32] principle to provide readable, unambiguous, and persistent URIs for resources. The SWRC domain model entities are the core elements exposed by the different methods of the API. This API allows access to these data for a variety of Science2.0 applications. Besides for More!, the API is also used by other research exploration applications like Muse [33] and Science Table [34].

4.3 Architecture

The architecture that supports the data sharing approach is presented in Figure 3. Different sources, like publication archives, institutions, and social media repositories (eg. user directories, such as Soharc [35]), feed a central repository that exposes the previously described model through the Research.fm API. These information sources expose their data through the Really Simple Syndication (RSS) format and/or the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [36], which allows automating the process. Both approaches expose XML representations of SWRC and FOAF ontologies to the central repository.

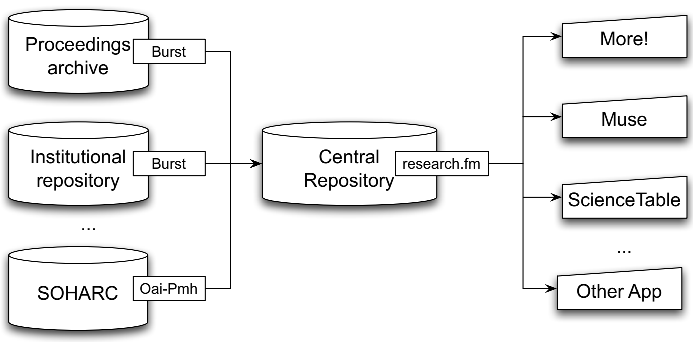


Fig. 3. Back-end architecture.

The central repository implementation is based on technology that we developed to store and manage learning objects and their metadata [37]. This flexible technology allows the consumption and management of different metadata schemas. The key feature of this technology is the ability to consume, store and expose any kind of XML document, which allowed us to easily develop the Research.fm API on top of it. Our repository technology has been evaluated on common software quality attributes, such as performance, reliability, interoperability, configurability and scalability [37].

The development of the back-end is ongoing work [29][31] that focuses on the adoption of a domain driven development methodology.

5 Evaluation

More! aims to increase awareness about different information sources, in order to help researchers to be more knowledgeable about related work. In order to assess the acceptance of the tool in real-life situations (i.e. at research events), we designed an evaluation in two steps.

Initially, we carried out two studies where the usability and functionality of the tool were tested in different contexts: a fictional scenario and a small real-life scenario. This approach provided us with feedback regarding the satisfaction level of the users and highlighted problems with the implementation. We used real data in order to reduce the artificial nature of the tests and increase the validity of the results from these initial evaluations [38]. The participants clearly agreed that More! is simple and easy to use but noted some concerns regarding the functionality, specifically related to the possibility of following the speaker's presentation via More!. Based on this feedback, the final version did not include extra functionality but re-arranged the presentation of the information for a final evaluation. This initial evaluation step is extensively described and discussed in [21].

In a second step, the application was tested in a research event, where researchers provided valuable feedback on the usability, functionality, and usefulness of the tool. This evaluation allowed us to obtain better insight in social interactions between researchers and how More! can enhance these. For this purpose, the tool was presented and promoted to all attendees of the EC-TEL 2010 conference [39] via mail, Twitter and during the opening session. From the previous evaluations, we observed that users had difficulties capturing QR codes during a presentation. In order to address this issue, the participants received QR codes for the different presentations on a leaflet added to the program of the conference. We also included a short explanation of what these codes were and how to use them. The participants were able to scan the QR code of a particular presentation and obtain the More! page of the presenter. Finally, evaluation data was gathered using 2 methods: a survey and usage tracking of the tool from all attendees.

For the survey, 10 users (8 male, 2 female) participated in the evaluation. The participants were between 28 and 50 years old, with the majority (n = 8) being less than 32 years old. All the participants were familiar with the Web2.0 concept, had prior knowledge of social tools and were active users of such tools for research purposes. 7 participants were researchers and 3 identified themselves as students. All the participants had a smartphone, powered by either Apple's iOS (n = 7) or Google's Android (n = 3) operating systems. The evaluations were focused on three dimensions: ease of use, satisfaction, and usefulness. In order to increase the accuracy and reliability of the questionnaire results compared to our previous evaluations, we used a seven-point Likert rating scale [40] and we added some extra questions to have more detailed feedback in the different evaluated dimensions. The questionnaires evaluated the usability of the application and were based on the USE questionnaire [41], where numerical values represent the agreement to a statement ranging from 1 to 7, with 7 being the highest agreement value. The results from the evaluation show us that the participants agree that More! is easy to use (M = 5,7 and SD = 1,06) and were convinced about its usefulness (M = 5,5 and SD = 1,43). Regarding the ease of use, the participants were a bit disappointed with a non flexible application that required some effort to be used successfully every time. We observed that these results were due to the problems experienced by the usage of QR codes. Also, it is important to mention that the participants found the tool as not being able to make them more productive.

In the questionnaire, participants had the opportunity to list the most positive and negative aspects of the tool. On the one hand, they were enthusiastic about the idea of combining information in one place and expressed the fact that the tool was easy to use. More! allowed the users to have a single point of access for different information sources, which provide the participants with a deeper view of the academic background of the presenter. On the other hand, the most important identified drawback of the application is the use of QR codes. Participants did not have any QR capturing and decoding application on their smartphone and expressed difficulties with finding and downloading one. Also, they were struggling with the quality of the printed codes, which made this a frustrating process. This means that the current workflow of the tool does not succeed in providing the frictionless experience to obtain the required data. This can also be observed from the results of the satisfaction dimension from the questionnaire, where participants express a bit less satisfaction with the tool (M = 5,2 and SD = 1,48), compared to the other dimensions – see Figure 4. While our sample size is too small to be conclusive, it is important to mention that our previous evaluation results [21] are consistent with these conclusions. In the previous evaluations a five-point Likert scale was used. In order to analyze and compare them, we used an equivalence of the Likert rating scales as presented in previous work [42]. Figure 4 summarizes and compares the results of the questionnaire with the initial studies. From the figure, we can observe that the trend among the different dimensions is descending when the context of the evaluation gets more realistic. Unexpectedly, the tool is not as easy and useful as we thought, and does not completely satisfy the users.

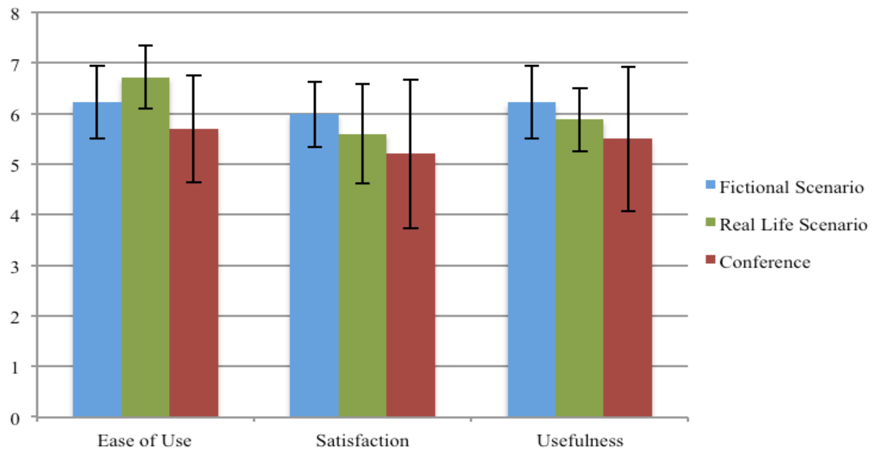


Fig. 4. Comparison of the different evaluations.

As stated before, the usage of the tool was tracked during the conference days. Figure 5 presents an overview of the visits and the devices that were used to access More!. There were a total 105 visits to the More! web application during the 4 days of the conference; which had around 250 participants in total. Out of the 105 visits, 62 were unique visitors over the 4 days, representing only 24.8% of the participants. In total, there were 260 page views with an average of 02:49 minutes per visit. Participants used More! less than two times and accessed around 4 pages during the conference. Furthermore, we were able to obtain an overview of the devices used to access the More! application. From the total number of visits, 63% of the visits came from personal computers and the rest from mobile devices. Unexpectedly, the application usage was significantly below than desired and its implementation for mobile devices was probably not necessary in these settings.

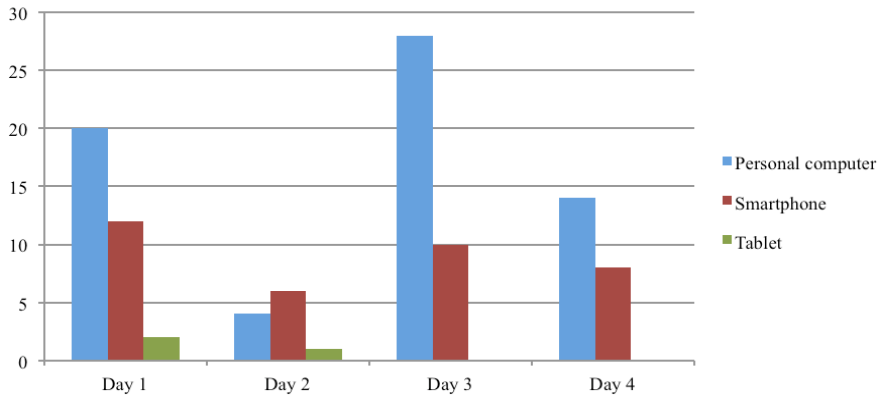


Fig. 5. Visits per device during the days of the conference.

6 Conclusion

In this paper, we have presented More!, a mobile web application that aims to increase awareness among researchers. Our application enables the enrichment of a face-to-face presentation with information that the Web2.0 environment provides. A back-end infrastructure was leveraged to support our mobile web tool. Our evaluation in different scenarios demonstrates that More! is simple, easy to use, and useful in a face-to-face scenario; but not widely accepted. The web application approach allowed participants to use different kind of devices to enjoy the benefits of the tool.

On the other hand, while the tool itself provides the expected functionality for the researchers, the initial fingerprint with QR codes is not an ideal solution and reduced the satisfaction level. Future work could include the test and comparison of different fingerprinting techniques to replace the QR codes, such as: shortened URLs, face recognition or location based services.

A deeper study of this type of discovery applications is needed, in order to better understand how awareness about ongoing relevant work, or even collaboration between researchers, can be improved. Moreover, there is a need to collect and connect the type of information that More! relies on. Research.fm aims to be the shared archive of data; consumed by More! and possibly other tools [29] [31].

Finally, we are working on a suite of Science2.0 tools for a wide range of devices (from handhelds, over laptops and desktops, to tabletops). Future work will further analyze these tools in a broader context. Indeed, it is important to understand more deeply the context in which applications like More! can bootstrap connections among researchers. Thus, we need to identify in more detail the specific requirements of researchers and the extent to which the research context influences "sensemaking" tasks [43] in the Science2.0 community.

Acknowledgments. We gratefully acknowledge the support of the STELLAR network-of-excellence, which is funded by the European Commission (grant agreement no. 231913).

References

1. Waldrop, M. M.: Science2.0. Scientific American, Vol. 298, No. 5, (2008), 68-73
2. Katzen, J.: Connecting researchers boosts collective intelligence. Research Information. June/July (2008)
3. Letierce, J., Passant, A., Breslin, J., Decker, S.: Understanding how Twitter is used to spread scientific messages. In Proceedings of WebSci10: Web Science Conference. Raleigh, NC, USA, (2010)
4. Research Information Network (RNI): If you build it, will they come? How researchers perceive and use web2.0. Report (2010)
5. University College London (UCL) CIBER Group: Social media and research workflow. Report (2010)
6. Heinze, N., Joubert, M., Gillet, D.: Connecting Early Career Researchers: Investigating the Needs of Ph.D. Candidates in TEL Working with Web2.0. In Proc. 2nd Intl. Workshop on Research2.0 at EC-TEL 2010, pp. 86--92. CEUR-WS (2010)
7. James, L., Norman, J., Baets, A.-S.D., Burchell- Hughes, I., Burchmore, H., Wilks, L., Wolffe, J.: The lives and technologies of early career researchers. (2009)
8. IBM Event Maps. https://researcher.ibm.com/eventmaps/
9. Wongchokprasitti, C., Brusilovsky, P., Parra, D.: Conference Navigator2.0: Community-Based Recommendation for Academic Conferences. In: Proc. of Workshop on Social Recommender Systems at IUI 2010, (2010)
10. Conference Guide, http://www.conferencegui.de/
11. Conference Compass, http://www.conference-compass.com/
12. CHI 2011 iOS App, http://itunes.apple.com/us/app/chi-2011/id434795508
13. Gist, http://gist.com/
14. Rapportive, http://rapportive.com/
15. Shhmooze, http://shhmooze.com/
16. Ban.jo, http://ban.jo/
17. Sonar, http://www.sonar.me/
18. JumpScan, http://jumpscan.com/
19. McConnell, S.: Rapid Development: Taming Wild Software Schedules, Microsoft Press Books (1996)
20. Shazam, http://www.shazam.com/

21. Parra, G, Duval, E.: More! A Social Discovery Tool for Researchers. In Proc. of World Conf. on EdMedia 2010, pp. 561--569. Chesapeake, VA: AACE, (2010)
22. ISO/IEC: ISO/IEC 18004:2006 Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification, (2006)
23. ScanLife, http://web.scanlife.com/en/
24. Evans, E.: Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley Professional (2003)
25. Sure, Y., Bloehdorn, S., Haase, P., Hartmann, J., Oberle, D.: The SWRC Ontology - Semantic Web for Research Communities. In Proc. 12th Portuguese Conf. on Artificial Intelligence. LNAI, vol. 3808, pp. 218--231. Springer Berlin, Heidelberg (2005)
26. The Friend of a Friend Project, http://www.foaf-project.org/
27. The Semantically-Interlinked Online Communities Project, http://sioc-project.org/
28. Representing vCard Objects in RDF, http://www.w3.org/Submission/vcard-rdf
29. Ullmann, T., Wild, F., Scott, P., Duval, E., Vandeputte, B., Parra, G., Reinhardt, W., Heinze, N., Kraker, P., Fessl, A., Lindstaedt, S., Nagel, T., Gillet, D.: Components of a Research2.0 infrastructure. In: ECTEL Conference 2010. LNCS, vol. 6383, pp. 590--595. Springer Berlin, Heidelberg (2010)
30. Page, K. R., De Roure, D. C., Martinez, K.: REST and Linked Data: a match made for domain driven development?. In: 2nd Intl. Workshop on RESTful Design, In Press (2011)
31. Parra, G., Duval, E.: Filling the Gaps to Know More! About a Researcher. In Proc. 2nd Intl. Workshop on Research2.0 at EC-TEL 2010, pp. 18--22. CEUR-WS (2010)
32. Cool URI, http://www.w3.org/TR/cooluris/
33. Nagel, T., Duval, E.: Muse: Visualizing the origins and connections of institutions based on co-authorship of publications. In Proc. 2nd Intl. Workshop on Research2.0 at EC-TEL 2010, pp. 48--52. CEUR-WS (2010)
34. Vandeputte, B., Duval, E.: Research at the table. In Proc. 2nd Intl. Workshop on Research2.0 at EC-TEL 2010, pp. 38—46. CEUR-WS (2010)
35. Social handle archive, http://soharc.upb.de/
36. Lagoze, C., Van de Sompel, H.: The open archives initiative: Building a low-barrier interoperability framework. In Proc. ACM/IEEE Joint Conf. on Digital Librarires. JCDL ’01, pp. 54–-62 (2001)
37. Klerkx, J., Vandeputte, B., Parra, G., Van Assche, F., Duval, E.: How to share and reuse learning resources: the ARIADNE experience. In: Proc. ECTEL Conference 2010. LNCS, vol. 6383, pp. 183--196. Springer Berlin, Heidelberg (2010)
38. Genov, A., Keavney, M., Zazelenchuk, T.: Usability Testing with Real Data. Journal of Usability Studies, vol. 4, no. 2, pp. 85--92. UPA (2009)
39. EC-TEL 2010, http://www.ectel2010.org/
40. Alwin, D., Krosnick, J.: The reliability of survey attitude measurement: The influence of questions and respondent attributes. Sociological Methods Research, vol. 20, no.1, pp. 139--181. SAGE (1991)
41. Lund, A.: Measuring Usability with the USE Questionnaire, Usability Interface, vol. 8, no. 2, (2001)
42. Colman, A., Norris, C., Preston, C.: Comparing rating scales of different lengths: Equivalence of scores from 5-point and 7-point scales. Psychological Reports no. 80, pp. 355--362, (1997)
43. Russell, D., Pirolli, P., Furnas, G., Card, S., Stefik, M.: Sensemaking workshop CHI 2009. In Ext. Abstracts CHI '09, pp. 4751--4754. ACM (2009)

Interacting with Semantic Data by Using X3S

Timo Stegemann, Tim Hussein, Werner Gaulke, and Jürgen Ziegler

University of Duisburg-Essen
Lotharstr. 65, 47057 Duisburg
firstname.lastname@uni-due.de

Abstract. The Internet has transformed increasingly from a document-centered web to a data-oriented one, enabling new opportunities to both users and developers. However, interacting with this data in a traditional way is often limited due to its amount and heterogeneity. In this paper, we describe X3S, an approach to filter and illustrate semantic content of data-webs, allowing the creation and use of components for data-exploration and visualization. In addition to that, we introduce an editor for X3S files, facilitating the creation of and the interaction with semantic data. The combination of the X3S format and the editor is compared to existing approaches and evaluated in a user study.

1 Introduction

Based on techniques and initiatives such as RDF and Linking Open Data, respectively, the available amount of semantic data steadily increases. Among others, electronic commerce intensely utilizes semantic representation of product data [4] and offers [5]. On the Internet, a shift from a rather document-centered web to a more data-oriented one can be observed as well. However, handling extensive, interconnected data may be cumbersome and not intuitive, which is why common methods of web engineering often reach their limits in this respect.

In this paper, we propose a method and a format for interacting with semantic data by defining templates following the X3S specification. X3S is an acronym for "XSL-transformed SPARQL results and Semantic Stylesheets". In contrast to CSS, known from web design, X3S includes directives for filtering data (by using SPARQL) as well as for decorating it (by using XSL transformations and CSS). In order to create and maintain X3S files, we developed an editor enabling the user to create complex templates for querying and presenting semantic data intuitively. Both the basics of the X3S format as well as the editor are introduced in this article.

2 Related Work

Several RDF or semantic web browsers have been developed to explore semantic data sources. However, presentation often is limited to tabular listings of all instances, properties, and property values. Examples are *Tabulator*[1] and *Disco*[2]. They work well for

[1] http:www.w3.org/2005/ajar/tab
[2] http://www.wiwiss.fu-berlin.de/bizer/ng4j/disco

getting a simple overview of data sets, yet they don't support styling or filtering of instances. As X3S is a solution with rich interaction for semantic data, we look at related projects with a similar focus.

The *Xenon* project [8] introduced an ontology described as "stylesheet ontology" or "RDF stylesheet lanuguage". The goal of *Xenon* is to display semantic data in a human usable way and facilitate the alteration of the representation according to the user's needs. Based on the concept of XML-Stylsheets (XSLT), the authors created a RDF-based stylesheet, which defines concepts for transforming RDF-data. The concept includes *lenses* and *views*. The purpose of lenses is data selection from instances whereas views describe the visualization of data elements. Just like with XSLT it is possible to embed HTML markup directly into the stylesheet for generating HTML representations of the data.

The RDF vocabulary *Fresnel* was developed as a successor of *Xenon*. Fresnel still uses lenses to select data but dropped the views concept and with it the possibility to embed visual markup such as HTML [7]. As a replacement, the *formats* concept is introduced. They are used for formatting data and enrich it with additional information for the renderer. For example, a format defines whether the selected lens data should be handled as a link, an image, or as a text. This added information is used by the browser for creating the final visualization. For example, the data could be represented as a table or graph, depending on the decision of the browser. To select data, lenses can use a special Fresnel Selector Language (FSL) or standardized SPARQL. Furthermore, lenses can be used in cascades for breaking data selection into several smaller pieces, thus improving reusability.

With *OWL-PL* [3], a language for transforming RDF/OWL data into (X)HTML is introduced. The language is strongly inspired by XLST and has the main goal to provide a simple transformation language for semantic data. *OWL-PL* allows the combination of transformational and representational markup. The language defines stylesheets, which are connected to semantic data by using a stylesheet ontology. The ontology describes how specific RDF classes are related to stylesheet elements. For example, the most universal relation for OWL:thing could be related to a tabular representation. If several relations are given for a class, the user can decide which graphical representation should be used. With help of a server-side Java application, stylesheets are converted into HTML.

With the introduction of *LESS* [1], a complete workflow from creating and processing templates for semantic data up to sharing templates between users is described. The declarative template language LeTL (LESS template language) is specified as a Smarty based templating language. It can process and transform semantic data from RDF documents or data requested by SPARQL queries. LESS provides an editor for creating LeTL templates. The editor shows available properties from the RDF data used or the SPARQL result. The properties can be directly used in LeTL code and combined with HTML markup to generate the final HTML output (Figure 1). The editor is not tied to HTML, other output formats can be created as well. The created LESS template is saved on the LESS server and can be accessed through a REST-based interface.

Contrary to the other approaches introduced in this section, *Dido* (Data-Interactive DOcument) [6], does not process semantic data, but is based on concepts used by *Xenon*

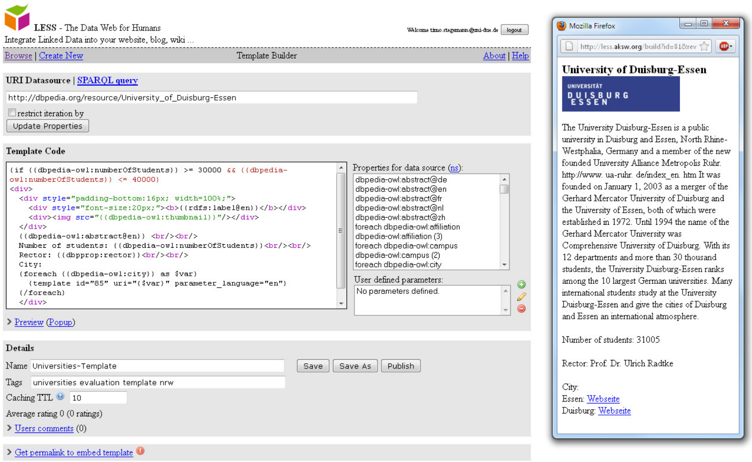
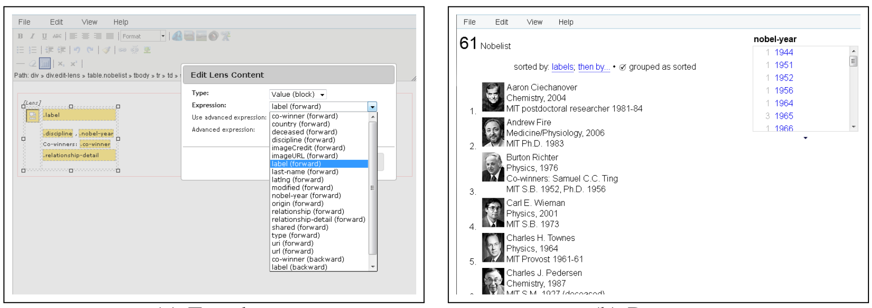


Fig. 1. LESS template editor (left) and the generated HTML output (right)

and *Fresnel*. *Dido* documents enable direct manipulation and creation of data as well as presentation directives. The embedded data is saved as key-value pairs in JavaScript-Object Notation (JSON). Selection and visualization are defined with lenses and views similar to *Xenon*. The user can directly choose in the editor, how the selected data should be displayed (Figure 2). For example, if the data should be represented as a list or table.



(a) Template (b) Document

Fig. 2. Dido's lens editor (left) and the generated result (rigth)

All approaches have in common, that they define their own templating languages for dealing with semantic data. Only *Dido* and *LESS* include editors facilitating the creation of new stylesheets. The editors focus on experts for semantic technologies, requiring a certain level of knowledge of the field. Editors for the other formats have not been developed so far.

X3S was designed to solve these shortcomings and provides a standard-based open format for semantic stylesheets. It can be used with arbitrary RDF-based data sources. The reference implementation provides an easy to use editor which enables non-experts the usage and creation of X3S documents.

3 The X3S Format

X3S is a format for describing semantic stylesheets. A semantic stylesheet is a template that can be applied to a RDF-based data source and leads to a filtered and styled subset of this data as a result. Its Goal is to transform semantic data into human-readable and exchangeable HTML documents. X3S defines a workflow consisting of the four steps *Querying*, *Restructuring*, *Transformation*, and *Styling*, most of which rely on established formats and technologies. Figure 3 illustrates the X3S workflow.

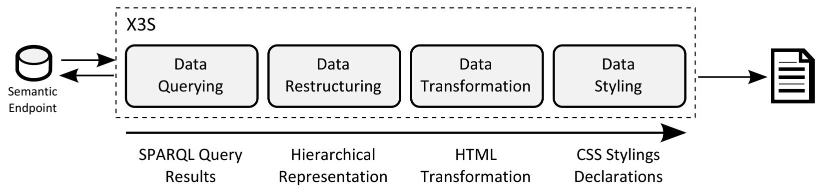


Fig. 3. X3S workflow to transform semantic data into a HTML document.

3.1 Querying

X3S uses SPARQL to retrieve content to be worked with. Any SPARQL compliant data source can be used such as *DBPedia*. With X3S, a single SPARQL query, that can request arbitrary properties, is deployed. The result of a SPARQL query is a set of property-value pairs, that are formatted in a W3C conform XML representation[3].

3.2 Restructuring

SPARQL results are not per se ordered hierarchically, leading to potentially redundancies. Thus, these results may have to be re-structured in order to remove redundancies and allow an unsophisticated way of processing the previously returned SPARQL-results. To achieve this, an additional XML-structure is used, onto which all SPARQL-results are mapped. Finally all result sets are aggregated into a single XML-structure, that contains all values as attributes of XML-nodes. Figure 4 shows a restructuring process.

[3] `http://www.w3.org/TR/rdf-sparql-XMLres`

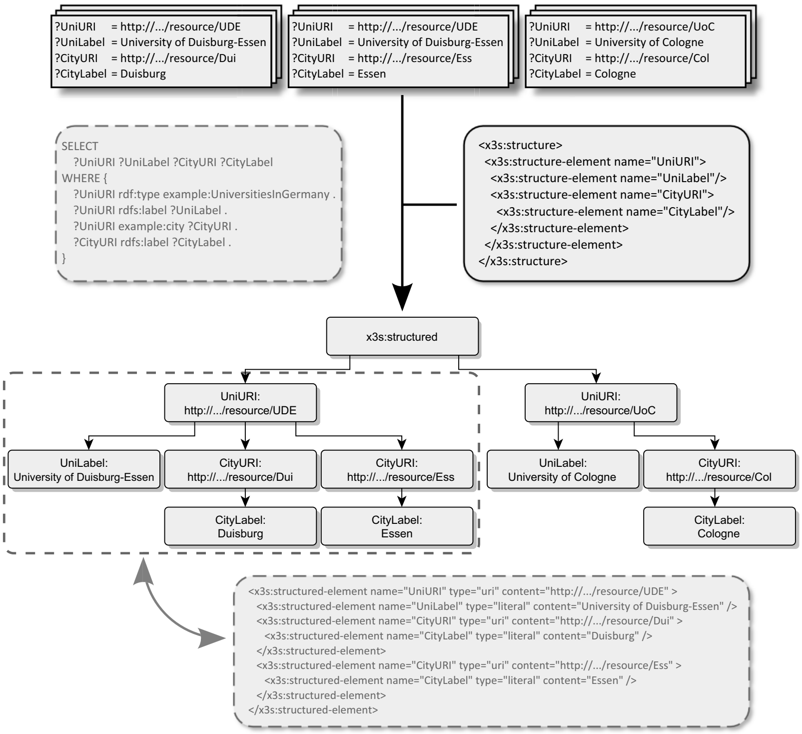


Fig. 4. SPARQL-results with redundancies are mapped onto a predefined hierarchical XML-structure, that meet the hierarchy of the primary SPARQL-request. All values are saved in the node's attributes.

In this example, information about German universities and their cities are requested from an notional SPARQL-endpoint. Because the University of Duisburg-Essen is located in two cities simultaneously – Duisburg and Essen – the SPALRQL-request returns two result sets with partial redundant data. The information about the university is equal, the information about the cities differs.

Direct processing of this data via XSLT, which is used in the next process step, would require some grouping mechanisms, that are possible but become very complex when nesting more than only one object into another. Therefore all result sets are mapped onto a predefined hierarchical XML-structure, that meets the hierarchy of the SPARQL-request without any redundancies.

3.3 Transformation

We use XSLT 1.0 for transforming the resulting XML trees or fragments into HTML, including CSS classes and IDs that can later be used to decorate the HTML (see Figure 5).

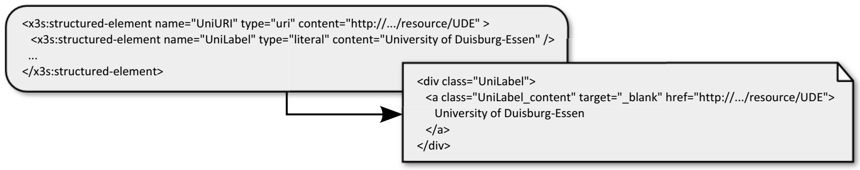


Fig. 5. The restructured Data is transformed via XSLT into an HTML document.

3.4 Styling

In order to specify the appearance of the documents, CSS can be embedded.

4 An Editor for Semantic Stylesheets

In order to create, edit, and preview X3S files, we developed an editor as a rich internet application based on Adobe Flex. The editor runs on any web browser with flash capabilities and can make use of existing semantic data repositories that provide a SPARQL interface.

Figure 6 shows the main window of the editor. In order to enable non-technical users to create and maintain stylesheets, we use point-and-click interactions instead of complex text-based directives: Drag-and-drop is used for selecting properties from a list of all possible attributes for the particular RDF class (Figure 6 A). The properties can be filtered and sorted by type, relevance[4], or in alphabetic order. In order to provide visual cues, datatype and object properties are displayed with different icons.

The user can drag elements from the list into the template located in the central workspace (Figure 6 B). When adding datatype properties to the template, an appropriate display format can be selected (for instance *depiction* in order to show the value of an URI as an image instead of the URI itself). As an additional feature, dynamic filters can be created based on datatype properties. This may be useful, if only a subset of all possible entities should be displayed (digital cameras with a price range from $100 to $200 for instance). Depending on the type of datatype property (String, boolean, integer, etc.), widgets such as sliders or date pickers for those filters are created dynamically.

The templates can finally be styled by using standardized CSS. The editor supports the user by directly offering graphical shortcuts to common CSS declaratives such as *bold* or *font size* (see Figure 7). Elements can have defined borders or margins, images can be adjusted in width and height, etc.

The elements in the workspace and the applied filters are internally used for generating a SPARQL query, which selects the respective result from the SPARQL source. This result is transformed into a hierarchical representation and transformed into HTML by the browser's XSLT processor (incorporating the CSS information). The results can be previewed in an IFrame (Figure 6 C), so that the user receives immediate feedback.

[4] *Relevance* is measured by the frequency of occurrence of this property among all instances of the class.

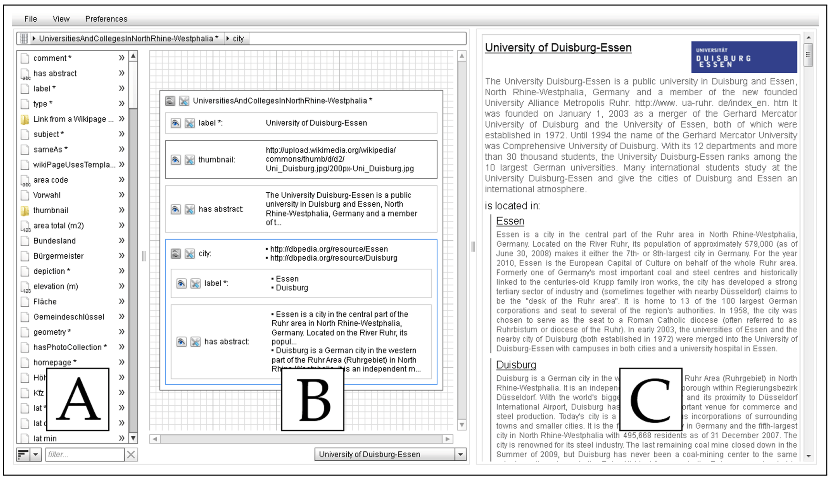


Fig. 6. X3S editor while creating a semantic stylesheet, working with data from DBpedia.

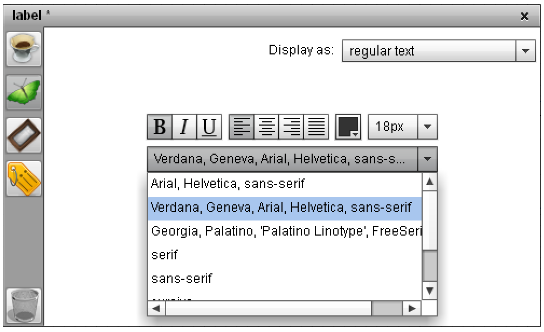


Fig. 7. Styling and formating options for regular text.

After the template has been created, it can be exported as an X3S file or archived as an HTML document.

5 Comparison

After introducing X3S and the corresponding editor, we now compare our approach to the ones described in the related work section (see Table 1). In terms of “Separation of Concerns”, all procedures but Dido are accessing their data from a separately saved source. Only X3S and LESS have a “SPARQL-Endpoint Support” and are able to request their data directly from an external semantic database. All approaches provide means for “Property Selection”, so only a subset of the data can be selected for presentation. “Filtering” of entries with regard to particular data values is also possible.

	X3S	Xenon	Fresnel	OWL-PL	LESS / LeTL	Dido
Separation of Concerns	●	●	●	●	●	○
SPARQL-Endpoint Support	●				●	
Property Selection	●	●	●	●	●	●
Filtering	●	●	●	●	●	●
Styling	●	●		●	●	○
Nested Object-Properties	●	○	○	●	○	
Usability (Developer)	●	○	○	○		
Usability (Author)	●	○	○	○	○	○
Usability (End User)	●	●	○	●	●	●
Templating	●	●	●	●	●	●

Table 1. Comparison of the approaches: ● support, ○ partial support, no entry: no support.

“Styling” of the data cannot be specified freely with Fresnel. All other candidates convert the data into HTML and can style them with CSS. It is simple with X3S and OWL-PL to display “Nested Object-Properties”, that are completely integrated into a single stylesheet. With the other procedures, a stylesheet author has to build a separate stylesheet that must be embedded into the primary one. All candidates are able to re-use the stylesheets by a “Templating” mechanism.

One main goal for X3S was a high usability for developers, stylesheet authors, and end users. To reduce the amount of work for “Developers” to implement X3S, we tried to restrict X3S mostly to familiar and approved techniques. Xenon, Fresnel and OWL-PL however use own languages to describe the stylesheets, which are created with them. The “Author’s” usability is principally determined by the tools and editors, with which the author can create a new stylesheet. LESS’ and Dido’s editor are capable to facilitate the author’s work, but in our opinion are to complex to be used by non-experts in the field of semantic data. The “End User” should normally not even notice the use of semantic data. So most approaches generate a HTML-document from the template or semantic stylesheet or, such as Fresnel, require a dedicated browser to view the data.

6 Evaluation

We compared the X3S-Editor with the LESS-Editor in an evaluation and evaluated their respective usability. The essential results of this evaluation are presented in this section. The evaluation was designed as a user study during which each test user had to solve five tasks with both editors (Within subject design). All five tasks combined created a semantic stylesheet, where each task should cover an essential part of the whole stylesheet creation process – property selection (T1), data styling (T2), adding static data (T3), filtering (T4), and nesting further objects (T5).

In order to derive the efficiency of the editor, we measured the time needed to to complete a single task. To find out it’s effectiveness, we recorded whether a test user solved a task, needed assistance to solve a task, or could not solve a task. To get information about the user's satisfaction, the test users had to rate ten statements on a 5 point Likert scale after solving all five tasks with an editor, defined by the System Usability Scale (SUS) [2].

Ten test users participated in this study. Two users stated that they have well HTML and CSS skills. The others described their skills as rather weak. Six participants rated their general programming skills as above average, while a single participant had no programming experiences. The editors were preconfigured to work with DBpedia's SPARQL endpoint. The test users' global task was to create a stylesheet, containing information about universities and the cities in which they are located. No participant had used one of the two editors previously. Before the test started, the users were briefly instructed about the basic functionalities of the respective editors. They also received a summary of the LESS-Editor's commands, needed to solve the tasks.

Most participants were able to solve the five tasks with the X3S-Editor without any assistance. However, few test users needed assistance while solving tasks 4 and 5, because they did not find the proper filter settings or did not realize, that they can embed further objects into an already existing one. The LESS-Editor however confronted the test users with more problems. In most cases, the users needed assistance. Few test users could not even solve some tasks with repeated assistance. Only task 3 was simple enough to be solved by every participant. Because the LESS-Editor produced an error while inserting a property with a number value, most participants were not able to solve task 1. Only one user was able to repair this error without assistance. Most participants lacked the necessary CSS commands, needed to solve task 2. Since the LESS-Editor did not support the users when filtering different values, only the six participants, that rated their general programming skills as above average, were able to solve task 4. Even though the users had an overview of LESS' necessary programming commands, most were not able to embed a further object into the main stylesheet, tested in task 5. Figure 8 shows the results in detail.

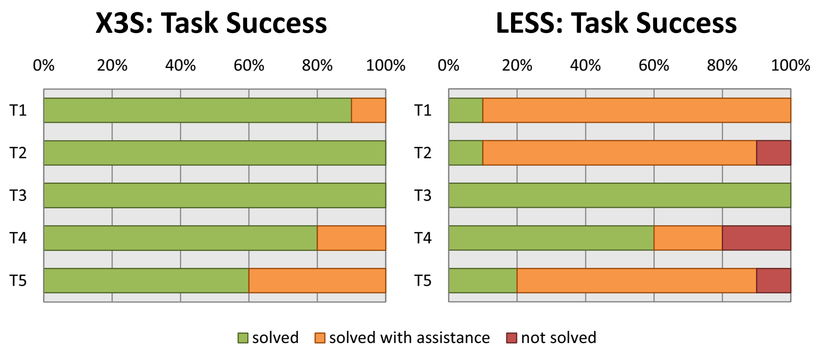


Fig. 8. We measured the both editors' efficiency by the user's task success. Most participants were able to solve all task with the X3S-Editor themselves. However, most test users needed assistance to solve the same tasks with the LESS-Editor.

In average the participants could solve most tasks significantly faster with the X3S-Editor than with the LESS-Editor (231% (T1) - 303% (T2) of the time needed with the X3S-Editor). Only Task 3 could be solved faster with the LESS-Editor. Summed up, the test users needed 7:26 minutes on average to create the whole stylesheet with the X3S-Editor. With 16:36 minutes on average, it took twice as long to do the same work with the LESS-Editor. All task times are shown in Figure 9.

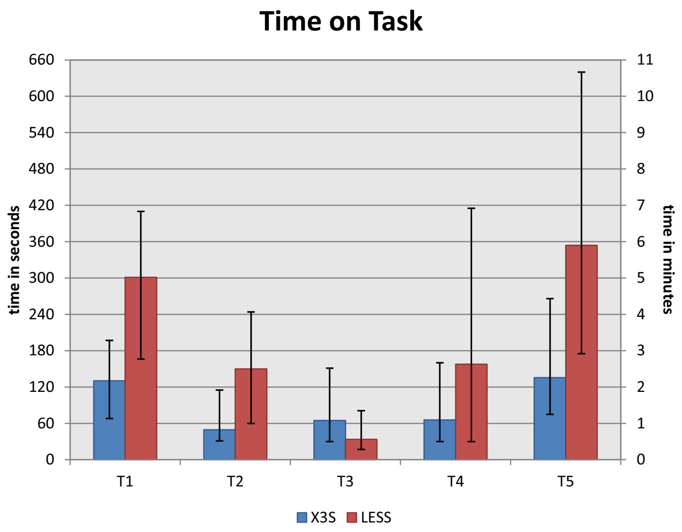


Fig. 9. Average times to solve the respective task. The error bars represent the minimum and maximum time, needed to solve a task.

Figure 10 shows the results of the System Usability Scale. The X3S-Editor achieved an average value of 90.5. One participant even rated the X3S-Editor with the maximum value of 100. However, the LESS-Editor only achieved an average value of 27.5. An average SUS score under about 60 points can be called as “relatively poor”, while a score over about 80 points could be considered “pretty good” [9].

7 Discussion

In this paper, we introduced X3S as well as a corresponding editor. X3S is a specification for filtering and displaying semantic data. The corresponding editor allows the creation of templates for data exploration and visualization, based on this X3S specification. X3S supports the separation of data and layout, arbitrary filtering of data and any kind of visualization via CSS. One main focus in the development of X3S was the ease of use for developers and end users.

The presented technique provides a variety of ways to use semantic information in various web applications. For example, predefined X3S templates in different config-

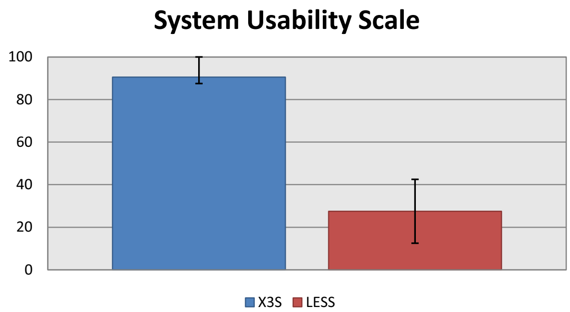


Fig. 10. Results from the System Usability Scale. The X3S-Editor achieved an average value of 90.5. One participant even rated the X3S-Editor with the maximum value of 100. The LESS-Editor only achieved an average value of 27.5.

urations can be used, helping to search a product. Due to their ease of use, they are suitable to build environments for intuitive exploration of semantic data such as customer advisory systems based on multi-touch surfaces.

The evaluation showed that the X3S-Editor is able to achieve the desired goal, to allow all users – even non-experts in the field of semantic data and web design – to create semantic stylesheets. None the less we revealed functions, that can be optimized. Especially the combination of dynamically retrieved semantic data and statically added text can be improved. Our future work tends to provide a generic component for processing X3S and to test it in the context of different application systems.

References

1. Sören Auer, Raphael Doehring, and Sebastian Dietzold. Less – template-based syndication and presentation of linked data. In *ESWC '10: Proceedings of the 7th Extended Semantic Web Conference*, number 6089 in Lecture Notes in Computer Science, pages 211 – 224, 2010.
2. John Brooke. SUS: A quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, *Usability evaluation in industry*. Taylor and Francis, London, 1996.
3. Matt Brophy. Owl-pl: A presentation language for displaying semantic data on the web. Master's thesis, Lehigh University, 2010.
4. Martin Hepp. Products and services ontologies: A methodology for deriving owl ontologies from industrial categorization standards. *International Journal on Semantic Web & Information Systems (IJSWIS)*, 2(1):72–99, 2006.
5. Martin Hepp. Goodrelations: An ontology for describing products and services offers on the web. In *EKAW '08: Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns*, volume 5268 of *Lecture Notes in Computer Science*, pages 329–346, 2008.
6. David R. Karger, Scott Ostler, and Ryan Lee. The web page as a wysiwyg end-user customizable database-backed information management application. In *UIST '09: Proceedings of the 22nd Annual ACM Symposium on User interface Software and Technology*, pages 257–260, New York, NY, USA, 2009. ACM.

7. Emmanuel Pietriga, Christian Bizer, David R. Karger, and Ryan Lee. Fresnel: A browser-independent presentation vocabulary for rdf. In *ISWC '06: Proceedings of the 5th International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 158–171. Springer, 2006.
8. Dennis Quan and David R. Karger. Xenon: An rdf stylesheet ontology. In *WWW '05: Proceedings of the 14th International World Wide Web Conference*, 2005.
9. Thomas Tullis and William Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann, San Francisco, CA, USA, 2008.