Technological infrastructure supporting the story network principle of the Atlas of the Ageing Society

Article *in* Geografie · January 2019

DOI: 10.37040/geografie2019124020217

Adriana Zanda
FHNW University of Applied Sciences and Arts Northwestern Switzerland, Muttenz

Jonas Lutz
University of Applied Sciences and Arts Northwestern Switzerland

Susanne Bleisch
FHNW University of Applied Sciences and Arts Northwestern Switzerland, Muttenz

Technological infrastructure supporting the story network principle of the Atlas of the Ageing Society

ADRIANA ZANDA[1], JONAS LUTZ[2], ALESYA HEYMANN[1], SUSANNE BLEISCH[1]

[1] FHNW University of Applied Sciences and Arts Northwestern Switzerland, Institute of Geomatics, Basel, Switzerland; e-mail: adriana.zanda@fhnw.ch, alesya.heymann@fhnw.ch, susanne.bleisch@fhnw.ch

[2] FHNW University of Applied Sciences and Arts Northwestern Switzerland, Institute for Information Systems, Basel, Switzerland; e-mail: jonas.lutz@fhnw.ch

ABSTRACT Atlases have a long tradition of showing and linking information for the exploration of various mostly spatially related topics. The Atlas of the Ageing Society is an interactive platform illustrating age related data. It enables individuals to explore facts and information related to age and the ageing society. To support content representation as well as a diverse audience, we developed the "story network principle", which embeds annotated visualizations into a network of information in order to allow storytelling with data. Enabling the exploration of such a multifaceted and highly interconnected data landscape, however, posed some technical challenges. This paper describes and discusses a back-end implementation that meets the requirements of the story network principle from a technical perspective. We detail and exemplify the design and implementation of the atlas infrastructure to enable others to benefit from our developments and approaches to the challenges. The story network principle is potentially applicable to a range of applications such as other atlases or digital portfolios.

KEY WORDS back-end implementation – technical infrastructure – network structure – atlas – storytelling – information visualization

ZANDA, A., LUTZ, J., HEYMANN, A., BLEISCH, S. (2019): Technological infrastructure supporting the story network principle of the Atlas of the Ageing Society. Geografie, 124, 2, 217–235.
Received November 2018, accepted April 2019.

1. Introduction

For the exploration of different spatially related topics, people long since use atlases as they compile and show information about a specific theme. In the past, atlases were mostly available in printed form. However, several collections of maps and other information are nowadays accessible as digital products, whereby two forms of digital atlases are distinguished: electronic atlases on CD-ROM/DVD and online atlases. The atlases on CD-ROM/DVD represent a more static form of documentation, mostly available in combination with a paperback guide, such as the "Atlas Digital de Costa Rica (CR 2014)" (Ortiz-Malavasi 2014), the "Digital Atlas of Indonesian History" (Cribb 2010) or the "Bilingual digital atlas of the geotectonic structure of Northwestern Germany and the German North-See sector" (Baldschuhn et al. 2001). They all show different themes and spatial as well as temporal resolutions. Sieber et al. (2016) stated a trend going towards service-oriented online atlases enabling broad access to (national spatial) data. One of the first online atlases was the "Atlas of Canada" which is online since 1994 as they changed directly from a printed to an online version of their atlas. Today, the Atlas of Canada allows exploring different interactive maps as well as downloading reference maps, wall maps or selected thematic maps (Natural Resources Canada 2016). In contrast, the "Atlas of Switzerland" (AoS) exemplifies the development of an atlas from a printed version (until 1997) to an interactive version on CD-ROM/DVD (2000–2010) to an online 3D atlas (since 2016). The concept of the AoS focuses on map visualizations based on a 3D globe "[…] where a 2D map is considered as a special case of a 3D map setting" (Sieber et al. 2013 in Sieber et al. 2016, p. 174). Furthermore, the AoS enables the exploration and comparison of different datasets showing structures and processes in Switzerland. Apart from interactivity and analyzing functionalities, digital atlases have one more big advantage compared to printed atlases; they "[…] have a characteristic capability for structuring the information flow" (Elzakker 1993 in Ormeling 1995, p. 12), as deployed in one of the newest online atlases – the "Indigenous Peoples Atlas of Canada" (Canadian Geographic 2018). In the last decade, this capability led towards a general trend of storytelling. With a focus on geographical storytelling, Cartwright (2004) examined the usefulness of gaming interfaces for accessing geographical information. Caquard et al. (2009) presented "geospatial storytelling" in a cybercartographic atlas where indigenous communities can contribute their (spatial) knowledge in order to preserve their heritage. Nowadays, a quite common approach of storytelling is the so-called "story map". For example, Cartwright and Field (2015) explored "cartographic storytelling" by showing a soldier's personal geography on an interactive map extended with cartographic artefacts, such as elements of a book or of other sources. However, this kind of framework does not allow users to make decisions about the course of the story, what would make the story experience even more intense (Thöny et al. 2018). In summary, several national and other thematic atlases can be named, which generally show a development over the last decades from static (printed or CD-ROM/DVD + paperback) to interactive online atlases. This trend enables users to increasingly interact with the represented atlas contents. Furthermore, this trend allows atlas producers to structure the information in order to tell stories that lead users through the topics. As mentioned, several researchers name different storytelling approaches and examine the implementation of various applications. But, in the end, the presented atlases and storytelling examples underpin the conventional definition that "an atlas is an intentional combination of specific maps or data sets [...] (which) relies on a certain rhetoric [...]" (Ormeling 1995, p. 12).

To create a modern Atlas of the Ageing Society that supports data exploration and storytelling we aimed at developing a flexible digital atlas platform. In order to do so we defined the following requirements that the platform needs to meet (numbered R1 to R5 for referral later in the text):

- Supporting a range of stories and being scalable (R1)
- Allowing users to define their course of the story (R2)
- Taking into account the requirements of a diverse audience from experts to the cursorily interested (R3)
- Supporting different data and visualization types (R4), and
- Separating the content and its structure from the representation (R5).

To meet these requirements conceptually, we developed the "story network principle" (schematically illustrated in Figure 1, cf. Bleisch, Zanda, Korkut 2018) and then enabled it technically with a suitable back-end and front-end implementation. The story network principle, as implemented in the Atlas of the Ageing Society (see Chapter 4), integrates annotated visualizations into a network of information where certain links can be emphasized in order to tell stories with data. An annotated visualization is called a "card", comparable to a digital index card. Such a card comprises several information entities, e.g. a title, a descriptive text and a visualization. All cards are linked based on thematic similarities or content-related connections (dashed lines in Figure 1), so that an initial "network" of information is built. Additionally, special relations between cards are added to the network in order to define "storylines" (bold lines in Figure 1). These storylines allow the authors to guide users through a set of selected cards in order to tell them a story. Nevertheless, the network aspect allows the users to leave a story anytime and to follow another network connection to a different card that seems interesting. By selecting information and presenting it on single cards, the story network principle enables the representation of information on different levels of abstraction and complexity. Additionally, the concept of cards supports structuring the contents as well as representing a range of data from different

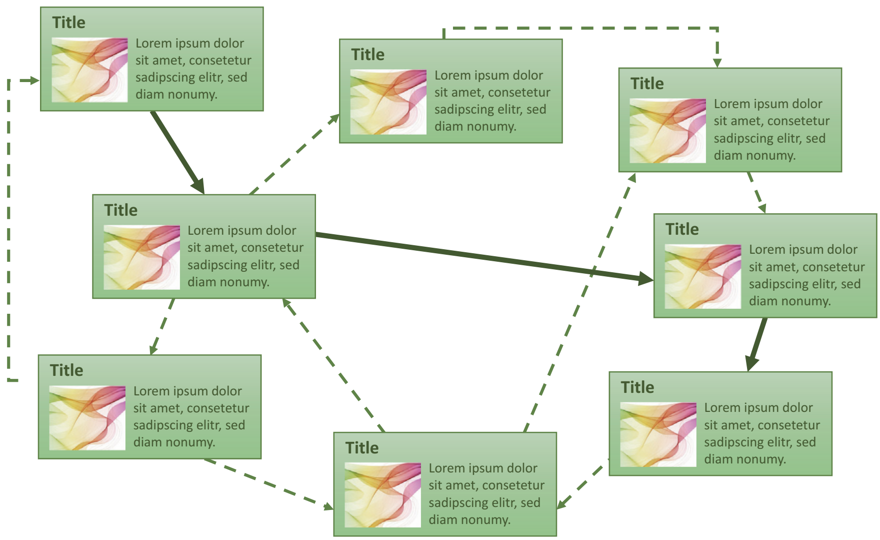


Fig. 1 – Schematic illustration of the story network principle – building a network by defining thematic connections between annotated visualizations. Adapted from Bleisch, Zanda, Korkut (2018).

sources, from quantitative tables to qualitative statements to image or film materials (Bleisch, Zanda, Korkut 2018).

The story network principle already meets the above stated requirements R1–R4 conceptually. First, it allows the continuous integration of new information in the form of cards (R1). Through the network and storyline features, the story network principle enables three ways of data exploration: guided (stories), semi-guided (network) and free navigation (randomly jumping across the network). These different navigation options allow users to define the course of the story or data exploration according to their interests (R2). Additionally, we accommodate the requirements of a diverse audience (R3) by offering information at different levels of granularity on the cards. The requirement R3 is also supported by the fragmentation of information into a card structure, which enables the integration of different data and visualization types (R4). What remains is the challenge of how those requirements as well as R5 – the separation of the data structure from the presentation – can be technically enabled.

This paper describes an infrastructure that supports the story network principle and meets the associated requirements (R1–R5) technically. The next chapter introduces the components, such as a suitable data model or usability dimensions, which regard the defined requirements and enable the technical implementation of the story network principle. Then, we detail and exemplify the design and implementation of the atlas infrastructure. This seems important, as the story network principle is potentially applicable to a range of application scenarios like other atlases or digital portfolios. Thereafter, we discuss the advantages and disadvantages of the employed technologies for interactive data displays. Finally, we conclude that the presented technical infrastructure is scalable and sketch future work on the story network principle and its implementation.

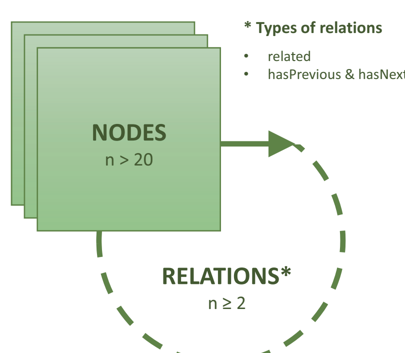
2. Components supporting the technical implementation of the story network principle

The requirements defined in the introduction are conceptually met by the story network principle. This conceptual model and its demands also define the components and architecture of the technical infrastructure. Thus, we present a data model, which considers the requirements R1, R2 & R4. Then, the aspects usability and performance are elucidated in order to meet the requirements imposed by the users (R3), as well as the model-view-controller architecture is introduced and its advantages for the technical infrastructure are discussed (R5).

2.1. The data model

The basic idea of the story network principle, namely linking information to a network and allowing the definition of special connections for storytelling, requires a suitable data model for storing the information as well as the connections (schematically shown in Figure 2). Relations connect nodes to a network, whereby each node represents selected content in form of an annotated visualization, a card. In order to create a reasonable network, the definition of at least 20 nodes is necessary; otherwise, the network structure is too small to unfold its potential. However, each node requires at least two relations to other nodes. This constraint

Fig. 2 – The data model consists of relations between a substantial number of nodes to construct a network. Different types of relations exist in order to establish the network ("related") and to allow the definition of storylines ("hasPrevious" and "hasNext").



	related	hasPrevious	hasNext
related	Network Node	End-of-Story Node	Start-of-Story Node
hasPrevious	End-of-Story Node	*Not allowed*	Story Node
hasNext	Start-of-Story Node	Story Node	*Not allowed*

Fig. 3 – Combinations of relation types for nodes with two relations

ensures that the nodes build a network without any "dead ends". Thus, a user could potentially infinitely navigate through the network and explore the data. In order to enable storytelling, the authors distinguish different types of relations: "related", "hasPrevious" and "hasNext". The relation type "related" creates the basic links for the underlying network. The two connection types "hasPrevious" and "hasNext" allow the definition of special "story" connections.

Assuming that a node has two relations, then there are seven possibilities for combining these three types of relations (see Figure 3). If a node has two "related" connections, then it is a network node, meaning that the node is part of the network but is not included in a storyline. Another plausible combination is "hasNext–related", this denotes a node, which represents the start of a storyline ("hasNext") and furthermore offers one or more other connections ("related") to explore. The opposite variant is "hasPrevious–related", which denotes the end of a story and offers related cards to explore as well. The node "hasPrevious–hasNext" is a story node, which is part of a storyline. When the number of a node's relations is greater than two, even more combinations than the exemplified above are possible. However, the combinations "hasPrevious–hasPrevious" and "hasNext–hasNext" are not allowed because they cannot be integrated into a logical network and story construction (see Figure 3).

An additional requirement was the support of different data and visualization types (R4). This was solved by adding several data attributes to each "node" in the data model, which represent the content of the card (listed within the nodes in Figure 4). These attributes can store different types of data and each data set represents only one specific node. For the cards of the Atlas of the Ageing Society the attributes are id, title, description and visualization (see Chapter 4 for more details on the specific implementation of the Atlas of the Ageing Society). The visualization is an attribute which accommodates a range of visualization types, such as the path to an image file or the data for a visualization that is later rendered by the front-end engine for the display. Another important aspect of the story network principle is the meta-nodes. Each of these nodes represents a real world

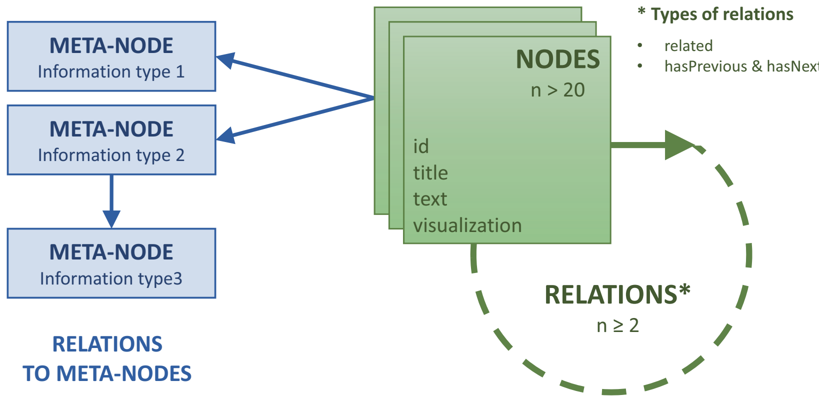


Fig. 4 – The extended network data model (cf. Fig. 2) illustrating further relations from nodes to meta-nodes

entity. Based on the idea of ontologies, where every entity is represented as its own instance, every real world object is represented on their own. Accordingly, the data model from Figure 2 is extended as illustrated in Figure 4, to accommodate the meta-nodes. Thus, two separate nodes could have a relation "hasAuthor" to the same meta-node of information type 1, or in this case "Author". In addition, a meta-node could have further meta-nodes, which specify the superordinate meta-node. As a result, this extended data model enables a flexible integration of new contents (nodes) and it can be adapted to multiple types of information by changing the structure of the related meta-nodes.

In summary, the presented data model covers the majority of the requirements of the story network principle. It allows the creation of a network and the definition of stories, what is the foundation for users being able to follow their own course of a story (R2). Furthermore, it is extendable in different dimensions by adding nodes and/or meta-nodes (R1) and it enables the support of different data and visualization types (R4).

2.2. Usability and performance

The representation of the story network principle should be accessible online by a wide range of users. Furthermore, it should take into account needs of a diverse audience (R3) in order that the website is appealing and users may revisit the atlas for further exploration. Generally, usability and user-friendly web design is of general interest and is of increasing importance (Mao et al. 2005). Usability mostly encompasses the following three aspects: effectiveness, efficiency and

Table 1 – Different query examples and possible responses enabling the communication between the view and the model

Query Example	Possible Response
getNode	A specific node
getRelations	The relations of a specific node
getMetaNodes	All the meta-nodes related to a specific node

satisfaction (Brooke 1996). In this paper, we do not explore the various options for front-end design and implementation that have the most influence on effectiveness and satisfaction. Some elements, such as symbol and interaction design for the Atlas of the Ageing Society are detailed in Bleisch, Zanda and Korkut (2018). However, several aspects of the back-end implementation directly influence the efficiency. One important aspect is the website's performance (Becker 2004). If a web response takes too long, users lose their interest and are unlikely to revisit the page. Accordingly, back-end and front-end efficiency is a general requirement to a technical infrastructure, meaning that the front-end requires having fast access to the information and the queries are required to be efficient. In our case, the back-end consists of a data model which allows a number of queries. Each query represents a question for the database and returns an answer to the front-end – a subset of the whole data stored in the database. More specifically, on a website representing the story network principle, a query returns the information needed for the illustration of a card. Based on the data model (see Section 2.1), Table 1 exemplifies general queries and their respective responses.

For meeting the requirement of supporting a good performance of the designed infrastructure, as well as separating the structure from the presentation (R5), the model-view-controller (MVC) architecture is employed as a guiding principle. The MVC approach emerged from software engineering when building interactive applications (Krasner, Pope 1988) and was adapted for several geospatial problems (e.g. Edwardes, Burghardt, Weibel 2003; Reichenbacher 2004). As the name of the MVC states, the architecture differentiates three components: the model, the view and the controller. The model contains functionalities to access and manipulate the data, the view is the visual representation of said data and the controller defines the interaction between the two (Edwardes 2007). The advantage of such an MVC architecture is the modularity, meaning that the data, the functionality and visualization are isolated from each other. This is advantageous for development, for the update as well as for the modification of an application (Krasner, Pope 1988). The MVC approach allows having different interfaces (views) with specific functionalities (controllers), which rely on the same data structure in the background (model). The technical infrastructure, as introduced in the following chapter, builds on these principles of functional separation of the MVC paradigm.

3. Technical implementation of the story network principle

To implement the mentioned requirements and to base the implementation on the model-view-controller (MVC) approach, we developed a technical infrastructure as shown in Figure 5. It follows the MVC architecture and separates the views (website) from the functionality (Node.js server) and the data (OrientDB graph database). The functionality and the data model implemented with Node.js (Node.js Foundation n.d.) and OrientDB (Callidus Software Inc. 2018a) are the foundation of the back-end. For the views, we developed a JavaScript based front-end to exploit the potential of the back-end and allow multifaceted user experiences. The front-end implementation can be adapted to different needs and is not described in detail here.

The data is stored in OrientDB – a multi-model, NoSQL and document-oriented database, which additionally supports graph database properties. The main argument for a graph database as content storage was that it naturally allows linking information. OrientDB was selected because it is an open source product and it promised a higher performance on graph operations than other graph databases such as Neo4j (Callidus Software Inc. 2018b, Neo4j Inc. 2018). A wide array of other popular graph databases were tested, however they failed to meet the project requirements in terms of functionality or they are not open source.

In general, a graph is composed of relations between nodes. This concept is directly comparable to the data model described in Section 2.1. Thus, the data

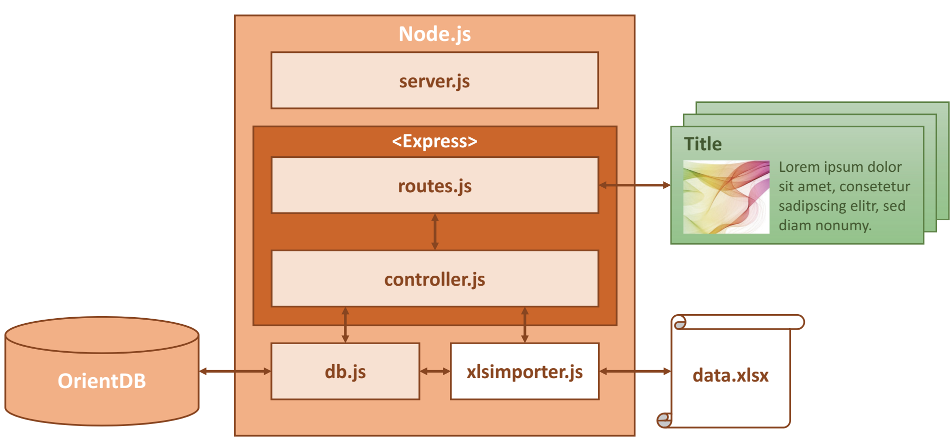


Fig. 5 – Technical infrastructure based on the MVC architecture. Left: the data in OrientDB, middle: the functionality in Node.js, upper right: the card views implemented as a website, and lower right (with white background): the optional "Excel importer" to support content definition in Excel sheets (data.xlsx) and importing it (xlsimporter.js) into the database.

model maps directly onto a graph within the database. Accordingly, this allows querying data natively for different data dimensions, such as nodes, node contents or relations, and to support a range of navigation and exploration scenarios in the final application, the views.

To allow dynamic queries of the data and data links, we developed a Node.js back-end application. Node.js is an open source server-side runtime environment based on JavaScript (JS), which enables running network applications. The event-driven architecture gives rise to the biggest advantage of Node.js, the "non-blocking I/O", meaning that each Input and Output operation is executed asynchronously. This increases the performance and leads to a fast response time for users as Node.js can handle a very large number of requests at the same time (Node.js Foundation n.d.). In addition, Node.js follows the ECMAScript standard (ECMA International 2018), which enriches the native JavaScript by adding diverse functionalities, e.g. the Promise functionality which eases the handling of asynchronous calls. Further, it has the advantage that existing modules are available for the Node Package Manager (npm), which simplifies their management and the handling of their dependencies. The npm hosts thousands of open source Node.js modules. Thus, faster development is possible due to many reusable modules and their functions (W3Schools 2018). Further advantages of Node.js are that it is a full-stack JavaScript technology by covering all areas (front-end and back-end) of an application. This can reduce the number of project collaborators and leads to fewer errors as well as higher productivity as only one language is used across the stack.

Figure 5 depicts the schematic structure of the server's back-end. It consists of the Node.js server, which is initialized by executing the "server.js" file. This file contains all server specifications and modules to be loaded. One of the employed modules is Express (Express 2017a), it handles the whole routing of the queries from the front-end to the back-end. It contains "routes.js", which handles the routing of the web application. This determines the answer to a request at a particular endpoint. A request is defined by an URI (or path) which triggers a specific HTTP method (GET). In non-technical terms this means that it maps an action of a user to an internal server function. For example, when a user requests the card with id 1 the query is sent to the server by using the URL: "http://www.altersatlas.ch/AA?story=1". At the back-end, the Express module then calls the "getNode" (subqueries listed in Table 3), and delivers the data to the front-end, which visualizes the data on the page. However, in reality the procedures to gather the node data are somewhat more complex. The procedure consists of several data queries, like gathering the node data, asking for connections and historical data. This is where the "controller.js" comes into play. It contains the detailed process of gathering all the data required and preparing it for the final visualization in the front-end application. More abstractly, the Express offers a set of possible functions to the user. Internally, it has a defined process for each of these functions on how to collect the data.

The separation of the different functionality dimensions is crucial for the efficiency of the development and the runtime (Section 2.2). Hence, the detailed queries of the database are not located in the controller, but in a specified "db.js" file. This file handles the whole data layer and contains all the database queries. To enable web security a further npm package was integrated: Helmet is a collection of nine smaller middleware features that set security-related HTTP headers. The module enables an xssFilter by default or hides the X-Powered-By header, which is usually set by Express and indicates, which technology the server uses (Express 2017b).

4. The Atlas of the Ageing Society: application of the story network principle

As mentioned in the introduction, the Atlas of the Ageing Society is an interactive platform aiming at illustrating age related data, such as the distribution of different age groups or the evolution of demographics in Switzerland. The first version is available online at www.altersatlas.ch. The platform was developed with support of the strategic initiative "Ageing Society" of the University of Applied Sciences and Arts Northwestern Switzerland (FHNW 2018). Consequently, a large part of the atlas consists of the results of the projects that were involved in the initiative and the atlas development advanced in collaboration with different project collaborators.

This chapter presents the first actual realization of the story network principle. We explain the implementation of the Atlas of the Ageing Society in order to show how the described back-end infrastructure is employed to create the atlas as well as to interact with different data and information contributors. To do so, we describe how and why the data model (Section 2.1) and the technical infrastructure (Chapter 3) have been adapted. Subsequently, an implementation of a front-end (view) is described as an example.

4.1. Realization of the data model

For the Atlas of the Ageing Society, the meta-nodes from the data model (Fig. 4) were implemented and assigned to real world entities, namely one or more data sources (optional), multiple tags (required) as well as one project (required), which is further connected to one or more institutes and a contact person (both required). The data sources define where the data for the visualization comes from. The tags label the content and assist in searching for specific nodes, for example, when all the nodes relating to a certain topic should be displayed. Moreover, the project defines which institutes were involved and who is the contact person.

In the Atlas of the Ageing Society, a node represents an annotated visualization (card). Accordingly, in the node itself, it was necessary to store the data for visualizations, different text elements (title and description), as well as metadata about all the data for a more detailed follow up. A special case is the visualization attribute, which accommodates a range of visualization types. This could be images (inserted as a link to the file) or automatically generated parameter-based diagrams. Those diagrams are stored as data and relevant visualization parameters in a JSON string that allows rendering them at display time. Hence, the implemented infrastructure allows creating automatically generated parameter-based diagrams (as shown in Fig. 6) as well as integrating static images, videos or pictures. This implementation of the technical infrastructure supports a range of different visualization types as is one of the requirements (R4).

4.2. Data input and queries

Once the data model is defined, it is necessary to fill it with content. As mentioned in the introduction of this chapter, several projects of the strategic initiative Ageing Society present their project's results in the atlas. Therefore, the project collaborators had to provide their content for importing it into the database. To simplify the process, we developed a structured Excel sheet where collaborators could provide their contents in a familiar environment and not having to interact with a graph database. In the Excel sheet, all the attributes of the nodes, the data stored in the meta-nodes as well as the connections between the nodes are defined. The column names in the first row designate the attributes. Accordingly, each row (except the first) contains all the information for creating one card. The tabular format provides a fixed scheme guiding a contributor in structuring the content for inclusion in the Atlas of the Ageing Society.

Each attribute needs to meet certain requirements that cannot be predetermined directly in the Excel sheet. As an example, each node requires a unique identifier. However, Excel does not automatically check the entries regarding this requirement. Accordingly, a separate list is needed, which describes the attributes and defines the requirements. The most relevant criteria is if an attribute is required or optional. For example, each card must have a title or a description. However, different aspects of the metadata depend on the used resources. As a result, they are optional. Moreover, special attributes can occur, such as the id or tags. The attribute "tags" can encompass multiple tags, which are separated by semicolons. Furthermore, this attribute should reduce the number of different

Table 2 – Queries and their responses enabling different functionalities in the view

Query	Response
searchNodes	All nodes containing a specific string
getRandomNode	A random node
getAllTags	All tags
getNodesByTag	All nodes with a specific tag

Table 3 – Subqueries of the "getNode" query, which returns all the information enabling the data representation of a complete single card in the front-end (view)

Query	Response
projectCall	The project assigned to the node (relation "hasProject")
contactCall	The contact of the project (relation "hasContact")
instituteCall	The institutes related to the project (relation "hasInstitute")
datasourceCall	All attributes of the data source of the node (relation "hasDatasource")
previousCall	The node which is previous in the story (relation "hasPrevious")
afterCall	The node which is after in the story (relation "hasAfter")
relatedCall	Nodes which are related to the actual node (relation "related")

words to a minimum by using always the same form of a word among the cards, e.g. always use "man" instead of mixing up "man" and "men". To date, these special cases must be handled and controlled manually.

In order to import the data from the Excel sheet into the database, the infrastructure offers an optional solution – the import of an Excel file (see Fig. 5). For the Atlas of the Ageing Society, the data input is controlled by the "xlsimporter.js" requiring the XLSX parser and writer module (NPM 2018). Then, as already described in Chapter 3, the db.js file manages the data handling with the therein-defined queries. Table 2 and Table 3 list some queries implemented in the Atlas of the Ageing Society in order to give an idea of the possible queries in the implemented infrastructure.

The front-end queries relevant data from the back-end when a user interacts with the interface. For example, the Atlas of the Ageing Society has a start page, which presents different entry points for exploring the atlas content. Table 2 shows the queries, which allow the front-end to query and output the content that is based on the user-selected entry point. If someone is searching for a specific term, the "searchNodes" query will return nodes containing the respective search term. On the other hand, for random browsing, the "getRandomNode" query returns any card from the network. The "getAllTags" and "getNodesByTag" enable the exploration of the data by tags. Underlying, there is one basic query for data and information illustration as a card. The "getNode" query is built up by several subqueries (listed in Table 3), which all together return the information needed

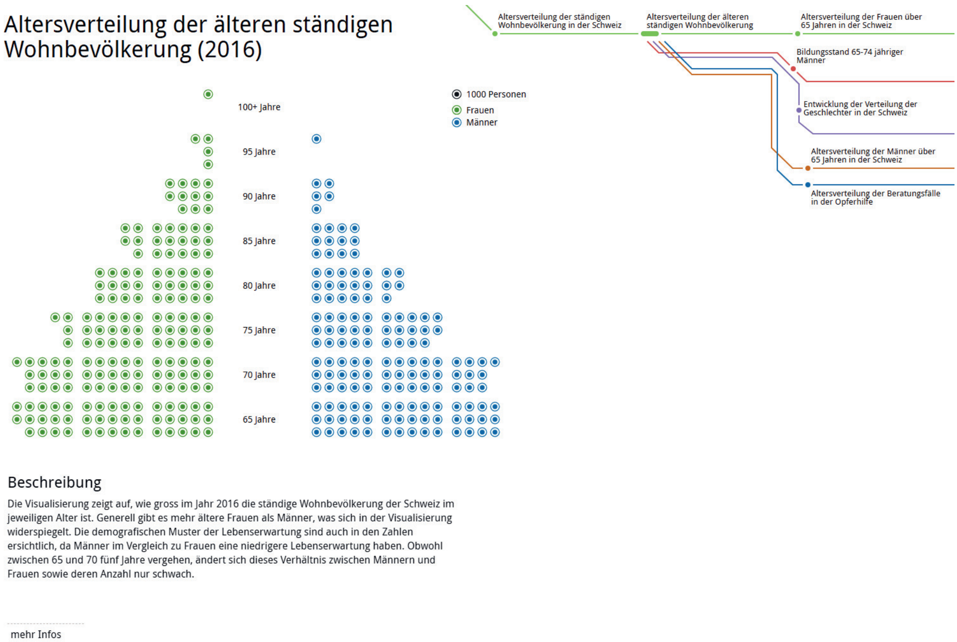


Fig. 6 – An example card of the Atlas of the Ageing Society showing a title, a visualization and a short description on the left-hand side. On the top right-hand side one example implementation of a navigation based on the story network principle with a storyline in green and related connections in other colors is shown.

for the rendering and display of a single card, the navigation to the linked cards – including the storylines – as well as the meta information.

4.3. The front-end of the Atlas of the Ageing Society

The Atlas of the Ageing Society (www.altersatlas.ch) is the first example showing how the output of the described back-end technical infrastructure can be represented on the client-side or the view. A JavaScript-based front-end was developed to exploit the options of the back-end and to allow multifaceted user experiences. Figure 6 shows a card of the Atlas of the Ageing Society. The entry point to each card is a single URL containing the identifier of the node. All the information used to create such a card is the result of the "getNode" query (see Table 3). On the left side of the card, there is always a title, a visualization, a description to the diagram and hidden metadata (the "back-side" of the card) illustrating the actual content of the card. On the top right, an extract of the network is represented as colored lines linking the related and the story nodes to the currently illustrated node. On the green line on the top of the network, the storyline is illustrated with the previous and the next node of the story. The colored lines below show the connections to the related nodes. Thus, the user can navigate through the network graph from node to node and explore the data. The data model requirement of each node linking to at least two other different nodes ensures that the navigation never ends in a dead-end. However, it still is possible to navigate in smaller or larger circles if the same links are chosen repeatedly.

5. Discussion

The introduced technical infrastructure as well as the exemplary implementation by the Atlas of the Ageing Society show that the demands of the story network principle can successfully be met technically. This chapter revisits the requirements (R1–R5) and discusses the employed technologies for interactive data displays, for example, for a modern atlas platform. One advantage is that the data model, as elaborated in Section 2.1, supports storytelling by linking nodes with different types of relations. Thus, it allows creating a basic network of connected nodes and the definition of special connections – the storylines. As the data model is directly mapped onto the graph database OrientDB, the implemented and described technical infrastructure is the foundation that enables the story network principle. Furthermore, the database enables saving attributes in a node and assigning metadata to it, which is stored in separate meta-nodes – as defined in the data model. As meta-nodes are linkable to multiple nodes, it is not necessary to store information twice or repeatedly, what simplifies the process of updating the database. A core strength of the story network principle and its technical implementation is its scalability (R1). Adding new content through additional nodes or meta-nodes is simple as long as the required connections are defined. Similarly, outdated nodes and the connections to them can be removed as long as each other card still features at least two connections.

The above-described technical infrastructure takes into account the needs of a diverse audience (R3) by meeting especially the user-oriented requirement of performance (efficiency), but also by enabling functionalities, which allow users to explore and interact with the data in different ways, i.e. guided, semi-guided or randomly (R2). Thanks to the asynchronous functioning of Node.js, a performant collaboration between the back-end and the front-end was implemented. The infrastructure enables back-end efficiency and thus allows an efficient front-end performance, to support the main goal of a smooth user experience and increased user retention. In addition, the technical infrastructure allows the implementation of different functionalities, such as listed in Table 2, which allow a range of access and interaction possibilities as exemplified by the Atlas of the Ageing Society.

The presented infrastructure supports different data and visualization formats (R4). This was realized through the specific data model, which encompasses multiple attributes, but also through the choice of OrientDB, which enables storing different data types. Accordingly, different attributes can represent different data types thus adding to the flexibility of using the technical infrastructure. Furthermore, as realized in the Atlas of the Ageing Society, the technical infrastructure was constructed in order to enable the creation of different visualization types from the same attribute. This visualization attribute can comprise a node's visualization data containing specific configurations to render itself, but also links to static image files. Consequently, it is possible to automatically generate parameter-based diagrams but also to show static images and figures.

The presented technical infrastructure is inspired by an MVC architecture, as described in Section 2.2, in order to enable the separation of the structure (back-end) from the representation (front-end; R5). This approach has the advantage that components, i.e. the view or the functionalities of a platform, can be changed or varied (for example, to allow for multiple views with different functionalities), but the infrastructure in the background can remain the same. This allows having a fixed data model while the queries and visualizations can always be adapted to the users' needs and implemented in a specific front-end application. The Atlas of the Ageing Society (Chapter 4) exemplifies one specific representation, which is based on the presented technical infrastructure. However, the database could be filled with different data for other applications of linked cards and, thus, the background implementation can be used for different views and applications. Additionally, in each application, new data nodes can be directly added to the network by adding a new network node and the respective links to the database. Adding new meta-nodes may require changing the query functionalities.

Another aspect to discuss is the data input with an Excel sheet. As described in Section 4.2, in the Atlas of the Ageing Society the approach was to fill the content in an Excel sheet and import it into the database. This has the advantage that most people can handle Excel sheets and can provide their content by filling out the different attributes per column. These columns can be adjusted and extended as required. Therefore, they can be directly adapted to changes in the data model. However, a disadvantage of this approach is that a separate list of specifications is needed to describe whether an attribute is optional or required. In contrast, most databases allow defining some of these specifications in order to prevent a user from forgetting compulsory attributes. Furthermore, databases enable the definition of identification attributes whereby unique identity numbers are generated automatically. A second disadvantage of the Excel file is that all entries have to be set or checked manually. The manual setting of values is often precarious, for example, for tags. As one tag can be related to multiple nodes in the data model (see Fig. 4), it needs to be entered in the sheet for multiple cards with exactly the same spelling; otherwise, two tags would be created for the same meaning (e.g. using "child" and "children"). Additionally, some manual checking has to be done regarding the link information to ensure that each node has at least two connections. Overall, the approach with the Excel sheet, while not ideal, allowed and simplified the collaboration with a diverse population of data and information contributors.

6. Conclusions and outlook

This paper presented the design and implementation of a technical infrastructure based on the story network principle, meeting the defined requirements of an information platform for data-based storytelling from a technical perspective. It introduced a framework that is scalable content- and network-wise by adding nodes and meta-nodes to the data model. The presented solution is fast and takes into account the common practices in web development. Furthermore, it can potentially be adapted to a range of projects that have similar requirements and want to present their contents to users through the story network principle of connected cards and storylines. The technical and conceptual system will be developed further in the future to enable more user interaction, such as adding comments or connecting cards to "personal" stories for recommendation, or even immersion, e.g. using personal data for comparison of information. Additionally, users could become registered and add content cards. Furthermore, storylines could be automatically generated through content analysis or through tracking user behavior as they follow interesting links through the card network. A first feedback process has shown that the users like the story network principle and its possibilities (Bleisch, Zanda, Korkut 2018). However, further user studies will be required to more completely evaluate the effectiveness of as well as the satisfaction with the implementation. Future developments will also replace the Excel sheet data input approach with a simple user interface that allows storing data directly into the database and automatic constraint checking while being similarly easy to handle for the collaborators.

References

BALDSCHUHN, R., BINOT, F., FLEIG, S., KOCKEL, F. (2001): Bilingual digital atlas of the geotectonic structure of Northwestern Germany and the German North-See sector. (English-German). Geologisches Jahrbuch Reihe A, Band A 153.

BECKER, S.A. (2004): A study of web usability for older adults seeking online health resources. ACM Transactions on Computer-Human Interaction, 11, 4, 387–406.

BLEISCH, S., ZANDA, A., KORKUT, S. (2018): Geschichten erzählen mit verlinkten Visualisierungen – ein Review des Alters-Atlas Konzepts. 38. Wissenschaftlich-Technische

Jahrestagung der DGPF und PFGK18 Tagung in München – Publikationen der DGPF, Band 27, 141–150.
BROOKE, J. (1996): SUS – A quick and dirty usability scale. In: Jordan, P. W., Thomas, B., McClelland, I. L., Weerdmeester, B. A. (eds.): Usability evaluation in industry. Taylor & Francis Ltd, London, 189–194.
CALLIDUS SOFTWARE INC. (2018a): OrientDB Enterprise Edition 3.0, https://orientdb.com/ (26.9.2018).
CALLIDUS SOFTWARE INC. (2018b): OrientDB vs Neo4j, https://orientdb.com/orientdb-vs-neo4j/ (14.8.2018).
CANADIAN GEOGRAPHIC (2018): Indigenous people atlas of Canada, https://indigenouspeoplesatlasofcanada.ca/ (25.9.2018).
CAQUARD, S., PYNE, S., IGLOLIORTE, H., MIERINS, K., HAYES, A., TYLER, D.F. (2009): A "living" atlas for geospatial storytelling: The cybercartographic atlas of indigenous perspectives and knowledge of the Great Lakes Region. Cartographica: The International Journal for Geographic Information and Geovisualization, 44, 2, 83–100.
CARTWRIGHT, W.E. (2004): Using the web for focused geographical storytelling via gameplay. In: Proceedings of the International Joint Workshop on Ubiquitous, Pervasive and Internet Mapping, 7–9 September 2004, Tokyo, Japan.
CARTWRIGHT, W.E., FIELD, K. (2015): Exploring cartographic storytelling. Reflections on mapping real-life and fictional stories. ICC International Cartographic Conference, 23–25 August 2015, Rio de Janeiro, Brazil.
CRIBB, R. (2010): Digital Atlas of Indonesian History. NIAS Press.
ECMA INTERNATIONAL (2018): Standard ECMA-262, https://www.ecma-international.org/publications/standards/Ecma-262.htm (6.3.2019).
EDWARDES, A. (2007): Re-placing location: Geographic perspectives in location based services. Ph.D. thesis, University of Zurich.
EDWARDES, A., BURGHARDT, D., WEIBEL, R. (2003): Webpark – Location based services for species search in recreation area. In: Proceedings of the 21st International Cartographic Conference (ICC), Durban, South Africa, 1012–1021.
EXPRESS (2017a): Express – Fast, unopinionated, minimalist web framework for Node.js, http://expressjs.com/ (2.10.2018).
EXPRESS (2017b): Production Best Practices: Security, http://expressjs.com/en/advanced/best-practice-security.html (14.8.2018).
FHNW (2018): Strategische Initiative Alternde Gesellschaft, https://www.fhnw.ch/de/die-fhnw/strategische-initiativen-15-17/alternde-gesellschaft (14.8.2018).
KRASNER, G.E., POPE, S.T. (1988): A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. Journal of Object-Oriented Programming, 1, 3, 26–49.
MAO, J.Y., VREDENBURG, K., SMITH, P.W., CAREY, T. (2005): The state of user-centered design practice. Communications of the ACM, 48, 3, 105–109.
NATURAL RESOURCES CANADA (2016): About the Atlas of Canada, http://www.nrcan.gc.ca/earth-sciences/geography/atlas-canada/about-atlas-canada/16890 (27.7.2018).
NEO4J INC. (2018): neo4j, https://neo4j.com/ (2.10.2018).
NODE.JS FOUNDATION (n.d.): About Node.js, https://nodejs.org/en/about/ (14.8.2018).
NPM (2018): XLSX, https://www.npmjs.com/package/xlsx (2.10.2018).
ORMELING, F. (1995): New forms, concepts, and structures for European national atlases. Cartographic Perspectives, 20, 12–20.

ORTIZ-MALAVASI, E. (2014): Atlas Digital de Costa Rica (CR 2014). Instituto Tecnológico de Costa Rica. Online access: http://hdl.handle.net/2238/6749.
REICHENBACHER, T. (2004): Mobile cartography – Adaptive visualisation of geographic information on mobile devices. Ph.D. thesis, Technical University of Munich.
SIEBER, R., SEREBRYAKOVA, M., SCHNÜRER, R., HURNI, L. (2016): Atlas of Switzerland goes online and 3D – concept, architecture and visualization methods. In: Gartner, G., Jobst, M., Huang, H. (eds.): Progress in Cartography. Springer, Switzerland, 171–184.
THÖNY, M., SCHNÜRER, R., SIEBER, R., HURNI, L., PAJAROLA, R. (2018): Storytelling in interactive 3D geographic visualization systems. ISPRS International Journal of Geo-Information, 7, 3, 123.
W3SCHOOLS (2018): Node.js NPM, https://www.w3schools.com/nodejs/nodejs_npm.asp (14.8.2018).

ACKNOWLEDGMENTS

This research was partly funded by the strategic initiative Ageing Society of the University of Applied Sciences and Arts Northwestern Switzerland as well as the national innovation network Age and Ageing in Society (AGE-NT) in Switzerland.

The content of this Special Issue of Geografie Journal has been prepared together with the International Cartographic Association, namely with the Commission on Cognitive Issues in Geographic Information Visualization, the Commission on Use, User and Usability Issues, and the Commission on Atlases. Selected authors were contacted during the ICA Commissions' joint workshop "Atlases, Cognition, Usability" held in Olomouc (Czechia) at Palacký University in April 2018. The content of the Special Issue reflects the current topics handled by the Commissions and we hope that the special issue increases awareness and knowledge of cognitive and usability issues with geographic information among the broader geographic community. Preliminary selection of manuscripts and pre-submission communication with authors was done by Petr Kubíček (Masaryk University, proposed incoming co-Chair of the Cognitive Issues in Geographic Information Visualization Commission) and Amy Griffin (RMIT University, current co-chair of the Cognitive Issues in Geographic Information Visualization Commission). We thank all of the authors who contributed their research to this special issue.