



RENDER
FP7-ICT-2009-5
Contract no.: 257790
www.render-project.eu

RENDER

Deliverable D4.2.2

Best Practices for Diversity-Aware Collaboration

Editor:	Fabian Flöck, KIT
Authors:	Andreas Thalhammer, UIBK; Javier Caminero, Carlos Picazo, Telefónica; Fabian Flöck, KIT, Daniele Pighin, Google; Mitja Trampus, JSI, Kai Nissen, Wikimedia
Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	30 September 2013
Actual delivery date:	30 September 2013
Suggested readers:	Developers of diversity-aware software tools
Version:	1.0
Keywords:	Deliverable, evaluation, best practices, diversity, opinion mining, social media, collaborative, web 2.0, extensions, RDF, semantic, community

Disclaimer

This document contains material, which is the copyright of certain RENDER consortium parties, and may not be reproduced or copied without permission.

In case of Public (PU):

All RENDER consortium parties have agreed to full publication of this document.

In case of Restricted to Programme (PP):

All RENDER consortium parties have agreed to make this document available on request to other framework programme participants.

In case of Restricted to Group (RE):

The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement. However, all RENDER consortium parties have agreed to make this document available to <group> / <purpose>.

In case of Consortium confidential (CO):

The information contained in this document is the proprietary confidential information of the RENDER consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the RENDER consortium as a whole, nor a certain party of the RENDER consortium warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

Full Project Title:	RENDER – Reflecting Knowledge Diversity
Short Project Title:	RENDER
Number and Title of Work package:	WP4 – Diversity Toolkit
Document Title:	D4.2.2 - Best Practices for Diversity-Aware Collaboration
Editor (Name, Affiliation)	Fabian Flöck, KIT
Work package Leader (Name, affiliation)	Fabian Flöck, KIT

Copyright notice

© 2010-2013 Participants in project RENDER

Executive Summary

In this deliverable, we describe the best practices when designing, developing and deploying different tools for knowledge diversity and the lessons learned in the process, including the interaction with the respective communities. We cover the Drupal extension “Diversity Enricher” in section 2, the Telefonica Opinion Mining Tool (T-OMT) in section 3, the Wikipedia Tools for diversity in section 4 and lastly the DiversiNews tool in section 5, before concluding with section 6. Each of the tool sections is divided into seven parts: **The concept of diversity and its actual interpretation and implementation in the use case:** Here, we describe how “knowledge diversity” was understood in each use case and how the concept was implemented in each application. **Design of Interfaces (and core functionalities):** This subsection describes what worked well in designing the interfaces for visualizing the diverse information and what was changed in the development process to optimize the usability. **Chosen technologies:** We reason why the technologies used for implementation were chosen and why this was or wasn’t a good choice in each specific case. **Way of development:** What methodologies were used for development, how well did they work, how was the interaction with the different stakeholders in the process and what were major challenges? **Approach to communities (e.g. developers, evaluators, end-users):** We describe when and how approached different communities, what problems we encountered and what we learned about the specific characteristics of each of them and their acceptance for the presented knowledge diversity concept and tools. **Way of introducing the technology to platforms, tools, or other technologies:** This subsection describes how the new technology was integrated into existing systems, how it communicated with other platforms and/or how it was made available to end-users for usage and installation. **Type of usage by the community:** This last segment describes how end-users actually employed the tools and where they deviated from what was foreseen by our initial concepts.

The following most important best practices and findings of putting these kinds of tools into application emerged as a communality of the presented use cases:

- When introducing the concept of knowledge diversity and its tangible implementation as a software tool to a community of either end-users or developers, it is important to find the right trade-off: On the one hand it might be harmful addressing the community with an immature product, confusing users (as there might be too many dimensions of diversity involved or the concept as such is not understood) and/or watering down the initial concept of the use case too much (because of large number of users that want to give input). On the other hand by not involving enough users early enough one might lose acceptance in the community and/or lose touch to what is important to them. Generally we tried to make contact to users early in the process, but mostly with some preconceived notion of how we wanted knowledge diversity to mean in the respective context. This could have been done earlier in some cases.
- In a similar vein, “knowledge diversity” as a concept is particularly hard to define ex-ante, outside real use and data and cannot be imposed on end-users. It is highly subjective to interpretation by different stakeholders (developers, users and also internally by different partners and developers) and will therefore always be subject to constant renegotiation and iterations. We highly recommend personal, informal contact to end-users even before any planned evaluation or testing starts to learn different points of view on the topic and get early “grounding” to the real use scenarios.
- Exploring knowledge diversity is actually a very popular theme with users. Although the testers might have not rated some tools as optimal during evaluation, the general goal of these tools and how they either try to generate more knowledge diversity or explore it was received very positively. Especially the possibility to navigate contents exhibiting different sentiments towards a common topic was highlighted as very valuable.
- Showing the diversity of different sources of knowledge and different viewpoints seems, on average, to increase the trustworthiness of the shown content. Users put – generally – more trust into the source if it discloses related, even non-agreeing content.

Table of Contents

Executive Summary	3
Table of Contents	4
List of Figures.....	5
Abbreviations.....	6
1 Introduction	7
2 Diversity Enricher: The RENDER Drupal extension.....	8
2.1 The concept of diversity and its actual interpretation and implementation in the use case.....	8
2.2 Design of Interfaces	8
2.3 Chosen Technologies	9
2.4 Way of Development	9
2.5 Approach to Communities	10
2.6 Way of Introducing the Technology to the Platform	11
2.7 Type of Usage by the Community.....	11
3 Telefónica Opinion Mining Tool	12
3.1 The Concept of Diversity and its Interpretation and Implementation in the Use Case.....	12
3.2 Design of Interfaces	12
3.3 Chosen Technologies & Way of Development.....	14
3.4 Approach to Communities	15
3.5 Integrating the Technology with Existing Services, Platforms and Technologies.....	16
3.6 Type of Usage by the Community.....	16
4 Wikipedia Tools.....	18
4.1 The Concept of Diversity and its Interpretation and Implementation in the use Case	18
4.2 Design of Interfaces and Core Functionality.....	18
4.3 Chosen Technologies & Way of Development.....	19
4.4 Integrating the Technology with Existing Services, Platforms and Technologies.....	19
4.5 Approach to Communities	20
4.6 Type of Usage by the Community.....	21
5 DiversiNews.....	22
5.1 The Concept of Diversity and its Interpretation and Implementation in the Use Case.....	22
5.2 Design of Interfaces	22
5.3 Chosen Technologies & Way of Development.....	22
5.4 Integrating the Technology with Existing Services, Platforms and Technologies.....	23
5.5 Approach to Communities	23
5.6 Type of Usage by the Community.....	23
6 Conclusions	24

List of Figures

Figure 1: Screenshot of the Basic Interface.....	9
Figure 2: T-OMT Initial Interface	13
Figure 3: T-OMT Final Interface.....	13
Figure 4: T-OMT Architecture.....	16

Abbreviations

CMF	Content Management Framework
CMS	Content Management System
KDO	Knowledge Discovery Ontology
KIT	Karlsruhe Institute of Technology
LOD	Linked Open Data
RDF	Resource Description Framework
SDK	Software Development Kit
TID	Telefónica Investigación y Desarrollo
T-OMT	Telefónica Opinion Mining Tool
TwDiViz	Twitter Diversity Visualization Tool
UI	User Interface
URI	Unique Resource Identifier
WP	Work Package
SMW	SemanticMediaWiki

1 Introduction

In this deliverable, we describe the best practices when designing, developing and deploying different tools for knowledge diversity and the lessons learned in the process, including the interaction with the respective communities. We cover the Drupal extension “Diversity Enricher” in section 2, the Telefonica Opinion Mining Tool (T-OMT) in section 3, the Wikipedia Tools for diversity in section 4 and lastly the DiversiNews tool in section 5, before concluding with section 6. Each of the tool sections is divided into seven parts:

1. **The concept of diversity and its actual interpretation and implementation in the use case:** Here, we describe how “knowledge diversity” was understood in each use case and how the concept was implemented in each application. This includes also learnings and change processes the concept went through in some cases while experiences were gathered during the technical development as well as the interactions with the community.
2. **Design of Interfaces (and core functionalities):** This subsection describes what worked well in designing the interfaces for visualizing the diverse information and what was changed in the development process to optimize the usability.
3. **Chosen technologies:** We reason why the technologies used for implementation were chosen and why this was or wasn’t a good choice in each specific case.
4. **Way of development:** What methodologies were used for development, how well did they work, how was the interaction with the different stakeholders in the process and what were major challenges?
5. **Approach to communities (e.g. developers, evaluators, end-users):** We describe when and how we approached different communities, what problems we encountered and what we learned about the specific characteristics of each of them and their acceptance for the presented knowledge diversity concept and tools. We also cover internal procedures proper to these communities and how best to deal with them.
6. **Way of introducing the technology to platforms, tools, or other technologies:** This subsection describes how the new technology was integrated into existing systems, how it communicated with other platforms and/or how it was made available to end-users for usage and installation. Here as well we describe what worked and didn’t and give recommendations for future
7. **Type of usage by the community:** This last segment describes how end-users actually employed the tools and where they deviated from what was foreseen by our initial concepts. It also provides some success stories and testimonials from the target audience.

2 Diversity Enricher: The RENDER Drupal extension

This section describes best practices regarding the RENDER Drupal extension, dubbed “Diversity Enricher”. It is available at:

- <https://drupal.org/sandbox/sti-innsbruck/1991696>

The lively discussion thread concerning the “full project” application is available at:

- <https://drupal.org/node/1996206>

As of the time of writing, the project is still in the code review process. The comments are merely of technical nature. This also reflects the complete <http://drupal.org> community, which attracts more developers rather than actual users.

In the following sections, we describe our learnings with respect to the design of interfaces, chosen technologies, way of development, approach of communities as well as the way of introducing the technology to the platform.

2.1 The concept of diversity and its actual interpretation and implementation in the use case

The Diversity Enricher Drupal extension was developed in order to support content authoring and consumption with diversity-aware features. The diversity concept was focused on interweaving topics (see D2.2.2) and sentiment towards them (see D2.1.2) produced by the Enrycher service. In the demo session at I-SEMANTICS 2013 in Graz, several users liked the idea that they could click on an identified topic and receive three lists respectively covering positive, neutral, and negative articles about the topic. This was pointed out to help identifying strong views and opinions in a single blog (e.g. the republic party is only mentioned in negative articles). The initial concept of diversity we had planned hence worked out well for this use case. As a learning it might be said that an informal discussion of how and why “diversity” should be brought to Drupal could have taken place earlier in the project in an informal way. This might have helped to steer the process better, although its lack didn’t prove to be a huge drawback in the development process.

2.2 Design of Interfaces

The interface is designed as a to combine optimal usability with a plain visual style that does not interfere with the main task of reading and writing. The professional evaluation of the interface with respect to diversity aspects (cf. D3.1.3) resulted in mostly “good” or average marks (no red flags at all). However, the following amendments were made on the way towards the current version:

- Redundancy: Topics originally were covered in the running text and as a topic listing on the right hand side: we removed this listing and replaced it with a tag cloud.
- The sentiment score is now displayed clearly at the top of each article. This helps users to make a clear relation to the related “positive, neutral and negative” articles that are shown on the right hand side.
- The export functionalities are displayed below the article rather than on the right hand side. This leaves additional space for the related articles and the tag cloud.
- The import function was moved to the administration interface.

Bringing the view of a design expert into the project hence clearly benefitted the interface development process of the extension. There is now a clear relation between the DiversiNews “Search for related entities” and the “related articles” feature of the Diversity Enricher Drupal module. Both interfaces recommend further articles. However, in Diversity Enricher, despite the usual topic overlap, we also

distinguish by sentiment scores and enable the users to get a nice overview about the conveyed sentiments of the recommended articles. These amendments were important improvements over the initial planning.



Figure 1: Screenshot of the Basic Interface.

2.3 Chosen Technologies

The used technology is a mix of classic PHP, the Drupal module interface, CSS, and jQuery. This mix enables seamless diversity-aware interaction and browsing. PHP and Drupal module interfaces is a requirement for developing a Drupal extension. However, the technologies are suitable for implementing diversity aspects. jQuery provides a JavaScript abstraction layer and supports unified interaction across different browsers. This helps to enable a unique interface for all users and browsers. The JavaScript interaction allows for seamless interaction and browsing, supporting pop-ups when clicking on topics or tags. For UIBK, especially the quick interaction enabled by jQuery can be regarded as a key feature in this diversity-enabled environment as users tend to perform a lot of exploring by clicking and opening new dialogs and windows to navigate the space of different articles, views and topics.

The communication with Enrycher and the Diversity-Aware Ranking Service is based on HTTP calls and as such enabled by PHP. During the code reviewing process, several recommendations were made that built-in Drupal services should be better exploited rather than taking the straight forward PHP way of implementation. This prevents unconscious manipulation of key security and stability services that are provided by the Drupal API.

2.4 Way of Development

The first prototype was developed by UIBK programmers. The further versions, including the alpha and beta versions included extensive testing, code reviews and iterative development cycles. The Drupal community is very active on code reviews and contributed useful suggestions as soon as project was

submitted for admission to “full project” status. The following lessons have been learnt from developing a Drupal module.

- Get in touch with the community as early as possible, as strict guidelines exists for code-reviews that are time-consuming
- Create the project as a sandbox project and apply for full project status as early as possible
- Early code reviews will help to stick to the Drupal code standards better and hence raise acceptance and avoid conflicts when installed in various Drupal settings

2.5 Approach to Communities

The Drupal community was initially approached by our colleague Ioan Toma in a Drupal Semantic Web group forum: <https://groups.drupal.org/node/273928> .

The initial comment by user no2e was about hosting the project at <http://drupal.org> portal. In the following, the team strived for pursuing such a release (alpha and beta versions) by fixing bugs, testing, and code reviews. In early May 2013 it was achieved to migrate the code from <http://github.com> to <http://drupal.org>. At the same time, the team started pursuing the full project status. Several rather technical discussions followed at: <https://drupal.org/node/1996206> .

At the end of May 2013, the Innsbruck team decided to approach the Drupal developer community with a session proposal at the Drupal Developer Days in Dublin.

The proposal was as follows:

SESSION TITLE

Content Linking, Enrichment, and Diversification

SPEAKERS

Andreas Thalhammer

DESCRIPTION

The amount of linked as well as open data is on the rise. As a consequence, more and more content enrichment services create better links as well as annotations. With those, explained links to new internal and external articles can be recommended leading to a more diversified user experience.

This session will give an overview on the state of the art of the following topics:

- data sources
- content enrichers
- content recommendation
- link explanation

In general, this session aims to point developers to new data sources, tools, and application scenarios.

However, unfortunately, this talk was probably valued as too unspecific from a technical perspective.

As a lesson learned, it can be said that the Drupal community is actually a very technical, developer community. There are, of course, much more Drupal users than developers, however, they don't seem to organize as a coherent "Drupal user community". Therefore, we need to draw a clear border to the Drupal developers. With respect to the topic of the extension, it became clear that the users could see the benefit of an enriched view with "diversified" information quite quickly and liked the idea, as also became apparent at the demo session at I-SEMANTICS 2013 in Graz.

We wanted to present a half-way "mature" product to the Drupal developer community to avoid putting them off to the idea of "diversifying" and to convey the concept better, as usually, the first impression also counts with these technical communities and to many discussions and changes can fray the initial idea to much. The reasoning behind this seems to have proved valid, but considering the experiences made, an earlier approach should still be considered given the long processing times for new the review process.

2.6 Way of Introducing the Technology to the Platform

The Diversity Enricher Drupal module is used in combination with Drupal. The installation procedure is straightforward and requires a minimum of external libraries. Other technologies the module is built on are Sesame/OWLIM, Enrycher and the Diversity-Aware Ranking Service. From our evaluation we saw that this add-on installation is in fact the most efficient and easily usable way to integrate the extension into Drupal.

2.7 Type of Usage by the Community

As mentioned earlier, the Drupal community is more focused on technical and developer aspects. The code reviewers expressed interest in the solution, e.g. the user Ayesh stated "Good job, and sorry if above suggestions sound any bad. I didn't test the module yet but trying to be helpful to other project reviewers". Another user, kersai, stated "Hello, pretty interesting seeing what that thing does. Seems to 'work,' here are a few things I noticed that you may want to clean up." These examples for feedback underline the technical orientation of the Drupal community and the general agreement to the idea. On the other hand, non-technical users that we approached did not seem to be willing to grapple with the new features. One line of comment was "I'm not sure I understand much of this..." A conclusion from this kind of comments could be to introduce diversity features in a simpler way to non-technical users in order to not to swamp them with previously unknown features. A step-by-step strategy might be the more appropriate way for future developments. In the demo session of the I-SEMANTICS 2013 conference, the participants – of whom most also have a strong technical background – were positively surprised by the "sentiment-directed" way of interaction and expressed their interest to use this or a similar application for blogging.

3 Telefónica Opinion Mining Tool

3.1 The Concept of Diversity and its Interpretation and Implementation in the Use Case

The main objective of T-OMT application is to deal with diversity, inherent to online customer data in a corporate domain, to improve customer relationship management. For this purpose, advantage can be taken of the information diversity extracted from non-structured information, collected in this case from Twitter as the preferred channel used by Telefónica customers to provide feedback, reviews and discussions about Telefónica products and services.

The use therefore aimed to leverage the diversity inherent to the data, expressed as a plurality of opinions and viewpoints on a particular topic; and to use it as a primary source of analysis for Telefónica's use case, fulfilling one of its important goals, the topic extraction and the evolution tracking to be applied over diversity in order to detect trends formation and customer satisfaction. In this way, management of customer feedback, management and marketing decisions can be based on the extracted information. One important learning regarding the adaption and refinement of the concept of "diversity" applied in the use case was certainly that not all dimensions possibly extractable and representable from the data should, in fact, be mined and represented, as users prefer a restriction to the most important information dimensions for a given task and not a maximized exploration space of dimensions. This became clear throughout the user interviews and evaluation; the tool was adapted accordingly.

3.2 Design of Interfaces

Initially, an interface separated in two main areas was designed (see Figure 2): one area (on the left side) was intended to generate a new report (the report filter options panel) and the other one (on the right part) is the report information panel where the active reports are shown. After some user testing most of them pointed out that the 'filter options panel' should not be constantly shown. The users did not consider this design acceptable for several reasons:

- Once the user has analyzed the data of the generated reports, the report filter options panel becomes useless. Therefore it was clear that this part should only be shown while the report was being generated, not later.
- In the initial interface, all options were shown at all times, while it turned out to be more useful to show only the active options at the different stages of the report generation process. Following this principle, the interface was redesigned and a single "menu button" with a drop-down list is now showed and in the drop-down list the active options are highlighted (see Figure 3).
- As another design dimension, the color palette was modified from the initial to the final version, basing it on the RENDER Project color palette.

For avoiding these kind of changes in future designs of diversifying software tools, a learning would be to start with some of the user interviews before producing the first working version, even in an unstructured, oral form, to avoid a "tunnel view" developing process where technical factors play a overly important role in comparison to end-user needs.

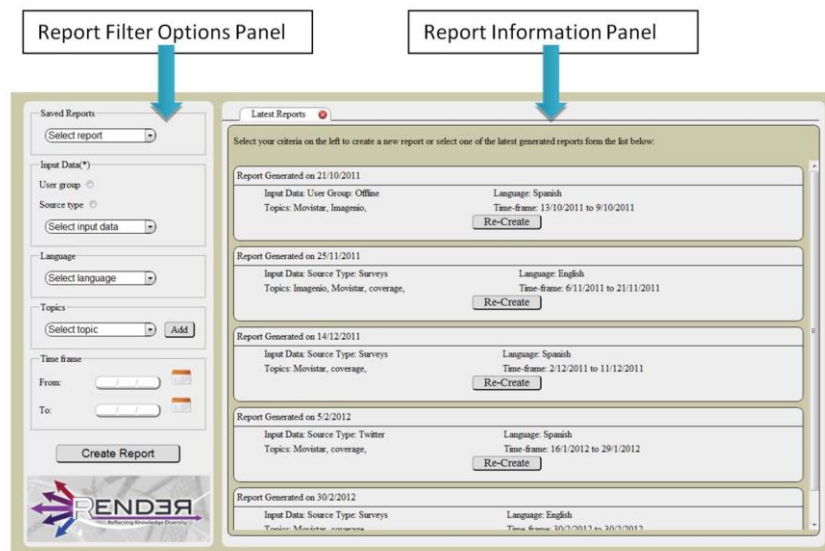


Figure 2: T-OMT Initial Interface

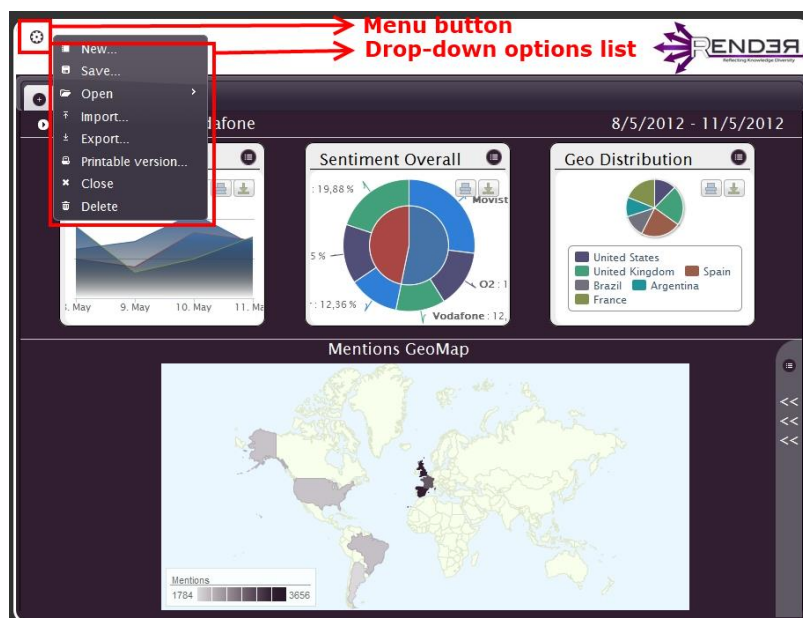


Figure 3: T-OMT Final Interface

The search, processing and classification of opinions expressed through free text messages in the frame of an unfiltered community (Twitter) is a complex task and current state-of-the-art approaches have not found a successful solution yet. During the development process we determined that the only way of extracting useful concrete information is setting up specific filters to limit the search results to match a certain goal. Filter granularity depends on the data source (Twitter in this case), but also both on the granularity of the selected use case and on the main scenario planned for the tool. For example, defining a time-frame, e.g., Christmas season, to extract information of a specific region, e.g., the UK, and for a certain topic, e.g., "O2" (the mobile operator of Telefónica in the UK).

Because of these changes, the performed analysis and the graphical representation of its results should be limited to certain variables in order to make them useful for the target audience.

The initial design of the interface was based on previous experiences in other works. However, in this case, the difficulty of representing the diversity in an easily understandable and self-explanatory way, useful for the community for taking strategic decisions made it necessary to limit the search parameters and filters to

concrete interests; i.e. it turned out to be a better strategy for the graphical representations to exhibit less parameters, resulting in sufficiently rich graphics that were still easy to read. When handling massive data, the user has to work with filters able to classify the information in a few groups to be able to produce easily understandable results. However, the groups obtained from the filtering process must represent features that can be useful to the user.

3.3 Chosen Technologies & Way of Development

For the tool implementation, state-of-the-art technologies have been selected, because they are widely used and highly compatible with the most popular browsers. These technologies are mainly jQuery and graphical libraries. In order to have good graphical features, a jQuery-ui library was included. It is a component of the jQuery framework and the natural solution for optimum graphical interfaces because of its full compatibility with the most important libraries integrated in the tool. We included the latest available stable release (*jquery-ui.1.10.2.js*), which uses a custom CSS3 template generated for T-OMT with RENDER project color set. Taking into account the feedback received in the initial evaluation of the tool, where many users expressed the necessity of having also a text version of the reports, *jquery.fileDownload.js* was included to take care of this, making it possible also the downloading of this text report onto the local user machines.

An intermediate version of Telefónica Opinion Mining Tool (T-OMT) used Google Chrome extensions. This, however, did not prove a good solution, as the software became dependent on possible changes in extensions out of our control, done periodically by Google without backward compatibility. Therefore, when developing this kind of diversifying tools, it is strongly recommended not to link the tools to external software going beyond the developer's own control.

Initially, we placed the data processing solely on the client side. We however quickly realized that the processing capacity of the potential user equipment proved insufficient in face of the heterogeneous and sizable twitter data, and thus made it necessary to create a backend service to take care of the highly CPU-demanding tasks of data mining, processing and classification.

Taking into account the opinion from the target audience, the functionalities, priorities and interface design several iterations were completed with different intermediate versions. Finally, the selected graphics and essential information to be displayed were chosen. A detailed description of the selected graphs can be found in "Deliverable D5.3.5 - Final version of diversity information extensions for Telefónica tools".

We can conclude that for these kind of tools it is better to start with a basic functionality and incrementally add features depending on the user's iterative feedbacks about whether the tool fulfils all his needs or not. Following this approach, at the end, the final version reached should perfectly match the user needs of what dimensions of "diversity" he prefers over others. It is important to not from the start overload the tool with too many dimensions that are not relevant and would merely be confusing.

The tool has been fully developed internally at TID. However, different knowledge groups of the company have participated and given advice for the development of the tool. The participating units have been the following:

- Software technology experts and architects, ensuring the best practices for software development. This group should be present through all the project lifecycle.
- User experience (UX) team for user testing and evaluation. The implication of UX team is recommended at specific milestones, in this case, when the intermediate versions should be evaluated.
- Quality assurance (QA) unit for software testing and deployment features. QA team presence in the project is recommended for the same situations than the mentioned for UX.

After the designing of the application and the preparation of mock-ups, scrum-agile based methodologies have been used, as the most adequate to determine which type of processing and data-mining is the most appropriate for obtaining valuable results needed by the future users. In this way, intermediate working

versions were made available and several loops involving software architects, designers, developers and users were performed before the final version was released.

The data representation must be compliant with the most suitable format according to the target user group preferences, which usually is a format familiar to them. The reason for this is that the acceptance of the tool will be much higher if the user does not need to deal with new formats and therefore he will be more able to concentrate in the final goal of the tool and not in minor details as data formats or learning how to interact with a complicated tool. In summary, make user's life as easy as possible. A constant upgrade and adaptation of the interfaces and their contents is mandatory given the multitude of possible ways to represent that kind of data.

3.4 Approach to Communities

Since the very beginning, when the application mock-up was done, the users were involved in order to adapt the application design. Once the tool integration was completed and real-data were used, the most strategic users were incorporated into the different iterations to reach a solution able to fulfil all their needs.

When designing a product, it is a good practice to involve users from the very beginning. Although for projects following a similar approach as our use case (mock-ups) this might sometimes not be possible in a formal evaluation or feedback-loop for whatever reason, we would recommend to involve users some other way, i.e. at least having personal unstructured interviews with some representative possible users, especially for a topic like data "diversity", where the specific implementation of the concept may vary highly with the interpretation and needs of the particular use case stakeholders.

It is also important to remark that for guaranteeing success, not only a good product is needed, but the product has as well to be made attractive to the users and gain their acceptance. This does not only hold true in an open market scenario but also in an intra-company set-up. The software was showed to key members of strategic business units of Telefónica, a group being very heterogeneous in nature. The presentation of the software to users instigated great interest, particularly for certain business units that needed a statistical processing of huge amounts of data and for those that wanted to know the global opinion about different products of the company expressed freely by their users. It is also important to mention that some users considered the possibility of extending the analysis to similar products done by our competitors as very important.

Essentially, the generated report design has taken into account the users vision since the very beginning. Finally, when incorporating the future heavy users in the detailed evaluation, these articulated a new need of having reports in a written, more detailed format to be able to perform a more in-depth analysis of the data and to incorporate that analysis into reports. Because of that we had to adapt our implementation, new functionality and additional charts were necessary (such as the sentiment geomap) to offer a broader scope from the analysis of the users opinions.

The mining tool is designed in a modular way, making it easy to add new functionalities in an encapsulated and coherent way. Figure 4 shows the T-OMT architecture. There are three main blocks: the frontend, the T-OMT backend and the data layer (FactForge). This architecture is detailed in Deliverable D5.3.5 "Final version of diversity information extensions for Telefónica Tools" The modular approach proved to be very appropriate, as it allowed us to adapt the tool more easily to emerging needs for changes.

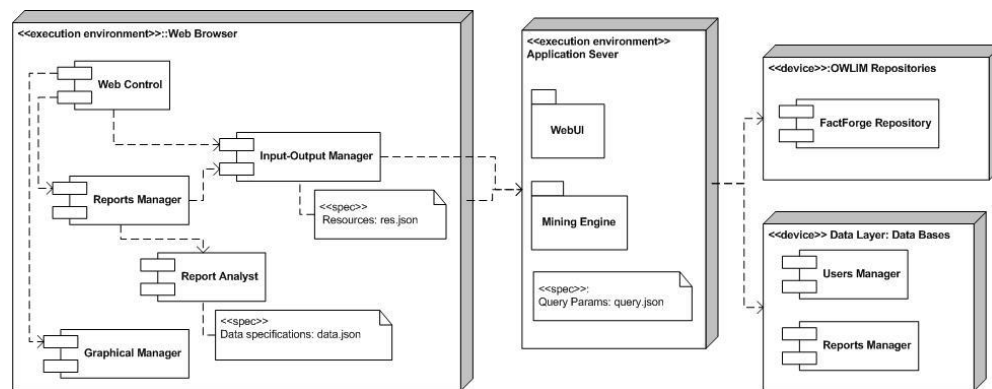


Figure 4: T-OMT Architecture

3.5 Integrating the Technology with Existing Services, Platforms and Technologies

In order to fulfil the ambitious goal of processing, classifying and summarizing the opinions from millions of users about different products and also to create statistics about their overall acceptance in the market, the T-OMT application must use the powerful tools available in RENDER framework:

The fundamentals of T-OMT application are the ability of the Enrycher software developed by JSI to obtain a ‘sentiment analysis’ of each published comment together with an accuracy score. In order to manage the complexity and diversity of opinions and the different ways of expressing them, it is mandatory training a specific model to make it possible the proper detection of the sentiment according to the defined goals. In this way, two essential factors of the unstructured opinions are taken into account (i.e. through open forms) together with the diversity of factor able to be statistically processed (like geo-positioning, time-frames ...). The collected data are massive, so for properly processing them, it turned out to be crucial to have suitable tools for the specific querying of these data type through different criteria. Therefore, the Ontotext repository solution was extremely useful to access and manage the data using different criteria in order to obtain results in a reasonable time in a distributed system. The same holds for the Enrycher software developed by JSI, where the model used to detect the sentiment was essential to focus the diversity processin; in the case of integrating the Ontotext repository, the construction of adequate queries was critical in order to speed up the data extraction, minimizing the required time for that task. A technique that has produced good results for that goal is splitting a query in groups of simpler queries to be processed in parallel, reducing the response time drastically and optimizing the available resources, both on the data repository side and in the statistical processing engine in charge of generating the reports. In this concrete scenario, a query is performed by each topic, day, sentiment polarity (either positive or negative) and country. In a second step, an integration of all individually obtained results must be done in order to generate a single report fulfilling all the requests from the user for the report generation.

Given the data diversity and the possible interpretations, T-OMT is a solution that tries to provide a report using different perspectives to march all possible users’ needs. However, the TwiDiViz tool offers the possibility of handling, ordering or classifying the data independently from the statistical processing of them performed by T-OMT backend. The visualization and interfacing is so radically different that the integration between the two systems is only made using the search criteria as the only common parameters shared by the two interfaces. To keep the data interface that simple turned out to be the most viable solution to switch between these too interfaces, as keeping the TwiDiViz application basically stand-alone and not integrating it to deep with the T-OMT allows to adapt it to further use cases beyond RENDER.

3.6 Type of Usage by the Community

The graphical (and textual) analysis performed by T-OMT provides its users with a powerful tool to make important business decisions.

In this case, the users would mainly be business people in charge of launching new products or tracking the situation of a product in the market to decide strategies about them. The main interesting uses for the testers were the following:

- Analysis of the situation of a product over time, in order to adapt the marketing strategies and react properly when a change of tendency was detected. For example, a new product can have a great interest at the beginning, but over the time, the interest is declining, so new marketing campaigns should be launched.
- Detection of problems in a particular country. In this case, a global strategy may not work and particular decisions should be adopted in that country. In the case of Telefónica, present in 25 countries, some countries of Latin America present peculiarities and for example, demand a more personal interaction with the customers.
- Detecting when the sentiment about a product is about to change. Decisions can be taken to solve the situation before the sentiment became in a negative one.
- Evaluating the impact of products of our competitors and comparing their impact with the impact of our own products.
- Detecting period of time with low activity, i.e. weekends, night/day. This could help to adapt the strategies to different periods of the day/week.

Apart from the mentioned uses, much more analyses can be performed, not only with the available graphs, but also interpreting the details given in the detailed textual report.

In addition to the user evaluation, some of the participating users took the opportunity of using the tool in a non-controlled way afterwards. In this way, preferred interaction ways could be detected and some conclusion derived:

- Certain users first create several reports simultaneously, trying different combinations of parameters in the entry filters, with little variations among them. They visualize the different obtained reports and compare their results.
- Following the procedure detailed above, some users, once they have obtained a successful combination of parameters, i.e. screening several dates until finding an appropriate range, they create a new report and save it into the database.
- Some users try to join information expressed in different graphical representations, then they, for example, visualize alternatively the mention and sentiment geocharts.
- Another users download the report into textual form and manually cut and paste some data into a spreadsheet to perform further analysis over them.

Some feedback we got from users of the tool was the response of a key decision-maker person in the Financial Services Unit from Telefónica, stating that “Thanks to this tool, I will be able to quickly analyze the acceptance of a recently launched product and quickly react to a possible unexpected evolution.”.

4 Wikipedia Tools

4.1 The Concept of Diversity and its Interpretation and Implementation in the use Case

We learned early that showing Wikipedians a simple measure of how “diverse” or “non-diverse” an article is turned out to be non-practical in this use case for different reasons: Firstly, it was impossible to define uniformly, accepted by all Wikipedians, what “diverse” actually means in the context of an article and how to accurately measure it. Secondly, the acceptance towards tools that do not address current, practical needs of the editors is hard to reach. Out of these preconditions we derived two aspects of diversity that are relevant to this use case and we were able to implement as day-to-day working tools for editors. One is to help each editor to form an assessment of how diverse an article is in her own interpretation and what is actually missing or overrepresented in it by showing a combination of simpler content and social metrics and inter-language comparisons, in order to give ideas what can be added or removed; and, in an extension to this, guide editors to articles that exhibit certain lacks in the first place. The other aspect is – a general goal of Wikimedia Deutschland e. V. – to attract readers to edit Wikipedia, to actually tap into their unused and, as known by studies, more diverse knowledge than what is represented by the minor percentage of users that edit nowadays. By summarizing different metrics about an article or giving suggestions about what may be missing or out-dated the idea is to keep raising the number of active authors contributing to Wikipedia. By adapting the “knowledge diversity” concept to the real needs of users we eventually achieved a bigger impact.

4.2 Design of Interfaces and Core Functionality

The integration process of new extensions into the productive system of Wikipedia is very complex and time consuming. It may only be sped up for tools that are highly accepted by a community or provide a huge benefit for all language versions or all projects within the Wikimedia environment. Besides, the introduction of new tools requires being very careful and at the same time respectful towards the Wikipedia community in general. In the past, several new functionalities were introduced into Wikipedia without involving the user communities. In some cases these tools had to be deactivated again because of raising highly controversial issues. So we decided to give the users a chance to test our tools and participate in the development process by giving feedback.

The aim of the Article Monitor is to give readers an overview of the currently viewed article's status and provide suggestions about what may be in need of being changed or updated. Diversity aspects are taken into account regarding the huge author community as well as considering different topical coverage throughout all language versions of an article. To make the tool visible close to the article's content, we decided to add a tab in the article's toolbar right above the article itself. We also did not want to draw the user away from the article's content, what led us to displaying a layer on top of the article. In that way users are able to access the information given by our tool with one click, but still see the article as the central element of the page.

The Article List Generator also respects the diversity of editors. While photographers are enabled to find articles lacking illustrations, experts can search their areas of expertise for articles that may need revision or completion. In this approach there is no need to bind its functionality close to an article. We decided to provide an application that is not integrated into Wikipedia's underlying MediaWiki software for the reasons mentioned above. Some of the filters included in the application require a lot of computational resources, the user may need to wait for minutes until the results are being put out - depending on the search depth or the complexity of the filters being chosen. For that reason we decided to build a JavaScript frontend application that is able to keep the user informed about the current status of the request.

Some people might argue that the diversity of content throughout language versions is vital to the project and depends on the relevance for readers in certain language versions. Articles about small towns in Denmark are more detailed in Danish than they are in Russian. But comparing different language versions

can help to complete information. Bringing different language versions closer together can also help to have articles being updated faster. The work of an expert in environmental technology editing the Danish Wikipedia for instance can very well be transferred into other language versions. WikiData¹ is a good example of that. Certain facts about a topic are stored in a centralized database. Once updated there it is distributed to all language versions.

The input from community members was very important for our work. The idea to implement a list generator tool which creates article lists was derived from the German community. The editorial for Biology has been tool-created lists of articles requiring maintenance for years. These tools were only used by authors with higher user rights. We decided to create a tool to be used by everybody without being part of a specific group of users. While some community members had problems understanding the purpose of some tools, others did not understand how to use them right away. We then simplified the UI, wrote detailed manuals, and implemented further explanations into UI components - all according to the feedback we collected.

While testing the Article List Generator we realized that we didn't provide a convenient way to enter categories to be crawled. At that time, the category names had to be entered into a simple input field. A spelling error in a category name led to empty results. So we decided to add an auto-completion feature.

4.3 Chosen Technologies & Way of Development

To fully integrate the tools into Wikipedia's underlying MediaWiki software we needed to develop the tools using technologies that are available in the MediaWiki software. To ensure best performance in highly available systems some of the (completely independent) backend software was written using other technologies, such as Python or C++. In terms of addressing the developer community to further develop the tools we realized that the personal preference of used technologies differs so much that it is not possible to meet everyone's favoured one. During application design we ensured that other developers could easily improve the tools.

We were aware that there would be a more convenient way to include the Article Monitor gadget aside from copying the complete code snippet to the user page, as we eventually did; namely offering the tool as an online service that would need no installation on the client side whatsoever would have been more convenient for the users. Moreover, having users install the complete code rather than embedding it from a central location makes it hard to maintain the gadget; versions of the gadget cannot be distributed easily, thus a version fragmentation of the code is likely. However, in order to be able to log and analyse the tool usage, as was needed to do for the evaluation, we needed to assign a unique ID that identifies each installation; this could only be achieved by local installation on the user profiles. Although the community was understanding in regard to this, we would recommend not to deploy in this way in future work as it limits the potential users of the tools. A viable alternative is for example to use the ResourceLoader provided by the MediaWiki software to keep the gadget code at one central location.

Although there were no developers directly contributing to the project, they were quite interested in participating in other ways. We set up contact forms to enable users to send us feedback or bug reports. When announcing the tools and the publishing of new tool versions on Wikipedia or the Meta-Wiki we always encouraged the users to make use of the discussion functionalities that are built into the MediaWiki software. The possibility to discuss feature requests, bug reports, or general opinions about the tools has generally produced more and better outcome than one-to-one communication.

4.4 Integrating the Technology with Existing Services, Platforms and Technologies

Although highly depending on the Wikimedia Toolserver or Labs infrastructure for running background services, the tools are designed as stand-alone applications. As mentioned earlier, we decided to use this

¹ <http://www.wikidata.org>, see D4.1.3

approach to have prototypes of the tools ready to show and to be usable rather than undergoing the time consuming and complex processes of integration into the productive Wikipedia systems. This has been proven a useful approach to begin early testing and iterations in development and has enabled us to reach the set milestones in RENDER in time. We plan to propose an integration of the tools into the productive systems after RENDER has finished, given a sufficient demand by the community.

4.5 Approach to Communities

Before going public with the Supporting Tools we wanted them to be at a presentable and already usable stage. In doing so, we were already able to show what the outcome of our ideas would be. As mentioned above an earlier involvement of the users may have led to more participation in general. Some users refused to use the tools simply because they were not able to contribute ideas to the project and didn't feel that the tools were aimed for them.

We always considered the results of the analyses provided by our tools to be suggestions to the users. There are a lot of users that manually assess the quality of articles and add corresponding maintenance templates to an article. The analysis approaches used in our tools were always meant to be a supplement to human decisions. To advertise them as such to the community and at the same time ask for input for improvement was an important success factor to gain acceptance. This might be different for platforms in enterprises or with paying end-users, but is definitely a very appropriate approach in open systems with a strong community like Wikipedia.

After publishing the tools we used many different communication channels to spread information about them. We also visited regional community members in different German cities and took part in several events to reach out to the community. Reaching out personally to the community members after publishing the prototypes and the final tool versions helped considerably to familiarize people with the tools and to communicate our intents to them.

After introducing the first showcase of the tools in the RENDER Toolkit we experienced a variety of reactions. We published information about the toolkit on mailing lists and on different central locations within Wikipedia and the Meta-Wiki. While some users showed a lot of interest, asked questions about the tools, wanted the Wikipedia Map being updated and available in more language versions, others were unfriendly or just ignored the toolkit. It led to a discussion about benefits, about advantages and disadvantages of research projects concerning Wikipedia in general. All that led to the conclusion that we should have involved the desired user groups earlier. We presented the first insights to our software developments within the RENDER Toolkit on the Wikimedia Toolserver.

The most important lesson we learned is the involvement of the community at a very early stage. Different authors have different requirements to a tool. If members of the community you are developing a tool for feel that their suggestions find attentive ears, they are generally open-minded about new things. Letting them participate actively in what the outcome will be, the acceptance is much higher. Community members - such as authors, photographers, oversighters, and also readers - are certainly qualified to provide input regarding tools that affect the usage or the daily work in Wikipedia. Respecting that, there is a high chance that tools are being used frequently, enhanced or improved by other developers, even to more deeply integrate them into the Wikipedia and have them become an inherent part of the MediaWiki software itself. Although involving too many users in a very early stage may well lead to problems in coordinating and agreeing on features of early prototypes, this trade-off seems necessary in order to reach a sufficient acceptance level of editors. And acceptance by a diverse group of stakeholders might be even more important in a case where software tools are meant to provide additional user and content diversity to a system.

4.6 Type of Usage by the Community

The tools are used to find articles which may need to be improved. The users create article lists to further work on Wikipedia and edit or add content. The Article List Generator supports the work of the authors in particular. They are able to create lists of articles according to their preferences and specific filters.

People use the Article Monitor to get more information about an article and to check if there are improvements suggested. They look up the available results for the article they started or contributed to. Some users find statistics or information they couldn't access before. All in all, the tools are used as envisioned to the way they were conceptualized, except some adjustments that need to be applied to specific tools, as became apparent in the evaluation.

We are in contact with some community developers who think about merging their tools/technologies with our supporting tools. This topic is very relevant to us, since the Wikimedia Toolserver will be replaced by the Wikimedia Labs and some Toolserver developers are not willing or do not have the time to move their tools which deal with similar analysis approaches or further tools which are related to information diversity. It also provides a unique chance to unify many, so far isolated tools in one central interface and thereby leverage their oftentimes very useful functions that are not "marketed" very much throughout the community.

During the several community events that followed the publication of the tools' final version we talked to interested users. Some of them were using the tools since their early beta release, others just got to know about them. The overall reception was very good, especially regarding the Article List Generator. The user M. from Munich stated that "The Article List Generator is absolutely great. A tool like this has been long-overdue". Photographer G. added that "having a tool for pointing out articles that don't contain any photos is a big help. I'm sure my fellow photographers will feel the same about that tool". User E. from Potsdam described the Article Monitor as "a big help to have all the statistics about an article at a glance. I'm glad about suggestions showing me where to start on editing an article".

5 DiversiNews

This section describes the lessons learned from the DiversiNews use case.

5.1 The Concept of Diversity and its Interpretation and Implementation in the Use Case

This use case deals with worldwide news articles/reports. Those are conceptually rich pieces of text that offer many different sources for diversity, stemming from the topic itself (what do articles write about), the metadata (who/when/where writes them) as well as the subtler aspects (what affect does the author hold towards the subject matter; what is the style of writing; how complex is the language; etc).

In the end, the aspects of diversity that we focused on were chosen primarily based on two criteria:

- 1) How well are we able to extract that attribute / aspect of diversity automatically?
- 2) How relevant and how easily understandable do we expect that aspect of diversity to be to the end user?

The design approach was hence primarily driven by the inherent, diverse dimensions of the data that we expected to be of interest and of use to readers. We settled for the following facets of diversity: topics/subtopics (keyword-based), place of origin (e.g. local news vs. international reports on the same topic) and author sentiment (positive vs. negative).

5.2 Design of Interfaces

DiversiNews allows the user to control for several types of diversity; we chose a custom interface for each of them. In increasing order of complexity, they are 1) a one-dimensional slider for sentiment/polarity, 2) a world map for exploring different news sources and 3) a topic map for exploring different content subtopics. The more complex the interface, the more problems users have using it and the more they shy away from it. “Complexity” is in fact relative; it is more the familiarity of an interface than its actual complexity or expressive power that determines how unfriendly the tool seems to users.

We developed the notion of diversity and the tools in parallel, as we were growing more familiar with the data and with the dimensions along which diversity could be identified. The most important insight might have been the following: An elaborate definition of diversity is not practical if we’re unable to capture it with tools and interfaces and if users cannot operate with it once it is put into an actual tool. Similarly, overly simplistic tools would not allow us to cover enough aspects of diversity.

The design was influenced by our own experience in news visualization, both as end users who sometimes wish for additional functionality in existing interfaces, as well as from our previous project collaborations with news agencies.

The results of DiversiNews User Experience (UX) evaluation (See D5.2.4) confirm that the adopted design solution are familiar enough to be self-explanatory and to ingenerate in the user correct expectations about each component’s behaviour. At the same time, they are powerful enough to let diversity emerge and to immerse the users in a pleasant and innovative news browsing experience.

5.3 Chosen Technologies & Way of Development

The main analytics part is based on our preexisting software stack, which performs preprocessing for most of the components. The stack is written in C++ and Java; some modules are written from scratch and some are wrappers around open-source libraries. The fact that the software stack brings together numerous analyses means that it also extracts many different types of attributes from text, enabling us to explore diversity along various dimensions.

We stuck with the chosen technology and interfaces because we believe they are a viable solution to the problem. Because they are solving a novel problem in a novel way, they were invariable met with some confusion among new users. This caused us to make minor adjustments to the interfaces that do not alter

their functionality but rather try to make their usage more apparent and intuitive. Concretely, we tweaked colors and shapes of controls, font sizes, added tooltips etc.

With diversity-centric software as with all novel user interfaces, it paid off to frequently check with new end-users and make sure they intuitively understand what the tools do, so that it was defined from the beginning what aspects of diversity should be mined and analysed by the backend services and at the same time displayed to users in a meaningful way by the UI components.

5.4 Integrating the Technology with Existing Services, Platforms and Technologies

Compatibility with other partners' software was, on the technical level, achieved by keeping the modules as loosely coupled as possible and APIs simple. This decision served us very well and saved many hours of negotiations and bug hunting. On the conceptual level, the different modules meshed very well: Ontotext's related entities service and Google's summarizer need the type of data the preprocessing pipeline is able to provide, and at the same time provide core functionality to the web service as a whole with their outputs. This can be largely attributed to the long discussions about what diversity is and what aspects of it we are able to capture.

5.5 Approach to Communities

DiversiNews is a technological preview (demo application) and as such it does not really insist on a community of users or developers. On the other hand, we carried out an extensive evaluation (see D5.2.4) in which we measured the technological adequacy of the platform, its ergonomics from a user experience point of view and the level of satisfaction of DiversiNews users. The evaluation shows a very high level of satisfaction of the users with the platform. We received very positive feedback from casual and professional users concerning both the originality of the paradigm and its practical utility as a means to let diversity in news emerge, and we collected extremely valuable feedback to further refine the demo and exploit its full potential.

5.6 Type of Usage by the Community

The User Experience evaluation has involved both casual and professional recipients of internet news. Both classes of users have expressed praise for concept and the implementation of the platform. Users find that the objective of stressing diversity in news is worth pursuing, and appreciated the summary-based interface which has the potential of letting diversity stand out more clearly. They have a practical interest in using a news browsing tool in which they are in control of the criteria by which the news are organized and presented, and appreciate the fact that the original sources are always available as this factor increases the trustworthiness of the results. Based on the collected feedback, we expect that the deployment of a platform based on DiversiNews experience would be a very successful initiative both within news professionals and lay people reading news for personal culture and pleasure.

6 Conclusions

We reported on a number of lessons learned and best practices for conceptualizing, developing and deploying tools for knowledge diversity, including best practices how to approach various communities. To sum up, we can outline the following most important best practices and findings of putting these kinds of tools into application that emerged as a communality of the presented use cases:

- When introducing the concept of knowledge diversity and its tangible implementation as a software tool to a community of either end-users or developers, it is important to find the right trade-off: On the one hand it might be harmful addressing the community with an immature product, confusing users (as there might be too many dimensions of diversity involved or the concept as such is not understood) and/or watering down the initial concept of the use case too much (because of large number of users that want to give input). On the other hand by not involving enough users early enough one might lose acceptance in the community and/or lose touch to what is important to them. Generally we tried to make contact to users early in the process, but mostly with some preconceived notion of how we wanted knowledge diversity to mean in the respective context. This could have been done earlier in some cases.
- In a similar vein, “knowledge diversity” as a concept is particularly hard to define ex-ante, outside real use and data and cannot be imposed on end-users. It is highly subjective to interpretation by different stakeholders (developers, users and also internally by different partners and developers) and will therefore always be subject to constant renegotiation and iterations. We highly recommend personal, informal contact to end-users even before any planned evaluation or testing starts to learn different points of view on the topic and get early “grounding” to the real use scenarios.
- Exploring knowledge diversity is actually a very popular theme with users. Although the testers might have not rated some tools as optimal during evaluation, the general goal of these tools and how they either try to generate more knowledge diversity or explore it was received very positively. Especially the possibility to navigate contents exhibiting different sentiments towards a common topic was highlighted as very valuable.
- Showing the diversity of different sources of knowledge and different viewpoints seems, on average, to increase the trustworthiness of the shown content. Users put – generally – more trust into the source if it discloses related, even non-agreeing content.

With these insights presented we hope to give assistance to future developers of tools for knowledge diversity and show them which guidelines may be helpful for them and which methods and approaches should better be avoided.