

Article

Building a Framework of Usability Patterns for Web Applications in Spatial Data Infrastructures

Christin Henzen

Department of Geosciences, Technische Universität Dresden, 01069 Dresden, Germany;
christin.henzen@tu-dresden.de

Received: 10 October 2018; Accepted: 4 November 2018; Published: 15 November 2018



Abstract: Web applications in spatial data infrastructures (SDIs) should provide robust and user-friendly user interfaces for geoinformation (GI) discovery, analysis, and usage. Poor usability, e.g., caused by unsuitable information presentation or inappropriate (non) availability of functions, can result in inefficient or faulty usage and can increase the acceptance of the application and provided geoinformation. Until now, a number of usability problems in GI web applications were identified; however, methods to summarize these problems, to provide (software-independent) solutions for them, and to find pairs of problems and related solutions hardly exist. We propose an adapted usability pattern concept for web applications in SDIs to map and categorize usability problems and best practice solutions and we enable a GI context-specific creation and discovery of these problems and solutions. The concept includes developed pattern types, relationships, and rules on how to use the relationships for the different pattern types.

Keywords: usability; usability pattern; spatial data infrastructure; GI web applications

1. Introduction

Today's implementations of spatial data infrastructures (SDIs) provide various web applications to discover, analyze, and use the provided geoinformation (GI). The number of available applications and the amount of literature on the usability of GI web applications are indicators for the increasing importance of usability (see Section 2).

The usability of a software describes its quality of use. Good usability results in an effective, efficient, and subjectively and objectively satisfying and probably enthusiastic usage of an application. In the GI domain, as in many other domains, the usability of an application can be rated differently resulting from the heterogeneity of users.

The usability concept can be illustrated as an iceberg metaphor, showing the usability as the visible part of the iceberg (Figure 1). Affecting aspects treat metadata and data problems (GI and data usability), e.g., missing metadata or data values, technical problems, such as incorrect interface implementations or unavailable web services, or software bugs and unimplemented functionality. Poor usability often leads to inefficient or even faulty usage, and reduces the reputation of the application. As a worst-case scenario, it affects the reputation of linked data and services or the SDI used by the application.

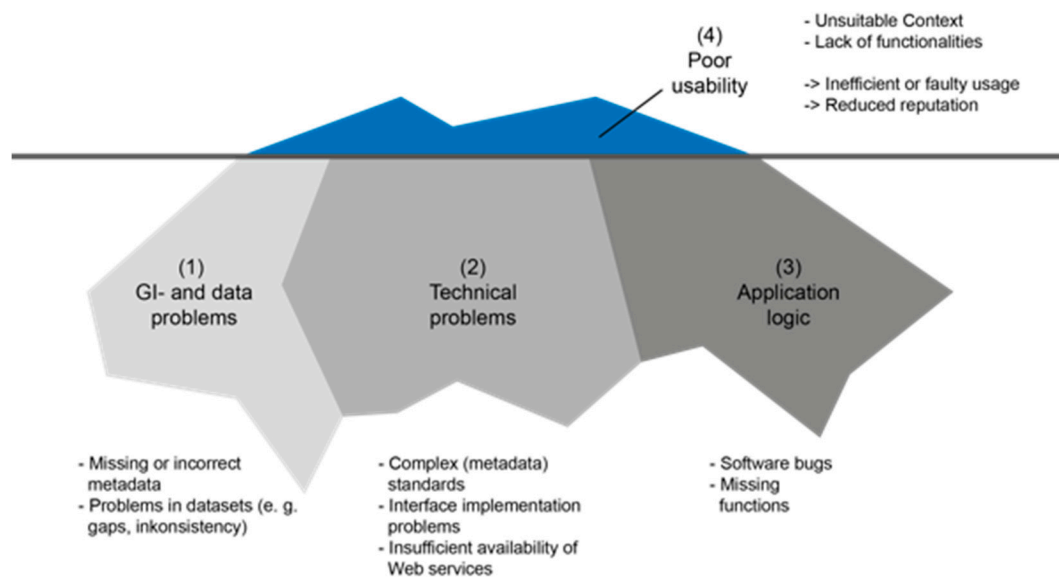


Figure 1. Aspects of usability in geoinformation (GI) web applications.

In GI web applications, several usability problems exist, which can be detected easily and occur frequently. These problems are typically unstructured and are sometimes only collected internally (e.g., in in-house ticket systems of software development companies). Thus, the development of a central usability problem knowledge base seems to be useful. Furthermore, it can be assumed that many problems can be solved by using already implemented solutions of other GI web applications. Although a number of usability problems are identified and documented in the literature (see Section 2), recommendations for problem elimination are rarely given or exist only as software-specific solutions—e.g., source code—and can hardly be transferred to other software projects.

The literature lacks methods (1) to summarize recurring usability problems, (2) to provide software-independent and established solutions enabling solution transfer across software projects, and (3) to find these pairs of problems and solutions.

Therefore, this paper addresses the following research questions: (1) What usability issues can be found in today's GI web applications? (2) How can we structure usability problems of GI web applications? (3) How can we define solutions and link them to the problems to enable a multidimensional, e.g., task- or problem-oriented, search of the solutions?

2. Related Work

In the GI domain, first evaluations and research on usability aspects, focusing on map visualizations or map-related tasks in geoinformation systems (GIS), as well as several reports on the lack of user-friendliness of considered applications can be found [1–3]. Furthermore, usability frameworks for geovisualization [4,5] and user-centered design approaches [6,7] were proposed. The need for specific usability evaluations for GI (web) applications was articulated to better cope with the domain's specificity and complexity [8].

The first usability studies were based on questionnaires to evaluate GIS, and later to evaluate WebGIS and web map sites, often combined with other methods, such as cognitive walkthrough (experts perform a task and act like a certain user group), or thinking aloud (users say what they think and why they act as they do) (Table 1). Unconventional tests, such as screenshot analysis [9] or long-term studies [5] can hardly be found.

Table 1. Usability studies and applied methods. GIS—geoinformation system.

GI Web Application	Usability Test or Inspection Method (List ¹ or Number of Evaluated Applications/Subjects)
Web map site	questionnaires [16] (2/20) + cognitive walkthrough [17] (2/30) + thinking aloud [18] (4 (1)/24 (8)) + workshops [19] (7/20+10) heuristics [20] (14/1)
WebGIS	questionnaires [10] (7/165) + group tests [11] (5,1/12,21,26,68) + workshops [6] (1/14,9,9) screenshot analysis [9] (3/-)
Geoportal	questionnaires [12] (1/14), [13] (3/5), [14] (-/105), [15] (1/22) (de facto) standard inspection [21] (6/1) cognitive walkthrough [21] (3/1) heuristics [21] (5/1)

¹ A comma-separated list indicates a sequence of usability tests or inspections described in one scientific paper.

Studies on WebGIS or web map sites are often more complex, e.g., more test persons are interviewed or more detailed evaluation results are provided, than studies with geoportals, which often focus on data search. Some findings of these studies can still be found in available web applications: lack of clarity in user interface (UI) structures, insufficient map sizes, and interactions, e.g., missing map functions or unstructured toolbars, and inconsistent terminology [9–15].

The results and settings of usability studies vary clearly. Test reports for geoportals often provide summaries of basic usability problems, but do not offer information on the relationships between measurements and test results or derived recommendations to improve the usability, as done in previous web map site tests [17,18]. Usability studies for geoportals are often based on user-centered questionnaires, which lead to subjective results. Objective expert tests (inspections) [20,21] have high potential for the reusability of study settings and, thus, have high potential for useful, but seldom performed comparisons of results of more tests (e.g., preparing long-term studies). Only a few usability studies analyzed complete user tasks, e.g., complex discovery or visualization tasks [13,21], while most studies focused on several aspects or parts of the task or functionality [12]. Finally, comparisons with frequently used and popular applications of other domains, as requested in some studies, e.g., geoportal users often demand search mechanisms such as those implemented in Google or Amazon, can hardly be found [21].

The reviewed studies underpin the change in GI web application targets. Firstly, GIS was implemented as an expert tool and provided a huge set of complex functionalities in complex UIs. Today's implementations often focus on certain user groups or themes and only provide a minimal set of meaningful functionalities to gain proper information. Personalization options for UI elements, content, or functions enable heterogeneous user groups to work with applications properly. In the GI domain, initial personalized UI features, e.g., saving maps or search settings for further usage, were implemented, but there is still potential for more.

The list below structures basic usability problems (see Table 2). Italicized problems are used as pattern examples. These problems were chosen because they occur during different user tasks, focus on different user groups, and address different components of GI web applications; thus, they provide a useful sample for explaining our concept. Underlined problems describe meta problems that cannot be described in one usability pattern. They serve as the input for our new structure of usability patterns for GI web applications, e.g., based on usability problems in problem 4 (too many different designs are used), we developed pattern relationships that show solutions with similar designs, as described later.

Table 2. Usability pattern examples. GI—geoinformation; UI—user interface.

Usability Pattern for GI Web Application	Pattern Description
Filter by geographic names	Users often need to find datasets or services for a certain place, which cannot be located on the map easily (e.g., requiring complex map interactions or the users not knowing the country). They need support in filtering the provided data pool by the place name.
Provide time control for map	Users like to have a time control UI to explore time-variant geodata on the map because, otherwise, they could not explore all provided information of the visualized geodata.
Provide visualization of dataset lineage	Users want to get brief visualizations of geodatasets' lineage to explore the relationship more efficiently than by reading long texts or checking complex tables.
Provide visualization of data hierarchy	If datasets of a given data pool are derived from other geodatasets, users want to get a visual overview of the dataset's relationships instead of working on separated metadata files with the related information.
Provide map link in dataset description	Users often evaluate discovered geodata by examining the related metadata and visualizations on a map. They want to navigate from geodata descriptions to the related map visualizations directly.
Provide link from map to result list	When a user explores geodata on a map and determines that the dataset does not fit their purpose, they want to get back to the latest search result list to find a more suitable dataset. Therefore, they need a direct navigation path from the map to the result list.

General

1. Context of provided functionality is not suitable
2. Lack of personalized/personalizable UI elements, e.g., frequently used combined maps or search filters
3. Lack of a save function for search settings
4. Excessive usage of visual design elements, e.g., colors, images, buttons, etc.
5. Not clearly identifiable icons, e.g., in map tool bars and boxes

Browse, filter, and search settings

6. Search filter, search options, and relevant/helpful information are not displayed
7. Lack of filter and sort functions, e.g., to filter results by city names, regions, or countries
8. Dynamically generated filter (based on result list) is missing
9. *Lack of time control functions (e.g., calendar) according to provided datasets and services*
10. Browsing of data pool can be improved
11. Display of active criteria and terms is not implemented

Hints

12. Labels are not understandable (e.g., technical terms as used in Open Geospatial Consortium (OGC) or International Organization for Standardization (ISO) metadata standards)
13. Inconsistent terminology
14. Lack of description of suitable date formats (e.g., YYYY-mm-dd)
15. Hints on how to edit incorrectly spelt temporal or spatial queries

Presentation of search results

16. Result lists are structured differently (within one or more applications); names for tabs and menus are different
17. Representations of result list cannot be adapted by users
18. Relevant metadata are not shown, e.g., title of a linked web service
19. Empty tabs are shown (instead of being collapsed and focusing on filled ones)

Search assistance

20. Lack of one-box search with autocompletion
21. Lack of information on previously performed search queries
22. *Spatial search is not implemented*
23. Recommendations for search results are missing, e.g., geodata with similar themes or spatial extent
24. Query on and in layers is not possible
25. Similar search functionality is not implemented
26. Automatic spell correction is missing
27. Search terms are not interpreted intelligently, e.g., only characters are compared, which causes problems with singular/plural forms or misspellings

Map visualization

28. Map size is too small
29. Unstructured tool bars with map functions
30. Detail information for objects (e.g., feature information) is not available
31. *Showing a map with discovered Web Map Service (WMS) link is not possible or much too complicated*

Navigation

32. Links between (sub)sites of the application are missing
33. *Lack of links from dataset descriptions to related service descriptions*
34. *Navigation to previous result list is not possible*
35. *Navigation between result list and map is too complicated*

Current usability studies critically discuss the search functionality in GI web applications. GI users take Google as the standard search user interface. They ask for applications (a) with simple UIs providing a single search box, which can be used for thematic, spatial, or temporal queries [15], and (b) which perform a Google-like intelligent (semantic) search [14], which leads to (c) Google-like access to resources [12] (Problems 14–19). Moreover, due to the lack of meaningful search logic and the presence of complex UIs, users think that they need more hints and proper filter and sorting functions to deal with (huge) result lists (Problems 4–13).

In the future, software developers and researchers should be aware of usability problems that address the look and feel of several applications and not only one application. There is a lack of commonly used terms within one or several applications. Using the same terms in every GI application would help users work with the application and reuse knowledge of known terms.

Furthermore, some GI web applications implement inconsistent interaction and user interface concepts. They often lack in proper navigation concepts, which help users avoid getting stuck on a dead-end page. Sometimes, GI web applications consist of already implemented applications (e.g., search clients and map clients), which are only linked to each other. Users then often miss links between subsites of these applications. Inconsistent layouts and color schemes, as well as unknown and complex terms and inappropriately provided UI elements, functions, and information, make the usage of the GI web applications more complicated.

Several usability problems of GI web applications can be detected easily and occur frequently. A collection of structured and linked usability problems and solutions would help software developers and architects avoid problems in early software development phases and would make their work more efficient, because they could use existing solutions to solve known problems without needing to develop new ones. Reusing existing solutions would help users work with different GI applications, because they could use already known interactions on known UIs.

3. Developing a Framework of GI Usability Patterns

Design patterns are a well-established concept in software development to tackle frequently occurring problems and to provide a solution blueprint. Usability patterns are a specialization of such design patterns to specifically address user interface issues and related software solutions. Belonging to human–computer interaction (HCI) patterns, they can also be defined as knowledge representations and functional or visual patterns for typical user interaction or user interface problems [22,23]. Usability patterns have eight attributes for structuring information: name, description, solution, example, context, rationale, consequences, and related patterns [23].

- In *description*, the problem situation is explained in a way that the suitability of the pattern for a certain use case can be derived.
- *Solution* describes the problem solution completely and contains additional hints (e.g., the user should not get distracted during input). It can contain several solutions for different contexts.
- *Example* illustrates at least one concrete implementation of the pattern's solution.
- *Context* defines conditions that need to be complied for the successful use of the pattern (e.g., required screen resolution or expertise).
- *Rationale* summarizes how a solution affects the usability of the application.
- *Cost* provides information for the project management and indicates risks, costs, and disadvantages which result from pattern usage.
- *Related patterns* describe whether and how other patterns relate to the described one.

In our research, we developed a concept of GI usability patterns. The concept is based on the results of the literature review and our studies on the usability of GI web applications (see Section 2). Our concept consists of three parts: an extended pattern context to describe the characteristics of usability problems in SDI web applications, a set of pattern super types and pattern relationships to model consistent interaction concepts, and a set of pattern combination rules to enable the proper application of several usability patterns.

To make the existing usability pattern concept applicable to GI web applications, the following aspects were considered:

1. Adapting to geoinformation: Typical aspects of the GI domain were integrated to enable users to discover suitable patterns (e.g., find all usability patterns that consider map visualization) and to create patterns more easily (e.g., using a given vocabulary).
2. Describing the context: The pattern context descriptions were extended to include the sequence of user interactions and the positioning of associated UI elements, thus linking the usability problem to a software solution context.
3. Defining GI usability pattern relationships: Relationship types were refined to ensure a consistent design and interaction concept (e.g., when users apply several usability patterns simultaneously for a software application).
4. Providing a framework to create, find, and compare GI usability patterns: The pattern content was structured using predefined values for pattern attributes to facilitate pattern creation, discovery, and comparison.

To satisfy the requirements of developing user-friendly GI web applications, the usability pattern concept needs to be reengineered to capture the pattern attributes and integrate domain-relevant information in a structured manner. Meanwhile, the usability pattern structure should stay as simple as possible to enable users to create new patterns easily. In maintaining the structure of existing usability patterns, the new patterns for GI web applications would be comparable to existing usability patterns from other domains. The particular structure of the usability pattern for GI web applications with a given vocabulary for pattern attributes and predefined attribute values should support users in creating new usability patterns, in classifying these patterns, and in comparing them with existing

ones. Supporting extensibility wherever possible should foster the establishment and easy uptake of usability patterns as the appreciated method of developing user-friendly GI web applications.

In the following sections, the concept is described in general and is based on the examples listed in Table 2. Examples were discussed and collected in an interdisciplinary workshop with geoinformation and usability experts.

4. Integrating the GI Domain Knowledge into the GI Usability Pattern Concept

The GI domain knowledge was integrated into the existing usability pattern concept to enable structures to better find, create, and compare problem and solution descriptions. The integration of GI domain knowledge into the GI usability pattern attribute context and into new pattern relationship types was driven by the general requirements below, which result from the systematic literature review on usability studies and experiences in software development of GI web applications.

Requirements for the structure of the pattern context

1. The domain knowledge should be structured in such a way that it enables users to find and filter patterns via several entry points, e.g., thematic or task-specific searches.
2. The patterns should be linked to each other, such that users can navigate from one pattern to a related pattern (e.g., to find similar or opposite patterns).
3. Users should be able to create new patterns and classify these based on existing pattern attributes. Thus, choosing pattern attributes from a fixed and unified set of values (vocabulary), which can then also be used for the (e.g., faceted) pattern discovery should be facilitated.

Requirements for the content of the pattern context

4. The pattern context should consider the characteristics of geoinformation. Software developers should be able to choose patterns based on these characteristics, e.g., finding all patterns that provide solutions for the visualization of time-variant geodata.
5. The pattern context should map typical GI resources (e.g., dataset and services) and their relationships as used in SDIs. They can be mapped to typical user tasks, e.g., visualization services are used in exploration tasks and influence task-specific functionality or information representations. Software engineers can filter patterns for certain information resources.

The context is a pattern attribute, which describes conditions of the pattern usage and thereby classifies the usability problem and related solutions of the pattern (Figure 2). In our concept, the context models the domain knowledge according to GI characteristics. Thus, context attributes classify patterns, form a unified vocabulary, and thereby support users in pattern discovery. The structure is kept simple to be easy to use and to be extendible for specific sets of GI usability patterns.

One of the five context sub-attributes is the *content*, which focuses on the general characteristics of geodata and builds on basic aspects of geodata [24,25] and essential attributes of geodatasets and services as described in the identification attribute of the related metadata of the ISO 19115-1:2014 standard [26] and in the geographic processing services taxonomy ISO19119:2015 [27]. To fit the terminology of the GI domain, the used concepts were mapped to the *attributes content.spatial_extent*, *content.temporal_extent*, *content.topic_category*, and *content.context*. Thus, the content uses relevant GI standards and, accordingly, further elements can be used to extend the content attribute if required.

A usability pattern for GI web applications can be described by one, several, or no content attribute values. The values' spatial extent, temporal extent, or topic category should only be used if they are relevant for the described usability problem or related solution. Table 3 shows that all given examples describe solutions for spatial data, but the spatial extent is just relevant for the patterns *Filter by geographic name* and *Provide time control for map* (context.content = spatial extent). The pattern *Provide time control for map* also requires the visualization of time-variant geodata (context.content = temporal extent). Furthermore, the geodata content is not relevant for the other example patterns.

Table 3. Context and sub-attribute content, resource, relationship, task, and user for six usability patterns for GI web applications. IT—information technology.

Context Sub-Attribute	Filter by Geographic Name	Provide Time Control for Map	Provide Visualization of Dataset Lineage	Provide Visualization of Data Hierarchy	Provide Map Link in Dataset Description	Provide Link from Map to Results
content	spatial extent	spatial extent, temporal extent				
resource	service	service, metadata	dataset, metadata	dataset, metadata	dataset, metadata, service	
relationship			dataset–dataset	dataset–dataset	dataset–service	
task	search	map visualization	search UI discovery	search UI discovery	search UI discovery, graph visualization	map visualization, search UI discovery
user	GI: inexperienced users, domain experts; IT: inexperienced users, IT experts	GI: domain experts; IT: inexperienced users, IT experts	GI: domain experts; IT: inexperienced users, IT experts	GI: domain experts; IT: inexperienced users, IT experts	GI: inexperienced users, domain experts; IT: inexperienced users, IT experts	GI: inexperienced users, domain experts; IT: inexperienced users, IT experts

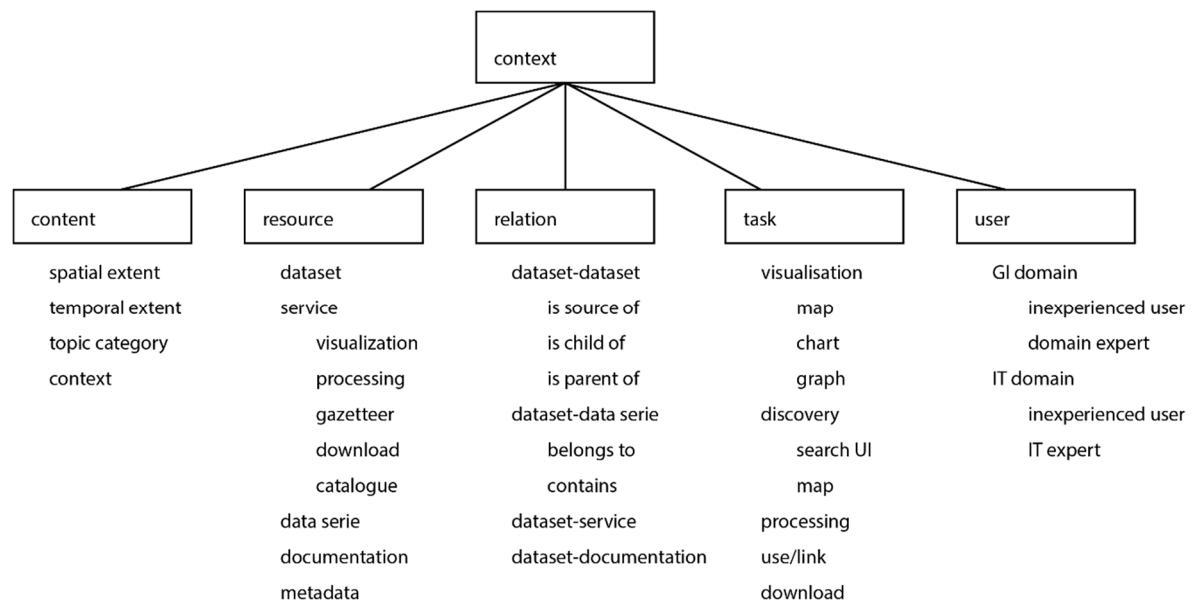


Figure 2. Sub-attributes of the pattern context.

The attribute *resource* discriminates the information resources dataset, service, data series, metadata, and documentation as defined in ISO19115-1:2014 [26]. Here, documentation is derived from the CI_Citation element (ISO19115-1:2014) and can be used to reference (scientific) publications that describe the assigned geodata and lineage in detail and to inform about observations, methods, or simulation models as the origins of the considered geodata [28,29]. The patterns *Provide visualization of dataset lineage* and *Provide visualization of data hierarchy* focus on dataset-related problems and visualize certain metadata elements. The patterns *Filter by geographic name* and *Provide time control for map visualizations* address gazetteer or visualization services in the pattern solutions. The pattern *Provide map link in dataset descriptions* focuses on datasets and related services (as described in the dataset's metadata).

The attribute *relationship* describes possible links between the information resources as defined in aforementioned standards. As these relationships are only modeled unidirectionally in existing GI web applications, they can be used one way only and often lead to usability problems. For example, information about hierarchically structured data (parent–child relationships) is only defined in the metadata of a child dataset (e.g., pattern *Provide visualization of data hierarchy*). Unidirectionally modeled relationships between information resources in SDI are also used to describe lineage or usage (e.g., pattern *Provide visualization of dataset lineage*), dataset–service relationships (e.g., to link a download or visualization service to a certain geodataset; pattern *Provide map link in dataset descriptions*), and dataset–publication relationships.

The *task* sub-attribute describes the context of usage. Typical user tasks performed in SDIs are discovery, visualization, process, use of data or services in other applications, and download. The discovery of suitable usability patterns by the task attribute facilitates designers, who are developing a new user interface, to find a pattern for a certain part of an application, e.g., for the map (visualization), and supports software engineers during the test phase to solve detected usability problems for a certain part of an application, e.g., discovery.

Many GI web applications were developed by linking several standalone applications (e.g., search client and map client). Then, usability problems occur due to different interaction concepts and missing navigation paths in the linked applications. The patterns *Provide map link in dataset description* and *Provide link from map to result list* address such usability problems (task search and visualization).

Furthermore, the sub-attribute *user* distinguishes users' experiences in the GI and information technology (IT) domain. Experiences in the IT domain include knowledge about data infrastructures and web services, whereas knowledge about spatial and spatio-temporal data modeling is described

in the GI domain expertise. Both domains then distinguish inexperienced and expert users. Usability patterns for GI web applications can be relevant for both user groups, e.g., pattern *Provide map link from dataset description*; it is expected that inexperienced users do not reach the map without being guided through direct navigation, while experts reach the map more efficiently. The patterns *Provide time control for map*, *Provide visualization of dataset lineage*, and *Provide visualization of data hierarchy* mainly focus on experts.

The attributes are meant to be extendible to also cover more detail. For instance, to more precisely describe a collection, which only contains patterns that deal with discovery aspects using a search UI, the task.discovery.searchui can be separated into the following:

```
task.discovery.searchui.formulate_search_query,
task.discovery.searchui.evaluate_results,
task.discovery.searchui.evaluate_a_result (metadata, visualization, or relationship evaluation),
task.discovery.searchui.reformulate_and_filter.
```

This enables refined pattern filtering. As each sub-task is defined as a child of the general discovery task via search UI, patterns of this collection are still compatible with patterns of other collections. Furthermore, specializations can be realized for other attributes, such as for the content to consider specific topics only (content.topic_category.water, content.topic_category.environment, etc.), e.g., Infrastructure for spatial information in Europe (INSPIRE) themes, or for the resource, e.g., to further discriminate service types (resource.service.wms, resource.service.wps, etc.).

5. Modeling a Consistent Interaction Concept with Pattern Types and Relationships

The usability pattern concept should not only reflect GI-specific aspects, but should also provide non-GI-specific solutions for usability problems that occur in current GI web applications. An example of a frequently occurring problem is the use of different and incompatible interactions in one application. This section describes our proposed concept of pattern relationships, which support modeling of a consistent interaction concept when applying several patterns simultaneously. Additionally, we developed pattern types that are introduced as additional options for grouping patterns. Since patterns cannot be combined freely or used independently, our developed rules on how to combine pattern relationships and types are introduced. The design follows the requirements below.

Requirements for pattern categorizations

1. It should be possible to map patterns to the tasks of the user interface engineering process (information design, visual design, and interaction design), thereby enabling information and interaction designers to filter pattern collections due to their tasks.
2. Patterns should be modeled as knowledge networks. Instead of providing isolated solutions, patterns should be linked to each other to explicitly show the potentials for exchange and reuse.
3. Pattern types and relationships should facilitate navigation through a pattern collection (browsing) at least by considering technological and problem-oriented aspects.
4. Pattern types and relationships should be designed in such way that rules on how to combine them can be derived. These rules should facilitate the creation and integration of patterns into an existing pattern collection.

Requirements for the content of pattern types and relationships

5. Users should be able to link described solutions of patterns via specific relationships, such that pattern exploration can be done from different perspectives in software design, e.g., interaction design. Thus, pattern solutions should be comparable on specific user interface design levels.
6. Relationships should allow the scalability of pattern solutions. It should be possible to model specializations of pattern solutions, as well as decompositions of the solutions, into reusable atomic components.

7. To avoid mistakes and their repetition during the software design phase, the relationship concept should enable the linking of established solutions (usability pattern) with inconvenient solutions (anti-pattern). This helps integrate knowledge as derived from usability evaluations into the pattern concepts.

Usability patterns can be linked via relationships to a knowledge network for usability problems and solutions. For different types of patterns, e.g., for design patterns, several relationship types already exist. These relationships can be supplemented with additional organization principles to facilitate pattern discovery and creation. Firstly, in our research, we developed three pattern super types that are introduced as such an organization principle. They serve, together with the pattern relationships, as foundation for a rule set on how to use and define patterns.

Usability patterns can generally be structured into three categories of solutions: (1) informative solutions, which describe the way of representing (new) information in a user interface; (2) functional solutions, which describe (new) functionalities and related user interface elements; and (3) navigational solutions, which describe new or optimized interaction paths (Table 4).

Table 4. Pattern super types and descriptions.

Pattern Super Type	Description
Information pattern	This describes the representation of (new) information, e.g., relationships between data, which are needed to fulfill one or several tasks. The subject of information patterns is not only the description of the displayed information (information design), but also the presentation form, e.g., the mapping of hierarchical information to a tree structure (visual design).
Function pattern	This describes new or optimized functionality, which is needed to fulfill one or several tasks. A function pattern does not only describe the functionality, but also the design and context of this functionality, which means that the user interface elements are needed to execute the functionality and the availability of the functionality (in relation to the user's workflow).
Navigation pattern	This describes new or optimized interaction paths, which are needed to fulfill one or several tasks. Thus, the solution of a navigation pattern describes new links between websites, components, or optimized navigation paths, e.g., expert mode, and information about their design, e.g., as a link or button.

For the introduced pattern super types, specific pattern relationships are proposed, which explicitly show the similarity of solutions and, thus, help implement consistent design and interaction concepts in GI web applications (Table 5). To model dependencies of concrete patterns, experts, e.g., domain or IT experts, choose proper relation types or at least check the otherwise defined pattern relationships.

The relationship types *same interaction* and *similar interaction* describe the similarity of two pattern interaction concepts. They also address the similarity of the interaction-related user interface elements. The relationships *same user interface design* and *similar user interface design* simply refer to similarities in the design of user interface elements (visual design).

Consistency in a GI web application can be achieved by implementing similar patterns in an analogous way, e.g., using the same source code. An example of patterns and relationships is given at the end of this section.

Additionally, existing relationship types are used: specialization (A refines B), aggregation (A consists of B and C), association (A is related to B), complement (A complements B), alternative (A is alternative to B), dependency (A is depend on B), incompatibility (A is incompatible with B), and the anti-pattern relationship (linking usability and an anti-pattern).

Relationships between patterns are typically modeled explicitly, but can also be expressed implicitly. Some relationship types, e.g., aggregation, specialization, and complement, indicate that the linked patterns realize a similar interaction or user interface. Knowledge about the implicitly modeled

relations and the implementation-oriented interaction and user interface relationships are useful for selecting and realizing pattern solutions, and support experts and inexperienced users in creating new pattern descriptions and in modeling relationships to existing patterns.

Table 5. Pattern relationships and descriptions.

Pattern Relationship	Description
Same interaction (A realizes same interaction as B)	Two patterns are linked since they describe solutions using the same interaction concept. Pattern A realizes the same interaction as pattern B, if and only if all atomic interaction elements that are used in the solution of pattern A are used in the same sequence in pattern B's solution, and if all user interface elements needed for the interaction are equal in look and feel.
Similar interaction (A realizes similar interaction as B)	Two patterns are linked since they describe solutions using similar interaction concepts. Pattern A realizes a similar interaction as pattern B, if and only if all atomic interaction elements that are used in the solution of pattern A are used in a similar sequence as in the solution of pattern B. Interaction types that are used in pattern B have to match those of pattern A. Differences between the pattern solutions could include design and labeling of user interface elements, e.g., differently labeled buttons, or the sequence of atomic interaction elements.
Same user interface (A realizes same user interface as B)	Two patterns are linked since they describe solutions using the same user interface concept. Pattern A realizes the same user interface as pattern B, if all atomic user interface elements used in pattern A's solution are used in the same representation form, number, and position.
Similar user interface (A realizes similar user interface as B)	Two patterns are linked since they describe solutions using similar user interface concepts. Pattern A realizes a similar user interface as pattern B, if and only if all atomic user interface elements have the same type as in B. Differences between the pattern solutions could include the design, labeling, and positioning.

The rules below summarize the knowledge about pattern relationship and type combinations, e.g., incompatibilities and implications. The list only contains rules that can be applied to the pattern examples illustrated in Figure 2; all other rules are listed in the Appendix A.

1. The relationships *same interaction* and *similar interaction*, as well as *same user interface* and *similar user interface*, can only link two patterns of the same type. Patterns that provide similar or same solutions in terms of user interaction or user interface must address a problem of the same type (information, navigation, or function).
2. The relationship *specialization* links two or more patterns in terms of a pattern-type-related refined aspect. Thus, the relationship can only link two patterns of the same super type.
3. The relationship *specialization* is used to model a pattern hierarchy. The relationship type implies (1) navigation patterns on the child level are linked with the relationships *same interaction* or *similar interaction*, (2) information patterns are linked with the relationships *similar user interface* or *same user interface*. Even if function patterns are thematically similar, they could sometimes describe different solutions on a more specialized child level (e.g., realize a date input as simple input or as calendar). Therefore, a *same as* or *similar* implication can, but does not have to be applied to function patterns.
4. The use of the relationship *same interaction* implies the use of the relationship *same user interface* or *similar user interface*. Patterns that describe solutions with same interaction concepts at least describe similar or equal user interface elements that are relevant for the interaction.

The 10 introduced rules support the creation of new patterns and the definition of relationships between new and existing patterns. For selecting patterns to be used in one concrete application, the relationships complement, dependency, incompatibility, alternative, and anti-pattern can especially be used to find useful pattern combinations. For software development, interactions and user interface relationships indicate the potential of reusing source code from other (linked) pattern solutions.

6. An Exemplary Set of GI Usability Patterns and Relationships

Figure 3 illustrates eight usability patterns for GI web applications (patterns P2, P5, and P6 were previously described in Table 3). The function pattern *Map visualization* is the center of this pattern collection. It describes the problem that some applications do not provide a visualization function. If users want to visualize time-series data, the map should also provide UI elements for time control, as described in pattern *Time control for maps* (P2). The time control should use a similar interaction element as existing map functions (e.g., similar buttons or sliders). The patterns *Provide map link in result list* (P8) and *Provide map link in details description* (P9) address the problem that, in some applications, the map functionality cannot be executed although it is available. They complement the pattern *Map visualization* (P7) and describe solutions with similarities in interaction and design concepts. Thus, the link in the result list and the link in the detailed description should look the same and should be used the same way.

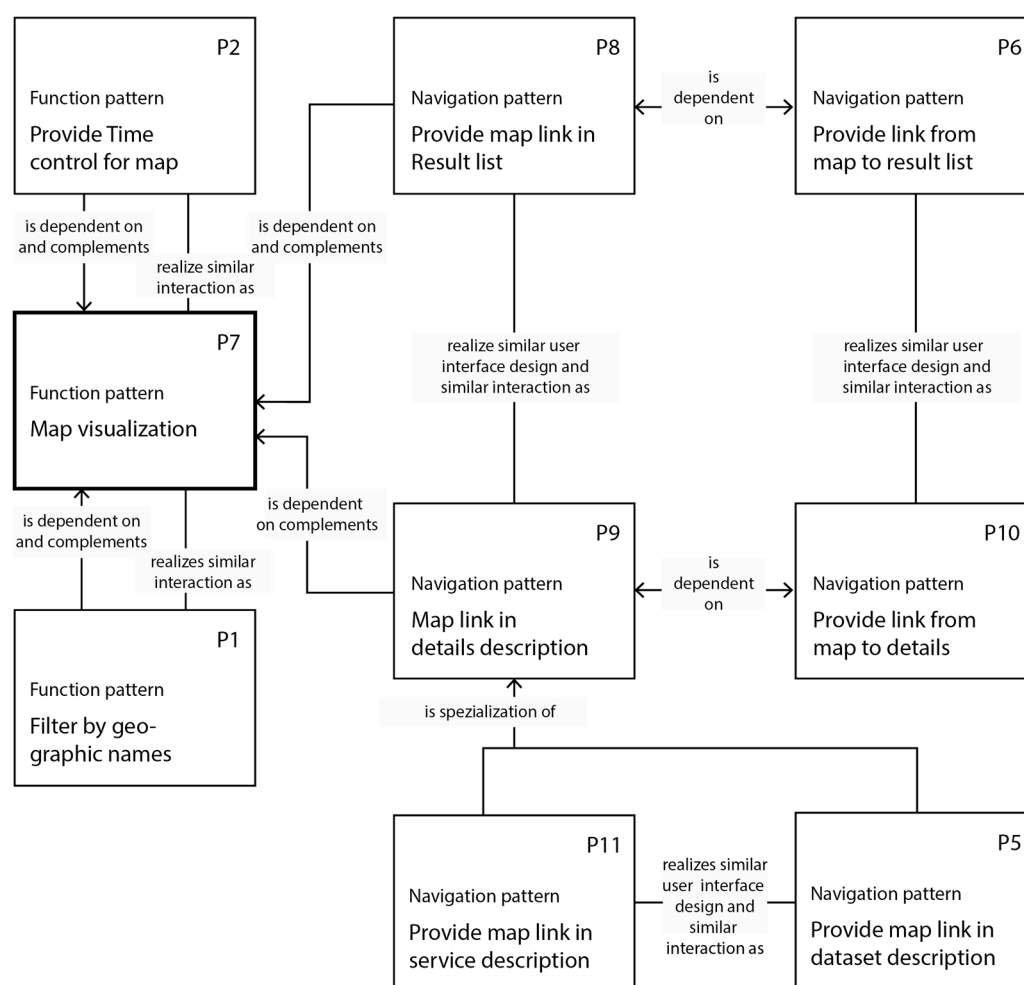
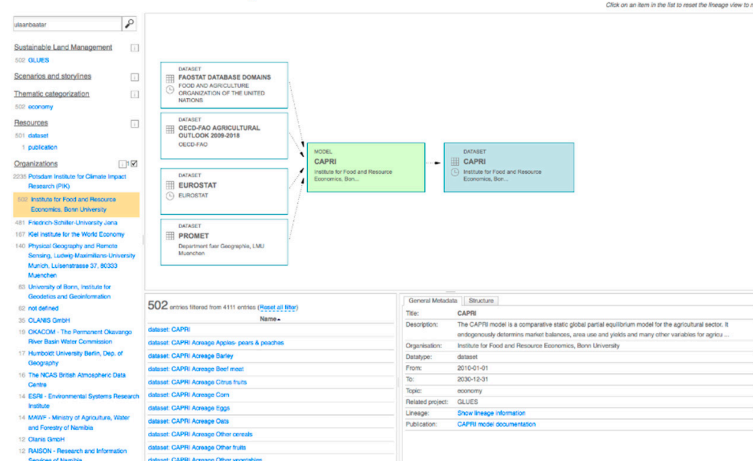


Figure 3. Eight usability patterns for GI web applications and their relationships.

The patterns *Provide visualization of dataset lineage* and *Provide map link in dataset description* illustrate the new structure of the usability patterns for GI web applications (Tables 6 and 7).

Table 6. Usability pattern Visualization of dataset.

Usability Pattern	Visualization of Dataset Lineage (Information Pattern)
Description	Users want to explore relationships of datasets visually as graph and avoid reading (long) texts or scan complex tables.
Solution	<p>Show the dataset lineage as a graph and enable the users to visually explore inputs, processes, and results.</p> <p>Show at least the title of the inputs and results. Add characteristics of the datasets, such as time variance, raster/vector, and characteristics of the process, e.g., source code format, if useful.</p> <p>Metadata elements should be used as presentable information, e.g., International Organization for Standardization (ISO) 19115-2 elements.</p> <p>The size of the graphical objects for inputs should be adopted to the number of visible objects automatically. If too many objects have to be displayed, the inputs should be grouped or shown and hidden by proper interaction methods.</p> <p>The application GeoMetaFacet (http://apps1.glues.geo.tu-dresden.de:8080/GeoMetaFacet/index.html) allows visualization of a simple input–process–output chain for a scientific dataset.</p> <p>The lineage graph uses boxes for datasets and process. White boxes show inputs, while the green box represents a process, in this case, a simulation model, and the blue box shows the output dataset.</p>



Example

The application MetaViz

(<http://geoportal-glues.ufz.de/MetaViz/detail.jsp?id=glues:lm:metadata:dataset:promet>)

allows illustration of a two-step lineage process for a scientific dataset.

For a certain dataset (blue box), its lineage (left part) and usage (right part) are illustrated. Inputs (white boxes on the left) are minimized, because of the amount of inputs.

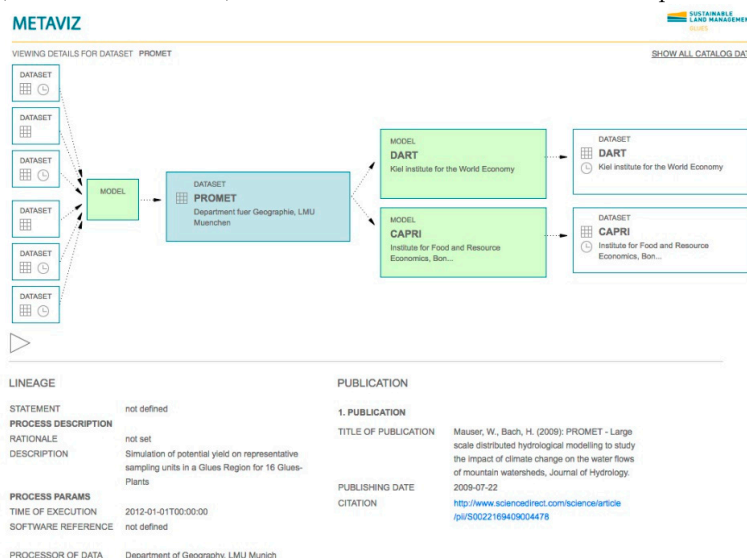
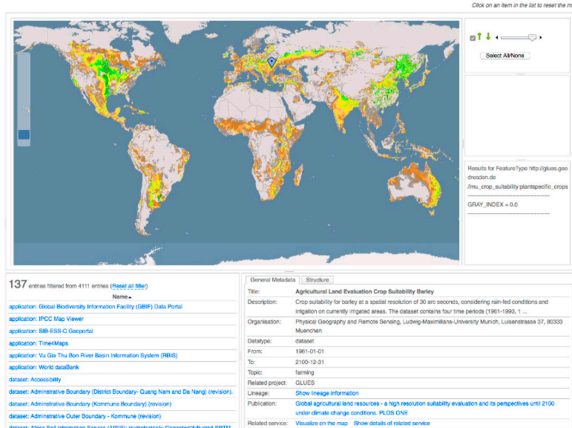


Table 6. Cont.

Usability Pattern	Visualization of Dataset Lineage (Information Pattern)
Context	Content Resource Relationship Task User dataset, metadata dataset–dataset search GI: experts IT: inexperienced users, experts
Rationale	When users work with lineage graphs, they quickly get a brief overview of the dataset’s lineage.
Costs	Implementation cost must be appropriate for the number of provided metadatasets with lineage information. The lineage information should be complex (more than one or two inputs) to make usage of the graph more efficient than the usage of a table.
Related patterns	-

Table 7. Usability pattern Provide map link in dataset description.

Usability Pattern	Provide Map Link in Dataset Description (Navigation Pattern)
Description	When users search for geodata, they often evaluate discovered geodata by examining the related metadata and visualizations on a map. If the evaluated metadata (described contents, information on data provider, etc.) hint to a suitable resource, users typically evaluate the geodata visually, if possible. They need a navigation path from the metadata to the visualization.
Solution	An application that enables users to discover geodata and related metadata should provide a navigation path from geodata descriptions to the related map visualization. Parse metadata to get information about related visualization services internally. The user is not interested in this interim result. Provide a link that enables the user to navigate directly to the visualization of the geodata. Do not ask the user to choose layers if information about related layers is available in the metadata. The application GeoMetaFacet (http://apps1.glues.geo.tu-dresden.de:8080/GeoMetaFacet/index.html) allows linking to the map; “Visualize on the map” is shown in the description of a scientific dataset. If the metadata of a dataset contains information about a related web service, the link “Visualize on the map” is automatically displayed. An alternative UI option is to provide the link to the map via button or icon. The UI should look and feel the same as that in the service description (see related patterns).
Example	 <p>The screenshot shows the GeoMetaFacet application interface. It features a world map with highlighted regions in yellow and orange. Below the map, there is a detailed description of a dataset titled 'Agricultural Land Evaluation Crop Suitability Barley'. The description includes information about the dataset's origin, purpose, and availability. A 'Visualize on the map' link is prominently displayed, allowing users to navigate directly to the map visualization.</p>
Context	Content Resource Relation Task User dataset, metadata, service dataset–service search, visualization GI: inexperienced users, experts IT: inexperienced users, experts
Rationale	When users can navigate directly from a dataset description to the map, they navigate more efficiently (use fewer clicks and need less time) than users who need to find the related service description first. Additionally, a direct link to the map enables users without expertise on dataset–service relationship concepts to open a map visualization.
Costs	Implementation costs must be evaluated regarding the amount of available dataset–service relationships.
Related patterns	Realizes same user interface design and same interaction as pattern <i>Provide map link in service description</i> Is a specialization of pattern <i>Provide map link in details description</i>

7. Conclusions and Future Work

Usability in GI web applications will be a pressing challenge in the upcoming years, e.g., due to software and hardware releases. The usability should be sustainably improved. Therefore, this paper presented an approach to map and categorize usability problems of web applications in SDIs and best practice solutions as usability patterns. We reengineered the existing usability pattern concept to enable a GI context-specific creation and discovery of these usability problems and solutions. Furthermore, we developed pattern types, relationships, and rules on how to use the relationships for the different pattern types to support software engineers in designing a GI web application with a consistent user interface and interaction concept.

Future work is intended to address three topics: (1) the design of a tool that supports users to find suitable patterns; (2) the development of an evaluation concept for pattern use; and (3) the design of a “crowdsourcing” strategy to collect usability patterns in a central web-based knowledge platform.

Usability patterns provide complex usability solutions for GI web applications, which are linked via pattern relationships to a complex pattern network. As described previously, it is not possible to combine all patterns in one software application. A tool, e.g., a recommendation system, could suggest suitable patterns based on existing pattern selections and prevent or at least hint to incompatible pattern combinations. In an extended version, such a tool could generate source code for the user interface based on the chosen patterns. A formal language of the usability patterns and a rule set on how to combine patterns could serve as a basis for such a tool.

In software engineering, the evaluation of applied solutions is vital and, in best cases, verifies the effectiveness of these solutions. Software improvements based on these solutions should become measurable with suitable metrics. Results of such measurements serve as the basis for cost–benefit analysis for suggested solutions. Hence, future research is required to analyze whether software evaluations help find suitable usability patterns and how to evaluate their successful application. The usability pattern structure could then also contain information about the usability metrics and the improvements, which could be achieved by the use of the proposed pattern.

The concept of usability patterns for GI web applications enables users to collect usability problems and solutions for various use cases in the GI domain. Currently, the concept does not contain strategies for a community-driven collection of these problems and their solutions as usability patterns. Usability patterns could serve as mediators between designers, developers, usability, and domain experts. Thus, this heterogeneous user group should be enabled to collaborate with GI usability patterns. This includes reviewing, rating, and complementing existing patterns, e.g., by adding links between patterns (with pattern relationships), keywords (structured as sub-attributes in the pattern context), or providing best practice solutions or anti-pattern examples. A community-driven approach to collect patterns, e.g., in a pre-structured wiki, could reflect the perspectives of different types of pattern users in an appropriate way.

Funding: This research received no external funding.

Acknowledgments: The author thanks Lars Bernard, Benno Schmidt and Wolfgang Reinhardt for their valuable input. The comments and suggestions of two anonymous reviewers and the journal editors helped to improve this submission and are gratefully acknowledged.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Appendix

Additional rules about usability pattern type and relationship combinations

1. The interaction relationships *same interaction* and *similar interaction* cannot be used to link information patterns, since information patterns exclusively describe solutions on how to prepare and present information with suitable user interface elements (and do not describe interactions).
2. The relationship *aggregation* divides a problem or solution into several sub-problems or solutions. Problems on the child level need to be considered individually (solve the sub-problems), but also

need joint consideration to ensure a consistent interaction or user interface concept. Thus, the usage of the relationship *aggregation* implies the use of *similar interaction* or *similar user interface* relationships on the child level.

3. The use of the relationship *complement* implies that complementary patterns realize a similar interaction or user interface concept. If the complementary patterns are navigation or function patterns, they describe interactions to solve a common task and, thus, use the same interaction types and realize a similar, but not same, interaction concept. Accordingly, complementary information patterns describe similar user interface concepts.
4. The use of the relationship *complement* implies that if the complementary and the complemented pattern are broken down into sub-patterns, the resulting pattern hierarchies (two trees) have the same structure. On each child level, the pattern on equivalent tree positions can be linked.
5. The use the relationship *alternative* implies that the alternative patterns do not describe solutions with the same interaction or user interface elements. Thus, a combination of the relationships *alternative* and *similar interaction* or *similar user interface* cannot be used to link patterns.
6. The use of the relationship *incompatibility* implies that the relationships *same interaction* or *same user interface* cannot be used to link incompatible patterns. In any case, the incompatibility refers to the lack of realizing a consistent interaction or user interface concept with patterns and, thus, the relationship can be interpreted as the opposite of the relationships *same interaction* or *same user interface*.

References

1. Traynor, C.; Williams, M.G. Why are geographic information systems hard to use? In Proceedings of the Mosaic of Creativity Conference on Human Factors in Computing Systems, Denver, CO, USA, 7–11 May 1995; pp. 288–289. [\[CrossRef\]](#)
2. Slocum, T.A.; Blok, C.; Jiang, B.; Koussoulakou, A.; Montello, D.R.; Fuhrmann, S.; Hedley, N.R. Cognitive and Usability Issues in Geovisualization. *Cartogr. Geogr. Inf. Sci.* **2001**, *28*, 61–75. [\[CrossRef\]](#)
3. Haklay, M.; Tobon, C. Usability Engineering and GIS: Towards a Learning-improving Cycle. In Proceedings of the 1st Annual Public Participation GIS Conference, New Brunswick, NJ, USA, 20–22 July 2002; pp. 1–18.
4. Koua, E.L.; Kraak, M.J. A usability framework for the design and evaluation of an exploratory geovisualization environment. In Proceedings of the IEEE International Conference on Information Visualisation, London, UK, 14–16 July 2004; pp. 153–158.
5. Marsh, S.L. Using and Evaluating HCI Techniques in Geovisualization. Ph.D. Thesis, City University Information Science, London, UK, 2007.
6. Haklay, M.; Tobon, C. Usability evaluation for web based GIS: Ensuring that portals are useful. In Proceedings of the Geographical Information Portals, London, UK, 9–11 April 2003.
7. Ellis, C.D.; Quiroga, C.; Shin, S.-Y.; Pina, R.J. GIS and Human-centered Systems Design: Using Ethnographic Data Collection and Analysis Methods to Design a Utility Permitting Support System. *URISA J.* **2007**, *15*, 5–22.
8. Wachowicz, M.; Vullings, W.; Bulens, J.; van den Broek, M. Uncovering the Main Elements of Geo-Web Usability. In Proceedings of the 8th AGILE Conference, Lisbon, Portugal, 26–28 May 2005.
9. Haklay, M.; Zafiri, A. Usability Engineering for GIS: Learning from a Screenshot. *Cartogr. J.* **2008**, *45*, 87–97. [\[CrossRef\]](#)
10. Komarkova, J.; Jakoubek, K.; Hub, M. Usability Evaluation of Web-based GIS—Case Study. In Proceedings of the Conference: iiWAS'2009—The Eleventh International Conference on Information Integration and Web-based Applications and Services, Kuala Lumpur, Malaysia, 14–16 December 2009; pp. 557–561.
11. Lathrop, R.; Auermuller, L.; Trimble, J.; Bognar, J. The Application of WebGIS Tools for Visualizing Coastal Flooding Vulnerability and Planning for Resiliency: The New Jersey Experience. *ISPRS Int. J. Geo-Inf.* **2014**, *3*, 408–429. [\[CrossRef\]](#)
12. He, X.; Persson, H.; Östman, A. Geoportal usability evaluation. *Int. J. Spat. Data Infrastruct. Res.* **2012**, *7*, 88–106. [\[CrossRef\]](#)

13. Bulens, J.; Vullings, W.; Houtkamp, J.; Vanmeulebrouck, B. Usability of Discovery Portals. In Proceedings of the AGILE Conference on Usability of Discovery Portals, Leuven, Belgium, 14–17 May 2013.
14. Resch, B.; Zimmer, B. User Experience Design in Professional Map-Based Geo-Portals. *ISPRS Int. J. Geo-Inf.* **2013**, *2*, 1015–1037. [[CrossRef](#)]
15. Henzen, C.; Bernard, L. Usability für Geoportale am Beispiel der Konzeption des Geoportal Sachsen. *Kartogr. Nachr.* **2013**, *5*, 262–269.
16. Joshi, A.; de Araújo Novaes, M.; Machiavelli, J.; Iyengar, M.S.; Vogler, R.; Johnson, C.W.; Zhang, J.; Hsu, C.E. Usability Evaluations of an Interactive, Internet Enabled Human Centered SanaViz Geovisualization Application. *HCI* **2014**, *8527*, 723–734. [[CrossRef](#)]
17. Çöltekin, A.; Heil, B.; Garlandini, S. Evaluating the effectiveness of interactive map interface designs: A case study integrating usability metrics with eye-movement analysis. *Cartogr. Geogr. Inf. Sci.* **2009**, *36*, 5–17. [[CrossRef](#)]
18. Nivala, A.-M.; Brewster, S.; Sarjakoski, T.L. Usability Evaluation of Web Mapping Sites. *Cartogr. J.* **2008**, *45*, 129–138. [[CrossRef](#)]
19. Skarlatidou, A.; Haklay, M. *Public Web Mapping: Preliminary Usability Evaluation*; GIS Research: Nottingham, UK, 2005.
20. Komarkova, J.; Visek, O.; Novak, M. Heuristic Evaluation of Usability of GeoWeb Sites. In *Web and Wireless Geographical Information Systems W2GIS*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4857, pp. 264–278.
21. Henzen, C. Usability von Webanwendungen in Geodateninfrastrukturen. *GIS Sci.* **2018**, in press.
22. Fincher, S.; Windsor, P. Why Patterns are not Enough: Some Suggestions Concerning an Organising Principle for Patterns of UI Design. 2000. Available online: <https://www.semanticscholar.org/paper/Why-patterns-are-not-enough-%3A-some-suggestions-an-Fincher/a98201d647cb3799139fbb1a924937531396e7c5> (accessed on 6 November 2017).
23. Röder, H. *Usability Patterns*; Cuvillier Verlag: Göttingen, Germany, 2012.
24. Aronoff, S. *Geographic Information Systems: A Management Perspective*; WDL Pubns: Ottawa, ON, Canada, 1991.
25. Chrisman, N. *Exploring Geographic Information Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2002.
26. ISO/TC 211. *DIN EN ISO 19115-1 Geoinformationen—Metadaten*; ISO: Geneva, Switzerland, 2014.
27. ISO/TC 211. *DIN EN ISO 19119—Geographic information—Services*; ISO: Geneva, Switzerland, 2015.
28. Bernard, L.; Mäs, S.; Müller, M.; Henzen, C.; Brauner, J. Scientific geodata infrastructures: Challenges, approaches and directions. *Int. J. Digit. Earth* **2013**, *7*, 613–633. [[CrossRef](#)]
29. Henzen, C.; Mäs, S.; Bernard, L. Provenance Information in Geodata Infrastructures. In *Geographic Information Science at the Heart of Europe, 2013. Lecture Notes in Geoinformation and Cartography*; Vandenbroucke, D., Bucher, B., Crompvoets, J., Eds.; Springer: Berlin, Germany, 2013; pp. 133–151.



© 2018 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).