2014 47th Hawaii International Conference on System Science

Applying Design Patterns in URI Strategies - Naming in Linked Geospatial Data Infrastructure

Sonya Abbas and Adegboyega Ojo
Digital Enterprise Research Institute, National University of Ireland, Galway
Galway, Republic of Ireland
{firstname.lastname}@deri.org

Abstract

The centrality of Uniform Resource Identifiers (URI) as names in Linked Data initiatives has led to the development of guidelines and best practices by World Wide Web Consortium (W3C) and other experts groups on how to design "good" URIs in general and for the government domain in particular. However, these URI design guidelines have had limited pragmatic value for several reasons including the underspecified nature of the rules, weak elaboration on nature of problems addressed and consequences of prescribed design decisions. With no conceptual or rigorous underpinning for existing design rules, checking for internal consistency or coverage when developing URI strategies is difficult. We tackle these problems in this paper by: 1) consolidating existing URI design rules, 2) distilling core URI design aspects or facets from these rules and 3) abstracting the rules into a set of consistent URI Design Patterns specifications. This process resulted in 8 Design Patterns from an initial set of 37 URI design rules. Following this, we show how the design patterns could be employed in developing a URI strategy to support the realization of a Linked Spatial Data Infrastructure. We conclude with an evaluation of the URI design patterns and implications of our work.

1. Introduction

The use of Uniform Resource Identifiers (URI) as names of physical and abstract concepts, and mechanisms for linking these entities constitute a fundamental tenet of Linked Data [1]. To support the design of persistent URIs in publishing linked data, several guidelines and best practices have been developed since the first guideline provided by Tim Berners-Lee in [2]. Typically, these guidelines identify the types of URIs and offer a set of informal rules to constrain the generation of URIs names.

In addition to the initial set of general URI design rules, sector specific rules have also been published. For instance, W3C provides in [3] a set of checklist for constructing Government URIs, while [4] contains the United Kingdom Public Sector Guidelines for designing URIs. Lately, domain-specific URI design needs such as those for the Geospatial domain have emerged to support the transition to Linked Data. In this sphere, guidelines or URI design principles are yet to be provided.

However, the reality is that most of the existing URI design guidelines have had limited pragmatic value for several reasons including the abstract nature of the rules [3], weak elaboration on nature of problems addressed and consequences of prescribed design decisions [5]. In addition, we observe that none of the existing guidelines have any conceptual or rigorous underpinning for the design rules as basis for determining the internal consistency or coverage of these rules when developing URI strategies.

Our goal in this paper is to address some of these problems by offering a rigorous process for describing the URI design rules and a framework for evaluating the adequacy of these rules in constructing large scale distributed URI naming scheme. To provide a rigorous framework for documenting design rules, we adopt the Design Pattern Approach. Design Patterns are general reusable solutions to commonly occurring problems within a given context [6]. Thus we aim to succinctly transform existing design rules as solutions to recurring URI problems. As part of the process for translating these rules into URI Design Pattern, we consolidate and streamline existing rules. Consequently, the resulting URI Design Patterns do not only provide a consistent and rigorous approach for documenting existing URI design rules, but also offer useful abstraction over the current unwieldy number of rules.

The rest of the paper is organized as follows: Section 2 presents the background of URI design and design patterns. Our approach of designing URI patterns is presented in Section 3. We present the obtained URI Design Patterns in Section 4 and applied them in a URI strategy for a Linked Spatial Data Infrastructure (LSDI) initiative in Section 5. Evaluation of the URI Design Patterns is presented in Section 6, while discussions and concluding remarks are presented in Section 7.

2. Background

2.1 URI Design and Linked Geospatial Data

Geospatial data describing information tied to some locations on Earth's surface [7] constitute an important and rapidly growing category of government data assets. This category of data is considered critical for planning, policy making and delivering innovative

978-1-4799-2504-9/14 $31.00 © 2014 IEEE
DOI 10.1109/HICSS.2014.265

IEEE computer society

location based services in domains including disaster mitigation, public health, geology, civil protection and agriculture [8].

An important aspect of managing geospatial data is the provision of the so-called Spatial Data Infrastructure (SDI); an information infrastructure providing access and enabling interoperability among spatial information based on standards, policies, regulations and coordination mechanisms [9].

The success of flagship Linked Open Data initiatives has raised the possibility of leveraging linked data for enabling global access to spatial data currently managed within national and regional SDIs. Many initiatives and studies on the adoption of linked data and semantic web in developing SDI have grown [10]. A reference model for Linked SDI (LSDI) has been developed to provide guideline and prescribes policy, standards and community requirements for a successful linked geospatial data strategy [11].

Given that URI design is one of the first steps for Linked Data initiatives, developing a URI strategy for a Linked SDI framework is a starting point for implementing the framework. A URI is a compact sequence of characters identifying an abstract or physical resource [12]. Tim Berners-Lee in [2] argued that resource identifiers should not only just provide descriptions for people and machines, but must be designed with simplicity, stability and manageability in mind. These URI design requirements are critical for the success of any linked data initiative. Hence, the UK Cabinet Office [13] addressed a set of challenges related to the need of establishing identifiers that will persist over time allowing data (re)users to extract, link and combine data using these identifiers.

Since the initial URI guidelines provided by [2], many technical guides, standards, specifications and best practices for URI design such as [2, 12, 14] and [4] have emerged. Entities that have contributed to URI guidelines and best practices include: 1) governments such as UK Public Sector [4], 2) semantic web communities such as W3C [2, 14] and 3) academia such as the work presented in study on persistent URIs [12]. Significant number of URI design rules are directed at naming a URI in a way that guarantee its persistence and readability [3, 4, 12]. Other rules prescribe how to manage the designed URIs, for instance proposing HTTP 303 or hash URIs as mechanisms for dereferenceable URIs [2], recommending human readable representations [4] and designing URIs using different languages [3]. The remaining rules prescribe how to handle linking, implementing and consumption aspects such as following content negotiation to resolve to the most appropriate representation URI for rendering information in a format as requested by the client [4] and using meaningful and mnemonic components in URI [14]. The genealogy of URI Design guidelines and best practices resulting from our analysis is shown in Figure 1.

However, our review of existing and current work on URI design rules reveals that these rules: 1) are in many cases too abstract and underspecified making their implementation difficult, e.g. "*Ensure that URIs do not have to change with every re-design*", 2) do not elaborate on nature of problems addressed and consequences of prescribed design decisions, e.g. "*if no auto-increment is to be allowed in generating URIs, how will URI's for a large dataset be generated?* 3) similar rules across documents without any explicit references to related rules in other sources, e.g. *"one rule specifying no file extension in URI and another rule from a second source indicating no mutable element in URI"*; 4) inconsistent when consolidated across different sources, e.g. "*one rule indicating having the right domain in URI and another rule specifying not having domain information in URI"*.

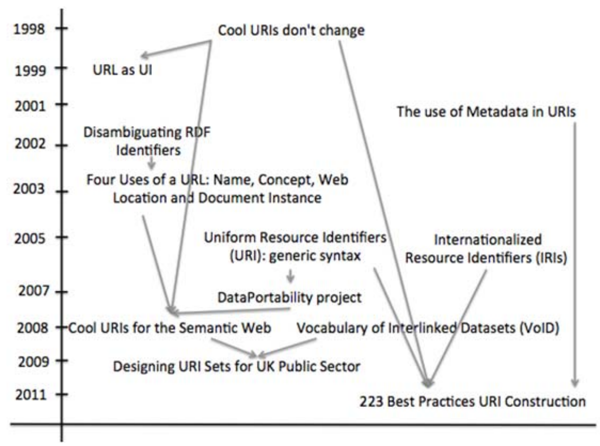

Figure 1: URI Designs' Dependencies

2.2 Design Patterns

Design patterns are a well–known method of reuse, applicable on analysis and design models as well as on implemented code in Software Engineering [15]. According to Alexander, "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" [6].

In general, a pattern has four essential elements: 1) the pattern name, 2) the problem, 3) the solution, and finally 4) the sequences [6].

Authors of [16] outline where the pattern idea comes from, and how it has been adapted for use in other disciplines. Application of design patterns span domains like urban architecture [17, 18] human-computer interaction (HCI) [19, 20], application domain [21] and software engineering [6].

Current existing URI rules do not provide enough details to easily determine the applicability or more importantly the consequence of using these rules. Since URI rules are design rules, the use of design patterns become an ideal approach for describing such best practices or design principles.

3. Approach

The problem being addressed in this work is how to increase the pragmatic value of existing URI design rules. To address this problem, we propose the use of Design Patterns [6] as a means for abstracting and rigorously documenting existing URI design rules. As the first step we identified major sources of URI guidelines such as [2–4], [12]. Second, we extracted all design rules in these documents and assigned unique numbers to each distinct rule. This step produced a catalog of 37 rules as shown in Table 2 below. Third, we identified the aspects of URI design that are addressed by each of the rules. This resulted in five (5) aspects – Naming, Linking, Implementation, Management and Consumption. Fourth, we manually clustered these 37 rules into groups of related or equivalent rules, producing eight different clusters. Each cluster consists of rules considered equivalent or could be arranged in a generalization hierarchy, e.g. see Figure 5.

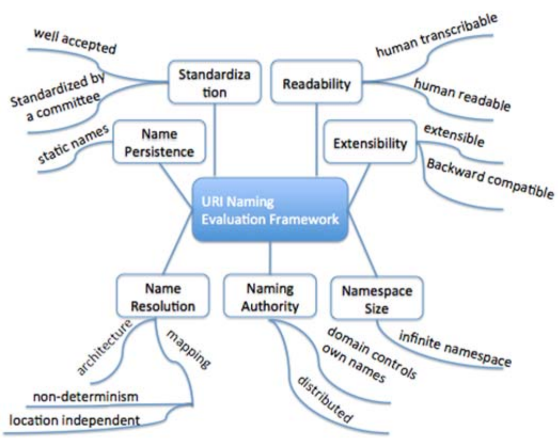

Figure 2: a set of criteria in naming schemes

The fifth step of our methodology involves documenting each of the eight clusters as a design pattern specifying the problem addressed by the pattern, solution proffered to the problem, consequence of the solution, the original URI rules contributing to definition of the pattern and the aspects of URI design impacted by the pattern and an example.

We evaluate the resulting URI design patterns three steps. First we check for the coverage of key issues in a naming scheme, we employed the set of criteria developed in [22]. Figure 2 summarizes the evaluation criteria. Second, we argue for the internal consistency of the resulting design patterns by checking for conflicts in in the design rules. Third, we seek feedbacks for validating and refining the URI design patterns from Linked Data experts.

4. URI Design Patterns

4.1 URI Types

URI design rules are associated with different URI types. Our analysis of existing rules produced six different but related set of URI types. Each of these types is described below.

1) *Thing:* represents any real-world entities or physical objects like people and cars that cannot be found on the web, except information about them [4, 14].
2) *Concept:* represents abstract ideas and non-physical entities in the world [14].
3) *Resource:* represents a documents on the web providing information about real-world things including objects and concepts [4].
4) *Representation:* represents one format of the resource. Each available resource format may be separately named by a Representation URI [4].
5) *Hierarchical:* represents a natural hierarchy exists between a set of resources [23].
6) *Onto:* represents a resource providing the meaning of things, concepts and relationships [4].

Table 1: URI Types

URI Type	Source				
	[3]	[4]	[14]	[23]	[24]
Thing	X	X	X		X
Concept	X				X
Resource	X	X	X	X	
Representation	X	X			
Hierarchical/List		X		X	
Ontological		X			

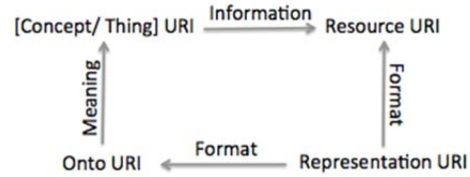

Figure 3: Connections between URI Types

The Table above provides the sources of the identified URI types, while Figure 3 indicates the relationship among these URI types.

4.2 URI Design Aspects

On analyzing existing rules, we identify a number of design perspectives or aspects. The design aspects are particularly useful in structuring the design rules. In general, group of patterns focuses on a specific perspective. These perspectives or aspects together were found to cover the URI life cycle; from naming through linking and implementation to management and consumption. The identified rule aspects are discussed below.

1) *Naming*: associated with rules that define the URI path itself and how to construct URI identifiers based on defined patterns.
2) *Linking*: deals with URIs rules that are related to linking one URI to another as well as to real-world object objects and concepts [14].
3) *Implementations*: associated with rules specifying how URIs will be implemented. It covers issues such as mechanisms, languages, standards and technologies for implementing URIs.
4) *URI Management*: deals with rules on management and governance of available URIs and URI sets.
5) *Consumption*: covers how to design URIs to enable effective use and re-use.

Rules could be associated with one or aspects as shown in Section 4.3. Overall, most of available rules are related to management and construction of URIs names for persistence and re-use as shown in Figure 5.

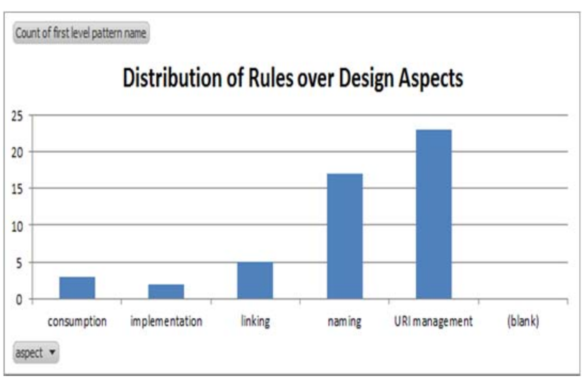

Figure 4: Distribution of Rules over Design Aspects

After studying different URI types and the design aspects, we streamlined the obtained 37 rules into the design patterns described in the next section.

4.3 Patterns

This section presents the patterns obtained from clustering the 37 rules presented above in Table 1. Clusters of rules were obtained by manually constructing generalization hierarchy and equivalence among rules (e.g. see Figure 4). Each of resulting clusters is documented as a design pattern specifying: 1) the problem addressed by the pattern, 2) the proposed solution, 3) consequence of the design decision, 4) sources of the rules streamlined to obtain the patterns, 5) aspects impacted by the pattern and 6) an example of the application of the pattern. The obtained 8 design patterns are: 1) De-referenceable URIs, 2) Guarantee URI Uniqueness, 3) Human Readable URI, 4) Immutable URI Elements, 5) URI Stability, 6) URI Longevity, 7) URI Multiple Representations, and 8) URI Quality.

Table 3: De-referenceable URIs Pattern

Dereferenceable URIs		
Problem		How to resolve a URI for a real-world objects or concept?
Solution		Use HTTP 303, hash URIs or combination of both
Consequences		HTTP: it defines only web resources and web documents. HTTP 303: leads to a number of HTTP round-trips. Hash URI: A client interested only in #product123 for example will inadvertently load the data for all other resources.
Source		[2–4, 12]
Aspect	Implementation	HTTP URIs HTTP 303 URIs Hash URIs Hash+303 URIs
Example		Combining Hash and 303 URI http://www.example.com/bob#this

Table 2: 37 URI Rules Extracted from Existing Guidelines

No.	Rule name	Description	Source
1	Content Negotiation	To return the most appropriate representation for a resource, use content negotiation mechanism	[4, 14]
2	Multiple URIs for Object Representations	When there are multiple representations for a URI referring to a single real-world object, link them	[12]
3	Multiple URIs for Single Document	When there are multiple URIs referring to a single document, provide a means of discovery.	[4]
4	Consistent URI representations	If persistence is required then provide consistent representations for URIs.	[3]
5	Linking URIs for a single real world object	If there are many URIs related to a single real-world object then explicitly link them with each other to help the consumers understand their relations.	[14, 24]
6	Readable URI path	Make URI path structure human readable and understandable.	[4]
7	No ownership in URI	To make URI less susceptible to change, avoid including the name of the organization or project that minted it in the URI.	[12]
8	No version in URI	To make a URI stable, avoid including version numbers and status information in the URI.	[12]
9	No department or agency name in URI	To make URI re-usable, avoid including any department or agency name in the URI	[4]
10	No mutable elements in URI	To make URI stable and reusable, do not include mutable elements in URI.	[2]
11	No file extension in URI	To avoid brittle URI, avoid URI names based on technologies.	[2–4]
12	No technical implementation in URI	To avoid brittle URI, avoid exposing the technical implementation of a URI in its structure	[12]
13	Reusing existing identifiers in URI	Reuse existing identifies in stable or persistent URIs	[4, 14]
14	No auto-increment in URI	When generating URIs automatically for large datasets, avoid the use of auto-increment as it could lead to the having same URIs for different objects or concepts.	[12]
15	Stable URIs	Ensure that URIs are do not have to change with every redesign	[12]
16	No topic in URI	If there is topic in URI to delegate the responsibility for sub-parts of a URI space then It's a medium term solution and has serious drawbacks in the long term	[2]
17	Consistent URI path	To obtain a consistent URI path, indicate URI type explicitly to help looking up data on the web about the resource that the URI names.	[4]
18	Pattern in URI	If the URI should be persistent then define patterns and follow them to name URIs	[12]
19	HTTP 303 URIs	When there is a generic documents then use a special HTTP status code 303 to give an indication that the requested resource is a generic document.	[12]
20	Hash URIs	When URI cannot be retrieved directly, use a hash symbol ("#").	[2]
21	Hash+303 URIs	If there is large dataset then use a combination of hash and 303 to allow, it to be separated into multiple parts and have an identifier for a non-document resource.	[2]
22	HTTP URIs	When defining web resources and web documents, use HTTP URIs.	[2, 4]
23	Html human-readable representation URI	To human-readable representation for a URI object, use HTML.	[4]
24	Meaningful or familiar components in URI	When URI is designed to be remembered by people then use meaningful or familiar components in URI.	[23]
25	Machine-readable representation for the resource	When a resource is to be understood by machine, provide machine-readable representation for it.	[3, 4]
26	Publish URI Longevity	To guarantee persistence of URIs set, publish the expected longevity and potential for re-use.	[4]
27	10 year persistence for reusable URI set	If the URI set is be promoted for re-use, design it for at least 10 years.	[4, 14]
28	Different languages for designing URIs	When there are multilingual URIs then use more than RFC 3986 for designing URIs	[3]
29	Right domain in URI	Use the right domain in URL to make the customer confident in using them.	[4]
30	No Domain in URI	Hide domain information such as web servers inside one apparent web server using redirection and proxying to avoid destroying many links during changing.	[2]
31	URI International domain	If you deal with multilingual domain names, use Internationalized Domain Name or IDN.	[3]
32	Short and mnemonic URI	If the goal is indicating the URI set quality then use short and mnemonic URI.	[14]

33	Metadata for URI	Track the following information about URIs for each document: authorization, authentication, data quality characteristics using a common vocabulary, its acceptable distribution, creation date and ideally its expiry date as metadata.	[2, 4]
34	URI dedicated service	If the URIs is designed to be persistent and independent of the data, use URI dedicated service	[12]
35	Decision of reusing URI set	If the set is promoted for re-use by other parts of government and/or the public then make it clear.	[4]
36	Naming real- world 'Thing' instances	If there is a department or agency responsible of real-world thing then it should be responsible for defining it and naming instances of it.	[4]
37	Abstract URI space	If we have to move the URI files then separate URI space from the used implementation	[2]

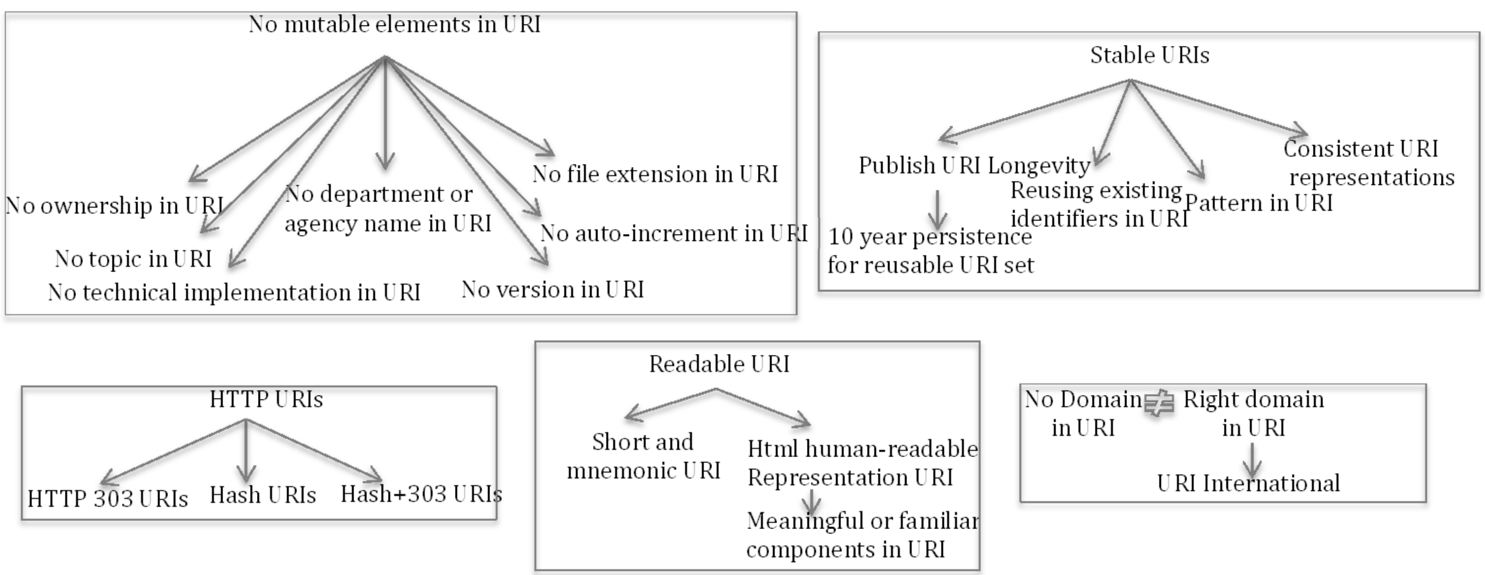

Figure 5: Examples of Design Rule Hierachies

Table 4: Guarantee URI Uniqueness Pattern

Guarantee URI uniqueness		
Problem		How to ensure that naming URIs automatically does not lead to same URIs for different objects?
Solution		Avoid using auto-increment in generating URIs
Consequences		Difficult to name URIs automatically in large datasets
Source		[12]
Aspect	Naming	No auto-increment in URI
Example		Using unique characteristics of the object to generate its URI, for instance a hash value.

Table 5: Human Readable URI Pattern

Human readable URI		
Problem		How to provide a human readable URI path?
Solution		Make the URI path structure readable so that a human has a reasonable understanding of its contents.
Consequences		The may require using changeable in URI path
Source		[4, 25]
Aspect	Naming	Readable URI path
	Consumption	Meaningful or familiar components in URI
Example		http://education.data.gov.uk/doc/school/78

Table 6: Immutable Elements in URI Pattern

Immutable Elements in URI		
Problem		How to design stable, reliable and reusable URIs?
Solution		Avoid using in URIs names of the organization, department, agency name project, version numbers, status information, topic, authors name, subject, status, access, file name extension, query string, technical implementation, software mechanisms or sessions tokens and hide many web servers inside one apparent web server.
Consequences		This will make the URI less human comprehensible and readable.
Source		[2–4, 12, 14]
Aspect	Naming	No stating ownership in URI No version in URI No department or agency name in URI No topic in URI No mutable elements in URI No file extension in URI No query strings in URI No technical implementation in URI
	URI Management	Domain in URI
Example		http://www.nsf.gov/cgi-bin/pubsys/browser/odbrowse.pl

Table 7: URI Stability Pattern

URI Stability		
Problem		How to design a stable URI?
Solution		Create URIs that are not dependent on the design of the information systems using them.
Consequences		
Source		[2, 3]
Aspect	Naming	Ability to use URIs after next redesign.
	Implementation	Different languages for designing URIs
Example		Like removing all changeable element from the URI.

Table 8: URI Longevity Pattern

URI Longevity		
Problem		How to increase URI set reusability, and increase confidence in consumers based on is longevity?
Solution		Publish the expected longevity, and potential for re-use and use the domain that conveys an assurance of quality and longevity.
Consequences		How precise should the temporal properties of a specific URI be? What if less than 10 years is required a URI?
Source		[4, 14]
Aspect	URI Management	10 years persistence for URI. Available expected longevity for URI. 10 years persistence for reusable URI set.
	Consumption	Right domain in URI

Table 9: Multiple Representations URIs Pattern

URI Multiple Representations	
Problem	How to discover available representation URIs for specific document?
Solution	Link all URIs related for a single real-world object explicitly and provide a means of discovering each of the available Representation URIs.
Consequences	Cost of storing and managing the representation URIs and links between them could be high.
Source	[3, 4, 12, 14, 24]

Aspect	Bind	Linking multiple representations for a URI Discovering available representations for a URI URI consistent representations Linking URIs for a single real world object
	Implementation	Content Negotiation
	URI Management	Html human-readable representation
Example		Supposing are two types of documents describing an entity, e.g. a distinct representation URI such as HTML: http://transport.data.gov.uk/doc/road/M5/junction/24/doc.html and the other representation URI as: http://transport.data.gov.uk/doc/road/M5/junction/24/doc.pdf. These two representations URIs must ne linked.

Table 10: URI Quality Pattern

URI Quality		
Problem		How to improve the quality of URIs?
Solution		Provide metadata about URI for instance its authorization, authentication, data quality characteristics using a common vocabulary, acceptable distribution, its creation date and ideally its expiry date.
Consequences		How to ensure they are sufficient to support a 'Web of Data', where each individual statement can be queried and linked.
Source		[2, 4, 14]
Aspect	URI Management	Metadata for URI. Patterns in URI.
	Consumption	Short and mnemonic URI.
Example		http://{domain}/{type}/{concept}/{reference}

4.4 Associating Patterns with Aspects

Extracted patterns cover three main types of design concerns - temporal, usability and functional. The immutable elements in URI, URI stability and longevity provide the temporal aspect. Usability is covered by human readable URI, URI quality and patterns in URI whereas de-referenceable URI and URI multiple representations provide the functional aspect.

To provide some guidance on when to use these patterns, we map to them the lifecycle phases of a URI design process discussed in section 4.2. The result is presented in Table 11.

Table 11: URI Phase Model

Naming	Guarantee URI uniqueness, Human readable URI, Immutable Elements in URI.
Linking	URI Multiple representations.
Implementation	Dereferencable URI, URI stability, URI Multiple representations.
Management	Human readable URI, Immutable elements in URI, URI longevity, URI quality.
Consumption	Human readable URI, URI quality, URI longevity.

Each of the design patterns discussed in Section 4.3 also includes the lifecycle it URI design phase supported.

5. Applying URI Patterns to Linked SDI

5.1 Overview

We show in this section as example how the patterns can be applied in the development of a URI strategy in the Linked SDI domain. The Linked SDI contains five core dimensions that includes Data, Network, People, Standards and Policy and its sub dimensions [11]. The overall reference architecture is presented in Figure 6. The data dimension describes the typical categories of datasets maintained by SDIs. Network Access dimension consists of services and clients applications. People dimension includes all stakeholders, both users and producers of spatial information interacting with the SDI. Infrastructure Standards dimension constitutes an important SDI component which provides technical guidance and enforceable rules. Policy dimension specifies important decisions on core aspects of the SDI including governance, role assignment to memberships, quality and funding.

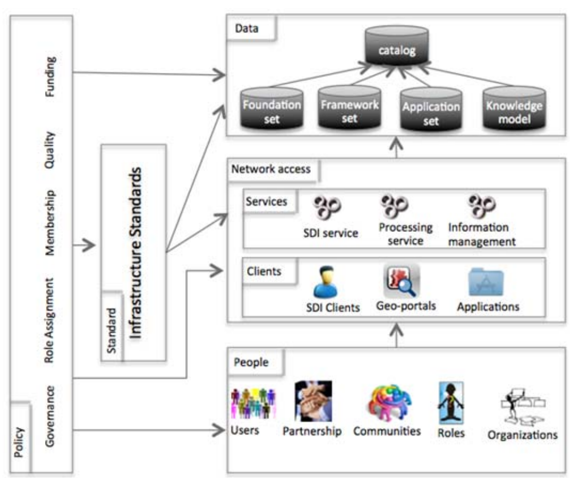

Figure 6: SDI Base Reference Architecture

The implementation of this LSDI reference architecture in requires the development of URIs for each of its of elements [1].

5.2 URI Types

As a first step, we identify the required URI types for each of the LSDI infrastructure elements. The Data component can be represented mainly by resource/data URI type. The People elements are mapped to physical/object URI type whereas services, standards and policies were mapped to Resource URI.

To represent the geospatial data element of LSDI requires URIs that reflects its hierarchical features. Hence, list/hierarchical URIs proposed by UK public sector could be used where list URI defines the identifier URIs that are contained within a set [4]. Table 12 shows how what type of URI could be used for identifying different LSDI entities.

Table 12: Mapping URI types to LSDI RA components

Entity\URI Type	Concept/ Thing URI	Onto URI	Resou rces URI	Represent ation URI
Dataset			X	X
Ontology		X		X
Service			X	X
Application			X	X
Organization	X		X	
Community	X		X	
Stakeholder	X		X	
Standard	X		X	X
Policy	X		X	X

5.3 URI Patterns in LSDI Context

Finally, determine how the URI design patterns described in Section 4 could be applied in elaborating a URI strategy for the Linked SDI. First we categorize the eights patterns into two categories: 1) domain invariant - patterns which are domain agnostic, for instance the De-referenceable and Human Readable URI patterns, and 2) domain variant patterns that are required to be elaborated in the context of different domains, for instance the Hierarchical URIs.

We summarize below some of the required elaborations for the different URI lifecycle stages based on Table 11 in Section 4.4:

- *Naming* – given the many registers and databases of physical objects and facilities maintained in the geospatial domain, more specific rules and guidelines on how to automate the generation of the URIs is important. Since the dominant practice involves the use serial numbers as identifiers in these register, how to avoid mutable elements in generated URI is an issue to consider.
- *Linking* – given the many possible geospatial representations for a single physical space, for example a river may be represented as a line in one context and as a polygon in another. These two requires different representation and thus requires elaboration on how to bind the physical space to URIs for these multiple representations. There is also the need to provide further guidelines on how to link these different representations.
- *Implementation* - guidelines will among others indicate the order of preference among multiple representations of the different physical objects when dereferencing.
- *Management* – the government organizations responsible for the LSDI needs to elaborate the different governance rule relating to URI longevity and lifetime through concrete policies.
- *Consumption* – to provide more information on the URI and represented resources, the URI Quality pattern needs further elaboration. For instance, the kind of information to be included in the URI metadata (e.g. on provenance) has to be decided.

6. Evaluation

Three types of evaluations were carried out in this work. The first involved checking for the internal consistency, the second is related to coverage of rules with respect to core naming scheme requirements [22]. The third involved the validation of rules in terms of the degree to which end-users and domain experts (both in Linked data and geospatial community) find the eight rules a useful abstraction and specification of the current unwieldy and overlapping sets of rules. Regarding the expert-validation, interviews were conducted with a number of experts requesting comments to serve as basis for improving the URI design strategy and determine detailed requirements for URI design for geospatial data.

Based on the naming scheme requirements described in [22] we argue that the catalog of 8 design patterns satisfy the criteria described in [22] as follows: 1) dereferenceable URI and URI multiple representations realize name resolution that maps a name to an address, 2) guarantee URI uniqueness should be assured by naming authority and the size of the namespace that determines how many unique entities can be named [22], 3) readability criteria is available by applying human readable URI pattern, 4) immutable elements in URI guarantees the name persistence, 5) extensibility is obtained by following URI stability and URI longevity patterns, 6) standardization criteria is covered by URI quality pattern.

Meetings with validation-experts and domain experts were organized. Feedback so far level of evaluation focuses on how much the proposed rules meet the requirements of the domains. Where the returned feedback will be essential for improving what we currently have.

7. Discussions and Conclusion

Many work have been done in URI design domain but these efforts are more domain-specific and don't provide any evaluation method [4]. Hence, the absence of concrete guidance for designing URIs motivated us to develop URI design abstract patterns in this work by structuring and then consolidating existing rules in order to abstract them into design patterns. The value of the developed URI design patterns include: 1) ensuring the consistency of existing rules, 2) guiding the process of designing URIs in any domain, 3) eliminating weak abstracted rules that leads to unwieldy number of rules.

This structured method in developing URI design pattern and the detailed documentation for each rule including the consequences show the possible interactions among rules such as human readability versus non-mutable elements patterns.

Since proposed URI design patterns based on existing rules, the validity of our approach depends directly on the ability of these patterns to represent existing URI design practices and guidelines. URI design patterns will be followed in order to implement LSDI. Geospatial data features are taken into consideration where the abstract level of patterns gives a wide room to include any domain and context such as geospatial domain. We note that most of the works considered are in English thereby leaving out multilingual issues as future work.

8. Acknowledgements

The work presented in this paper has been funded by the European Union under Grant No. 286714 (Linked2Media – Research for SME Associations, SME-2011-2).

9. References

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, pp. 1–22, 2009.

[2] T. Berners-Lee, "Cool URIs don't Change," 1998.

[3] "223 Best Practices URI Construction," 2012. [Online]. Available: http://www.w3.org/2011/gld/wiki/223_Best_Practices_URI_Construction.

[4] P. Davidson, "Designing URI Sets for the UK Public Sector," *UK Chief Technology Officer Council*, no. October, 2009.

[5] D. U. R. I. Strategy and N. L. P. Sector, "Draft URI Strategy for the NL Public Sector," 2013. [Online]. Available: http://www.w3.org/2013/04/odw/odw13_submission_14.pdf.

[6] J. Vlissides, R. Helm, R. Johnson, and E. Gamma, "Design patterns: Elements of reusable object-oriented software," *Reading: Addison-Wesley*, 1995.

[7] R. Béjar and M. Latre, "An RM-ODP enterprise view for spatial data infrastructures," *Computer Standards & …*, vol. 34, no. 2, pp. 263–272, Feb. 2012.

[8] K. Tóth, C. Portele, A. Illert, M. Lutz, and M. N. De Lima, *A conceptual Model for developing Interoperability Specifications in Spatial Data Infrastructures*. 2012.

[9] R. Groot, "Spatial data infrastructure (SDI) for sustainable land management," *ITC journal*, 1997.

[10] C. Becker and C. Bizer, "Dbpedia mobile: A location-enabled linked data browser," *Linked Data on the Web (LDOW2008)*, pp. 6–7, 2008.

[11] S. Abbas and A. Ojo, "Towards a Linked Geospatial Data Infrastructure," *EGOVIS & EDEM '13*, p. 15, 2013.

[12] P. Archer, *Study on Persistent URIs with identification of best practices and recommendations on the topic for the Member States and the European Commission*. 2013.

[13] UK Cabinet Office, "Challenge- Persistent resolvable identifiers." [Online]. Available: http://standards.data.gov.uk/challenge/persistent-resolvable-identifiers.

[14] L. Sauermann, R. Cyganiak, and M. Völkel, "Cool URIs for the semantic web," pp. 1–15, 2011.

[15] V. Svatek, "Design Patterns for Semantic Web Ontologies : Motivation and Discussion," *Proceedings of the 7th Conference on Business …*, 2004.

[16] J. Borchers, "A pattern approach to interaction design," *Proceedings of the 3rd conference on Designing …*, pp. 369–378, 2000.

[17] C. Alexander, S. Ishikawa, and M. Silverstein, *A pattern language: towns, buildings, construction*. 1977.

[18] C. Alexander, *the timeless way of building*. New York: Oxford University Press, 1979.

[19] S. S. T. M. Mills, Don and Bonn, England Amsterdam and San Juan, *Macintosh Human Interface Guidelines*. 1992.

[20] R. Griffiths and L. Pemberton, "Pattern languages for interaction design: Building momentum," *CHI'00 extended abstracts …*, no. April, p. 2000, 2000.

[21] A. Granlund, D. Lafrenière, and D. Carr, "A pattern-supported approach to the user interface design process," *Proceedings of HCI International*, pp. 1–5, 2001.

[22] S. E. N. Aming, I. N. L. Arge, and M. U. O. N. Etworks, "Service Naming in Large-Scale and Multi-domain Networks," *IEEE Communications Surveys*, vol. 7, no. 3, pp. 38–54, 2005.

[23] L. Dodds and I. Davis, "Linked Data Patterns," *… , and consuming Linked Data*, 2011.

[24] D. Booth, "Four Uses of a URL: Name, Concept, Web Location and Document Instance," *Retrieved January*, vol. 28, p. 2003, 2003.

[25] L. Masinter, T. Berners-Lee, and R. T. Fielding, "Uniform resource identifier (URI): Generic syntax," *RFC 2396, August*, 2005.