

# Visualizing Provenance using Comics

Andreas Schreiber

German Aerospace Center (DLR)  
andreas.schreiber@dlr.de

Regina Struminski

University of Applied Sciences Düsseldorf  
regina.struminski@study.hs-duesseldorf.de

## Abstract

Understanding how a piece of data was produced, where it was stored, and by whom it was accessed, is crucial information in many processes. To understand the trace of data, the provenance of that data can be recorded and analyzed. But it is sometimes hard to understand this provenance information, especially for people who are not familiar with software or computer science. To close this gap, we present a visualization technique for data provenance using comics strips. Each strip of the comic represents an activity of the provenance graph, for example, using an app, storing or retrieving data on a cloud service, or generating a diagram. The comic strips are generated automatically using recorded provenance graphs. These *provenance comics* are intended to enable people to understand the provenance of their data and realize crucial points more easily.

**Keywords** provenance, visualization, comics, self-tracking

## 1. Introduction

In many applications, it is necessary to understand how a piece of data has been produced or processed by the underlying process. Insights into the data processing are important for gaining trust in the data; for example, trust in its quality, its integrity, or trust that it has not unwantedly been accessed by organizations. Especially, detecting and investigating privacy violations of personal data is a relevant issue for many people and companies.

For example, the integrity and privacy of personal health data is crucial. Health-related data should not be manipulated, if doctors base a medical diagnosis on that data. Personal data from self-tracking should not be available to other people or companies, as this might lead to commercial exploitation or even disadvantages for people. Such data is often generated by medical sensors or wearable devices, then

processed and transmitted by smartphone and desktop applications, and finally stored and analyzed using services. Following the trace of data through the various distributed devices, apps, and services is not easy. Especially, people who are not familiar with software or computer science are often not able to understand where their data is stored and accessed.

To understand the trace of data, the provenance of that data can be recorded and analyzed. Provenance information is represented by a directed acyclic property graph, which is recorded during generation, manipulation, and transmission of data. The provenance can be analyzed using a variety of graph analytics and visualization methods [10]. Presenting provenance to non-experts is an ongoing research topic (“*Provenance for people*”). As a new presentation and visualization technique for provenance, we introduce *provenance comics*:

- We explain the general idea of *provenance comics* for visualizing provenance documents as a set of comic strips representing single activities (Sect. 2).
- As a use case to show the feasibility, we describe a specific visual mapping between the provenance of self-tracking data [15, 16] and their graphical representations in comics (Sect. 3).
- We briefly describe our prototype for *automatically generating provenance comics* from provenance documents compliant with the PROV standard [12] (Sect. 4).

## 2. Provenance Comics

The basic idea of *provenance comics* is to present the provenance information of data processes in a visual representation that people can understand without prior instruction or training. A general advantage of comics over conventional visualizations, such as node-link diagrams, is their familiarity: Almost anyone has probably seen some comics in their life. No training is required to read them, and they can transport meaning with minimal textual annotation. They are easy to interpret and not as strenuous to read as, for example, a graph or a long paragraph of continuous text.

Data provenance has a temporal aspect: origin, manipulation, transformation, and other activities happen sequentially over time. The directed, acyclic provenance graph guaran-



**Figure 1.** Example: Comic depicting manual input of blood pressure values.

tees that, while moving through its nodes, one always moves linearly forward or backward in time. It is therefore possible to derive a temporal sequence of happenings from the graph that can be narrated like a story.

We generate a comic strip for each basic activity in the provenance data. Each strip consists of a varying number of panels, which are small drawings that provide further details about the activity (see Fig. 1 as an example for a single comic strip). The comic strip for the earliest activity in the provenance document is at the top, while the strip for the newest, most recent activity is at the bottom. The complete set of comic strips shows the “story” of the data. Of course, when there are many activities, the collection of comic strips could become quite large. In this case, one could choose a subset of the provenance, containing only those activities that are relevant in real use cases.

Some questions that the provenance comics should answer and explain are *When was data generated or changed?*, *What software tools have been used?*, or *Where was the user’s data stored?* At this time, the comics do not contain the actual data. They only visualize information contained in the provenance of the user’s data. This might be extended in the future by using (parts of the) data for representing the real measurements, geographical coordinates, etc.

## 2.1 Visual Mapping

To generate the provenance comics for a certain use case, one has to define a visual mapping that specifies how elements of the provenance are rendered within the comics. For evaluation, we defined a consistent visual language for the use case of self-tracking [18] (see Sect. 3). Such a visual language allows to “translate” the provenance data into corresponding drawings. Generally speaking, the elements of the PROV standard (*Entity*, *Activity*, *Agent*) maps onto three distinctive features: *shapes*, *colors*, and *icons or labels*.

**Shapes** The Shapes represent PROV *Agents* and *Entities* with easy-to-recognize and common pictograms. *Agents* of type *Person* are represented by a human silhouette, *Agents* of type *SoftwareAgent* by symbols for smartphones or computers, and *Agents* of type *Organization* by office buildings. *Entities* are represented by “standard” pictograms such as folder icons or document icons.

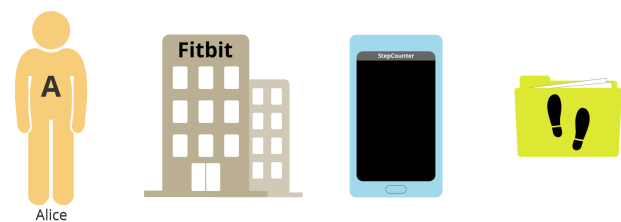
We designed and selected shapes according to several criteria. Most importantly, the shapes do not show much detail or “fancy” decorations, which could cause distraction. Instead, they have a “flat” look without any textures, ornamental decorations, shadows, or three-dimensional elements.

**Colors** Different colors distinguish the *Entities* as well as the different types of *Agents*. For example, per default *Persons* use a light orange color, while *SoftwareAgents* have a light blue and *Organization* agents a tan color. *Entities* are always colored in a bright yellowy green. The selected colors ensures that similar elements always use similar color hues, yet remain well distinguishable even by people suffering from different kinds of color-blindness (Protanopia, Deuteranopia, Tritanopia, or Achromatopsia).






The distinctiveness between the colors of different object types is not as important as that between colors of the same types of objects. That is to say: Color is more important for distinguishing two items that have the same shape than it is for two items with different shapes.

**Icons, Letters, and Labels** All main actors (*Entities* or *Agents*) in the comics carry some kind of symbol on them, whether it be an icon, a single letter, or a whole word (Fig.2).

- Person agents wear the first letter of their name on the chest.
- Organization agents display their name at the top of the office building.
- SoftwareAgents show an application name on the screen.
- *Entities* are marked by an icon representing the type of data they contain.



**Figure 2.** Icons, letters, and labels for different shapes.

Element type	Shape	Example
<b>Agent</b> type: Person	human silhouette	
<b>Agent</b> type: SoftwareAgent	smartphone, computer, ... (depending on the agents "device" attribute)	
<b>Agent</b> type: Organization	office building	
<b>Entity</b>	file folder, document, chart, ... (depending on the entities type attribute)	
<b>Activity-related objects</b>	button, icon, ... (depending on the activity's name or "role" attribute)	

**Table 1.** Shapes defined for different types of PROV elements.

**Captions and text** Additional captions and text support PROV activities. Most of the information should be conveyed by the graphics to provide an effortless “reading” experience. However, in certain cases, a few words are useful to support the interpretation of symbols. For example, when uploading or downloading data, the words “Uploading...” or “Downloading...” are added below the cloud icon. These short annotations take only little cognitive capacity to read, but may greatly help understand certain icons.

### 3. Use Case: Self-Tracking

As a use case for evaluation and for performing a user study, we have chosen *self-tracking* (“Quantified Self” [5, 7]). In this field, data is often generated by medical sensors or wearable devices, then processed and transmitted by smartphone and desktop applications, and finally stored and analyzed using services (e.g., web or cloud services operated by commercial vendors). Integrity and privacy of this data is crucial, since it should not be available to other people or companies, as this might lead to commercial exploitation or even disadvantages for people.

For self-tracking, we use a provenance model that consists of several sub-models for basic activities [15], such as input data manually, get data from sensors, store, retrieve, or synchronize data with cloud or web services, visualize the data, etc. The provenance should be recorded automatically, which can be quite complicated—especially for legacy software—but many strategies and techniques have already been developed [16].

For the self-tracking activities, we defined a visual mapping [17]. Table 1 gives an overview of the shapes we selected to reflect the different types of elements in the Quantified-Self PROV model [15]. Fig. 3 shows an example of two comic strips that correspond to the provenance graph in Fig. 4. The example contains the consecutive strips for two user actions: *downloading steps count data from a cloud service to the user’s smart phone*, and *visualizing the steps data in a line chart*.

### 4. Implementation

For generating the comic strips, we developed the web application PROV COMICS in JavaScript [19] (Fig. 5). This web application fetches provenance documents directly from a provenance store. The current prototype supports ProvStore [9] using the ProvStore jQuery API<sup>1</sup> to retrieve documents from the ProvStore for a certain user.

Within the provenance document, the script first looks for activities to determine what kinds of panels need to be displayed. If there is more than one activity, the correct order can be derived from the activities’ timestamps.

As already mentioned earlier, activities will not be represented by a single graphic, but by a sequence of three to five comic panels. Similar activities should be illustrated by similar sets of panels, making use of recurring image compositions. For example, the activities *Export*, *Aggregate*, and

<sup>1</sup><https://provenance.ecs.soton.ac.uk/store/help/api/#jquery>

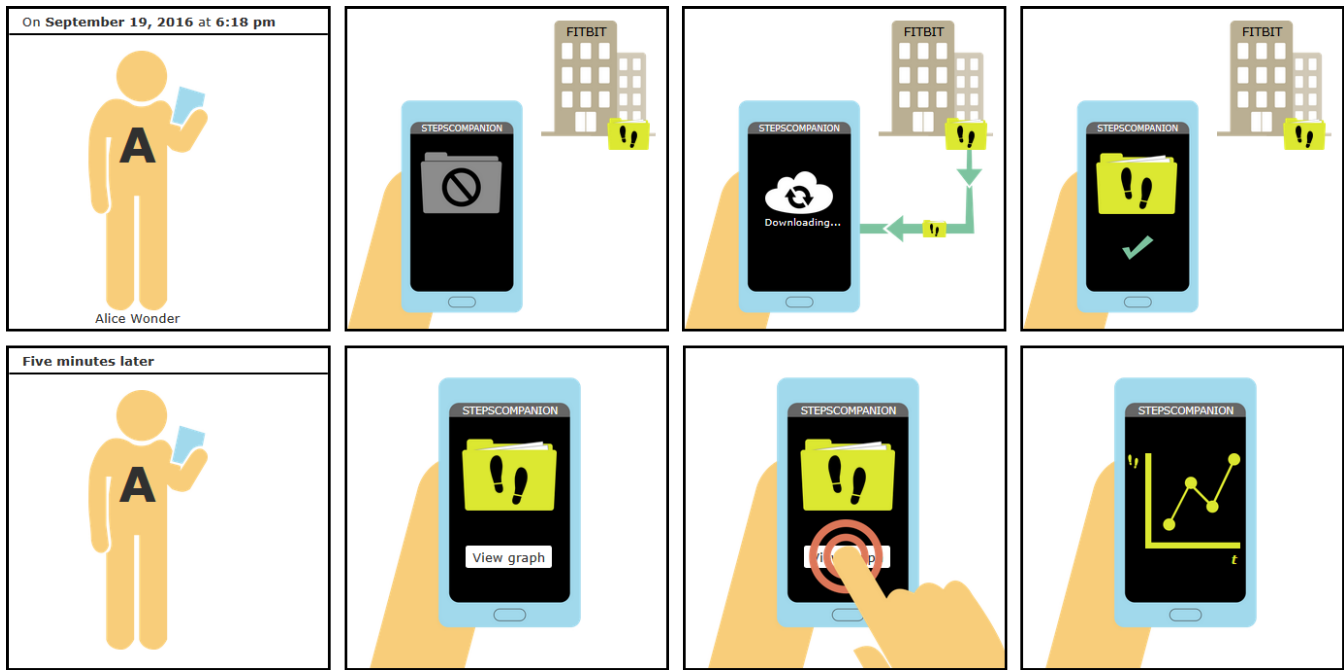


Figure 3. Generated provenance comics strip for two consecutive user actions.

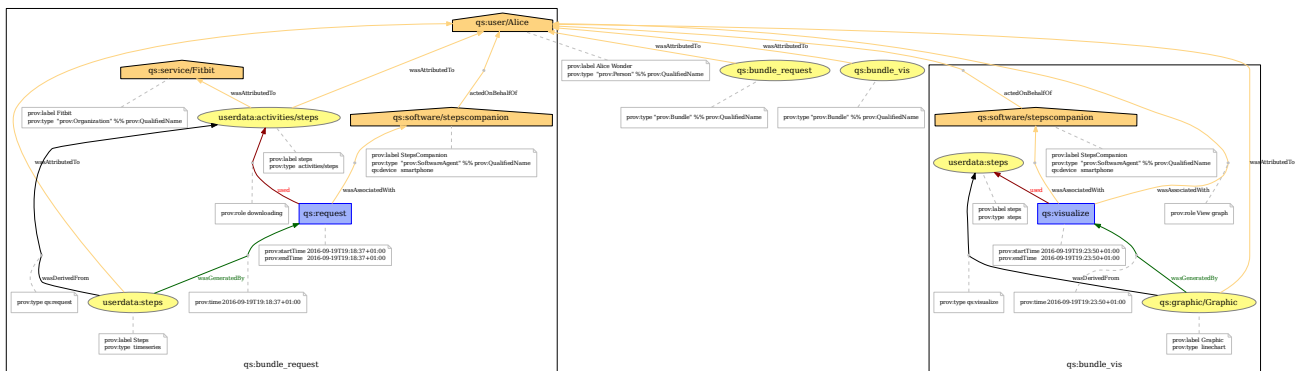


Figure 4. Provenance graph of two user actions (see <https://provenance.ecs.soton.ac.uk/store/documents/115642/>)

*Visualize* are comparable since they all take one kind of data and create a different kind of data from it.

After that, the script reads the attributes of involved agents, entities, and relations to decide which graphics to include in these panels. For example, the attributes indicate whether to display a smartphone or a computer, a folder or a single document, a steps icon or a weight icon, etc.

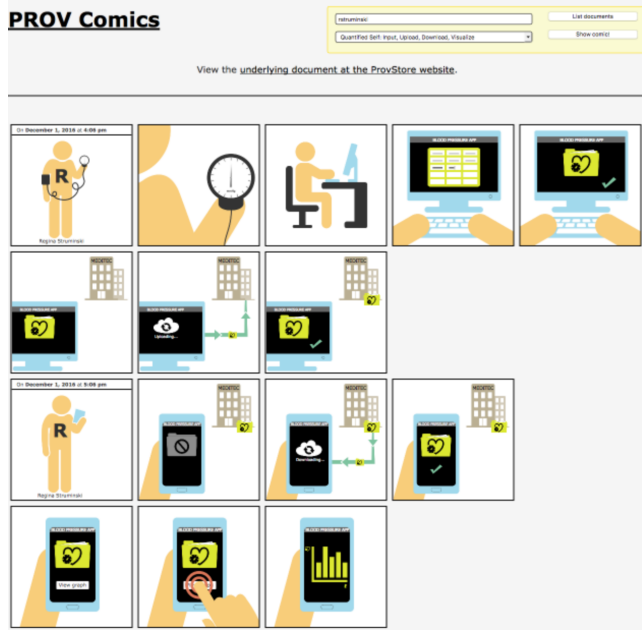
For generating the comics, the *ProvComics.js* script defines three JavaScript prototypes (“classes”):

**ProvComic** serves as a frame to contain all comic panels. It is also the general starting point for creating a *provenance Comic* inside a given HTML element. For example, if there is a `<div id="comic">` tag in the HTML, a new *provenance comic* may be started

within the `div` element by declaring `var comic = new ProvComic("#comic")`.

**Panel** represents a single comic panel and has all necessary abilities to create any of the panels described in the concept. For example, it provides functions to add captions, Persons, SoftwareAgents, Organizations, different types of entities, etc.

**PanelGroup** represents a predefined sequence of panels. They make it easier to insert recurring panel sequences. For example, it provides a function to add all panels depicting a download *Request* at once.



**Figure 5.** Screenshot of the PROV COMICS web application (<http://www.struminski.de/provcomics/>).

#### 4.1 Reading Study

Using the generated provenance comics, we conducted a reading study [17, 18]. The study, involving ten test readers, has shown that a non-expert audience is mostly able to understand the provenance of Quantified-Self data through provenance comics without any prior instruction or training. The overall percentage of 77% for findings verbalized by participants is deemed a good result, given that the checklists were very detailed and contained findings that some readers probably omitted, because they seemed too obvious and self-evident to them.

### 5. Related Work

For *provenance* visualization, most tools found in literature visualize provenance graphs using ordinary node-link diagrams, or tree representations similar to node-link diagrams. Provenance Map Orbiter [11], Provenance Browser [1], and Provenance Explorer [8] are based upon node-link diagrams. Large provenance graphs are then simplified by combining or collapsing sub-nodes or hiding nodes that are not of interest right now. The user can interactively explore the graph by expanding or zooming into these nodes.

Other tools, such as VisTrails [3], use a tree representation similar to node-link diagrams. Visual clutter is reduced by hiding certain nodes, limiting the depth of the tree, or displaying only the nodes that are related to the selected node.

Probe-It! [6] and Cytoscape [4] basically display provenance as ordinary graphs. However, Probe-It! does not only show the *provenance* of data, but also the *actual* data that resulted from process executions. In Cytoscape, users can cre-

ate their own visual styles, mapping certain data attributes onto visual properties like color, size, transparency, or font type.

There are some more related works, even though they are not directly concerned with provenance visualization. A non-visual approach to communicating provenance is natural language generation by Richardson and Moreau [14]. In this case, PROV documents are translated into complete English sentences.

Quite similar to provenance comics are *Graph Comics* by Bach et al. [2], which are used to visualize and communicate changes in dynamic networks using comic strips.

### 6. Conclusions and Future Work

The goal of this work was to develop a self-explaining, easy-to-understand visualization of data provenance that can be understood by non-expert end users.

We created a concept that defines a consistent visual language, which includes graphics for PROV elements such as different agents and entities. Sequences of comic panels represent different activities and the defined sequence of symbols, icons, and panel in an exact and uniform manner enables the automatic generation of comics.

As proof of concept, we developed a prototypical website, which is able to automatically generate comics from PROV documents compliant with the existing Quantified-Self data model. The documents are loaded from the ProvStore.

Future work will focus on graphical improvements. This includes suggested improvement measures that resulted from the reading study. Further future work will improve some technical challenges, such as adding flexibility for adding or selecting icons and images for PROV elements or dealing with uncomplete or very large provenance graphs. A major step will be quantitative comics, which also show actual measured values. For example, diagrams on depicted devices could show real plots of health data, and single comic panels may include real geographical information.

A useful improvement of the provenance comics would be to make them application-generic to some extent, (i.e., not restricted to the Quantified Self domain). We plan to explore whether provenance comics might be useful for other application domains, such as electronic laboratory notebooks [13] or writing news stories in journalism.

### References

- [1] M. K. Anand, S. Bowers, I. Altintas, and B. Ludäscher. Approaches for exploring and querying scientific workflow provenance graphs. In D. L. McGuinness, J. R. Michaelis, and L. Moreau, editors, *Provenance and Annotation of Data and Processes: Third International Provenance and Annotation Workshop, IPAW 2010, Troy, NY, USA, June 15-16, 2010. Revised Selected Papers*, pages 17–26, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-17819-1. doi: 10.1007/978-3-642-17819-1\_3.

- [2] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy, and N. Henry Riche. Telling stories about dynamic networks with graph comics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3670–3682, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858387.
- [3] L. Bavoil, S. P. Callahan, P. J. Crossno, J. Freire, and H. T. Vo. *VisTrails: enabling interactive multiple-view visualizations*, pages 135–142. IEEE, 2005. ISBN 0-7803-9462-3. doi: 10.1109/VISUAL.2005.1532788.
- [4] P. Chen, B. Plale, Y.-W. Cheah, D. Ghoshal, S. Jensen, and Y. Luo. Visualization of network data provenance. In *2012 19th International Conference on High Performance Computing*, pages 1–9, Dec 2012. doi: 10.1109/HiPC.2012.6507517.
- [5] E. K. Choe, N. B. Lee, B. Lee, W. Pratt, and J. A. Kientz. Understanding quantified-selfers’ practices in collecting and exploring personal data. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1143–1152. ACM, 2014. doi: 10.1145/2556288.2557372.
- [6] N. Del Rio and P. P. da Silva. *Probe-It! Visualization Support for Provenance*, pages 732–741. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-76856-2. doi: 10.1007/978-3-540-76856-2\_72.
- [7] M. B. Hoy. Personal activity trackers and the quantified self. *Med Ref Serv Q*, 35(1):94–100, 2016. ISSN 1540-9597 (Electronic), 0276-3869 (Linking). doi: 10.1080/02763869.2016.1117300.
- [8] J. Hunter and K. Cheung. Provenance explorer—a graphical interface for constructing scientific publication packages from provenance trails. *International Journal on Digital Libraries*, 7(1-2):99–107, 2007. ISSN 1432-5012. doi: 10.1007/s00799-007-0018-5.
- [9] T. D. Huynh and L. Moreau. ProvStore: A public provenance repository. In B. Ludäscher and B. Plale, editors, *Provenance and Annotation of Data and Processes: 5th International Provenance and Annotation Workshop, IPAW 2014, Cologne, Germany, June 9-13, 2014. Revised Selected Papers*, pages 275–277, Cham, 2015. Springer International Publishing. ISBN 978-3-319-16462-5. doi: 10.1007/978-3-319-16462-5\_32.
- [10] M. Kunde, H. Bergmeyer, and A. Schreiber. Requirements for a provenance visualization component. In J. Freire, D. Koop, and L. Moreau, editors, *Provenance and Annotation of Data and Processes: Second International Provenance and Annotation Workshop, IPAW 2008, Salt Lake City, UT, USA, June 17-18, 2008. Revised Selected Papers*, pages 241–252, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-89965-5. doi: 10.1007/978-3-540-89965-5\_25.
- [11] P. Macko and M. Seltzer. Provenance map orbiter: Interactive exploration of large provenance graphs. In *Proceedings of the 3rd Workshop on the Theory and Practice of Provenance (TaPP)*, USENIX Association, 2011.
- [12] L. Moreau, P. Missier, K. Belhajjame, R. B’Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, and C. Tilmes. PROV-DM: The PROV data model, 30 April 2013 2013. URL <http://www.w3.org/TR/2013/REC-prov-dm-20130430/>.
- [13] M. Ney, G. K. Kloss, and A. Schreiber. Using provenance to support good laboratory practice in grid environments. In Q. Liu, Q. Bai, S. Giugni, D. Williamson, and J. Taylor, editors, *Data Provenance and Data Management in eScience*, volume 426 of *Studies in Computational Intelligence*, pages 157–180. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-29930-8. doi: 10.1007/978-3-642-29931-5\_7.
- [14] D. P. Richardson and L. Moreau. Towards the domain agnostic generation of natural language explanations from provenance graphs for casual users. In M. Mattoso and B. Glavic, editors, *Provenance and Annotation of Data and Processes: 6th International Provenance and Annotation Workshop, IPAW 2016, McLean, VA, USA, June 7-8, 2016, Proceedings*, pages 95–106, Cham, 2016. Springer International Publishing. ISBN 978-3-319-40593-3. doi: 10.1007/978-3-319-40593-3\_8.
- [15] A. Schreiber. A provenance model for quantified self data. In M. Antona and C. Stephanidis, editors, *Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices: 10th International Conference, UAHCI 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings, Part I*, pages 382–393, Cham, 2016. Springer International Publishing. ISBN 978-3-319-40250-5. doi: 10.1007/978-3-319-40250-5\_37.
- [16] A. Schreiber and D. Seider. Towards provenance capturing of quantified self data. In M. Mattoso and B. Glavic, editors, *Provenance and Annotation of Data and Processes: 6th International Provenance and Annotation Workshop, IPAW 2016, McLean, VA, USA, June 7-8, 2016, Proceedings*, pages 218–221, Cham, 2016. Springer International Publishing. ISBN 978-3-319-40593-3. doi: 10.1007/978-3-319-40593-3\_25.
- [17] A. Schreiber and R. Struminski. Tracing personal data using comics. In *Universal Access in Human-Computer Interaction. Methods, Techniques, and Best Practices: 11th International Conference, UAHCI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9-14, 2016, Proceedings, Part I*, 2017.
- [18] R. Struminski. Visualization of the provenance of quantified self data. Master thesis, Hochschule Düsseldorf, 2017. URL <http://elib.dlr.de/110996/>.
- [19] R. Struminski, S. Bieliauskas, and A. Schreiber. DLR-SC/prov-comics: QS PROV Comics prototype - big fixes [data set]. Zenodo, 2017. URL <http://doi.org/10.5281/zenodo.555927>.