FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

User Interaction Patterns for Linked Data

Mariana Barbosa Aguiar

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Sérgio Nunes

Second Supervisor: Bruno Giesteira

June 29, 2021

User Interaction Patterns for Linked Data

Mariana Barbosa Aguiar

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Rui Rodrigues
External Examiner: Prof. Cátia Pesquita
Supervisor: Prof. Sérgio Nunes

June 29, 2021

Abstract

Since the beginning of the Semantic Web and the coining of the term Linked Data, more than one thousand datasets with over sixteen thousand links have been published to the Linked Open Data Cloud. Linked Data refers to a set of principles for publishing structured data on the web, using vocabularies that allow the interconnection of data from different sources. With the rise of interest in structured data, we start to see more schemas and platforms developed by the community, resulting in the creation of a global data space - the web of data. However, Linked Data is still frequently perceived as data that will only be created and consumed by machines, not by humans. As a result, Linked Data applications often use the more traditional graph-based visualisations that come with usability issues. However, alternative user interaction approaches have been developed, studied, and evaluated in the literature since the early developments of the Semantic Web. Many of these less common approaches have proven to be better solutions for inexperienced users.

One technique to formalise and document these user interaction techniques and best practices is in the form of patterns. While many pattern collections in the literature cover user interaction problems, there are no works targeted at user interaction problems for Linked Data interfaces. A survey conducted with Linked Data researchers and developers proved this need for user interaction patterns in the community, with over 90% of participants rating them as useful. Here, we propose a pattern collection of 20 novel user interaction patterns for Linked Data as a result of the abstraction of common problems and solutions for visualising, searching, browsing, and authoring Linked Data. This pattern collection supports both experienced and inexperienced developers and designers in creating applications that are easy to use by all users. The proposed patterns are the combined result of the most common problems reported by members of the community in the conducted survey, the knowledge and experience gathered in 10 pattern mining interviews with Linked Data researchers and developers, and the recurrent approaches collected through an extensive literature review of the solutions for user interaction with Linked Data. To validate the structure of the pattern collection, we followed the Pattern Classification method to identify and discuss its strong and weak points. To validate each proposed pattern, we conducted a pattern usage survey to evaluate if the problems covered are commonly encountered and if the solutions are considered good by the community. From this evaluation, we obtained positive scores for 18 out of the 20 patterns. Lastly, we performed an analysis on Wikidata's user interfaces, identifying which patterns are in use and which could be implemented to improve the users' experience. We believe that the main contribution of this work — the proposed pattern collection, can be a valid and helpful tool for developers, regardless of experience, to improve the user interfaces of their Linked Data applications. The future goals include improving the currently written patterns and adding new ones to cover all the sub-problems encompassed in the design of user interaction for Linked Data, aiming to create a complete pattern language on this topic.

Keywords: Linked Data, User Interaction, Pattern, Linked Data User Interaction

Resumo

Desde o início da web semântica e do uso do termo dados ligados, mais de mil coleções de dados com mais de dezasseis mil ligações foram publicadas na Linked Open Data Cloud. O termo dados ligados refere-se a um conjunto de princípios definidos para publicar dados estruturados na web. Com o aumento do interesse em dados estruturados, um número elevado de vocabulários e plataformas começaram a ser desenvolvidos, criando aquilo que se designa por web dos dados. No entanto, coleções de dados ligados continuam a ser vistas como sendo consumidas e criadas apenas por computadores, e não também por humanos. Em consequência, aplicações com dados ligados usam frequentemente as tradicionais visualizações em grafo, que suscitam vários problemas de usabilidade. Contudo, outras soluções de interação têm vindo a ser desenvolvidas, estudadas, e avaliadas desde o início da web semântica. Muitas destas soluções provaram ser melhores opções para utilizadores com pouca experiência na área.

Uma técnica usada para formalizar e documentar soluções e boas práticas de interação é a escrita de padrões. Embora existam coleções de padrões que cobrem problemas de interação, ainda não existem trabalhos focados em interação com dados ligados. Um inquérito realizado junto de programadores e investigadores em dados ligados mostrou a necessidade de padrões nesta área, justificada pelos cerca de 90% de participantes que os consideraram potencialmente úteis. Este estudo propõe uma coleção de 20 padrões de interação com dados ligados, resultado da abstração de problemas recorrentes e de soluções para visualizar, pesquisar, navegar, e criar coleções de dados ligados. Os padrões propostos permitem apoiar programadores e designers com diferentes níveis de experiência, na criação de aplicações dirigidas a todo o tipo de utilizadores. Os padrões propostos neste trabalho foram criados com base nos problemas mais comuns reportados por membros da comunidade no inquérito realizado, no conhecimento e experiência recolhida em 10 entrevistas com programadores e investigadores em dados ligados, e nas abordagens recorrentes colecionadas através de uma revisão da literatura. Para validar a estrutura da coleção de padrões, foi utilizado um método de classificação para identificar os seus pontos fortes e fracos. Para validar cada padrão, foi realizado um inquérito por forma a avaliar se os problemas incluídos são recorrentes e se as soluções propostas são apropriadas. Nesta avaliação, 18 dos 20 padrões propostos obtiveram pontuação positiva. Por fim, foi realizada uma análise das interfaces da plataforma Wikidata com o objetivo de identificar quais os padrões usados e quais os que poderiam ser aplicados para melhorar a experiência dos utilizadores. A principal contribuição deste estudo — a colecção de padrões proposta, é considerada uma ferramenta válida e útil para programadores de dados ligados aperfeiçoarem as interfaces das suas aplicações. Os objetivos futuros para este trabalho incluem a revisão dos padrões escritos, e a adição de novos por forma a cobrir todos os sub-problemas do design de interação com dados ligados, tendo como finalidade a criação de uma linguagem de padrões completa.

Keywords: Dados Ligados, Interação com o Utilizador, Padrões, Interação com Dados Ligados

Acknowledgements

Thank you to Professor Sérgio Nunes for the constant guidance, support, and availability throughout this last chapter. I'm incredibly grateful for your trust in me and in this work and for always trying to push me further. Thank you to Professor Bruno Giesteira for the advice and all the different and challenging perspectives and ideas.

A word of appreciation to all the Linked Data researchers and developers who took the time to participate in the interviews. Without your willingness to share your knowledge and contribute with your experience, this work wouldn't have been possible.

Thank you to the friends that have accompanied me through these years, for sticking with me through good and bad, for all the laughter and great discussions, for the company on long nights, and the beer breaks on sunny afternoons.

Thank you to my family for always believing in me and providing a safe space for me to grow and create my own path. A special word to my sister Leonor, my forever companion - thank you for your support, for all the gardening escapes, cat walks, and coffee runs.

Mariana Aguiar

This work was developed within the scope of the EPISA project - DSAIPA/DS/0023/2018, financed by National Funds through FCT - Foundation for Science and Technology I.P..

"Yet still, there are those special secret moments in our lives,
when we smile unexpectedly - when all our forces are resolved...
whatever it is you are doing at such a moment, hold on to it,
repeat it - for that certain smile is the best knowledge that
we ever have of what our hidden forces are, and where they lie,
and how they can be loosed."

Christopher Alexander

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 2
- 1.3 Goals and Contributions . . . 3
- 1.4 Structure . . . 3

2 Linked Data **5**
- 2.1 Semantic Web Technologies . . . 6
- 2.2 Linked Data Principles . . . 8
- 2.3 Linked Data Open Cloud . . . 9
- 2.4 Linked Data Applications . . . 10

3 User Interaction Patterns **13**
- 3.1 User Interaction Patterns . . . 14
- 3.2 Pattern Design . . . 16
  - 3.2.1 Format . . . 17
  - 3.2.2 Methodology . . . 20
  - 3.2.3 Evaluation and Validation Methods . . . 21

4 Interaction with Linked Data **25**
- 4.1 Methodology . . . 25
- 4.2 Related Work . . . 27
- 4.3 Full-featured tools . . . 28
- 4.4 Browsers and exploratory tools . . . 31
- 4.5 Models and Frameworks . . . 34
- 4.6 Discussion . . . 37

5 Problem Definition **45**
- 5.1 Problem Statement . . . 45
- 5.2 Research Questions . . . 46

6 Methods and Materials **47**
- 6.1 Experience with Linked Data Survey . . . 47
- 6.2 Pattern Mining Interviews . . . 48
- 6.3 Pattern Validation . . . 48
- 6.4 Wikidata Practical Case . . . 48

7 Linked Data Experience Survey **51**
7.1 Design and Execution . . . 51
7.1.1 Questionnaire . . . 52
7.2 Data Analysis . . . 54
7.2.1 Participants' Experience with Linked Data . . . 55
7.2.2 Common Problems . . . 56
7.2.3 User Interaction Patterns Usefulness . . . 60
7.2.4 Final Comments . . . 62

8 Pattern Mining Interviews **65**
8.1 Design and Execution . . . 65
8.1.1 Interview Guide . . . 66
8.2 Thematic Data Analysis . . . 67
8.2.1 Problems and Solutions . . . 68
8.2.2 Other topics . . . 70
8.3 Conclusions . . . 71

9 User Interaction Patterns for Linked Data **73**
9.1 Pattern Structure . . . 74
9.2 Pattern Collection Overview . . . 74
9.3 Visualisation Patterns . . . 76
9.3.1 Template . . . 76
9.3.2 Multi-Visualisation . . . 78
9.3.3 Show Definitions . . . 80
9.3.4 Graph Visualisation . . . 82
9.3.5 Search Results Visualisation . . . 84
9.3.6 Raw Data . . . 86
9.3.7 Complete Information . . . 88
9.3.8 Visual Information . . . 90
9.3.9 Collection Overview . . . 92
9.3.10 Hierarchical Tree . . . 94
9.4 Searching Patterns . . . 96
9.4.1 Keyword Queries . . . 96
9.4.2 SPARQL Queries . . . 98
9.4.3 Advanced Search . . . 100
9.5 Browsing Patterns . . . 102
9.5.1 Link Browsing . . . 102
9.5.2 Browse Path . . . 104
9.5.3 Faceted Browsing . . . 106
9.6 Authoring Patterns . . . 108
9.6.1 Multiple Input Boxes . . . 108
9.6.2 Input Recommendations . . . 110
9.6.3 Input Examples . . . 112
9.6.4 Download Data . . . 114

10 Patterns Evaluation **117**
10.1 Pattern Classification . . . 117
10.1.1 Results and Discussion . . . 118
10.2 Pattern Usage Survey . . . 121
10.2.1 Design and Execution . . . 121
10.2.2 Survey Answers . . . 122
10.2.3 Pattern Usage Results . . . 124
10.2.4 Conclusions . . . 126

11 Wikidata Practical Case **127**
11.1 Goals . . . 127
11.2 Analysis Results . . . 128
11.3 Discussion . . . 133

12 Conclusions and Future Work **137**
12.1 Main Contributions . . . 139
12.2 Future Work . . . 139

A Experience with Linked Data Survey **141**
A.1 Questionnaire . . . 141

B Pattern Mining Guide **147**
B.1 Interview Guide . . . 147

C Patterns Problem Solution Summaries **151**

D Patterns Patlets **153**

E Pattern Usage Survey **155**
E.1 Questionnaire . . . 155

References **169**

List of Figures

2.1 Informal graph of an example set of triples. . . . 7
2.2 Linked Open Data Cloud diagram. . . . 10

3.1 Street Cafe pattern. . . . 18
3.2 Organisation of a partial pattern language for web design. . . . 20
3.3 Schema on how to process the elaboration of patterns and pattern language. . . . 22
3.4 Examples of sketches from the “Pattern Sketches”. . . . 23

4.1 Example of a single individual visualisation and edit mode in OKM. . . . 30
4.2 Example of a single individual visualisation in RDF Surveyor. . . . 31
4.3 Example of the visualisation of a single individual, group of individuals, and search options from PepeSearch. . . . 34
4.4 Example from Conference Explorer, a platform built with Information Workbench. 36
4.5 Examples from WarVictimSampo, a platform built with Sampo-UI. . . . 38

7.1 Results of the participants experience with Linked Data. . . . 55
7.2 Results of the participants experience developing Linked Data applications. . . . 56
7.3 Results of the participants experience using Linked Data applications. . . . 57
7.4 Results of the frequency and importance of problems in Linked Data applications. 58
7.5 Number of problems reported per participant. . . . 58
7.6 Results on the need for guidelines and usefulness of user interaction patterns for Linked Data. . . . 61
7.7 Usefulness of user interaction patterns and developing experience. . . . 61
7.8 Usefulness of user interaction patterns and developing user interfaces experience. 62

9.1 Pattern collection structure overview. . . . 75
9.2 TEMPLATE diagram. . . . 77
9.3 TEMPLATE examples. . . . 77
9.4 MULTI-VISUALISATION diagram. . . . 79
9.5 MULTI-VISUALISATION examples. . . . 79
9.6 SHOW DEFINITIONS diagram. . . . 81
9.7 SHOW DEFINITIONS examples. . . . 81
9.8 GRAPH VISUALISATION diagram. . . . 83
9.9 GRAPH VISUALISATION example - AkatemiaSampo. . . . 83
9.10 SEARCH RESULTS VISUALISATION diagram. . . . 85
9.11 SEARCH RESULTS VISUALISATION examples. . . . 85
9.12 RAW DATA diagram. . . . 87
9.13 RAW DATA example - ImageSnippets. . . . 87
9.14 COMPLETE INFORMATION diagram. . . . 89

9.15 COMPLETE INFORMATION examples. 89
9.16 VISUAL INFORMATION diagram. 91
9.17 VISUAL INFORMATION examples. 91
9.18 COLLECTION OVERVIEW diagram. 93
9.19 COLLECTION OVERVIEW examples. 93
9.20 HIERARCHICAL TREE diagram. 95
9.21 HIERARCHICAL TREE examples. 95
9.22 KEYWORD QUERIES diagram. 97
9.23 KEYWORD QUERIES examples. 97
9.24 SPARQL QUERIES diagram. 99
9.25 SPARQL QUERIES examples. 99
9.26 ADVANCED SEARCH diagram. 101
9.27 ADVANCED SEARCH examples. 101
9.28 LINK BROWSING diagram. 103
9.29 LINK BROWSING examples. 103
9.30 BROWSE PATH diagram. 104
9.31 BROWSE PATH example - inWalk. 105
9.32 FACETED BROWSING diagram. 107
9.33 FACETED BROWSING examples. 107
9.34 MULTIPLE INPUT BOXES diagram. 109
9.35 MULTIPLE INPUT BOXES examples. 109
9.36 INPUT RECOMMENDATIONS diagram. 111
9.37 INPUT RECOMMENDATIONS examples. 111
9.38 INPUT EXAMPLES diagram. 113
9.39 INPUT EXAMPLES example - Wikidata. 113
9.40 DOWNLOAD DATA diagram. 115
9.41 DOWNLOAD DATA examples. 115

10.1 Pattern usage results. 123

11.1 Wikidata resource page. 130
11.2 Wikidata property page. 130
11.3 Wikidata search form and search results page. 130
11.4 Wikidata SPARQL editor page. 131
11.5 Wikidata edit resource page. 131
11.6 Draft of a possible implementation of the TEMPLATE, HIERARCHICAL TREE, and BROWSE PATH patterns in Wikidata's user interface. 134

List of Tables

3.1 Overview of the reviewed user interaction pattern collections. 16
3.2 Overview of the pattern formats in the reviewed user interaction pattern collections. 19
3.3 Overview of the reviewed pattern miming methods. 21
3.4 Overview of the reviewed pattern language evaluation methods. 24

4.1 Overview of the interaction actions provided by the works reviewed. 40
4.2 Overview of the visualisation techniques used in the reviewed works. 41
4.3 Overview of the evaluation methods used. 42
4.4 Overview of reviewed works domain. 43
4.5 Overview of the search techniques in the reviewed works. 44

7.1 Overview of the categorisation of the reported problems. 59
7.2 Familiarity with user interaction patterns and number of problems reported. . . . 60

8.1 Overview of the thematic analysis results of the pattern mining interviews. 68
8.2 Summary of the thematic analysis results of the pattern mining interviews. 70

10.1 Universal Model of a User Interface Layers. 118
10.2 Pattern classification results. 119
10.3 Pattern classification results comparison. 120
10.4 Pattern usage survey answer options. 122
10.5 Pattern classification results. 125

C.1 Problems and solutions of the user interaction patterns for Linked Data. 152

D.1 Pattern collection patlets. 154

Abbreviations

HCI	Human-Computer Interaction
LD	Linked Data
LOD	Linked Open Data
OWL	Web Ontology Language
PLoP	Pattern Language of Programming
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
UI	User Interface
URI	Uniform Resource Identifier
WWW	*World Wide Web*

Chapter 1

Introduction

1.1 Context

The Internet's impact on our lives has outgrown what was initially envisioned when it was created. The World Wide Web (WWW), in particular, allows us to shop for groceries, make trip reservations, talk to family and friends, search for various topics, and, most importantly, share information and knowledge with the world. The WWW is a large collection of documents that differ in type, size, language, and format and are part of a global information space. Users can traverse this web of documents using web browsers and following hypertext links present in the documents [6].

After the burst of the dot-com bubble and the failure of many of the powerful companies of the first years of the web, a turning point was marked and alongside it a new term appeared, web 2.0. The term web 2.0 helped in outlining the major changes of what the web meant, and the differences to the previous web 1.0. Web 2.0 companies started to use the web as a platform to offer services, not packaged software; and their value was now considered proportional to the scale of the data that they helped to manage. So, many companies control unique and hard to recreate datasets, that are harvested from the users as a side effect of the ordinary use of the application [56].

Despite all of this, in 2001 the proposal of the Semantic Web marked a step forward towards Tim Berners-Lee initial idea for the web [7]: a web with more semantics than what was initially conceived. The Semantic Web, also referred to as the web of data, was proposed as an extension of the traditional web, where data is machine-readable and linked, and computers are capable of analysing the data, the content, links, and interactions from humans and computers.

Documents published on the traditional web are usually available in the HTML format. While they have some structure, this language does not provide means for structuring data and annotating individual entities that can be connected to other related entities [39]. The Semantic Web proposal also aims to tackle the problem of interlinking data on the web by providing a set of principles for publishing structured data denominated Linked Data (LD), a term coined by Tim Berners-Lee in 2006 [5]. Linked Data provides a way of using the web to create typed links between data from different sources [6] with the use of the Resource Description Framework (RDF) language. RDF

is used to describe things in the world and how they relate to each other, allowing data to have explicit meaning, be machine-readable, and be linked to and from external datasets [58].

While the classic web of documents and the newer web of data form, together, a navigable information space, applications and systems based on Linked Data pose new challenges and problems. Developers and designers are faced with new obstacles in many phases of software development, such as user interface and interaction design tasks. Questions like: how can LD applications make the most of the linked aspect of the data and still present it to the user in a clean, understandable way; and how can LD applications lead users to correctly create and edit data in a familiar, intuitive manner, remain unanswered.

1.2 Motivation

A Linked Data application is an application that operates over semantically annotated, interconnected, structured data. It exploits the advantages of using linked machine-readable data to respond to advanced use-case scenarios, and at the same time outputs it for human consumption. With the rising interest in structured linked data on the web and the growth of the web of data, we start to see more of these applications and platforms being developed by the community. This interest is fuelled by the benefits that information understandable by machines and computers can have in several types of systems, like search engines or recommendation systems. Linked Data was initially portrayed as data to be created and consumed solely by machines and computers, and not also by humans. Nonetheless, alongside this new rising interest in Linked Data, we observe a rise in humans creating and consuming Linked Data.

There are numerous reusable solutions for common design problems in user interfaces, formalised in published pattern collections. However, Linked Data applications arise with new and more complex user interaction problems. Designers and developers may wonder how to output Linked Data in a way that takes advantage of its graph format and inter-connectivity, but still make it intuitive, familiar, and easy to navigate by a common user. There are several studies on visualisation tools and methods for browsing RDF data or Linked Data in the literature. However, due to the initial idea that humans would not interact with Linked Data, there is a lack of formalised and documented guidelines on how to model user interaction with Linked Data. With all this in mind, this work focuses on surveying frequent Linked Data user interaction problems and abstracting and formalising their solutions. We aim to study interaction solutions targeted at common web users, for searching, exploring, visualising, and authoring Linked Data. The result is a pattern collection that gathers collective experience and good practices in this field.

A pattern can be simply defined as a reusable solution to a common problem in a specific context. This modern notion of a pattern was first introduced in 1977 by Christopher Alexander, a building architect, to share his experience and knowledge on designing towns and buildings with others [2]. A pattern language aims to tackle a high-level problem by providing a set of interconnected patterns with proven solutions to the smaller problems.

The proposed pattern collection serves as a way of documenting and formalising best practices and experiences in modelling user interactions with Linked Data. It aims to supply novices and experts alike, with proven solutions and empirically tested knowledge that will aid them to make informed decisions while developing and designing Linked Data based systems.

1.3 Goals and Contributions

With this work we aim to write a pattern collection that identifies best practices in user interaction design and that facilitates the decision-making process when building Linked Data based environments. The patterns and the pattern collection are evaluated according to their completeness and usage. The contributions of this research are:

A review of the state of the art of user interaction with Linked Data. Reviews the studies in user interaction with Linked Data and Semantic Web applications. Identifies and evaluates the interaction features and solutions in existing Linked Data visualization tools.

A pattern collection for user interaction with Linked Data. Describes common design problems in Linked Data environments, and their reusable solutions, helping developers and designers in developing applications.

Pattern collection evaluation. Evaluates the patterns and pattern language assessing their completeness and usage, following the pattern classification method, and by performing a pattern adoption survey on the Linked Data community.

Practical case study. Identifies the patterns already employed in the Wikidata platform and describes the implementation of some of the missing ones.

1.4 Structure

This document is divided into 12 chapters. Chapter 2 (p. 5) provides an overview of the fundamentals of Linked Data by introducing the Semantic Web concept and addressing LD main technologies, LD principles and the Linked Data Open Cloud. This chapter also presents a series of examples of LD applications relevant to the problem at hand. Chapter 3 (p. 13) gives a brief history of patterns and pattern languages, followed by methodologies, evaluation and validation methods for the development of pattern languages. This chapter also introduces the concept of user interaction design, and studies the current available collections of user interaction patterns. Chapter 4 (p. 25) presents a study of the state of the art on interaction with Linked Data, focusing on research conducted to explore and evaluate the main interaction scenarios with Linked Data: visualise, search, navigate and author. Chapter 5 (p. 45) formalises the problem at hand, by presenting the hypothesis and, research questions. Chapter 6 (p. 47) presents the methodology followed to collect information, create, and evaluate the patterns; and describes the methods used

in each phase of this research. Chapter 7 (p. 51) details and presents the results of the online questionnaire conducted to survey the participants experience with Linked Data, common problems when using and developing Linked Data applications, and the usefulness of user interaction patterns for this topic. Chapter 8 (p. 65) describes the guide followed for the pattern mining sessions with Linked Data researchers and developers, and presents the results from a thematic analysis of the gathered data. Chapter 9 (p. 73) proposes a collection of 20 user interaction patterns for Linked Data, divided in four categories, visualising, searching, browsing, and authoring. Chapter 10 (p. 117) details the methods conducted to evaluate the patterns, and the obtained results. Chapter 11 (p. 127) describes the analysis performed on the Wikidata user interface, according to the patterns implemented. Chapter 12 (p. 137) exposes the conclusions of this research on user interaction with Linked Data and summarises the future work.

Chapter 2

Linked Data

The term Linked Data (LD) refers to a set of principles for publishing and interlinking structured data on the Web, proposed in 2006 by Tim Berners-Lee, the creator of the World Wide Web (WWW). LD allows for annotating data in a machine-readable format so that it is understandable and browsable by both humans and computers alike. This is accomplished by publishing and sharing content alongside metadata, which provides a context to the main content in the form of machine-readable descriptions. Then, machines can extract knowledge, and not only text like in the traditional web of documents, by using methods similar to human deduction and inference. This leads to better and more meaningful results and opens doors to systems with automated information gathering and searching, and intelligent search engines with a meaning-based approach, as opposed to the traditional keyword-based approach. Another benefit of using LD is that it provides ways to link data from different sources on the web, allowing humans or machines to traverse and explore it. By following the links between data, users can very quickly and effectively discover other related data, and applications can exploit the knowledge from other datasets by integrating these connections, providing a more complete result and a better experience to the user.

The WWW [6] is a collection of documents connected by hypertext links to other documents with related information. We rely on it to perform various tasks, such as buying groceries, making trip reservations, connecting with friends and family, attending classes, performing our jobs, and constantly sharing information with others and the world. During the 1990 decade there was an idea that the web was for publishing and not participating, and that the advertisers and top websites controlled and dominated it, instead of the consumers. However, the burst of the dot-com bubble marked a turning point for the web, with the disappearance of many of the big companies that controlled it, and a change in the way that the web is perceived. The term web 2.0 originated in a conference brainstorming session between Tim O'Reilly and MediaLiveInternational, and even with some people initially rejecting it, the term helped to outline the major changes in what the web meant, and to clarify the differences to the previous web 1.0. According to O'Reilly [56] the companies that thrived in the web 2.0 era followed a set of principles, not common to the web 1.0

companies: these companies use the web as a platform to provide services to the users; the value of the software is proportional to the scale of the data that it helps to manage, so many companies own unique datasets that are harvested from the users as a side effect of the ordinary use of the application; and try to reach out to the edges of the web, not just the center.

Despite all of this, Tim Berners-Lee initial idea for the web included more semantics than what was initially conceived. In 2001, with the proposal of the Semantic Web a step towards Berners-Lee's initial idea was taken [7]. The Semantic Web, also referred to as the web of data, is an extension of the traditional web where data is machine-readable and linked, and computers are capable of analyzing the data, the content, links, and interactions between humans and computers.

To make the Semantic Web a reality, it is important that huge amounts of data and their relationships are available and published on the web in a standard format, as opposed to a simple collection of different datasets. Linked Data plays a massive role in connecting and keeping alive the Semantic Web. It provides ways to publish and connect data from different sources at a large scale, using a common format that is still open enough to accommodate every possible data domain. Since the Linked Data principles were published, numerous data providers and application developers have started publishing LD, transforming the web of data into a space as diverse as the traditional web, where several topical domains can be found like scientific and governmental data [71, 48].

2.1 Semantic Web Technologies

For the Semantic Web to be able to exist, common data formats and technologies to define and query structured data should be made available. In order to grow, the Semantic Web had to use and be developed upon different document formats than the ones used in the traditional web. Nowadays, most pages on the web have only some degree of structure, such as a title, sections and comments. Since HTML, the language in which they are created, is intended for structuring textual documents and not data. Thus, it is difficult for applications to extract structured data from HTML pages, as the data is intermingled into the surrounding text [39].

The Semantic Web is built upon different technologies that have been piled up, some common to the traditional web and some not. These technologies form what is called as the Semantic Web Stack [58], and some have contributed to the birth and evolution of Linked Data, namely RDF and RDF Schema, SPARQL and OWL.

RDF and RDF Schema. Resource Description Framework (RDF) [29] is a framework for describing information about resources, such as documents, people, objects or even abstract concepts. RDF was developed to allow information on the Web to be processed by applications. It provides its users with a framework for describing information that can be exchanged between applications or machines without losing its meaning [66]. At the core of RDF are the RDF statements, a way to express relationships between two resources. A statement is composed of three components, subjects and objects, the related resources, and predicates, that specify the nature of the relationship. Due to being consisted of three components, RDF statements are called triples.

In RDF the same resource can be referenced in infinite statements, both in the subject and object position. This feature is what gives RDF great importance, because it allows machines to find connections between triples. A set of connected triples can be visualised as a graph, like the example in Figure 2.1. The RDF Schema [11] is an extension of the basic RDF vocabulary, and it provides mechanisms for describing groups of related resources and their relationships.

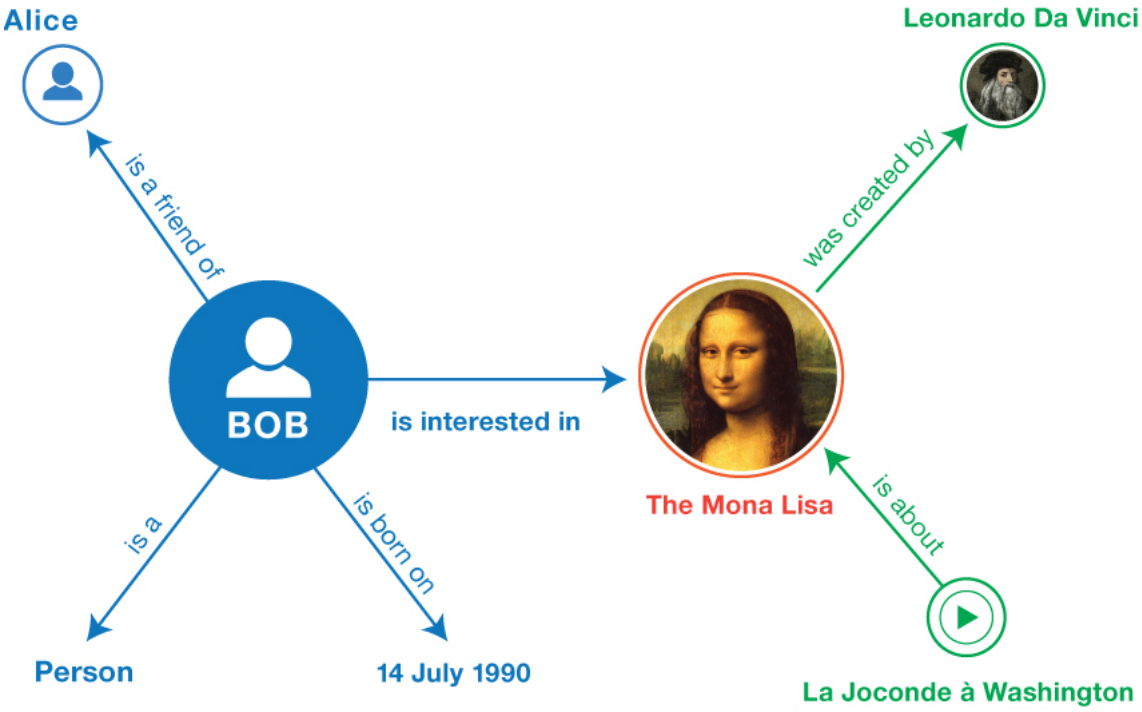


Figure 2.1: Informal graph of an example set of triples (Image from Schreiber and Raimond [66]).

SPARQL. The SPARQL Protocol and RDF Query Language (SPARQL) [59] is a set of specifications that allow protocols and languages to retrieve and manipulate RDF graph content. SPARQL queries are based on graph pattern matching; graph patterns are templates of sets of triples that the SPARQL engine looks for [30].

OWL. The Web Ontology Language (OWL) is a language designed to represent and describe complex knowledge about things and their relationships. An important feature of OWL is that knowledge expressed using this language can be reasoned using computers, either to verify its consistency or to make explicit implicit knowledge [41]. OWL is used to express ontologies, sets of precise descriptive statements about a domain. The use of precise descriptions has several advantages, such as the prevention of misunderstandings in human communications and the assurance that software works in a correct, predictable way. Compared to other vocabularies, OWL provides ways to express richer semantics in data.

Each of the previously described technologies plays a different part in maintaining the use of Linked Data datasets and the development of Semantic Web applications. RDF provides mechanisms for adding machine readable information to web pages, enriching datasets by linking them

to other third party datasets, and publishing structured linked data on the web. SPARQL is the language that allows applications and systems to retrieve and manipulate the structured RDF data from knowledge graphs available on the web. Finally, OWL provides a vocabulary to express knowledge about a certain domain, and promotes the reuse of these ontologies by different people resulting in the interlinking of data from different sources.

As mentioned previously, these technologies are the backbone of the Semantic Web as they provide the mechanisms to fulfil its main goals. Linked Data also makes use of some of these technologies to promote the publishing of datasets with RDF structured data, with machine-readable semantics, and interlinked to other published datasets.

2.2 Linked Data Principles

At the same time of the coining of the term Linked Data, Tim Berners-Lee published a note with a set of 4 principles for publishing LD on the web [5].

1. Use URIs as names for things.
2. Use HTTP URIs, so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs so that they can discover more things.

The first LD principle promotes the use of URI references to identify web documents, other digital content, and real-world things or abstract concepts such as a person, a car, a place, the type of relationship between two people, or a specific color. The generalised use of this principle results in the extension of the content on the web from only online resources to any object or concept in the world. The HTTP protocol is the web's universal communication mechanism, used in the traditional web to combine globally unique identification with a simple and commonly understood retrieval method [39]. The second LD principle advocates for the use of HTTP URIs to identify objects and concepts, like the ones mentioned previously. These URIs can be looked up over the HTTP protocol, providing the user with a description of the object or concept. As mentioned previously in Section 2.1, the Semantic Web relies on the use of common standardized formats for its content, just as the traditional web growth was due to the agreement of HTML as its dominant document format. The third LD principle reflects this by promoting a single data model for publishing data on the web and providing information on the URI annotated concepts, the Resource Description Framework. The fourth and last LD principle aims to promote links to connect different types of things or documents on the web; just like in the traditional web, the value of information or data is reliant on what it is linked to. However, contrary to the classic web, links between data are typed to provide a description of the relationships between the things. They are specified using RDF links [39].

As previously discussed, the Semantic Web relies on standards and common technologies to exist, as standard formats are needed to connect data on the web. The Linked Data principles support the existence and connection of the web of data and consequently allow for the implementation of generic applications that make use of the resulting global data space. Not following these principles is not considered wrong but a missed opportunity to interconnect data and contribute to a web where information can be reused over and over again [5].

2.3 Linked Data Open Cloud

Around the same time of the birth of the LD principles, Open Data, a term dating back to 1995 [58], started to grow. Open data started out to be promoted as a complete and open exchange of scientific information between different countries. However, a formal definition appeared only in 2005 with the Open Definition 2.1[1]. This document outlines the requirements for data to be considered open and can be summarised as knowledge that is free to be accessed, used, modified and shared.

With the benefits of merging Linked Data and Open Data in mind, Tim Berners-Lee published in 2010, an extension of his original note, adding a star rating system for publishing Linked Open Data (LOD) [5]. LOD is Linked Data that is published with an open license, and so considered Open Data as well. The star rating system for LOD, named 5 Star Model, provides guidelines and promotes publishing good open Linked Data to the web. LOD plays a important part in connecting data on the web. The 5 Star Model raking system tiers are described next [5]:

- **1 star:** Available on the web (whatever format) but with an open licence, to be Open Data.
- **2 stars:** Available as machine-readable structured data.
- **3 stars:** Available in a non-proprietary format.
- **4 stars:** Use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff.
- **5 stars:** Link your data to other people's data to provide context.

The Linked Open Data Cloud[2] is the direct result of publishing LOD on the web, it is composed of the Linked Data datasets that are available publicly. As of May 2020, the LOD cloud contains 1,269 datasets with 16,201 links between them, a high-level representation can be consulted in Figure 2.2. In this figure, the LOD Cloud is represented as a partially connected graph, each dataset is portrayed as a node, and a link between nodes means that at least one property connects elements from two datasets. To improve the user's experience while navigating the LOD cloud, the size of a node is dictated by its total number of triples, and its color represents the domain it covers. Anyone can submit datasets to the cloud, however they will only be accepted if they follow the LOD cloud principles, a slightly different version of LD principles.

[1] https://opendefinition.org/od/2.1/en/
[2] https://lod-cloud.net/

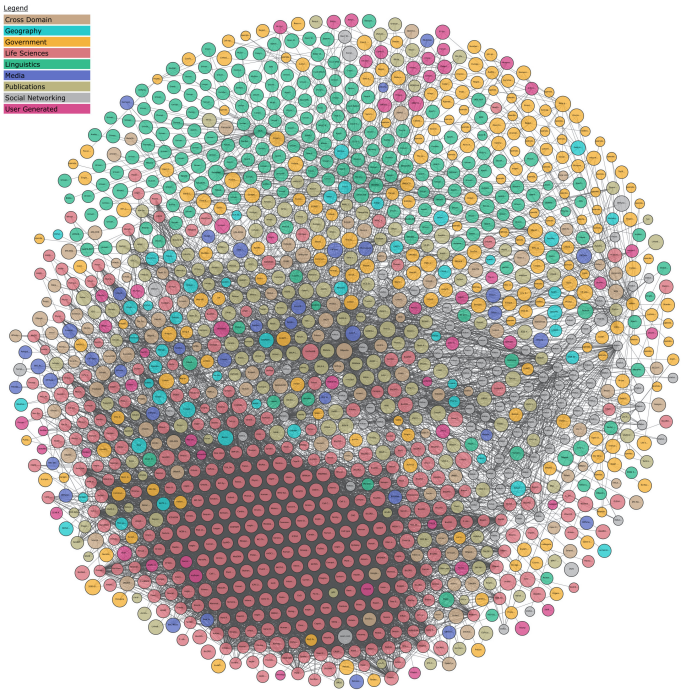


Figure 2.2: Linked Open Data Cloud diagram.

The LOD cloud diagram is considered the best representation of the web of data [58]. It provides a way for LD authors to quickly navigate the available datasets and find some that could be related to their own, promoting the interlinking of datasets from different sources. The LOD cloud further simplifies this search by providing domain subsets of the whole graph, and details such as, a description, contact information, example resources, and SPARQL endpoints, for every dataset.

2.4 Linked Data Applications

Nowadays, Linked Data and Linked Open Data datasets are used in all kinds of applications. Here, we present four different LD datasets that are currently used in web applications: DBpedia, the biggest LOD dataset on the web; the BBC LOD datasets used in their applications; Van Gogh Worldwide, an application that uses a private LD dataset to provide information about Van Gogh's work; and Wikidata, a collection of all the Wikipedia's information in RDF format.

DBpedia. DBpedia[3] is a crowd-sourced project that generates structured information extracted from automatic Wikipedia dumps. The DBpedia Linked Data dataset provides information about numerous resources from different knowledge fields, and is available as Open Data on the web. DBpedia also provides users with ways to semantically query the resources and relationships in the dataset, as well as links to other related datasets. The DBpedia dataset is still be biggest LD

[3] https://wiki.dbpedia.org

dataset published openly on the web, since the begining of the LOD cloud project. It plays a massive role in interlinking the web of data as it acts like a hub for connecting other datasets.

BBC. The British Broadcasting Corporation (BBC) is a public radio and television corporation, that uses open Linked Data datasets[4] in some of its applications, such as BBC Sport, BBC Education and BBC Music. Additionally, all these datasets are modelled using OWL ontologies created by teams at BBC, to cover the concepts featured in their applications. All the ontologies are publicly available and generic enough to allow other users to reuse them to model their data and link other concepts. The ontologies cover topics from sports, food, and wildlife to politics and journalism.

Van Gogh Worldwide. Van Gogh Worldwide[5] is a web application that provides information about the work of the artist Van Gogh. This application aggregates information from different museums that host Van Gogh collections, in the form of Linked Data. The use of structured data allows the application to describe all artworks in great detail, to quickly add all kinds of objects related to Van Gogh's work, and to make uniform the data from different institutions. The application also provides its users with a map visualisation of the works.

Wikidata. Wikidata [80] is a free knowledge base that integrates into a single location the Wikipedia's data. Wikidata allows every user to extend and edit the structured, semantically annotated, and linked RDF data provided in the platform, enabling it to be searched, analysed, and reused. Wikidata was first launched in 2012, with limited features and contributors only being able to create items and connect them to Wikipedia pages. However, until March of 2013, Wikidata counted with more than three million items and all Wikipedias connected to it. In 2014, Wikidata was edited 150 to 500 times per minute, or a half million times per day. At that time, 90% [80] of the edits were done by bots created by contributers for task automation, even so, Wikidata counted with one million edits per month made by humans. By the end of February 2014, Wikidata was composed of 14.5 million items and all Wikipedia articles were connected to it. All Wikidata data follows the Linked Data publishing principles, and so contributes to the integration of other data sources, and the connection of the Semantic Web.

[4] https://www.bbc.co.uk/ontologies
[5] https://vangoghworldwide.org

Chapter 3

User Interaction Patterns

The first use and formalisation of the term pattern, as a way to document and describe common problems and their proven solutions, was by Christopher Alexander in his work "A Pattern Language" [2]. Alexander introduces this notion of patterns as the description of recurring solutions to common problems in a way that allows the solutions to be reused in different cases. Alexander presents a hierarchical collection of architectural design patterns to help other architects make future buildings and urban environments more usable and pleasing. To this interconnected set of patterns Alexander calls a pattern language. A pattern language is essentially a set of linked patterns in the form of a hierarchy or network, that together aim to tackle a high-level problem by providing proven solutions to each one of its smaller problems.

Alexander introduced this idea of pattern language as a mechanism, not only for experts to formalise and share their knowledge, but for laypeople to use as a vocabulary to express their ideas and designs. However, not all sets of patterns available can be considered pattern languages. Pattern catalogues are smaller sets of patterns that only cover a fraction of the common solutions for a certain domain. Pattern collections are sets of related or unrelated patterns that can be grouped together from other languages, collections or catalogues. Finally, pattern repositories are systems that provide access to a set of patterns from different pattern languages or catalogues.

Patterns and pattern languages serve a multitude of purposes, they, as mentioned previously, are a mechanism for experts in a certain domain to formalise and document their knowledge and experience. They also provide first time creators in an area with knowledge of best practices and guidelines for their work. Previous creations in the domain can also be evaluated against the pattern language and consequently improved by applying some pattern solutions. From a communication perspective, pattern languages can provide a vocabulary to a research group of a specific field, allowing it to more clearly refer to certain aspects of a creation using the patterns names applied. All in all, pattern languages serve many proposes and can aid people in their daily lives.

Two decades after the publication of "A Pattern Language: Towns, Building, Construction",

the software engineering field picked up on the idea of creating patterns and pattern languages, and decided to try and transfer this way of documenting knowledge to the Computer Science area. The first publication of a set of patterns for Computer Science was by Gamma et al. [23] in "Design Patterns: Elements of Reusable Object-Oriented Software". Eventually the software pattern community growth lead to the creation of the annual Pattern Language of Programming (PLoP) conferences. In the PLoP events, anyone can submit a pattern language on a topic of the software engineering field, and have it reviewed and discussed by several experienced pattern language authors. The goal is to provide an environment where proven and generalised solutions to common software design problems can be exchanged, discussed, and improved.

Due to the popularity that the software engineering pattern community acquired, the idea of creating pattern languages in other fields of computer science started to appear. Workshops popped up in Human-Computer Interaction (HCI) conferences to discuss the importance of pattern languages in the field and to shape what a design pattern in HCI would mean [10]. Meanwhile, pattern collections in interaction design started to be published and some were even featured in PLoP conferences [76, 69].

3.1 User Interaction Patterns

One of the areas of research in the field of HCI is the study of the user interaction with computers and machines. Interaction design is focused on modelling and constructing the way people interact with all kinds of systems and products. The results of user interaction design have a direct impact in the quality of the user experience while using the system [37]. Nowadays, the value of a piece of software is not entirely dependent on its quality, but also on how well it is interpreted and interacted with by users. So, studying best practices to model the user interaction with different types of systems is becoming more and more important to ensure that users can extract valuable information for the presented content, and quickly and correctly perform important tasks. The study of user interaction with more complex and uncommon data types, like Linked Data, is even more crucial, as they are an unfamiliar way of structuring data for most users and often require special and novel mental models.

One way of formalising and documenting the knowledge acquired studying human computer interaction is to create patterns and pattern languages. Collecting experiences and best practices, and composing a pattern language in user interaction is a great way to share knowledge in this area. Patterns are capable of presenting problems and explaining possible solutions in the form of an abstraction, providing the pattern's user with freedom to incorporate the solution in its design in a unique way. In "A Pattern Approach to Interaction Design", Borchers [10] provides a formal description of what a pattern in the HCI community should look like and justifies why the use of pattern languages aids not just a single design phase, but the whole design process. Borchers also proposes to apply the use of patterns and pattern languages to different and new application domains, explaining that they provide a vocabulary and ease communication in interdisciplinary

teams. Additionally, the use of pattern languages inherently offers justifications for certain design options.

Over the years, a substantial number of works with suggestions of user interaction patterns were published. Here, we highlight 7 works considered more relevant from the literature on user interaction design patterns, summarisation of these works can be consulted in Table 3.1. "Designing Interfaces: Patterns for Effective Interaction Design" [73] by Jenifer Tidwell is one of the first pattern collections published in the HCI community, with its first edition dating back to 1999. This pattern collection is organised in several sections, covering different parts or types of interaction with a system, from page organisation and layout, navigation and search, user input, actions and commands, to more specific topics like interaction with complex data, social media and mobile applications. In its first edition, this collection was targeted mostly at desktop applications, like email clients and browsers, however with the growth of the web, more recent editions include new patterns specific for web-based applications and mobile design. Tidwell's work features many patterns that are commonly encountered in numerous applications, and thus may now seem trivial. Nevertheless they once were not as widely used as right now, which comes to prove their importance and impact.

In "Interaction patterns in user interfaces", van Welie and Traetteberg [76] propose a pattern collection for interaction with user interfaces. van Welie and Traetterberg take on a different approach, compared to Tidwell's to identify patterns and deem them as "good" and "accepted" solutions. The patterns proposed are focused on the user perspective, meaning that they are targeting problems faced by end-users, as opposed to designers. In order to do this, the pattern collection is organised in categories representing the kinds of problems users have while interacting with a system: visibility, affordance, natural mapping, constraints, conceptual models, feedback, safety and flexibility [55]. Then, each pattern is focused on improving the usability of a system, and so, each one states the usability indicator it tries to improve. van Welie and Traetteberg [76] defend that a user interaction design pattern must improve at least one usability usage indicator, otherwise it is not considered one.

Another pattern collection was published by Yahoo in 2012, "Yahoo Design Pattern Library" [83], it isn't only focused on user interaction but also on the design of user interfaces. It features several patterns that we can now identify in many recent websites and applications. Another work focused on web design is "Designing Social Interfaces: Principles, Patterns, and Practices for Improving the User Experience" [15] which is a pattern collection focused on gathering best practices and principles from applications and websites with social features or focus.

Other works like Laakso [50]'s and Toxboe [74]'s also propose pattern collections for interaction and interface design problems. These two works are not focused on a specific medium, as patterns can be applied to all types of applications, desktop, web-based or mobile. However, the pattern collection presented in "Designing Web Interfaces" [67] is targeted at tackling common problems in the design of web interfaces.

While some of the previously presented works differ on the types of systems that they are focused on, none are targeted at a certain data type. Most of the patterns featured in the pattern

	Problem	Medium	Number of Patterns	Related Patterns
Tidwell [73]	Interface design	Digital interfaces	125	Yes
van Welie and Traetteberg [76]	Interaction design	Digital interfaces	22	Yes
Yahoo [83]	Web design	Web interfaces	59	No
Crumlish and Malone [15]	Social interface design	Social web interfaces	92	Yes
Laakso [50]	Interface design	Digial interfaces	21	Yes
Scott and Neil [67]	Web design	Web interfaces	65	No
Toxboe [74]	Web design	Web interfaces	90	Yes

Table 3.1: Overview of the reviewed user interaction pattern collections.

collections presented could be applied to systems with different data types. However, more complex data types emerge with more difficult interactions that could be confusing and not intuitive to users when not modelled correctly. Linked Data is still an unfamiliar data modelling paradigm for most web users. A pattern language on how to model interaction systems that work with Linked Data could result in Semantic Web applications that are more familiar and friendlier to the common user.

There isn't yet a pattern language or pattern collection formalised for interacting with Linked Data. However, we can point out one work proposing patterns for other LD related problems. In "Linked Data Patterns: A pattern catalogue for modelling, publishing, and consuming Linked Data" [21], a pattern catalogue is proposed for tackling problems when modelling, publishing and consuming LD. The pattern catalogue presents a section for application patterns, these are targeted at architectural and code design problems while developing the system, rather than visualisation and user interface design problems.

3.2 Pattern Design

When writing a pattern language one has to be mindful of its structure and organisation, and the structure of the patterns it contains. What sets patterns apart from other ways of documenting solutions is the ability to explain the rationale of using the described solution in a specific context. In order to achieve this, patterns must be constituted of elements that ensure an easy comprehension of all these aspects to the reader. There are several studies and pattern languages with recommendations and best practices for writing pattern languages, touching on topics like pattern format and the requirements for each element, techniques for linking a pattern to other patterns,

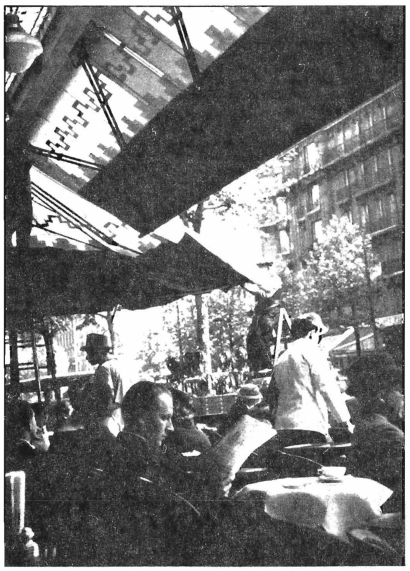
the organisation of pattern languages, and recommendations on how to make a pattern language easily transversed and functional to the user.

3.2.1 Format

The first patterns written by Alexander [2] for the building architecture domain, are composed of the following elements, as can be seen in the example of Figure 3.1: the pattern title; a picture with an archetypal example of the pattern; an introductory paragraph outlining the context; a headline with the essence of the problem followed by the main body where the empirical background of the pattern and the evidence for its validity are described; the solution; a diagram with a visual representation of the solution; and finally the section with the related patterns. Over the years this classic pattern format evolved and suffered some changes, as different domains required different components. In "A Pattern Language for Pattern Writing", Meszaros and Doble [54] outline the following as mandatory elements of a pattern: pattern name, context, problem, forces, and solution. In "How to write a pattern?: a rough guide for first-time pattern authors", Wellhausen and Fiesser [82] formalise some guidelines for writing these mandatory pattern elements, the context of a pattern should contain important aspects and requirements that without them the problem may not exist; the problem statement describes the goal to be accomplished by the solution and can explain why a certain situation is worse without the solution applied; the forces are what shows that finding a solution for the given problem is not trivial, and must restrict the other available solutions to just the one described in the pattern; at last the solution is a description of how to tackle the problem and should be clear enough that someone familiar with the domain is able to understand it quickly.

Looking at the user interaction pattern collections introduced in Section 3.1 we can quickly conclude by looking at Table 3.2, that they share a somewhat similar choice of pattern format. In "Designing Interfaces: Patterns for Effective Interaction Design" [73] the pattern elements are the following: pattern name, a section named "What" corresponding to the problem statement, followed by "Use When" describing the context, "Why" outlining the pattern forces, "How" with the solution and a final "Examples" section. In "Interaction patterns in user interfaces" [76] the patterns are structured as follows: pattern name, problem statement and solution, followed by a "Use When" section with the context of the problem, a "How" section describing how to apply the solution, section "Why" stating the forces and a "More Examples" section with interfaces where the pattern was used. We can observe that both works follow a similar pattern format to what is recommended by Meszaros and Doble [54] with the addition of a section with practical examples of the pattern. In a "Pattern Approach to Interaction Design", Borchers [10] formalises a domain-independent definition of design patterns that can be integrated into the usability engineering life cycle, for the HCI field. One of the aspects described is the pattern format to be used. According to the author, user interaction patterns must be composed of the following: pattern name; context; a ranking, showing how valid the pattern author believes it to be; problem; forces; solution; diagram, to graphically summarise the solution; and examples, showing how the pattern can be used and providing a visual justification to the recommended solution.

88 STREET CAFE**



436

TOWNS

States. And the existence of such places provides social glue for the community. They become like clubs—people tend to return to their favorite, the faces become familiar. When there is a successful cafe within walking distance of your home, in the neighborhood, so much the better. It helps enormously to increase the identity of a neighborhood. It is one of the few settings where a newcomer to the neighborhood can start learning the ropes and meeting the people who have been there many years.

The ingredients of a successful street cafe seem to be:

1. There is an established local clientele. That is, by name, location, and staff, the cafe is very much anchored in the neighborhood in which it is situated.

2. In addition to the terrace which is open to the street, the cafe contains several other spaces: with games, fire, soft chairs, newspapers. . . . This allows a variety of people to start using it, according to slightly different social styles.

3. The cafe serves simple food and drinks—some alcoholic drinks, but it is not a bar. It is a place where you are as likely to go in the morning, to start the day, as in the evening, for a nightcap.

When these conditions are present, and the cafe takes hold, it offers something unique to the lives of the people who use it: it offers a setting for discussions of great spirit—talks, two-bit lectures, half-public, half-private, learning, exchange of thought.

When we worked for the University of Oregon, we compared the importance of such discussion in cafes and cafe-like places, with the instruction students receive in the classroom. We interviewed 30 students to measure the extent that shops and cafes contributed to their intellectual and emotional growth at the University. We found that "talking with a small group of students in a coffee shop" and "discussion over a glass of beer" scored as high and higher than "examinations" and "laboratory study." Apparently the informal activities of shops and cafes contribute as much to the growth of students, as the more formal educational activities.

We believe this phenomenon is general. The quality that we tried to capture in these interviews, and which is present in a neighborhood cafe, is essential to all neighborhoods—not only student neighborhoods. It is part of their life-blood.

438

. . . neighborhoods are defined by IDENTIFIABLE NEIGHBORHOOD (14); their natural points of focus are given by ACTIVITY NODES (30) and SMALL PUBLIC SQUARES (61). This pattern, and the ones which follow it, give the neighborhood and its points of focus, their identity.

✣ ✣ ✣

The street cafe provides a unique setting, special to cities: a place where people can sit lazily, legitimately, be on view, and watch the world go by.

The most humane cities are always full of street cafes. Let us try to understand the experience which makes these places so attractive.

We know that people enjoy mixing in public, in parks, squares, along promenades and avenues, in street cafes. The preconditions seem to be: the setting gives you the right to be there, by custom; there are a few things to do that are part of the scene, almost ritual: reading the newspaper, strolling, nursing a beer, playing catch; and people feel safe enough to relax, nod at each other, perhaps even meet. A good cafe terrace meets these conditions. But it has in addition, special qualities of its own: a person may sit there for hours—in public! Strolling, a person must keep up a pace; loitering is only for a few minutes. You can sit still in a park, but there is not the volume of people passing, it is more a private, peaceful experience. And sitting at home on one's porch is again different: it is far more protected; and there is not the mix of people passing by. But on the cafe terrace, you can sit still, relax, and be very public. As an experience it has special possibilities; "perhaps the next person . . ."; it is a risky place.

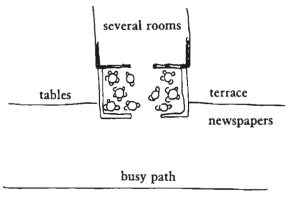
It is this experience that the street cafe supports. And it is one of the attractions of cities, for only in cities do we have the concentration of people required to bring it off. But this experience need not be confined to the special, extraordinary parts of town. In European cities and towns, there is a street cafe in every neighborhood—they are as ordinary as gas stations are in the United

437

88 STREET CAFE

Therefore:

Encourage local cafes to spring up in each neighborhood. Make them intimate places, with several rooms, open to a busy path, where people can sit with coffee or a drink and watch the world go by. Build the front of the cafe so that a set of tables stretch out of the cafe, right into the street.



✣ ✣ ✣

Build a wide, substantial opening between the terrace and the indoors—OPENING TO THE STREET (165); make the terrace double as A PLACE TO WAIT (150) for nearby bus stops and offices; both indoors and on the terrace use a great variety of different kinds of chairs and tables—DIFFERENT CHAIRS (251); and give the terrace some low definition at the street edge if it is in danger of being interrupted by street action—STAIR SEATS (125), SITTING WALL (243), perhaps A CANVAS ROOF (244). For the shape of the building, the terrace, and the surroundings, begin with BUILDING COMPLEX (95). . . .

439

Figure 3.1: Street Cafe pattern by Alexander [2].

	Title	Context	Problem	Solution	Forces	Example(s)	Diagram
Alexander [2]	✓	✓	✓	✓	✓		✓
Tidwell [73]	✓	✓	✓	✓	✓	✓	
van Welie and Traetteberg [76]	✓	✓	✓	✓	✓	✓	
Yahoo [83]	✓	✓	✓	✓	✓	✓	
Crumlish and Malone [15]	✓	✓	✓	✓	✓	✓	
Laakso [50]	✓	✓	✓	✓		✓	
Scott and Neil [67]	✓		✓	✓		✓	
Toxboe [74]	✓	✓	✓	✓	✓	✓	
Borchers [10]	✓	✓	✓	✓	✓	✓	✓

Table 3.2: Overview of the pattern formats in the reviewed user interaction pattern collections.

Pattern languages differ from pattern collections in the fact that they provide patterns for all the sub-problems of the topic and try to guide the reader to a solution by leading them from one pattern to another. Apart from best practices to write stand alone patterns, Meszaros and Doble [54] also provide guides to organise and interlink a pattern language. Here, we highlight some of the solutions documented by these patterns. Related patterns must represent other solutions to the same problem, internal or external to the pattern language, more generic or specific variations of the pattern, or other patterns that solve a part of the problem in the described context. Regarding the structure and organisation of the pattern language, one should attempt to make each pattern "single-pass readable", meaning that the reader can understand the pattern after consulting it the first time. To achieve this, it is recommended to provide a concise and clear summary of the pattern language with a brief description of each pattern's problem and solution, a glossary, and an intent catalogue, where the patterns intents are summarised to facilitate cross-references between patterns. It is also noted by Meszaros and Doble that a pattern language should be introduced by explaining the high-level problem it aims to tackle and the patterns that contribute to it, along with why and how the patterns can be used together to solve the high-level problem.

Regarding the organisation of user interaction pattern languages, van Welie and van der Veer [77] outline one based on user interaction design problems. The idea is to present the patterns categorised in a top-down approach of the user interaction design problems, organising the language in a way that all design levels are covered by the patterns. Following this approach, four categories of patterns are defined: posture type patterns, that cover problems such as the interface structure and the main experiences offered; experience patterns, that describe the common user experiences and which lower level patterns must be used to achieve them; task patterns, that describe solutions to small user problems, such as a series of interactions on one or more objects to reach a goal; and action patterns, that describe common uses of well-known or custom widgets and typically

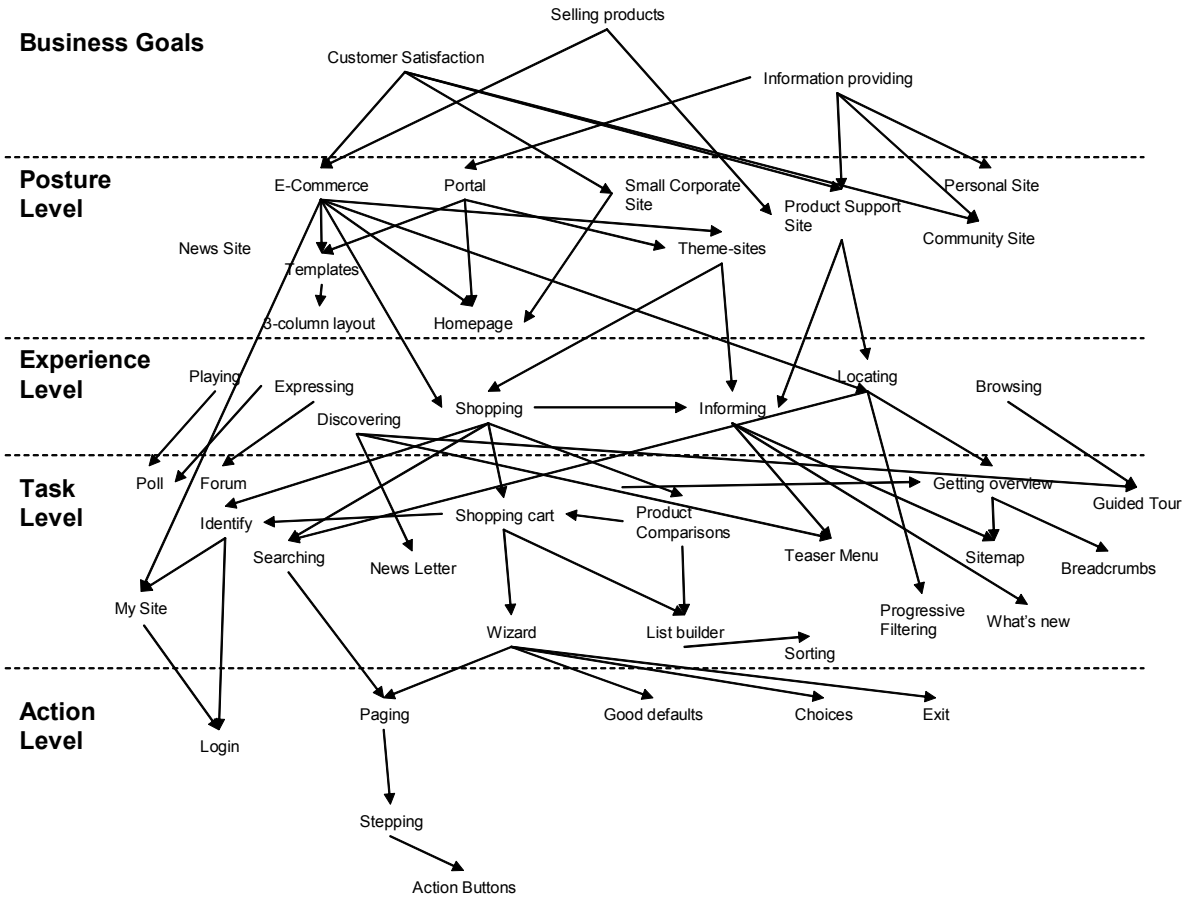


Figure 3.2: Organisation of a partial pattern language for web design (Image from van Welie and van der Veer [77]).

not related to a specific user goal. The example in Figure 3.2 shows the organisation of a pattern language on web design, following the previously described approach.

3.2.2 Methodology

Before trying to write patterns and build a pattern language, one has to collect knowledge about the domain at hand and create the pattern seeds. According to Sasabe et al. [65] the pattern seeds are pieces of knowledge and information that give origin to the written patterns. This phase of creating a pattern language is called pattern mining, where valuable information is collected from good practices and knowledge is gathered to outline the required elements of a pattern. It is considered of great importance as it directly impacts the writing phase. One cannot write a good pattern without good pattern seeds.

Iba and Isaku [45] outline three main types of pattern mining, mining based on: individual contributions, meaning knowledge collected is from the patterns' author own experience; second-hand contributions, the knowledge is gathered from other experts in the topic; and collaborative mining, when a group of experienced people mine their collective knowledge and experience.

Here, we will focus on pattern mining methods from secondhand and collaborative contributions, and highlight four approaches in the literature, to conduct the mining phase. These approaches are summarised in Table 3.3. The method proposed by Iba and Isaku [45] describes how to mine patterns in work groups by using sticky notes to write tips or rules that are important to the domain being explored. In "Patterns Mining" [19] a method is outlined to mine tips and valuable information from interviewing experts and researchers of the area. An approach to perform the mining phase in a workshop-like environment is described by Akado et al. [1]. This method allows the participants to reflect on and recognise their experience in the field through the conducted dialogue. A more straight forward yet trivial approach to pattern mining is the one used by Seffah [68], specific to user interaction and design patterns. This work details a method where common solutions can be identified by studying examples of user interfaces and identifying the solutions that are used with some frequency. However, while we consider this approach a way to start the process of pattern mining and a great mechanism to obtain knowledge on the domain at hand, we also believe it shouldn't be used on its own, but accompanied with one of the previously mentioned approaches. The solo use of the examples approach conducted by Seffah [68] could result in trivial patterns that do not add value to the problem one is trying to tackle.

	Description	**Contributions**	**Number of Participants**
Iba and Isaku [45]	Group pattern mining with sticky notes.	Collaborative	>2
Akado et al. [1]	Pattern mining workshop.	Collaborative	>2
Delano [19]	Web pattern mining interview.	Secondhand	>1
Seffah [68]	Examples collection.	Secondhand	1

Table 3.3: Overview of the reviewed pattern miming methods.

3.2.3 Evaluation and Validation Methods

After the pattern mining phase, the elaboration process of a pattern language is, usually, an iterative cycle of writing, evaluating and revising the patterns, as can be seen in Figure 3.3. These iterations take place on two different levels, the first phase is focused on shaping the patterns, and the second phase on evaluating and revising the pattern language. The evaluations and revisions conducted on the first level are commonly conducted through Shepherding [35] and in Writers' Workshops at PLoP conferences. However, Seidel [70] states that while the first level evaluations are necessary and important, additional methods to obtain feedback and orientation on the pattern language as a whole are necessary. The methods presented next belong to the second level of the evaluation process, and are summarised in Table 3.4.

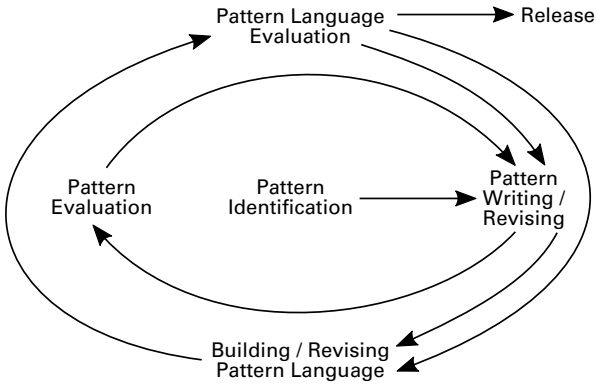


Figure 3.3: Schema on how to process the elaboration of patterns and pattern language (Image from Seidel [70]).

In "Empirical Evaluation Methods for Pattern Languages: Sketches, Classification, and Network Analysis", Seidel [70] outlines two main focus of pattern language evaluation, one is the evaluation of the written representation of the patterns, in order to assess their "completeness, briefness, and validity"; the other is the evaluation of the application and use of the patterns to assess the pattern language's "usability, feasibility, and its impact".

Three different methods are proposed for evaluating and analysing patterns and pattern languages with the intent of providing mechanisms to improve them. The first method is "Pattern Sketches" proposed to empirically evaluate the written comprehension of user interaction and design patterns, by assessing their ability to transfer and visualise the information in a practical way. It consists of asking the participants, which should represent the target audience for the patterns, to draw, as a sketch, the solution for the user interface design proposed by the pattern, like the examples in Figure 3.4. The patterns should be stripped down to its core components: name, context, problem and solution. Finally, the resulting sketches should be evaluated by the language authors on two dimensions: its correctness, meaning the sketch contains only valid interpretations of the patterns; and its completeness, meaning the sketch contains all parts of the solution. The patterns with the lowest results should be revisited and improved.

The second method proposed is "Pattern Classification", a method consisting of filing the patterns of a pattern language into a category system targeted at the domain at hand, and deducted by scientific theories. This method intends to identify which areas of the pattern language have more emphasis and which areas are underrepresented. Before using this method, a quality category system for the pattern language's domain must be chosen. The author recommends the use of a category system used for similar pattern languages. The result of this method is the distribution of the patterns in the categories as it can help the language's author to identify the ones with a low amount of assigned patterns. These categories must be reflected upon, focusing on if the language is not targeting them on purpose or if there are patterns missing.

The final method is a technique for analysing groups of patterns in a pattern language, named "Network Analysis". It should be used to identify if patterns are missing and which ones are dis-

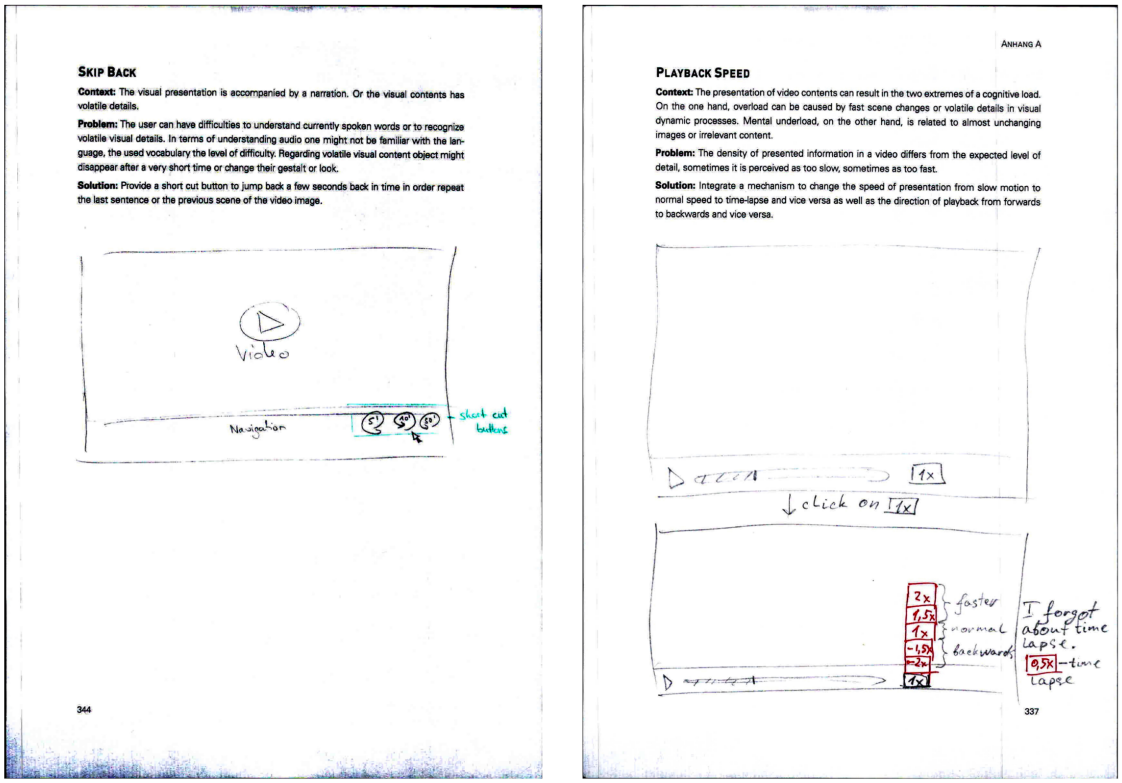


Figure 3.4: Examples of sketches from the “Pattern Sketches” method (Image from Seidel [70]).

pensable. This method analyses the interconnections between the patterns in a pattern language using network analysis methods by considering the pattern a graph. This graph represents the intended flow of information when reading or browsing the language. The results are network analysis metrics such as graph density, degree centrality, out-degree centrality, closeness centrality and betweenness centrality. Most of these metrics absolute values can only be assessed by comparing to other similar pattern languages. Nonetheless, they allow the language’s author to identify abstract patterns that are connected to a high number of patterns, or patterns considered a hub, because they connect different parts of the language.

Another work focused on providing methods for evaluating pattern languages is “An Evaluation Process for Pattern Languages” by Guerra and Fernandes [31]. The proposed method assesses the patterns on two levels, it evaluates their capability of representing a recurrent solution used in existent works of the field, and their usefulness in the development of new solutions. In order to achieve this, two different questionnaires, containing questions about the usage of each pattern and the usefulness of the pattern language as a whole, must be conducted. The answers must be considered positive, negative, or neutral to create a score for each pattern. One of the questionnaires must be answered by inexperienced people in the language’s field that have read the patterns previously and developed a solution using them. The second questionnaire must be answered by experienced people that have read the patterns and compared them to their previous experience. After the questionnaires, the answers must be analysed and a rating assigned to each

pattern, representing its usefulness to both the inexperienced and the experienced group of people.

	Description	**Goal**
Pattern Sketches [70]	Ask users to sketch each pattern's solution.	Comprehension of the written patterns.
Pattern Classification [70]	File all the patterns in a category system.	Completeness of the pattern language.
Network Analysis [70]	Perform network analysis methods to the pattern language graph.	Structure of the pattern language.
Guerra and Fernandes [31]	Conduct questionnaires with inexperienced and experienced users in the pattern language's field.	Pattern's usage and pattern language's usefulness.

Table 3.4: Overview of the reviewed pattern language evaluation methods.

Chapter 4

Interaction with Linked Data

In this chapter we present a review of the state of the art in user interaction solutions for Linked Data. With this review we aim to acquire knowledge about commonly used and evaluated techniques for interacting and visualising Linked Data. The study will cover different types of interaction with content, analysing works that allow users to navigate, search, create and edit Linked Data.

The remainder of this chapter is divided into 6 sections, in the first we detail the methodology adopted to perform this literature review and the criteria followed for including or excluding a work; in the second section we list related work that analyse and classify visualisation tools for Linked Data; in the third to fifth sections of this chapter we will present a brief description about each work, presenting them according to the type of application; and finally in the sixth section we will categorise the reviewed works according to their domain, evaluation methods, visualisation techniques, interaction techniques, and implemented search solutions.

4.1 Methodology

Aiming to extract useful knowledge from this review to build a pattern collection in user interaction design with Linked Data, we opted to only feature Linked Data tools and applications developed for a common user, a user with no previous knowledge of Linked Data technologies. We believe to be imperative the study of tools only targeted at non-expert users, due to one of the end goals of this work. We aim to present formalised and documented solutions to build Linked Data applications targeted at lay users in the form of a pattern language, to motivate the development of intuitive applications and consequently the general use of the Semantic Web.

Visualisation tools play a big part in studying user interaction with Linked Data. Even tough by definition visualisation tools are developed to offer its users features to explore the data to identify patterns, infer correlations, and causalities [9]; many tools featured in LD visualisation studies [58, 20] are in fact developed not only for people with previous experience with LD.

The works featured in this literature review are tools and applications developed for lay users and intended for navigating, searching, authoring, and visualising data, as opposed to analytical pattern and statistics extraction.

To conduct this literature review, we started out by defining 5 survey research questions by which we will guide our research. We aim to acquire knowledge about what visualisation techniques are commonly used to display Linked Data, what actions can the users perform in the applications, what are the most recurring search solutions, and information about the application domains and the interface evaluations. To answer these issues we defined the following survey research questions:

SRQ1: Which interaction actions can the user perform?

SRQ2: What visualisation techniques are commonly used?

SRQ3: How is the user interface proposed evaluated?

SRQ4: Which are the recurring domains covered?

SRQ5: Which search solutions are provided?

Next, we established inclusion and exclusion criteria to filter the works to be reviewed. We included works considered as LD visualisation tools, applications, or models to aid the development of user interfaces for LD applications; works targeted at users with no previous knowledge of Semantic Web technologies; and prioritised works that explicitly state the user interface as the main focus and that conducted usability tests for it. We excluded works not written in English and focused on tools for the extraction of statistics and analytical patterns from data.

Finally, we entered the following search query, `[("linked data" OR "semantic web") interaction]`, in several scientific databases, such as Google Scholar[1], ResearchGate[2], and ACM Digital Library[3], and selected works according to the previously described criteria. Additionally, we collected works by recursively analysing for each article the citations from it and the citations to it.

In the end, 18 works were identified. The selected 18 works are presented in chronological order, in the following three sections: Section 4.3, where we present tools and applications focused on navigation and visualisation of the data, providing the user with a complete experience; Section 4.4, where tools focused on improving the user experience while performing search tasks and creating queries are presented; Section 4.5, where we present formalised models for user interfaces and frameworks focused on facilitating the design and development of LD user interfaces.

[1] https://scholar.google.com
[2] https://www.researchgate.net
[3] https://dl.acm.org

4.2 Related Work

Several works have been published aiming to study and analyse Linked Data visualisation tools. Some surveying all the visualisation tools found in the literature [9, 16, 64, 20], others focusing on more specific subsets of Linked Data tools like Linked Data consumption tools [49], ontology visualisation tools [47], and Linked Data exploration systems [53].

In "Approaches to Visualising Linked Data: A Survey" [16] the authors start out by describing the key requirements that visualisation approaches for Linked Data must follow to be accessible to non-domain and non-technical users. It is defended that clear and coherent visualisation of Linked Data is essential to the use of the Semantic Web by everyone. In this survey the existing approaches to visualise Linked Data are reviewed and analysed according to the requirements for interactive visualisation outlined before and that the authors believe will lead to intuitive knowledge discovery for all users.

Similar to the previous work, in "Exploration and Visualization in the Web of Big Linked Data: A Survey of the State of the Art and Challenges Ahead" [9] the authors outline the prerequisites and challenges that should be followed by exploration and visualisation systems, and review the solutions developed by the Semantic Web community and analyse their alignment to the requirements. With this work the focus is the Web of Big Linked Data, and no requirements are outlined to ensure these systems are accessible and usable by users with no previous experience.

Another work surveying the state of the art of Linked Data visualisation tools is "Empirical evaluation of Linked Data Visualization Tools" [20]. In this work the authors formalise a set of goals that the tools must fulfil in order to "provide clear and convincing visualisations" and "encourage user comprehension". Each tool is then evaluated according to these goals to determine which ones can be used successfully by any Linked Data consumers.

"Interacting with Linked Data: A Survey from the SIGCHI Perspective" [64] is focused on studying the current interaction and visualisation approaches for Linked Data. This survey reviews Human-Computer Interaction (HCI) works on Linked Data and analyses them according to the following aspects: the end-users, the context, the design contributions, and how the research is validated. The authors highlight that there is little research supporting human end-users in querying, browsing, and visualising Linked Data, and that the ones that exist rarely include validation methods or theoretical conclusions.

In "Ontology Visualization Methods - A Survey" [47] a more specific set of Linked Data tools is analysed, specifically tools for visualising ontologies. A set of requirements is previously defined for this type of visualisation tools and the reviewed approaches were analysed according to it. The surveying tools were also categorised according to the different characteristics of the presentation, interaction techniques, functionalities supported, and visualisation dimensions. Another two surveys focused on smaller sets of Linked Data tools are a "Survey of Tools for Linked Data Consumption" [49] and "Survey of Linked Data based exploration systems" [53] where tools for Linked Data consumption and Linked Data exploration and search tools are reviewed, respectively.

Some of the previously mentioned works are focused on defining requirements that Linked

Data visualisation tools must follow to be accessible to non-technical and lay users, and then reviewing all the state of the art approaches according to those requirements. However, none set their goal as surveying the state of the art of Linked Data visualisation tools and applications targeted at inexperienced users. With this survey on solutions for user interaction with Linked Data, we set out to analyse and review the Linked Data applications and visualisation tools developed with inexperienced users in mind to find their most commonly used visualisation techniques, interaction features, and search approaches. Instead of focusing on evaluating and determining which tools from the set of published solutions could be used by lay-users, we aim to find the most used visualisation and interaction techniques and the evaluation methods in the subset of tools explicitly designed for inexperienced users.

4.3 Full-featured tools

In this section we describe the works that present platforms and applications for Linked Data that are focused on providing the user with a complete experience. These tools provide ways for users to explore and navigate the dataset, perform search queries, visualise individuals or groups of resources, and, in some cases create and edit data.

CS AKTiveSpace (CAS) [25] is a platform developed to take advantage of the semantic and distributed content collected related to Computer Science research in the United Kingdom. This application was developed, in part, to test the applicability to any domain of the mSpace model [24], described later in Section 4.5. Due to being based on mSpace, CAS user interface allows a semantic exploration of the domain tailored to its unique attributes. Users can explore the dataset by filtering down the instances by research area and location. An overview of the main instances, in this case researchers, is displayed to the user using a map, where they can select a specific area. When a researcher is selected, detailed information is also displayed in a simple textual format. While this application provides its users with a strong navigation approach, the authors state that it should be used as a complement to a more traditional keyword search, something not implemented in CS AKTiveSpace. No formal tests have been conducted to evaluate the user interface implemented, however it is stated that it was well received and motivated the development of other industrial and governmental funded platforms.

MuseumFinland [44] is a semantic portal for publishing heterogeneous museum collections in the Semantic Web. This application provides its users with several features, such as a global view of the collections, a content-based information retrieval system, and semantically linked and browsable content. Regarding the search features, MuseumFinland is equipped with a multi-faceted search system, that organises the resources in categories, as well as a simple keyword-based and geographical search. Additionally to helping the user perform search tasks, the categories used in the faceted search provide an overview of the kind of information present in the system, and can be used as a way of browsing the collections. When selecting an instance in a result list, the user is redirected to a global view of that instance, composed of the instance image, metadata, and a set of hyperlinks to semantically related instances. While no evaluation

of the choices taken during the user interface design was conducted, the authors highlight that the user interface and user interaction logic were based on previous works, namely Flamenco's multi-faceted search [38].

Ontowiki [3] is a Semantic Web tool developed for facilitating the visual representation and creation of a knowledge base for end users. This tool is focused on social collaboration aspects, and so it presents features such as change tracking, comments and discussions forums, rating and popularity metrics, and analysis of the users activity. To visualise a single instance, Ontowiki provides its users with an instance view that combines all of the properties and presents information about similar instances and incoming links. Additionally, if the properties of an instance have geographic values they can be visualised in a map, or if they have date values they can be visualised in the form of a calendar. Regarding the search features provided, Ontowiki allows for facet-based browsing and a semantically enhanced full-text search. In the first approach users can select objects according to certain property values, resulting in a search that will never lead to empty results. For the second approach, users can search for one or multiple keywords that occur in literal property values, the results can then be filtered by individuals of a class or literal values of a certain property. To author and edit the data an inline edit mode is present in all information resulting from statements, following the "What You See Is What You Get" approach. The users can quickly edit or add statements by clicking a small edit or add button next to the information.

DBin [75] is a tool for contributing and exploring the Semantic Web by exchanging data in P2P channels. DBin is a tool that allows users to browse data from different domains, and each domain is equipped with predefined settings for visualising, querying and annotating the data with the help of "community configurable user interfaces", called Brainlets. A Brainlet defines which classes and relations between items are featured in DBin's tree layout navigator. This allows for the organisation of the information displayed to be targeted at the needs and common tasks of a certain domain. For each one of the instances displayed in the tree layout, the user is presented with a view of its relationships, properties, related images and metadata. To aid inexperienced users to query the data, a Brainlet provides predefined common and useful queries to the domain, these are entitled as "precooked queries". Finally, to author and share the data, Brainlet lets its creator define custom domain dependent annotation templates, simple or more complex, and keep track of the annotations authors. No formal evaluation was conducted. However, DBin was compared to similar tools, and according to the authors stands out as an end user centered Semantic Web application.

OKM Knowledge Management (OKM) [18] is a Linked Data authoring tool developed to analyse and test how lay users can interact with Linked Data systems. OKM is based on a traditional user interface with "resource-centric" pages, where each resource has its own page with information and related resources, as can be seen in Figure 4.1. The users can follow these object properties to navigate the whole dataset. On top of this interface, the authors implemented some atypical features that were studied and analysed in order to assess their benefits to inexperienced users. One of the features was the inclusion of roles and templates. Instead of presenting a possibly exhaustive list of all the properties of a resource, the authors defend that a role must be defined

for each class of the domain, where the most relevant information is displayed first. When adding a resource to the system the user must also assign one or more roles to it. Additionally, a template is provided so that all relevant information for that role is filled by the user. To evaluate the impact of these features, user experiments were conducted with participants with no previous knowledge of Linked Data. From this study it can be concluded that while roles and templates facilitate the navigation and implicit creation of resources, they do not tackle all problems that arise by authoring LD. Firstly, for a user to benefit from a template they must select the right role and in general they are inconsistent unless relevant objects already have an appropriate role. Additionally, templates do not improve the correctness of the data entered by users and in some cases even lead to semantic incorrectness.

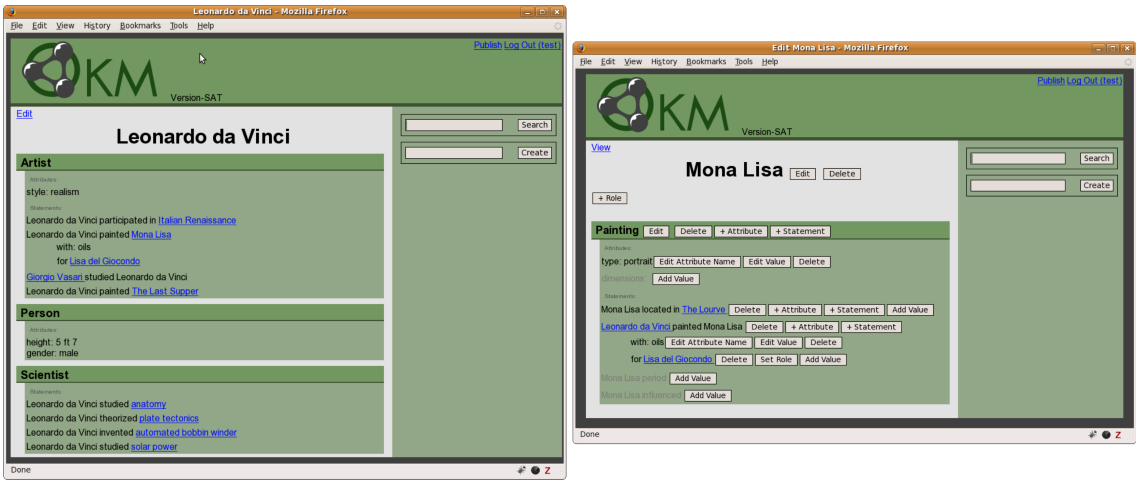


Figure 4.1: Example of a single individual visualisation and edit mode in OKM (Image from Davies et al. [18]).

The development of RDF Surveyor [79] was motivated by the lack of simple and intuitive solutions for user interfaces capable of being used by lay users to explore RDF data. The tool was developed to provide mechanisms to explore RDF datasets that hide the complex details of this kind of data. RDF Surveyor allows its users to navigate and explore the dataset, visualise an instance, and search using a facet-based approach and a keyword-based approach. In the page of an instance, shown in the example in Figure 4.2, the class types, data properties, and related individuals are displayed, when applicable a picture and a map view are also provided. The user can follow the links of the related instances to semantically browse the dataset. The interface

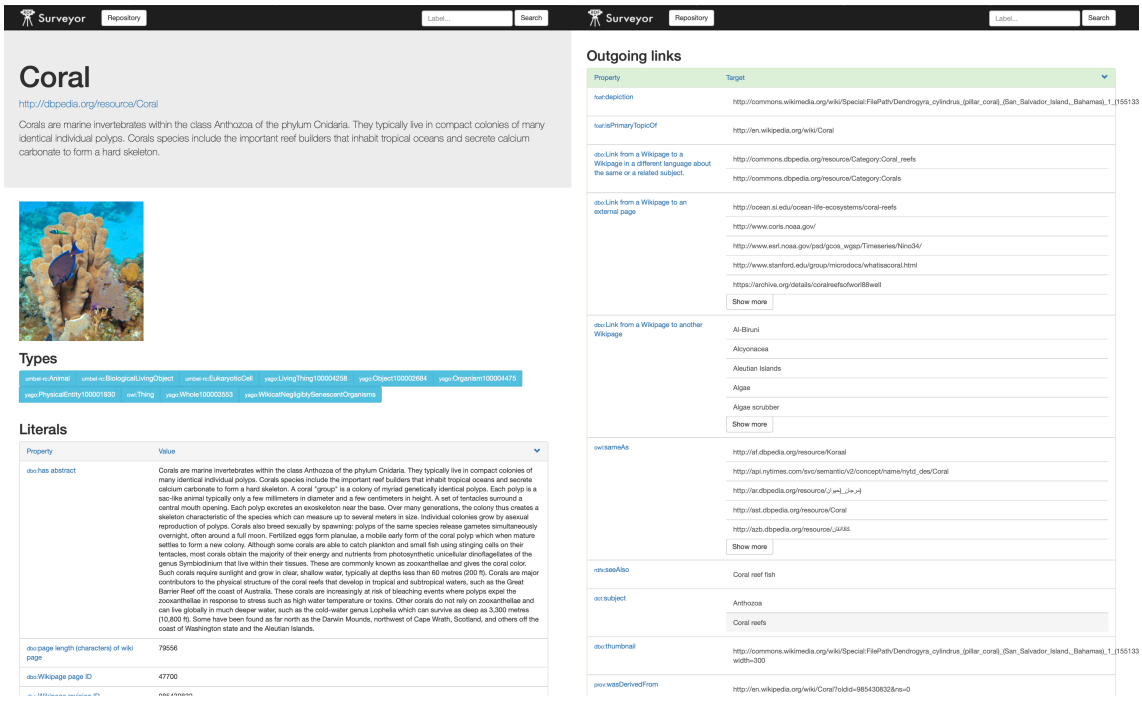


Figure 4.2: Example of a single individual visualisation in RDF Surveyor [79].

usability was evaluated through a questionnaire, involving 14 participants. However, only 4 of the 14 did not have previous experience with Linked Data, so we believe that the usability scores do not provide information about the intuitiveness of the platform for lay users. Even though most of the participants commented the tool was easy to use, RDF Surveyor's usability score was considered low when compared to similar tools. As expected, the participants with previous experience with Linked Data rated the interface higher than those with no experience. Nonetheless, it is considered as a valid and practical solution for exploring and navigating Linked Data datasets.

4.4 Browsers and exploratory tools

Here we present browser and exploratory tools, meaning Linked Data tools and applications that are focused on providing the user with ways to navigate the datasets, or perform specific search tasks. Most tools presented next offer both navigation and search features, as well as, visualisation features.

QuizRDF [17] is a search engine proposed in 2004, to combine the advantages of free-text search with the advantages of browsing and searching for RDF metadata. The authors of this work defend that a typical user usually benefits more from starting with a single node or resource and exploring the remainder of the dataset from there; as opposed to starting with a general overview of the dataset. In order to do this, QuizRDF provides its users with an initial keyword search query,

of ideally one or two terms, and a resulting ordered list of the results. Then, the users can further specify the search by selecting only the instances of a certain class, or the instances with a certain range of values for a given property. All the hyperlinks of properties relating the current class to others, or the super classes are clickable allowing the user to continue browsing and exploring the dataset. One of the limitations of this work, acknowledged by the authors, is the implemented user interface's lack of scalability. Currently, in QuizRDF there is no way of displaying intuitively a large number of classes and properties, resulting from highly interconnected datasets. Nevertheless, the authors believe QuizRDF provides a more intuitive and familiar way of searching and browsing for semantic information, compared to a traditional search engine or browser tool.

Piggy Bank [43] is a browser extension developed to provide users with ways of using Semantic Web content while exploring and browsing the web. The main feature of this tool is that it allows users to extract information from web pages and transform it into RDF data. Piggy Bank also allows the user to visualise, browse and search for the data that they have saved. Upon saving a retrieved item, considered as a RDF resource that belongs to a class, has property values, and other retrieved resources, the user can assign tags and keywords to it. The items of each tag are then organised in collections, and faceted-search can be performed using them. The visualisation provided by Piggy Bank in an instance view follows a very simple approach, displaying the property and their values in a table view.

Tabulator [8] is a RDF browser developed for both inexperienced and experienced users to interact with the Semantic Web, and post and refine RDF data. Tabulator's main goal was to provide a generic Linked Data browser, that at the same time lets users benefit from user interface tools commonly encountered in domain specific applications. Similarly to other works, Tabulator allows the users to semantically browse the web of data, following the links between different resources. These resources are displayed in a tree view, where nodes can be expanded to obtain more information about them. Tabulator also allows the users to query for resources by providing values to different fields. The results can be visualised in different ways, with a table, timeline, map, or calendar view. In either mode, searching or exploring, when a resource is selected, the user is directed to a page with an outline view of the resource featuring more detailed information about it. Even tough no formal usability tests were conducted for the user interface, the authors informally received feedback from technical and Linked Data experienced users, most of their comments and remarks were considered and incorporated in Tabulator's interface. While feedback from field experts is still important, we are unsure their requirements and problems are shared by or useful to inexperienced users.

TreePlus [51] is a tool for visualising graphs that can be applied to several domains and different types of graphs, like the ones resulting from Linked Data datasets. This tool presents the graph to the users in a tree layout, allowing them to interactively explore it by starting with a single node and incrementally expanding it. The main goals while developing TreePlus were to keep the relationship labels readable and the tree layout as steady as possible. TreePlus offers its users features like a preview of the adjacent nodes, zooming and panning, simple keyword search and browsing and sorting options. To evaluate the approach chosen in the development of TreePlus, a controlled

experiment was conducted to evaluate the completion time, success rate, errors and user confidence, in comparison with a traditional graph visualisation approach. The tests concluded that TreePlus performed better in most of the tasks and that its benefits increase with the density of the graphs.

Ozone Browser [12] is a tool to make use of the semantics embedded in web documents in order to improve the experience of the users while browsing the web. The Ozone Browser is a graphical overlay that provides contextual knowledge of the hyperlinks in a web document. Regarding its user interface, Ozone Browser dynamically generates different views according to the domain of the RDF data presented. The default visualisation is composed of the RDF triples and related entities, if geographical data is available a map view is presented.

RelFinder [40] takes on a different approach than other browsing focused Linked Data tools. RelFinder's goal is to extract a graph that covers all the relationships between two predefined objects, and present it to the user. To this very simple graph visualisation, some interactive features were implemented to reduce the number of nodes and relations displayed. To start using this tool the user must select the two main objects using a basic keyword search, then the tool displays the graph connecting these two options. To filter the displayed relationships and nodes, users can select a subset according to their length or the classes they belong to, allowing them to better cater the visualisation to their interests and needs. Finally, if a node is selected in the graph, a simple visualisation of the instance information is displayed, this includes an image, short description and classes it belongs to. While the approach taken in RelFinder can fulfil some of the needs of common web users, we believe it does not cover all use cases, nor is generic enough to be a solution capable of being implemented and integrated in every Linked Data application. One of its limitations is the inability to browse to a different part of the dataset not encompassed in relations between the two main objects.

The approach for visual SPARQL queries based on filter/flow graphs [33] was developed for users with no knowledge or experience with RDF data to be able to create complex queries and obtain in depth knowledge. This approach is composed of filters on the data or object properties of the data, that are connected as a graph. These filters include restrictions on ordinal values and strings, or resource identifiers. This tool underwent a qualitative user evaluation to assess the users ability to comprehend and compose SPARQL queries. While the results show that most users were able to quickly complete the tasks, half of the participants already had previous basic knowledge of Semantic Web technologies. With these results we cannot be sure the tool is suited for users with no experience with Linked Data. Nonetheless the nature of the filter/flow graphs approach results in a complex and odd shaped interface, that we believe not suitable for a generic Linked Data application.

PepeSearch [78] is a Linked Data tool for searching semantic datasets, targeted at mainstream users, and thus requires no understanding or previous experience with Semantic Web data formats. PepeSearch provides its users with searching and instance visualisation features, represented in Figure 4.3. Similar to other works, Pepesearch also allows its users to use a multi-faceted search, where they can apply different constraints to the data and object properties of all the classes in the

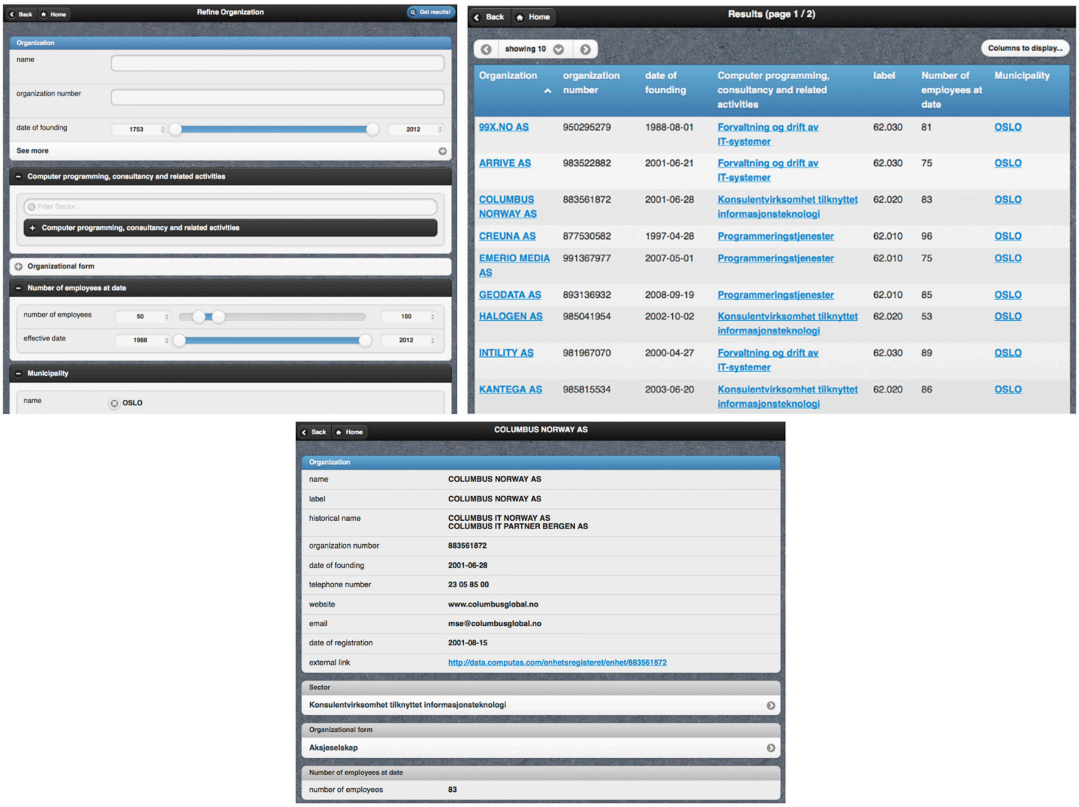


Figure 4.3: Example of the visualisation of a single individual, group of individuals, and search options from PepeSearch (Image from Vega-Gorgojo et al. [78]).

dataset. The results are presented in a table with the values of all properties, and when an instance is selected the user is redirected to a page dedicated to that instance. In this page the data and object properties of the instance are displayed and the user can follow the links to the pages of the related instances. To evaluate the usability of the interface developed, a usability questionnaire was conducted with participants with no previous experience in Linked Data. The results showed the interface had a higher usability score when compared to similar tools, and proved common web users are able to perform, without previous training, non-trivial search tasks using PepeSearch.

4.5 Models and Frameworks

In this section, we feature works that present formalised interaction models created for, or adapted to Linked Data environments. Additionally, we present software frameworks to develop user interfaces and model user interaction for semantic platforms.

The mSpace model [24] is an interaction model that can be used for interactions with Semantic Web applications. This model provides developers and designers with an intuitive and effective user interface that accommodates alternatives to the commonly used keyword search. Initially, the mSpace model was not developed for Semantic Web applications, but motivated by the lack of research in user interaction with Linked Data, the authors decided on the formalisation of mSpace as

an interaction model for the Semantic Web. mSpace is focused on facilitating the interaction with Semantic Web data to perform user-determined exploration of a certain domain. The main advantage of the model developed is that it enables users to first understand the scope, relations, and entities of a certain domain, and then explore it following the direction best suited for their needs and interests. The organisation of the different classes and relations in the interface results from a flattened projection of the domain. The default visualisation of this model is a multi-column table view, where data classes are ordered from left to right. The users can then browse through the data selecting an instance in each column. By selecting sequentially one instance from each column, the instances are filtered according to their relations, this browsing mechanism allows the users to chose their path while navigating the domain. Additionally, to better accommodate the user interests, the classes presented in the table can be rearranged, or substituted by others. To complement the browsing approach, mSpace provides users with a detailed view for the last selected instance, that displays contextual information about it. While, the mSpace model for Semantic Web has not been formally evaluated, platforms like the previously presented CS AKTiveSpace [25], based on the mSpace model have had a positive feedback from users.

Haystack [60] is a platform to facilitate the development of Linked Data applications that allow lay users to create, manipulate, and visualise RDF data. Haystack focuses on providing developers and designers with user interface components that were designed specially to handle the heterogeneity of RDF data and contribute to a good user experience. Haystack allows designers to specify different types of views, differing in size and location, for every type of resource in the domain. Additionally, developers can also add customisable operations tailored to a resource, available in the view displayed to the user. This tool provides users with a keyword-based search, a simple textual visualisation of each resource, and mechanisms to author, edit, and annotate the resources. Finally, users can semantically explore the domain by following the links of the related resources.

The Information Workbench [34] is a platform that supports the whole development process of developing a Linked Data application. Here, we will be focused on reviewing the options and chosen solutions for the user interface design. Firstly, the platform supports the implementation of a basic keyword search, as well as graph pattern search and faceted search. The Information Workbench provides the developers with a wide range of different visualisation widgets to accommodate the high heterogeneity of Linked Data datasets, as demonstrated in Figure 4.4. For a set of instances or resources, views like maps, lists, timelines, and graphs are provided. For a single instance, information can be consulted in a generic textual view, tabular view, or graph view, where only the neighbourhood of the instance selected is displayed. In all these visualisation options the user can semantically browse the dataset by clicking in the related instances hyperlinks. Additionally to searching, visualising and browsing the data, users are provided with simple widgets to edit and annotate the presented data. All in all, the Information Workbench provides developers with easy to implement widgets that provide users with visualisation techniques and search solutions commonly present in other Linked Data tools, like the ones previously presented.

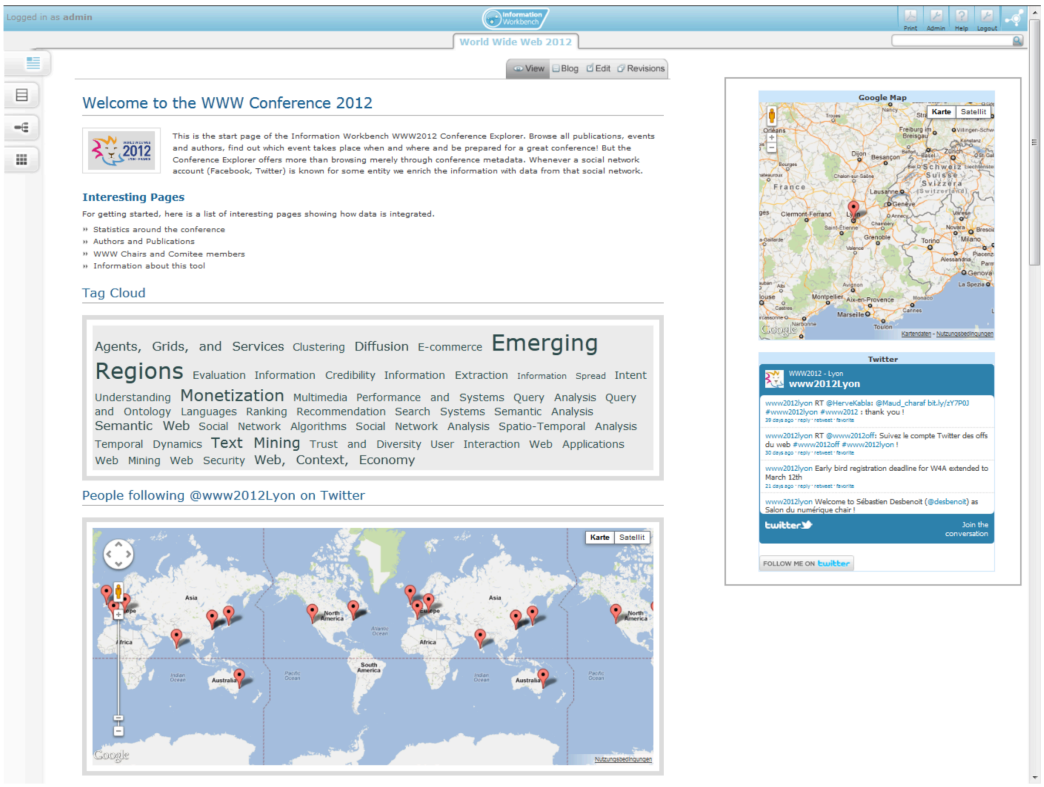


Figure 4.4: Example from Conference Explorer, a platform built with Information Workbench (Image from Haase et al. [34]).

Sampo-UI [46] is a software framework for developing customisable and responsive user interfaces for semantic portals. Ikkala et al. [46] started out by outlining the requirements for the user interfaces for Semantic Web applications, based on their experience in developing these semantic portals. The first requirement presented is providing the user with a variety of different ways of exploring the data. For this, the authors highlight some search paradigms: free-text search, faceted search, geospatial search, and temporal search. Another requirement is providing the users with different views of the search results, in the form of tables, lists, geospatial, or temporal visualisations. The final requirement outlined is that the user interface should support semantic browsing, meaning semantic links should allow the users to browse through the dataset, from one instance's page to another. These instance pages are described as a landing page for an entity, providing information about its metadata, and internal and external links to other related entities. The development of Sampo-UI was based on these requirements and components to implement these features were developed. The resulting framework allows developers to build a semantic portal that follows the previously described requirements and that is composed of three main types of pages: a landing page for the portal, where free-text search can be performed; a faceted search page; and customised entity landing pages for each different class in the application domain. Examples of the interface of WarVictimSampo[4], a platform built with Sampo-UI, can be consulted in Figure 4.5. The authors defend, based on their experience developing semantic portals, that the framework is viable, useful, and easy to use for developers wanting to create semantic portals for end-users.

4.6 Discussion

To answer the survey research questions previously stated, each reviewed work was classified according to its interaction techniques, visualisation techniques, search solutions, user interface evaluation, and domain. The information gathered was then systematised in 5 tables and analysed. Regarding SRQ1 (Which interaction actions can the user perform?) we categorised the works according to the main interaction actions provided to the users, summarised in Table 4.1. These were divided in 4 categories: (1) exploration/navigation, meaning the users were able to semantically browse the data by following related resources, and without the need to always start a new search task; (2) search, where users can query the system for a single or a group of resources; (3) author/edit, where the users are provided with mechanisms to edit, create, or annotate resources; (4) visualisation, applications where users have access to the visualisation of a single resource or a group of related resources. We can conclude that the majority of works provide ways for users to explore, search and visualise Linked Data, with only a small group of works providing features to author and edit. We can assume the lack of tools for authoring data is due to the fact that most targeted at lay users are being focused on browsing and visualising it, as opposed to focused on users contributing to the dataset. Another reason might be the complexity of designing an interface that motivates the users to input semantically correct RDF data, with no previously

[4] https://sotasurmat.narc.fi/en

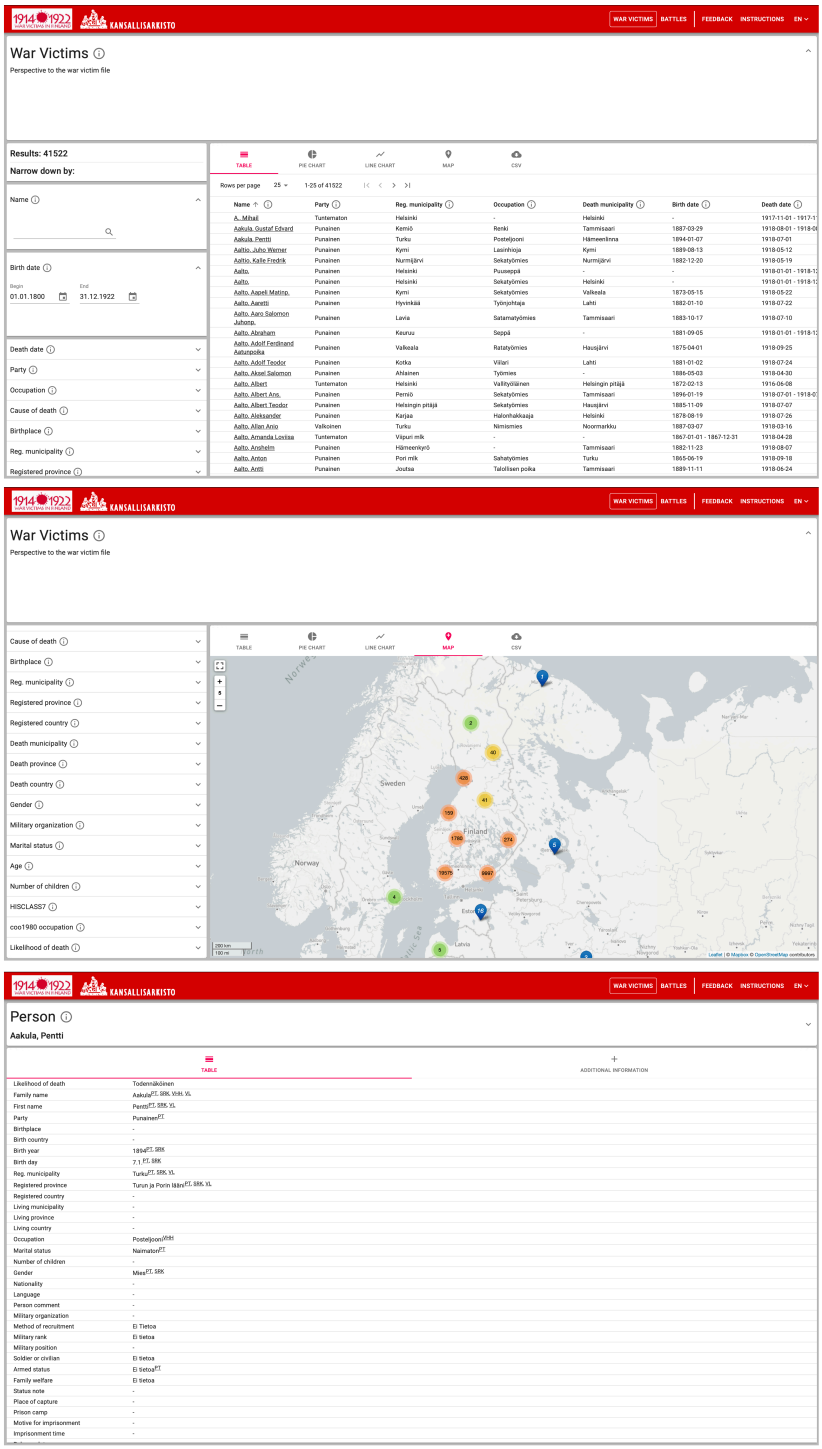


Figure 4.5: Examples from WarVictimSampo, a platform built with Sampo-UI [46].

knowledge of these technologies. It is also worth mentioning that the two works with no visualisation features [51, 33] are solely focused on the development of a novel approach for lay users to construct SPARQL queries and exploring graph-based data, respectively.

To survey the visualisation techniques in the reviewed works, we decided to analyse two different kinds of visualisation, the visualisation of a single instance, and the visualisation of a set of instances. The gathered information to answer SRQ2 (What visualisation techniques are commonly used?) is summarised in Table 4.2. For the single instance visualisation, the most common technique is a textual view of the resource, where the related information, data properties, object properties, and other related resources are listed. However, several works highlight the fact that this approach can quickly make the interface too cluttered and dense, when the dataset used is highly connected and the classes have a large number of properties. To overcome this, some approaches use collapsible sections, to hide and show more information upon a user's need. The map and calendar techniques consist of only displaying the related resources with geographical and date data, respectively. The graph view for a single instance, only adopted in the Information Workbench model [34], is used to display the neighborhood of related resources, suitable to visualise social networks or family trees. Regarding the visualisation techniques for a set of instances, the most common approaches are lists and tables, the only difference between the two being that a table usually displays some of the instance's attributes, while a list only displays the instance's label. Surprisingly, the least common approach was the graph visualisation, usually deeply connected to the visualisation of Linked Data, due to its natural network format. One explanation for this observation is that all tools are developed to be used by anyone with no previous experience with Linked Data, and graph visualisations, specially when large and highly connected, usually result in usability issues for these users.

To answer SRQ3 (How is the user interface proposed evaluated?) we categorised the works according to the type of evaluation conducted to validate the user interface, summarised in Table 4.3. A large majority of around 70% of the works reviewed did not formally test their interfaces, which proves the opportunity to develop and evaluate solutions for user interaction with Linked Data systems. The domains of the reviewed tools were assessed to determine which solutions could be converted into other domains. The information to answer SRQ4 (Which are the reoccurring domains covered?) is detailed in Table 4.4, where we can see that a large majority of tools do not have a specific domain associated. We can point out that all the user tested works are not limited to a specific domain, and with most of them showing positive results we can conclude that they could be applied to improve LD interfaces in all domains.

Finally, regarding SRQ5 (Which search solutions are provided?) we collected the search solutions available in the subset of works that provided users with features to search for a single or a set of resources. Table 4.5 summarises the gathered information, and clearly shows that the two most common approaches are faceted-based search and keyword-based search. Tools that offer a faceted-based search allow users to search for resources specifying the class they belong to, data property values, or other related resources. We believe a faceted-based search is the most promising solution for semantic datasets, as it allows users to benefit from the semantic knowledge of the

data. However, we find it best used together with the more traditional keyword-based approach, where users can search for resources by matching the entered keywords with the resources' label or other literal attributes.

From this survey of solutions for user interaction with Linked Data, we can point out the use of faceted search solutions, in addition to keyword-based approaches as a recurrent solution. In terms of visualisation, we found several applications that provide multiple visualisation techniques when displaying a resource to allow the users to customise their experience. We also observed a low number of applications that provided users with solutions to create and edit data. All in all, we noted that researchers have started to deviate from more traditional visualisation techniques, with graph visualisations being the least used approach, which justifies the need for documentation of these alternative solutions in order to broaden their use. Furthermore, as most solutions encountered are not connected to a specific domain or user task, they may prove easy to document and abstract to be used by others in various domains.

	Exploration/ Navigation	**Search**	**Author/Edit**	**Visualisation**
CS AKTiveSpace (2004) [25]	✓			✓
MuseumFinland (2005) [44]	✓	✓		✓
Ontowiki (2006) [3]		✓	✓	✓
DBin (2006) [75]	✓	✓	✓	✓
OKM (2010) [18]	✓	✓	✓	✓
RDF Surveyor (2019) [79]	✓	✓		✓
QuizRDF (2004) [17]	✓	✓		✓
Piggy Bank (2005) [43]	✓	✓		✓
Tabulator (2006) [8]	✓	✓		✓
TreePlus (2006) [51]	✓	✓		
Ozone Browser (2009) [12]	✓			✓
RelFinder (2009) [40]	✓	✓		✓
Filter/Flow (2014) [33]		✓		
PepeSearch (2016) [78]		✓		✓
mSpace (2003) [24]	✓			✓
Haystack (2003) [60]	✓	✓	✓	✓
Information Workbench (2011) [34]	✓	✓	✓	✓
Sampo-UI (2020) [46]	✓	✓		✓

Table 4.1: Overview of the interaction actions provided by the works reviewed.

	Instance View					Instance Set View					
	Textual	Map	Calendar	Table	Graph	List	Table	Tree	Map	Timeline	Graph
CS AKTiveSpace (2004) [25]	✓					✓	✓		✓		
MuseumFinland (2005) [44]	✓					✓					
Ontowiki (2006) [3]	✓	✓	✓			✓	✓				
DBin (2006) [75]	✓							✓			
OKM (2010) [18]	✓					✓					
RDF Surveyor (2019) [79]	✓	✓				✓					
QuizRDF (2004) [17]	✓					✓					
Piggy Bank (2005) [43]	✓					✓	✓				
Tabulator (2006) [8]	✓					✓	✓	✓	✓	✓	
TreePlus (2006) [51]								✓			
Ozone Browser (2009) [12]	✓	✓									
RelFinder (2009) [40]	✓					✓	✓		✓	✓	✓
Filter/Flow (2014) [33]	✓				✓						
PepeSearch (2016) [78]	✓					✓					
mSpace (2003) [24]	✓						✓				
Haystack (2003) [60]	✓					✓					
Information Workbench (2011) [34]	✓			✓	✓	✓	✓		✓	✓	✓
Sampo-UI (2020) [46]	✓					✓	✓		✓	✓	

Table 4.2: Overview of the visualisation techniques used in the reviewed works.

	None	Qualitative	Quantitative
CS AKTiveSpace (2004) [25]	✓		
MuseumFinland (2005) [44]	✓		
Ontowiki (2006) [3]	✓		
DBin (2006) [75]	✓		
OKM (2010) [18]		Satisfaction Study (71)	User Study (71)
RDF Surveyor (2019) [79]			Usability Study (14)
QuizRDF (2004) [17]	✓		
Piggy Bank (2005) [43]	✓		
Tabulator (2006) [8]	✓		
TreePlus (2006) [51]		User Study (28)	
Ozone Browser (2009) [12]	✓		
RelFinder (2009) [40]	✓		
Filter/Flow (2014) [33]		User Study (10)	
PepeSearch (2016) [78]			Usability Study (15)
mSpace (2003) [24]	✓		
Haystack (2003) [60]	✓		
Information Workbench (2011) [34]	✓		
Sampo-UI (2020) [46]	✓		

Table 4.3: Overview of the evaluation methods used.

	Generic	Domain Specific	Dataset
CS AKTiveSpace (2004) [25]		✓	UK Computer Science Research
MuseumFinland (2005) [44]		✓	MAO
Ontowiki (2006) [3]		✓	Scientific Conferences
DBin (2006) [75]	✓		–
OKM (2010) [18]	✓		–
RDF Surveyor (2019) [79]	✓		DBPedia
QuizRDF (2004) [17]	✓		–
Piggy Bank (2005) [43]	✓		–
Tabulator (2006) [8]	✓		–
TreePlus (2006) [51]	✓		–
Ozone Browser (2009) [12]	✓		–
RelFinder (2009) [40]	✓		DBPedia
Filter/Flow (2014) [33]	✓		DBPedia
PepeSearch (2016) [78]	✓		Norwegian Business Entrepises
mSpace (2003) [24]	✓		–
Haystack (2003) [60]	✓		–
Information Workbench (2011) [34]	✓		DBLP
Sampo-UI (2020) [46]	✓		War Victims 1914-1922

Table 4.4: Overview of reviewed works domain.

	Facet-based	**Keyword-based**	**Predefined queries**	**Geographical**	**Temporal**
MuseumFinland (2005) [44]	✓	✓		✓	
Ontowiki (2006) [3]	✓	✓			
DBin (2006) [75]			✓		
OKM (2010) [18]	✓				
RDF Surveyor (2019) [79]	✓	✓			
QuizRDF (2004) [17]		✓			
Piggy Bank (2005) [43]	✓				
Tabulator (2006) [8]			✓		
TreePlus (2006) [51]		✓			
RelFinder (2009) [40]		✓			
Filter/Flow (2014) [33]	✓				
PepeSearch (2016) [78]	✓				
mSpace (2003) [24]	✓				
Haystack (2003) [60]	✓				
Information Workbench (2011) [34]	✓	✓			
Sampo-UI (2020) [46]	✓	✓		✓	✓

Table 4.5: Overview of the search techniques in the reviewed works.

Chapter 5

Problem Definition

In previous chapters we have presented an overview of what is Linked Data and the Semantic Web, what are user interaction patterns and pattern languages, a review of the methodologies to create and validate them, and a survey of the most used techniques and solutions for visualising and designing user interaction in Linked Data environments. We sought to justify how this research can contribute to both Linked Data and user interaction design research, while at the same time motivate the development of good user interfaces that help grow the use of the Semantic Web by users with no previous experience. Based on the conducted research, we argue that there is a lack of scientific literature on how to build a user interface for a Linked Data application, that is supported by a systematic collection of knowledge and a formalised evaluation. Here, we describe the hypothesis and associated research questions that will guide this research.

5.1 Problem Statement

As the central hypothesis for this work, we aim to provide evidence that

> While designing user interaction for Linked Data applications, there are categories of recurring problems, for which solutions emerge from good design principles and best practices. The solutions are abstract enough to adjust to different contexts and, can be described using a pattern language.

We believe that developers and designers, even working independently tend to converge into similar solutions by applying knowledge and design principles common in the field. However, the resulted solutions are tailored and adapted to each unique scenario, and so can be abstracted to fit every context. We propose to research these practices and recurring problems, mined through observation, literature review, and interviews with experienced developers. We aim to systematise these practices in a pattern collection, so that they can be easily applied by other Linked Data developers and designers.

5.2 Research Questions

To provide aid in the answering of the previously described hypothesis, we outline 3 research questions (RQs) that will guide this research.

RQ1: What are the recurring problems when designing user interaction for Linked Data applications?

What recurring problems can we observe from the state of the art Linked Data applications? Which consequences arise when wrong design choices are taken?

RQ2: What techniques or strategies are used to tackle these problems?

Have developers and designers converged in similar solutions when approaching the problems pointed in RQ1? Is it possible to formalise and abstract these decisions? Can we identify solutions that are related to other solutions?

RQ3: How can the abstracted solutions be applied to a specific context?

How can a developer or designer apply in a practical way the abstracted solutions? Which possible difficulties must be taken into consideration?

Chapter 6

Methods and Materials

To reach this study's goals and answer the described hypothesis, we structured this research into three stages: pattern mining, pattern writing, and validation. We will start by conducting an online survey of the participants' experience with Linked Data. Then, to mine the pattern seeds, we will perform interview sessions with experienced researchers and Linked Data developers to gather knowledge of the main recurring problems and their solutions. To complement the results from these interviews, we will rely on the study of LD applications to identify common solutions that could be abstracted. We will follow the best practices for writing patterns and pattern collections collected for the pattern writing phase and choose a pattern format suited for user interaction patterns. To validate the resulting pattern collection, we will perform three main tasks: (1) analyse the pattern collection structure and identify its strong and weak points; (2) evaluate the usage and adoption of each pattern by the Linked Data community; (3) and lastly, we will apply the pattern language to a specific use case and provide a description of the process and results. This chapter presents the main steps in the methodology outlined for this work, describing the goals and methods followed for each step.

6.1 Experience with Linked Data Survey

Online questionnaires facilitate the dissemination of surveys without the geographical barriers and with less temporal limitations. We used an online questionnaire to reach Linked Data researchers, professionals, and users to gather quantitative and qualitative data on their experience.

The goal of this online questionnaire was to survey the participants experience with Linked Data, the most common problems when using and developing Linked Data applications, the usefulness of user interaction patterns for this topic, and ultimately the recruitment of participants for the pattern mining interviews. Thus, the questionnaire is targeted at both people with experience developing with and modelling Linked Data, and people experienced in using Linked Data applications and tools.

6.2 Pattern Mining Interviews

The first step in the process of creating and writing patterns is the collection of knowledge about the domain at hand. This phase is called pattern mining, as it consists of gathering patterns seeds, pieces of knowledge and information that will give origin to the patterns. There are a variety of methods in the literature on how to perform pattern mining, such as in work groups [65] or in workshops [1] with other pattern authors. However, these methods require a previous strong experience and knowledge on the domain by the author, as well as, a large group of people available at the same time and with experience on the domain. Due to the time restrictions, difficulty in finding experienced people on the field available to participate, and lack of deep knowledge and strong experience on the domain by the author, we decided to conduct expert interviews as a method of pattern mining. A pattern mining interview [19] consists on an individual interview with an experienced Linked Data researcher or developer, with the goal to gather common user interaction problems, and their knowledge on how to solve and overcome them.

6.3 Pattern Validation

After conducting the pattern mining and writing phases in the creation of a pattern collection, usually the patterns undergo an evaluation phase on two levels. The first level is targeted at shaping the patterns and the second on evaluating the pattern collection. The evaluations on the first level are commonly conducted through Shepherding [35] and in Writers' Workshops at PLoP conferences. However, due to the time constraints of this study, we chose not to perform these types of evaluations on the pattern collection.

For the evaluations on the second level, we surveyed 4 different methods to evaluate the patterns and pattern collection: Pattern Sketches [70] to evaluate the comprehension of the written patterns; Pattern Classification [70] to assess the completeness of the pattern language; Network Analysis [70] to analyse the structure of the pattern language; and the method outlined by Guerra and Fernandes [31] to evaluate the patterns' adoption and usage, and the pattern language's usefulness. From this set of methods, we chose to conduct the Pattern Classification, and the described questionnaire with experienced people to assess the patterns' usage. These two methods best suit the needs of the pattern collection and the constraints imposed by this study.

6.4 Wikidata Practical Case

As a final method of validation for the proposed pattern collection, we conducted an analysis of Wikidata's [80] user interfaces. Wikidata is a free knowledge base that integrates into a single location the Wikipedia's data, using Linked Data. Our main goal for this analysis was to present to developers and designers how the proposed patterns can be implemented and used on a specific context. In order to do this, we started out by identifying the patterns that are already being applied in Wikidata's user interface. Next, we identified which patterns could be additionally implemented

and described why and how, and presented mockups that illustrate a possible concrete application of them on the interface. Finally, we described this experience, focusing on the difficulties and problems that were faced, and that could occur with other applications.

Chapter 7

Linked Data Experience Survey

Before starting the pattern mining phase of the development of user interaction patterns for Linked Data, we set out to survey the opinion of members of the Linked Data community on the usefulness of user interaction patterns for the field, and to find Linked Data experts and developers willing to participate in pattern mining interviews. In order to do this, we developed and published an online questionnaire surveying the experience with Linked Data of the participants, common problems when using Linked Data applications, and the usefulness of user interaction patterns. This chapter describes the conducted web survey with over 30 Linked Data researchers and developers, highlighting the design of the questionnaire, the analysis of the results obtained, and the conclusions taken from it.

7.1 Design and Execution

The survey performed was carried out as an investigation tool for this research on user interaction with Linked Data. The goal of the online survey was to collect information on the common problems of user interaction with Linked Data by asking the participants to share their experience with Linked Data/RDF/Semantic Web applications. Additionally, our intent was also to gather contacts of experienced researchers and developers willing to participate in a more thorough interview about their experience and findings. The goal of these more in depth interviews is to perform pattern mining, which means the collection of knowledge that will originate possible user interaction patterns for Linked Data.

The questionnaire used for this online survey was designed aligned with the goals mentioned previously, its contents and rationale are presented and described in Section 7.1.1. After designing the questionnaire we set it up using Google Forms and disseminated it via email to Linked Data researchers, online forums, and mailing lists on the topic. Some of these mailing lists were

the following: W3C's RDF-DEV Community Group[1], Linked Open Data[2] mailing list, and the Knowledge Graph Conference[3] community on Slack. The form was online from the 5th of March 2021 to the 30th of April 2021, during which time we got 32 responses.

7.1.1 Questionnaire

The questionnaire is composed of a total of 15 questions, to be answered in an average total time of 5 minutes. The first section comprises 7 questions about the participants' experience with Linked Data as users or as developers. Section 2, composed of 4 questions, surveys the participants opinion on the importance and need for design patterns for user interaction with Linked Data. Section 3, with 5 questions, aims to identify participants interested in participating in a pattern mining interview. Given that the participants were all volunteers, we took into account that their availability and willingness to spend time answering the survey was limited. Due to this, we made an effort in balancing the number of questions with the time required to answer them, allowing us to extract useful information without the questionnaire answer time becoming too long.

In the next sections, we describe each question and their response options, along with its rationale. The original questionnaire is available in Appendix A.

7.1.1.1 Experience with Linked Data (ELD)

This section of the questionnaire is composed of questions on the participants experience using or developing Linked Data applications and tools.

ELD1. Experience with Linked Data The users must rate their experience with Linked Data on three different levels: Studying, Developing, and Using. They can rate their experience from 0 - None, 1 - Low to 5 - High. This question was included to survey the participant's background experience with linked Data. The participant's background was a sorting criteria when choosing and contacting people for the pattern mining interviews, as people that have studied and developed using Linked Data would be preferable to people that have only used Linked Data.

ELD2. Experience developing Linked Data applications The users must rate their experience developing Linked Data on three different aspects: Modelling, User Interface, and Backend. They can rate their experience from 0 - None, 1 - Low to 5 - High. As the development of Linked Data applications and tools encompasses many different steps and areas of expertise, knowing the participant's experience in each one was important when analysing their mentioned problems and struggles. When trying to collect knowledge on how to design user interaction, people with a strong experience in user interfaces for Linked Data are preferable.

[1] https://lists.w3.org/Archives/Public/semantic-web/
[2] https://lists.w3.org/Archives/Public/public-lod/
[3] https://www.knowledgegraph.tech

ELD3. What are some of the Linked Data projects/research studies you have been involved with (using or developing)? This question was included as a way to give the participants the space to highlight any research studies on projects that they have been involved with. It was important for us to collect more resources that we haven't come in contact which, especially from people with experience developing user interfaces for Linked Data.

ELD4. As a user of Linked Data applications, how do you rate your usage frequency for each of the following tasks? The users must rate their usage frequency from 0 - None, 1 - Low to 5 - High, in each of the following tasks: visualising, performing search tasks, browsing and navigating, creating and editing data. From our previous research on Linked Data applications, we found that the features offered to users usually fall under one of these categories, and it was important to gather knowledge on the participants' usage frequency of each one of the tasks, especially correlated to their experience.

ELD5. As a user of Linked Data, how do you rate the frequency and importance of problems encountered for each of the following tasks? The users must rate the frequency and importance of the problems encountered from 0 - None, 1 - Low to 5 - High, in each of the following tasks: visualising, performing search tasks, browsing and navigating, creating and editing data. In line with the previous question, it was also important to understand which main tasks commonly pose more problems to the users. This question allowed us to start to understand where most of the user interaction problems in Linked Data would be focused on.

ELD6. Which are some of the most common problems you have encountered when using Linked Data applications? This question was included as a way to start mining and gathering common problems people encountered when using linked Data applications. We kept the question open to any kind or category of problems to give a chance for all participants to share their experience, whether they are familiar with the concept of user interaction or not.

ELD7. Which Linked Data applications do you consider to be excellent examples in terms of supporting user interaction? The goal with this question was to collect and gather applications and tools that the participants considered as good examples regarding user interaction.

7.1.1.2 Importance of problems (IP)

This section of the questionnaire is composed of questions on the participants opinion on the importance of design patterns for user interaction with Linked Data.

IP1. How do you rate the need for guidelines for designing user interaction with Linked Data? The users must rate this need from 1 - Low to 5 - High. Understanding if the participants feel the need or consider important the existence of guidelines for designing user interaction with Linked Data was one of the main goals of this questionnaire. Due to the fact

that patterns are a way of documenting best practices and can serve as guidelines for anyone developing Linked Data applications, this question allowed us to form an idea on whether people on the Linked Data community consider this lack of guidelines a real problem.

IP2. Are you familiar with user interaction patterns or other pattern collections? The users must answer yes or no. Surveying the participants previous knowledge of user interaction patterns allowed us to have an idea on the percentage of people that have used or studied user interaction patterns before.

IP3. To what extent do you think patterns for user interaction with Linked Data would be useful for developers? The users must rate the usefulness of user interaction patterns for Linked Data from 1 - Low to 5 - High. One of the goals of this questionnaire was to validate our research with members of the Linked Data community, and the patterns end-users. Surveying the participants opinion on the usefulness of user interaction patterns for developers allowed us to form an idea about the potential adoption of this research within the Linked Data community.

7.1.1.3 Contacts and Comments (CC)

This section designed is to collect the contacts of the participants interested in participating in a thorough interview, and for the participants to leave any comments or suggestions.

CC1. Are you interested and available for a follow-up interview? The users must answer yes or no. One of the main goals of this questionnaire was to collect a group of contacts of Linked Data developers or researchers willing to participate in a pattern mining interview about their experience.

CC2. Do you have suggestions on other contacts, mailing lists, projects, or researchers that you think would be relevant to contact? This question comes as a way of gathering other contacts and communication channels within the Linked Data community to share the questionnaire with the largest number of relevant people possible.

CC3. Other comments? This question provides a space for the participants to leave any comments, suggestions, or critics on the research study we are conducting.

7.2 Data Analysis

The next sections present a data analysis of the results obtained in the conducted survey on experience with Linked Data. The results described next were collected from the 32 responses of Linked Data experts and developers. We will characterise the participants according to their experience with Linked Data, perform an analysis on the problems reported by the participants, and analyse the results on the usefulness of Linked Data interaction patterns and how they relate to the participants experience.

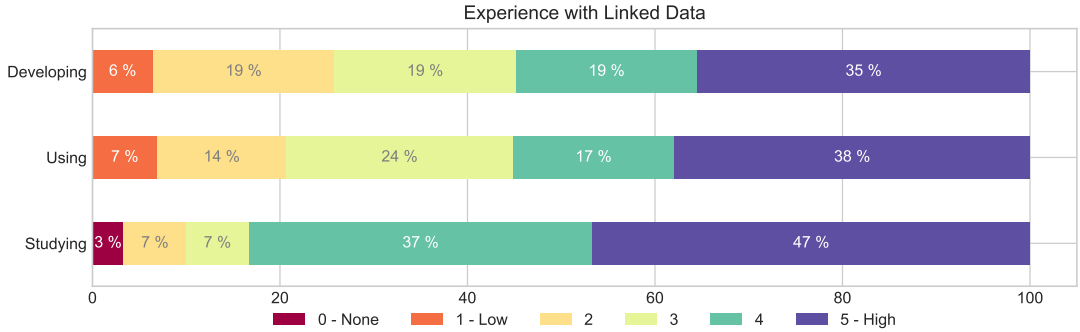


Figure 7.1: Results of the participants experience with Linked Data.

7.2.1 Participants' Experience with Linked Data

We asked every participant to rate their experience studying, using, and developing Linked Data from 0 to 5, 0 corresponding to no experience, and 5 corresponding to a high level of experience. The results are presented in Figure 7.1, where we can consult the percentage of each level of experience for the tree categories. We can point out that a large majority of about 80% of the participants rated their experience studying Linked Data with a high level (4 and 5). This leads us to assume that most of the surveys' participants are likely to be experienced researchers and have academic work developed in the Linked Data field. The results for using and developing Linked Data are quite similar, with about 50% of participants rating their experience as high (4 and 5), and about 20 to 25 percent of participants rating their experience as low (1 and 2). Even though a large majority of participants rated their experience studying Linked Data as high (4 and 5), this category is the only one out of the three where a very small percentage of participants reported having no experience. All in all, the results obtained about the participants experience with Linked Data are in line with what we expected and hoped to obtain, as a large majority of users reported having a medium to high level of experience for the three categories, studying, using, and developing.

After surveying the participants' general experience with Linked Data, we wanted to gather more specific information on the participants' experience developing Linked Data applications and tools. Thus, we asked every participant to rate their experience in each area of the development of Linked Data applications, modelling, user interface, and backend. The results are presented in Figure 7.2, where we can consult the percentage of each level of experience for these three main areas. One of the reasons to collect information about the participants' experience developing with Linked Data, was to gather knowledge on the participants experience with user interfaces, and possibly with the design of user interaction. We can point out that modelling is the category with the highest percentage, of about 75%, of participants rating their experience as high (4 and 5), followed by user interface with about 50% of participants, and finally backend with 40% of participants. Even though user interface is not the category with the smallest percentage of participants with a high level of experience, it is by far the category with the largest percentage of participants with a low level of experience (1 and 2). The discrepancy of the results obtained for

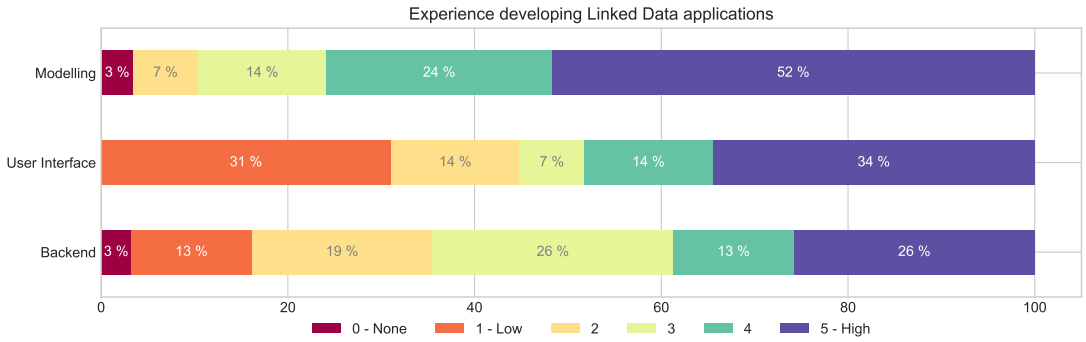


Figure 7.2: Results of the participants experience developing Linked Data applications.

the user interface category, compared to the other two categories, are in line with the small number of research and studies regarding user interfaces and user interaction with Linked Data, and show that this is still a neglected aspect of the development of Linked Data applications.

Lastly, we surveyed more in depth information about the participants experience using Linked Data applications. Figure 7.3 presents the results obtained, where we can consult the percentage of each level of experience for the four main tasks performed in applications, visualising, performing search tasks, browsing and navigating, and creating and editing data. The results obtained are not very different between the four tasks, but we can point out that browsing and navigating is the task with the highest percentage of users rating their experience as medium and high (3, 4, and 5) with about 75%. It is followed by creating and editing data with about 70%, and performing search tasks with close to 70% of participants. The task with the lowest percentage of participants is visualising with about 55%. These results inform us on which tasks are more important to the majority of users and will be helpful later on when gathering knowledge and creating the patterns.

7.2.2 Common Problems

One of the goals of this survey is to start to gather information and knowledge on some of the most common problems, users and developers encounter in Linked Data applications. To start this analysis we asked every participant to rate the frequency and importance of problems encountered when performing the following tasks in Linked Data applications: visualising, performing search tasks, browsing and navigating, and creating and editing data. The results are presented in Figure 7.4 where we can consult the percentage of each frequency and importance level for the four tasks. The results obtained for each one of the tasks are quite similar to each other, we can point out that browsing and navigating has the highest percentage of participants rating with a high level (4 and 5) of frequency and importance, with about 55%. Creating and editing follows with over 50% and visualising with close to 50%. Finally, performing search tasks is the task with the lowest percentage, with about 45%. All in all, the results are very similar to each other, much like the ones obtained for the participants' experience while using Linked Data applications. From these results we can conclude that there is no task that is distinctly more important than the others regarding the

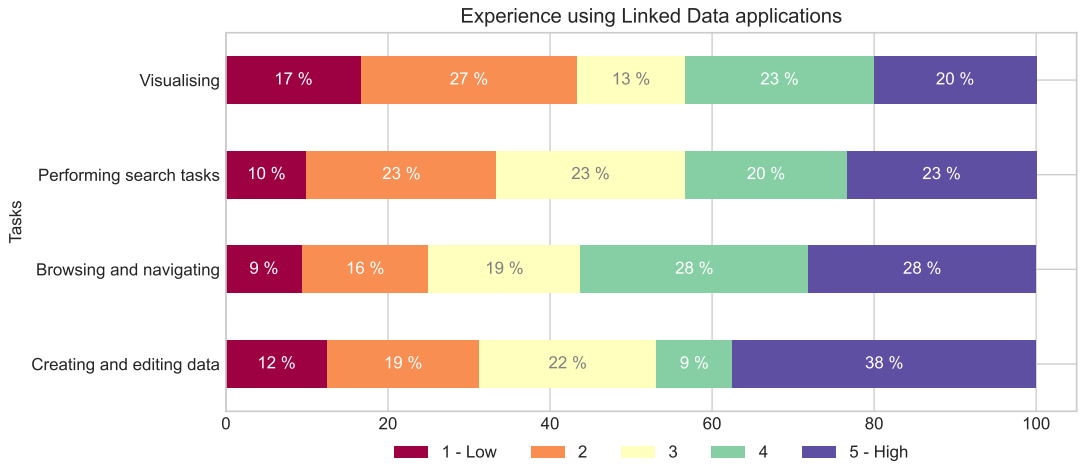


Figure 7.3: Results of the participants experience using Linked Data applications.

problems encountered. Thus, when performing the pattern mining sessions and creating the user interaction patterns, we should try to cover all the four tasks as much as possible.

Regarding the common problems encountered in Linked Data applications, we also provided the users with an open question where they could report problems encountered in Linked Data applications. The answers to this question were processed and analysed. We obtained a total of 28 answers with 25 considered valid. From this 25 answers, we extracted 47 problems. The distribution of the number of problems per answer or participant is very well balanced, and can be consulted in Figure 7.5. The maximum number of problems per participant is 4, and only with a total of 3 answers.

After the problems were extracted from the answers, we performed a manual analysis of these problems and categorised them according to the three parts of the development of Linked Data applications, modelling, user interface, and backend. As the focus of this research is user interaction, we performed an additional analysis on the user interface problems. The user interface problems were categorised with the previously mentioned four main tasks performed in Linked Data applications.

The results from this analysis are summarised in Table 7.1. We can point out that user interface is the category with the largest number of problems associated, a total of 24, followed by backend and then modelling. This reflects the fact that many Linked Data applications are still neglecting their user interface and user interaction problems, proving that this topic is still not sufficiently explored and analysed. These results also prove that applications, and consequently their users, will likely benefit from guidelines and studies performed to identify best practices when designing user interfaces and user interaction for Linked Data.

Regarding the tasks associated to the user interface problems, we can clearly see that visualising has the vast majority of problems associated, with a total of 17. Next, we can point out browsing and navigating with 7 problems, and performing search tasks and creating and editing data with a single problem each. The distribution of problems amongst the four tasks does not

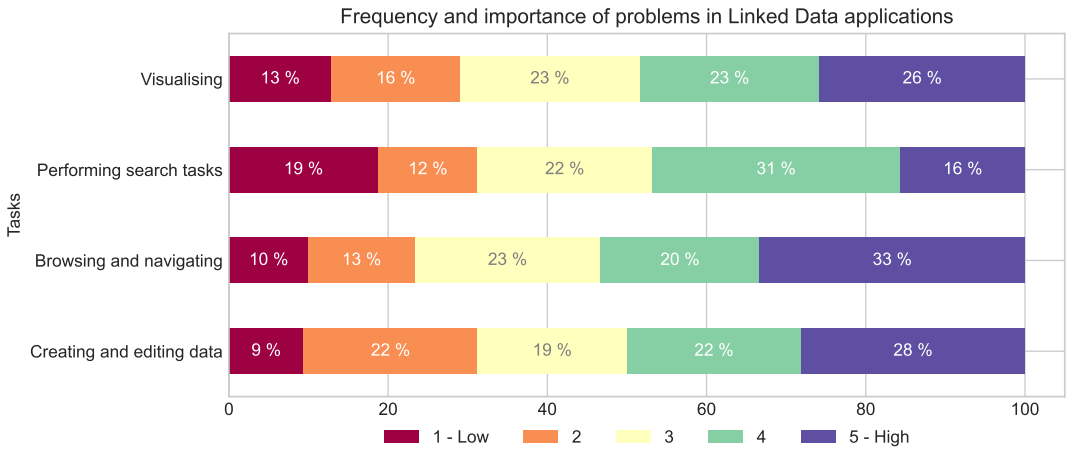


Figure 7.4: Results of the frequency and importance of problems in Linked Data applications.

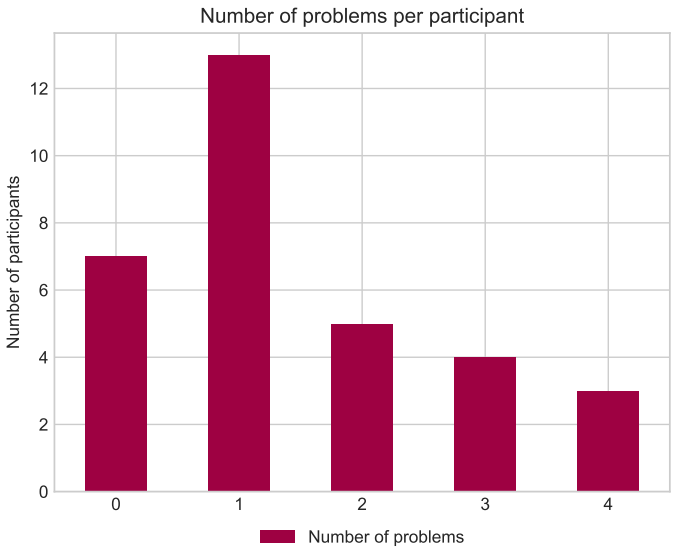


Figure 7.5: Number of problems reported per participant.

align with the results obtained for the frequency and importance of problems, where visualising had the lowest percentage of high ratings. This could mean that participants rated more according to the importance of problems and less with their frequency, thus resulting in a higher number of visualising problems, but a lower importance associated to them. Regardless, these mixed results in the frequency of problems justify a pattern collection that tries to cover the four tasks as much as possible.

Task	**Number of problems**
User Interface	24
Visualising	17
Browsing and navigating	7
Performing search tasks	1
Creating and editing data	1
Modelling	12
Backend	19

Table 7.1: Overview of the categorisation of the reported problems.

As shown previously most of the reported user interface problems were related to the visualisation of Linked Data, and most of them report the difficulty of choosing visualisation techniques for the data, especially when the datasets are large and dense; the lack of mechanisms that provide the user with a clear overview of the available data and their vocabularies. Regarding the problems reported for browsing and navigating the dataset, these are focused on the fact that browsing new datasets is often hard and cumbersome due to the time it takes to understand the new ontology or vocabulary, and the difficulty to understand the data distribution. Another problem commonly reported is the fact that there are often too many links or irrelevant nodes in the interfaces; and that exploring relationships is difficult, especially when they are complex, causing a poor understanding of the data.

Lastly, we related the user interface problems with the familiarity with user interaction patterns of their author, and compiled the total number for each task in Table 7.2. The total percentage of participants that answered the survey that are familiar with user interaction patterns is of about 53%, and 47% for participants that are not familiar. We chose not to include the results for the performing search tasks and creating and editing data tasks, due to the small number of problems associated with them that could not reflect a correct distribution regarding the familiarity with user interaction problems. For the browsing and navigating tasks we can observe a very similar distribution to the one with the full set of participants. Regarding the visualising task we can see that a larger number of problems were reported by participants that are familiar with user interaction problems. This could mean that they were more inclined to participate and share them, or are more sensible to these kinds of problems when using applications.

	User Interaction Patterns	
Task	**Familiar**	**Not familiar**
Visualising	12	5
Browsing and navigating	2	4

Table 7.2: Familiarity with user interaction patterns and number of problems reported.

7.2.3 User Interaction Patterns Usefulness

Another main goal of this online survey was to gather information on the opinion of members of the Linked Data and Semantic Web community about the usefulness of user interaction patterns for Linked Data. For this, we asked the participants to rate the need for guidelines on how to design user interaction with Linked Data, and to rate the usefulness of user interaction patterns for Linked Data. The results obtained for these two questions are presented in Figure 7.6. We can point out that, in both questions, a majority of the participants answered with a high rating (4 and 5), with the usefulness of patterns having the highest percentage of 85%, and the need for guidelines with 70%. By analysing Figure 7.6, we can see that very few participants gave these questions a low rating (1 and 2), and that no participants gave the lowest rating to neither questions. We can conclude from these results that the lack of guidelines for this topic is a valid and relevant problem, and that the development and creation of user interaction problems to tackle it is a useful and needed solution.

To perform a more in depth analysis on the results regarding the usefulness of user interaction patterns, we related the results obtained regarding it and the participants' experience with developing Linked Data applications. The results can be consulted in Figure 7.7, where we can observe that the ratings on the usefulness of patterns greatly differ between the levels of experience developing Linked Data applications. We can point out that participants with a rating of 1 (low) in developing experience were the ones to rate most frequently the usefulness of patterns as low and medium, with 50% of participants rating it with 1, 2, and 3. Participants with a high rating of 5 in developing experience follow the ones with 1, as the second group that most frequently rated the usefulness of patterns as low and medium, with about 30% of participants. Finally, 15% of the group of participants with a rating of 3 in developing experience rated the usefulness of patterns as low and medium. The groups of participants with a rating of 4 and 2 in developing experience were the only ones to only rate the usefulness of patterns as high (4 or 5).

Finally, we also related the results obtained regarding the usefulness of patterns with the participants' experience with developing user interfaces for Linked Data applications. The results are present in Figure 7.8, where we can again observe that the ratings on the usefulness of patterns greatly differ amongst the different levels of user interface development experience. In this case, the group with the highest percentage, with 50% of low and medium ratings (1,2, and 3) on the usefulness of patterns, is the group with an experience of 2. This group is followed by the group of participants with an experience of 4, with 25%, and finally the group with an experience of 5,

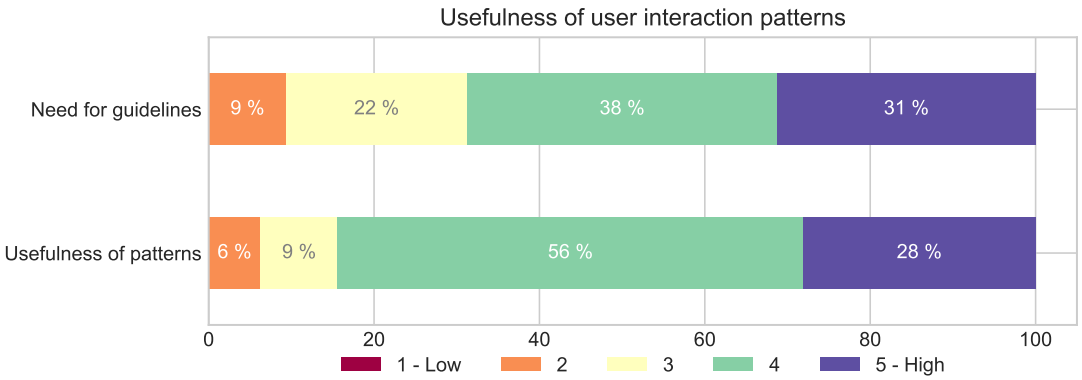


Figure 7.6: Results on the need for guidelines and usefulness of user interaction patterns for Linked Data.

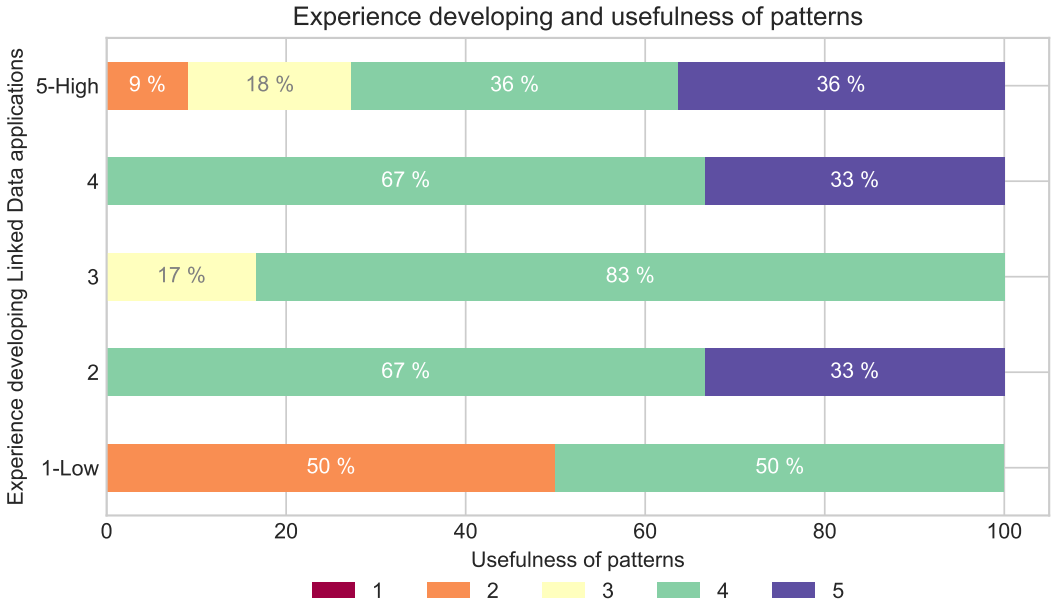


Figure 7.7: Usefulness of user interaction patterns and developing experience.

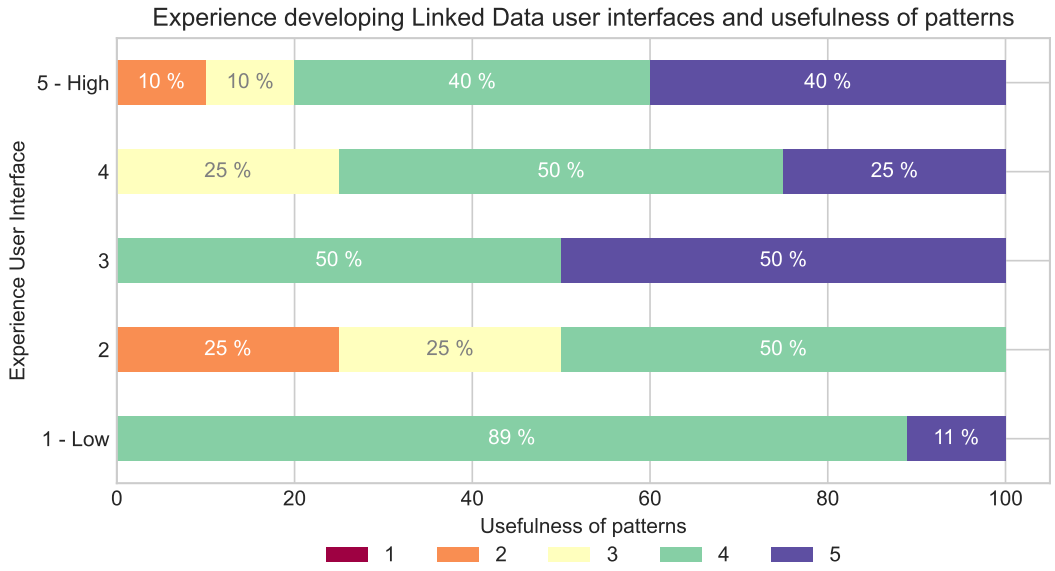


Figure 7.8: Usefulness of user interaction patterns and developing user interfaces experience.

with 20%. Again, we have two groups with only high ratings for the usefulness of patterns, the group with an experience of 3 and 1. From these results we can conclude that despite the irregularities between the different levels of experience, the participants rated the patterns as useful for all levels. This leads us to believe that user interaction patterns could be useful for all developers designing user interfaces for Linked Data applications, regardless of experience.

7.2.4 Final Comments

The final section of this online survey was dedicated to collect the contacts of participants willing to participate in a follow-up interview, part of the pattern mining sessions, and to give the space for the participants to leave comments and their opinions on the topic of this research. From the 32 participants we collected the contacts of a total of 18 people willing to participate in a follow-up interview.

Regarding the comments and opinions of the participants of the survey, some mentioned that it is a very interesting research topic and that there is a need for more studies at the intersection of Linked Data and Human-Computer Interaction to better inform other researchers and professionals about this pertinent subject of Linked Data. Other participants mentioned that developers and designers generally have little experience in this topic, which tends to translate in a lack of capacity to create new functionalities for users that take advantage of the capabilities of Linked Data. Two participants mentioned a possible reason for the large amount of user interaction issues related to Linked Data. One argued that most of the these issues come from the fact that Linked Data based applications are built by a niche group of researchers and developers, that are more eager to push Linked Data capabilities on their users rather than focus on the users' tasks and only adding the needed modifications on top of more traditional UIs. This participant even

suggested that we start to study the generic user interaction guidelines, and then analyse how they can be modified when taking in account Linked Data. The other participant stated that most of the Linked Data specific user interaction problems stem from the relatively complicated history of how the Semantic Web standards have been built. Only one participant rejected the core idea of this research, user interaction with Linked Data, arguing that it puts the storage mechanism and communication protocols at the heart of the interaction, and that is what has been holding back Linked Data. This participant defends that applications should be built to satisfy a user need and not for a researcher or developer to simply publish their data. This would result in the exploitation of Linked Data resources only because they have data that is required for an application or particular website. Essentially this participant argues that there are no new user interaction problems that come with the use of Linked Data, because it should not be transparent to the user interface. This is a very pertinent point that we sought to further explore in the one-on-one interviews with Linked Data researchers and developers, by asking the participants' opinion on this topic.

Chapter 8

Pattern Mining Interviews

Pattern mining is the phase in the development and creation of patterns where the knowledge and information to write them is collected and gathered. Wanting to mine this knowledge from second-hand contributions, we decided to conduct expert interviews as a method of pattern mining. After disseminating the Linked Data experience survey and obtaining a solid number of responses and some people interested in participating in the pattern mining interviews, we contacted them and scheduled interviews. The goal of these interviews was to listen to researchers and developers experience with Linked Data, using, developing, or studying, and extract the common struggles and problems encountered regarding user interaction. When possible, solutions and approaches to those problems were also collected, as well as any general tips and recommendations on how to model user interaction with Linked Data. We conducted a total of 10 pattern mining interviews with both Linked Data researchers and developers, in the time span of about one month and a half. This chapter describes these pattern mining interviews, including the design of the interview guide, a thematic analysis of the results and their presentation, and a final reflection on the knowledge gathered and importance of these sessions for the development of the patterns.

8.1 Design and Execution

After researching and studying how to conduct expert interviews, and how to decide which type of interview to perform, open, semi-structured, or structured, we decided on semi-structured interviews for the pattern mining sessions. Semi-structured interviews have a predefined structure and the interviewer has a general guide [57]. However, they are open to follow and investigate any other issues or topics raised by the participants [63], and give the interviewers the flexibility to ask the questions in any order, omit questions, or ask questions that are not in the guide. This style of interview better suits to perform pattern mining, as it conducts the interviewees to focus on their experience and problems and solutions, while at the same time provides the space for them

to highlight what they consider more relevant and important. The interview guide followed and the rationale for the questions included are presented and described in Section 8.1.1.

We carried out a total of 10 pattern mining interviews, with Linked Data researchers and developers from around the world, 5 from Europe, 3 from North America, and 2 from South America. The interview pool was made up of 8 men, 1 woman, and 1 participant identifying themselves as non-binary. The interviews had an average time of around 60 minutes and were conducted online, via Google Meets. The participants were encouraged to share any examples and applications using the screen sharing feature, as a big part of the interview consisted of their previous projects and works in the Linked Data field. All the interviews were audio and video recorded, upon previous consent from the interviewees, and later transcribed. After the interviews, a thematic analysis was conducted on the interviews' transcriptions and notes taken.

8.1.1 Interview Guide

To write the guide for the semi-structured pattern mining interviews we followed the steps recommended in "Writing an Effective Guide for a UX Interview" [63]: write the research questions, brainstorm interview questions, broaden your questions, fill in for unaccounted research questions, arrange the questions, prepare additional probing and followup questions, and finally pilot your guide.

The research questions outlined for the pattern mining interviews are focused on extracting user interaction problems and their possible solutions. The interview research questions (IRQ) identified are the following:

IRQ1: What are the recurrent user interaction problems in Linked Data applications?

IRQ2: Do these problems fit in the following categories: visualising, browsing, searching, authoring?

IRQ3: What techniques or solutions are used to tackle those problems?

IRQ4: What aspects lead to the use of one solution instead of another?

IRQ5: What are some of the consequences of using the solution? (positive and negative)

IRQ6: What are some of the consequences of not using the solution? (positive and negative)

From these interview research questions, an interview guide was designed to collect useful and clear information for the creation of the user interaction patterns. The interview guide is composed of 3 parts and designed for an expected duration of 45 to 60 minutes. The first section covers the participants' introduction, consent for recordings, interview and research context, and a brief overview of the data collected on the need and usefulness of patterns and guidelines for the topic at hand. The second part is composed of 6 main questions focused on the interviewee's experience with Linked Data, as a user and as a developer, and on the problems encountered. The third and final part of the interview is composed of 4 main questions covering the solutions for the

problems mentioned previously; context, examples, and consequences of the solutions; and final tips and comments on the topic. In addition, a total of 12 follow-up questions were defined for some of the main questions. A full version of the interview guide can be consulted in Appendix B.

As is usual with semi-structured interviews, not all questions were asked in most of the interviews conducted. However, some questions were consistently asked across the sessions and played an important part in the extraction of the knowledge and information for the patterns. The main questions asked in all interview session are the following:

- Can you talk a little about your opinion on the impact, future, preset, or past, of Linked Data on the web?
- Can you tell me about how you started to work on or be interested in Linked Data?
- What are some of the projects or research studies that you have been involved with that you would like to highlight?
- Do you use Linked Data based applications in your day to day life, for personal or work related tasks?
- Can you point out some common problems you have faced when developing Linked Data applications? If possible in specific user interaction problems?
- What do you usually do to tackle them? What are your most recurrent approaches?

8.2 Thematic Data Analysis

After conducting the pattern mining phase in the form of expert interviews with Linked Data researchers and developers, we set out to analyse the qualitative data gathered in the interviews' transcriptions and notes. These pattern mining interviews are a tool to perform exploratory research on the topic of user interaction with Linked Data, and thus produce a large amount of qualitative attitudinal data that corresponds to people's thoughts, experience, and believes that are reported to the interviewer [62]. Aiming to extract clear and valuable information from the interviews, in order to feed possible pattern ideas and the patterns themselves, we decided to conduct a thematic analysis on the interviews' transcriptions and notes. Thematic analysis is a method of systematically breaking down and organising rich data produced by qualitative research methods, and consists of tagging and annotating observations and quotations with different codes, in order to discover significant themes amongst the several interviews [32]. The three most common methods of performing thematic analysis are the use of specialised software, journaling, or the use of affinity diagramming techniques. We decided to use the journaling method that consists of the manual annotation and highlighting of the data, followed by a record of the researchers' ideas and thought processes on the interviews' data.

Following the interview research questions defined and mentioned previously, we categorised the data according to any mention of possible problems when developing Linked Data applications, and solutions used to tackle them. We tagged the data as a problem, a solution, or a pair of problem and related solution. Additionally we categorised these problems and solutions as regarding user interface, modelling, and backend. Even though the questions asked during the pattern mining interviews were targeted at user interaction and user interface problems and solutions, most of the interviewees still mentioned some problems regarding the other aspects of Linked Data application development, the backend and modelling phases. Due to the fact that the main focus of these interviews is the collection of knowledge regarding experience with user interface and user interaction for Linked Data, we further classified the problems and solutions tagged as regarding user interface using the 4 main tasks performed in applications as categories: visualising, searching, browsing, and authoring. The results of this thematic analysis are described in more detail in the following sections, with Section 8.2.1 focusing on the reported problems and solutions, and Section 8.2.2 highlighting other topics mentioned in the interviews that were considered worth mentioning.

8.2.1 Problems and Solutions

The thematic analysis performed on the 10 pattern mining interviews identified a total of 38 problems, 22 solutions, and 35 pairs of problems and solutions. As expected, the category with more problems and solutions associated is user interface with a total of 70, followed by backend with 15, and modelling with 7. Similar to the distribution of the problems reported in the Linked Data experience survey in Chapter 7, from the user interface problems, the task with more problems and solutions associated is visualising, and with the least is authoring. An overview of these results can be consulted with more detail in Table 8.1.

Area	Problems	Solutions	Problem and Solution Pairs	Total
User Interface	22	22	26	70
Visualising	12	13	15	40
Searching	0	5	5	10
Browsing	6	2	4	12
Authoring	1	4	3	8
Modelling	6	0	1	7
Backend	8	0	7	15
All	38	22	35	95

Table 8.1: Overview of the thematic analysis results of the pattern mining interviews.

As mentioned previously, the goal of these interviews, and consequently of the thematic analysis, is to gather and collect knowledge of user interface and user interaction problems and solutions

for Linked Data. Thus, we filtered out the reported problems and solutions for modelling and developing backend, and performed a more in depth analysis on the ones tagged as regarding user interface. We collected the problems and solutions that were more frequently mentioned and the ones that were considered more relevant and useful for the creation of patterns, and aggregated and summarised them in Table 8.2. We can observe that the set of problems and solutions is composed of 3 categorised as authoring, 7 as visualising, 2 as browsing, and 1 as searching. This set is also comprised of 8 problem and solution pairs, 4 solutions, and 1 problem.

The information collected regarding the authoring of data in Linked Data applications is focused on aiding, specially inexperienced users, in the creation and edition of data, by providing familiar features, such as input areas for each part of the triple or resource with clear labels that lead the users to enter the correct information in the correct place, as can be consulted in Problem 1. Another solution frequently mentioned, detailed in Problem 3, is the inclusion of valid recommendations for each input area, to help the user choose and enter valid and correct data. Regarding authoring solutions, such as the one in Problem 2, some interviewees also mentioned that it is important to provide the user with the option to download the data that is presented to them, specially when one of the goals of the data collection is to be used by others in the Linked Data community.

Regarding the problems and solutions targeted at the visualisation of Linked Data, they are mainly focused on trying to help the user to understand the data better and more quickly. These include solutions like in Problem 4, defining a set of the most relevant and important information about a resource and the outline of how to present it to the user, in order to reduce the large number of triples displayed in a single page; and in Problem 6, aggregating a subset of triples to present the user with complete information that may be needed for a user task or need. Other solutions like Problem 5, also include the use of primitive formats of visualisation for each data type, such as maps for geographical data, and calendars for temporal data; in Problem 8, the ability by the user to choose from one of these provided visualisations to better suit their needs; and the inclusion of raw data in the visualisation options to allow the users to analyse it further if needed, in Problem 9. To help the users gain a better and more complete understanding of the data collections, some solutions were highlighted, such as providing an overview of the structure of the data, and the definitions of all internal and external resources and properties, detailed in Problem 10 and 7.

In regards to browsing the dataset, some interviewees mentioned that providing the users with links to the internal and external related resources is an easy to use solution, as detailed in Problem 12. However, other interviewees mentioned a problem that arises from it, due to the fact that everything is linked, it is easy for the users to loose track of where they are and where they started, while browsing the dataset via links, making them feel lost in the data collection, as can be consulted in Problem 11.

Finally, only one solution was highlighted regarding the ability to perform search tasks in the data collection, as is presented in Problem 13. Some interviewees mentioned the importance of providing a SPARQL query editor, in addition to more simple solutions like keyword-based approaches. This would allow the experienced users to create more complex queries when the

more traditional approaches are insufficient. It was also mentioned that it should be possible to visualise the corresponding SPARQL query of the search task previously performed, to allow the users to edit it, or save it to be used in another occasion.

Number	**Problem**	**Solution**	**Area**
1	Allow inexperienced users to create and edit Linked Data.	Provide features to create and edit data in the similar and familiar ways present in other web applications, and transform it in the background in RDF triples. Present clear boxes to users, so that the correct information is entered in the correct place.	Authoring
2		Provide a feature to download the data present in search results, single resources, graphs, or tables.	Authoring
3	Aid users when choosing what to input, while creating or editing data.	Provide recommendations of valid options for each text area available. Filter the recommendations to the most relevant according to the resource and domain, or by the previously used by the user.	Authoring
4	Visualising all the triples associated with a resource can be too much information at once for the users.	Define for a resource the set of the most important properties and related resources to be displayed to the user. Divide them in different sections and order according to importance. Provide the remaining properties and related resources in closed sections for the user to consult if needed.	Visualising
5	Mapping the data to visual information that can be easily understandable by the users.	Use primitive intuitive formats of visualisation (geographic data into maps, time data into calendars or timelines) and combine those different types of visualisations in one component.	Visualising
6	Information is presented to the user scattered over several pages or sections, and thus can be hard to fully comprehend.	Aggregate sub-graphs of the data that contain the complete information for a use case or user task and create a visualisation component for that complex data aggregation. Know when information should be presented together and when information should be in another page navigable to by a link.	Visualising
7	Finding a way to make the user understand the underlying structure of the dataset.	If not too dense, show a graph visualisation with the different classes in the dataset and the relations between them. Not easy if the application is not constrained to a specific domain or dataset, because an overview could mean an overview of the web of data.	Visualising
8		Allow the user to choose from different visualisations of the displayed data (table, list, graph, map, calendar, picture, video).	Visualising
9		Always have raw data available in addition to other kinds of visualisations. Have the option for users to be able to dig deeper into whatever direction they want.	Visualising
10	Users have difficulty interpreting the labels of the resources or properties being displayed.	The definition of something, a resource or a property, should be available, for example when you hover the name.	Visualising
11	Due to the fact that everything is linked, it's difficult to provide effective navigation that is intuitive and keeps track of where the users come from. Otherwise, they can feel lost while browsing the data.		Browsing
12	Allow the users to browse the data.	Provide links to the internal and external resources and properties.	Browsing
13		Provide users with a SPARQL query editor in addition to the simple keyword-based search approach. Allow the user to visualise the corresponding SPARQL query of the search task performed.	Searching

Table 8.2: Summary of the thematic analysis results of the pattern mining interviews.

8.2.2 Other topics

In addition to the problems and solutions identified previously, by performing a thematic analysis we identified other topics regarding the presentation of Linked Data, and the validity of this research study that we found worth mentioning. Several interviewees identified the open world assumption as one of their biggest struggles while developing Linked Data applications, as it isn't a user interface problem it was not mentioned in the previous section. According to these interviewees, the open world assumption is very difficult to work around when an application uses mainly local data, and when it doesn't the developers don't know what to expect regarding the structure and the volume of the data. This problem is aggravated when the application needs to present the

unknown data to the user in a meaningful way, which is difficult to deal as a developer or as a designer.

One topic discussed during one of the interviews concerns moral issues related to the use and creation of Linked Data. This interviewee exposed the problem of the lack of diversity of voices in the Linked Data field, as its research tends to be concentrated on the Northern Hemisphere, specially Europe and North America. While at the same time there is a belief that Linked Data collections and datasets are a mirror of reality that people create and put online to be explored by everyone. However, as this interviewee reported, all the knowledge comprised in Linked Data collections is only partly true, due to the fact that it is contributed by individual people, with individual bias and individual geographical contexts. Even though, this is a problem hard to solve and tackle, it is important to create awareness of developers and data scientists about this issue, and the need to make clear when displaying data to the user that it represent a situated knowledge.

Following the comment presented in Chapter 7.1.1 rejecting the core idea of this research studies, we posed this question to some of the researchers interviewed, by asking if they agree that the use of Linked Data poses new user interaction problems. Some of the interviewees reported that it depends, stating that in regards to user interfaces many components could be reused from traditional interfaces when using Linked Data and that in most cases it won't stray from the use of tables and lists, because that is what people are used to and familiar with. It was also mentioned that it depends on what the application wants to show to the user, it could be an application in some domain that chooses to present the data in a more traditional way, without it ever being transparent to the user that the application uses Linked Data. However, if the application wants to provide tools to explore large knowledge graphs, or Linked Data collections, new issues are going to arise from that. Another interviewee argued that new user interaction problems only arise if the application wants to show to the user that it is using Linked Data, and that it makes the most of its capabilities. The other half of interviewees reported that they believe Linked Data poses new user interaction problems, for example in providing ways to browse the dataset due to the fact that everything is linked, which makes it differ from more traditional approaches. It was also argued that Linked Data poses new problems, because developers and designers have to deal with the fact that they don't know what to expect when accessing the data, and that there isn't always immediate access to the description of everything on the web of data. However, it was also reported that even though it poses new problems, we can use the same heuristics to solve them, by taking the traditional approaches in as inspiration.

8.3 Conclusions

We can identify the conducted pattern mining interviews as the most important phase in the collection and gathering of knowledge to create the user interaction patterns. The results obtained from these interviews are rich and full of problems and solutions reported by developers and researchers in the field, and will prove very important when deciding which problems to cover and which solutions to propose in the patterns. We can conclude that the chosen interview format, as well as

the developed guide helped in extracting the right information. One aspect of the interviews, that proved to be more useful than initially thought was the sharing of concrete examples. This allowed us to view and explore new examples that we hadn't come in contact with yet, as well as, listen to the rationale behind some of the features and user interaction decisions.

From the 10 people interviewed in this pattern mining phase, we found some more useful than others, usually due to the number of examples discussed during the interview, and the type of examples. We found very useful to interview Linked Data developers that work in more commercial companies, in addition to academic researchers. The former group tends to work more with applications targeted at end-users, and not applications or tools solely developed to post and share datasets or data collections. In future and additional pattern mining sessions for this pattern collection, we would consciously make an effort to include more Linked Data developers that work outside of the academic world.

Even though the pattern mining interviews resulted in a collection of information rich in solutions, topics, and themes, it still portrays the opinions and experience of a very small group of people, that may not be representative of the Linked Data community. As mentioned previously, an effort was made to include in these sessions people from different locations and with different backgrounds, which was constrained by the time limitations and number of people available. When handling the information gathered, there needs to exist an understanding that it may be biased and not represent the general opinion of the Linked Data community. As such, thorough evaluations need to be conducted on the results that come from or are based on the knowledge and information gathered in these interviews.

Chapter 9

User Interaction Patterns for Linked Data

Pattern languages and pattern collections are used to document and summarise best practices in a field of expertise, by abstracting common problems and solutions found in the literature and in the industry for that topic. Research in the Linked Data field regarding user interaction and user interface design is lacking studies that gather these best practices and guidelines. This may be due to the fact that Linked Data is still perceived as data only to be created and consumed by computers, and not by humans as well. However, with the recent growth of humans consuming Linked Data, the problem of designing good user interaction for the users, regardless of experience or technical background gains even more relevance. As a way of validating the problem of the need for guidelines and best practices for user interaction with Linked Data, and the use of a pattern collection to tackle it, we conducted an online survey in the Linked Data community, described previously in Chapter 7. This survey presented positive results in regards to this problem, and the need for creation of patterns to solve it and help developers and designers build Linked Data user interfaces, regardless of experience level.

This chapter introduces a pattern collection for user interaction with Linked Data, where we propose twenty novel patterns divided in four different categories: visualising, browsing, searching, and authoring. The patterns proposed in this chapter are a result of the previously presented study of the state of the art of Linked Data applications and tools, the common problems reported in the online survey, and the pattern mining sessions conducted with Linked Data researchers and developers. A catalog of the patterns problems and solutions can be consulted in Appendix C, and a list of the patterns' patlets [14], a term known as a short description of a pattern, can be consulted in Appendix D.

This chapter is divided into 6 sections: in Section 9.1 we present the pattern structure chosen for the user interaction patterns and detail each component; in Section 9.2, we present an overview of the structure of the pattern collection and describe each one of the categories; and in

Section 9.3, 9.4, 9.5, and 9.6 we present the 20 user interaction patterns for Linked Data, divided in the four categories, visualising, browsing, searching, and authoring.

9.1 Pattern Structure

The proposed patterns use a structure based on the classical pattern structure proposed in "A Pattern Language for Pattern Writing" [54] and "How to write a pattern? A rough guide for first-time pattern authors" [82], which include the context, problem, forces, and solution. We made some alterations to this proposed pattern structure to better fit the needs of the user interaction patterns. The pattern structure we opted for is composed of the following sections:

Context. Describes the situation in which the problem occurs, and details the circumstances that constraint the solution. Sometimes the context is described in terms of the patterns that have already been applied.

Problem. Presents the specific problem that needs to be solved.

Solution. Presents the proposed solution to the problem, describes the implementation details that need to be considered, and explains some of the rationale behind it.

Diagram. Shows a simple illustration that helps the reader to understand the pattern more quickly.

Examples. Presents concrete examples, aligned with the context, that illustrate how to apply the pattern.

9.2 Pattern Collection Overview

The pattern collection proposed in this chapter is composed of 20 new user interaction patterns for Linked Data, that are organised into 4 different categories: visualising, searching, browsing, and authoring. The visualising category comprises 10 patterns and is focused on problems and solutions regarding the visualisation and display of Linked Data. The searching category is composed of 3 patterns and covers problems when performing search tasks in Linked Data datasets and data collections. The browsing category is also composed of 3 patterns and targets problems that arise from browsing and navigating Linked Data. The final category, authoring, comprises 4 patterns and covers problems and solutions for creating and editing Linked Data. Figure 9.1 presents an overview of the structure of the pattern collection, focused on the relationships between the patterns.

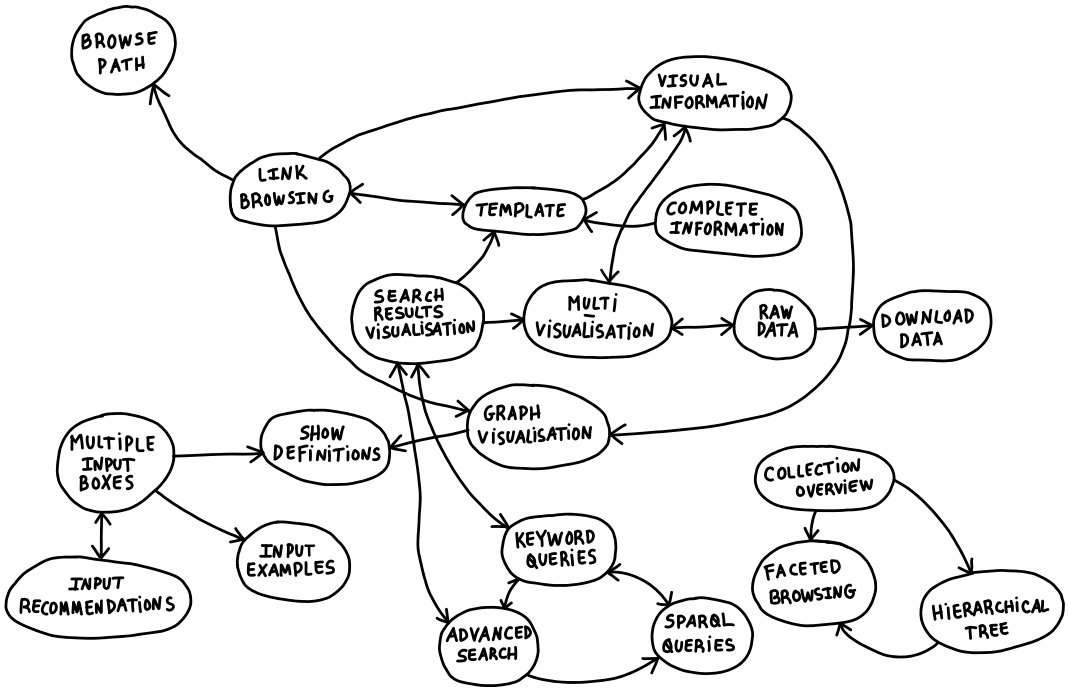


Figure 9.1: Pattern collection structure overview.

9.3 Visualisation Patterns

9.3.1 Template

Context

Resources and concepts represented in Linked Data usually have a huge number of triples related to them, either as primitive data or as other related resources. One of the easiest approaches to present resources and concepts is to visualise the triples in lists or tables, resulting in indifferentiable information at first glance and visualisations that can be frightening to certain users.

Problem

When a resource's page presents a long list of visually similar triples containing information or related resources, users can feel overwhelmed at first glance and struggle to find the wanted information.

Solution

Define a set of the most important properties and related resources for each one of the resources in the data present in the application.

Analyse the data and identify the different resources or classes. If the data has a large number of resources or classes start with the most important ones or the ones more frequently visited by the users. For each one of the resources, define a template containing a set of the most important properties and related resources according to its domain and concept that it represents. This template must also include an outline dictating how the information should be presented to the user. The properties and related resources must be ordered according to importance and divided in sections according to the different kinds of information presented. The properties and related resources not present in the defined template, must still be available to the user at the end of the page, in closed or collapsible sections.

When deciding on how to present the data and the different types of information in the template, opt for simple lists for textual data or single numeric values. On how to visualise and present other types of data, like temporal or geographical, refer to VISUAL INFORMATION (p. 90). The related resources, either internal or external, should be presented as clickable links to aid the navigation of the users through the data, via LINK BROWSING (p. 102).

Curating the information to present to the user and how to present it is crucial in helping the user complete their tasks and fulfill their needs as quick as possible. However, still providing the least relevant or important information in a less prominent manner is very important, as it allows the users to explore the presented data further if they choose to, making no assumptions on the tasks or needs of the users.

A diagram illustrating the solution described can be consulted in Figure 9.2. Concrete examples on how to apply this pattern are present in Figure 9.3.

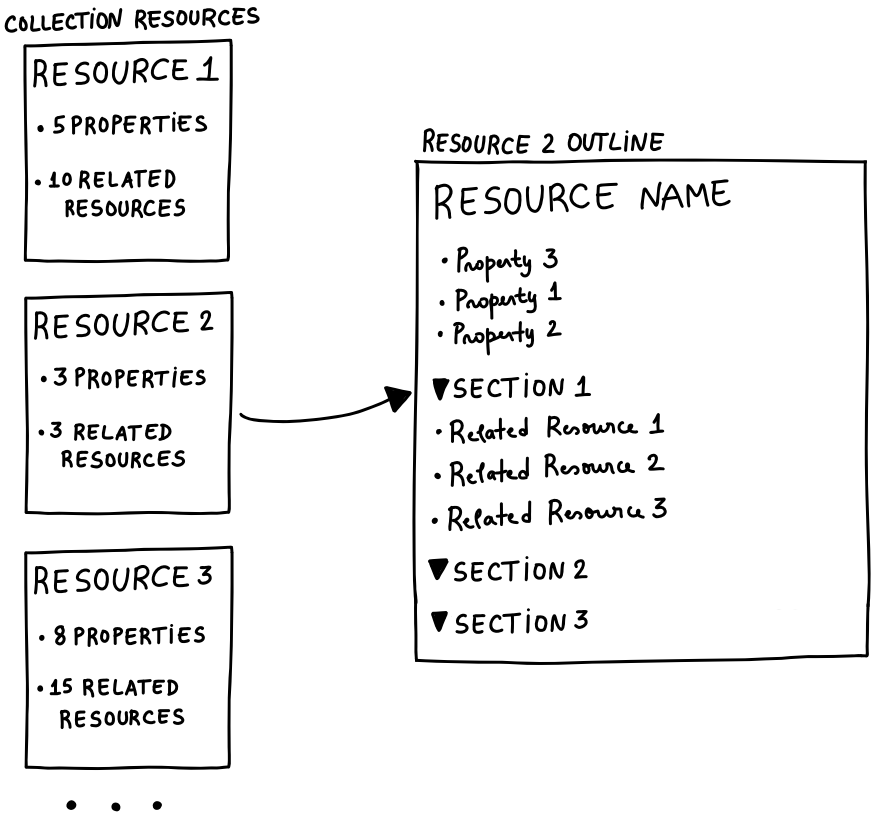
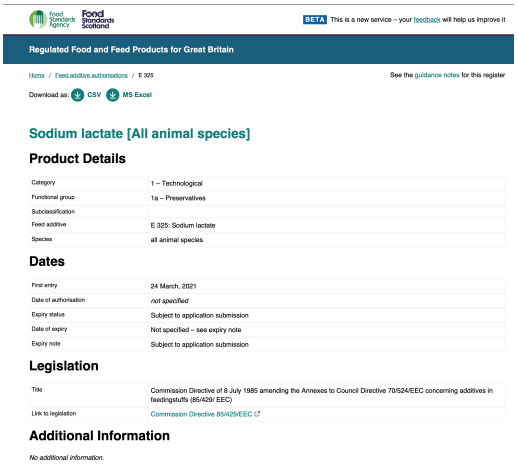
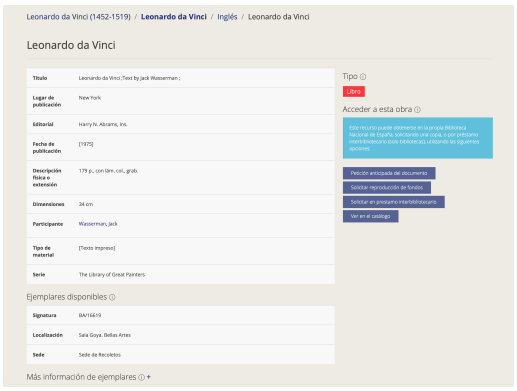


Figure 9.2: TEMPLATE diagram.



(a) Datos Biblioteca Nacional de España [28].



(b) Register of regulated food and feed products for Great Britain [26].

Figure 9.3: TEMPLATE examples.

9.3.2 Multi-Visualisation

Context

Concepts represented using Linked Data usually contain or have associated different types of data, such as temporal data, statistical data, or spatial data. Each of these types of data is best presented to the users with different visualisation techniques.

Problem

Finding and choosing the best visualisation technique for the users, that best suits the types of data and aids the user in their needs and tasks.

Solution

Provide the user with different types of visualisations for the data that is displayed.

According to the types of the data presented choose from tables, lists, maps, graphs, calendars, plots, or timelines. Refer to VISUAL INFORMATION (p. 90) for how to find the best visualisation technique for each type of data. Despite the types of the data or the visualisation technique that best suits the resource and its domain, make sure to always provide the option to visualise the information in a textual format, using tables or lists. This ensures the accessibility of the application and consequently the data, as it allows visually impaired users to consult it using text-to-speech software.

If one of the goals of the data is not only to be presented and consulted, but also to be reused in other domains and datasets of other users, provide RAW DATA (p. 86).

Providing different options to the user when visualising data allows them to choose which visualisation best suits their needs, tasks, or even personal background. It also allows the user to choose between inspecting the data further and with more detail, or just consulting an overview.

A diagram illustrating the solution described can be consulted in Figure 9.4. Concrete examples on how to apply this pattern are present in Figure 9.5.

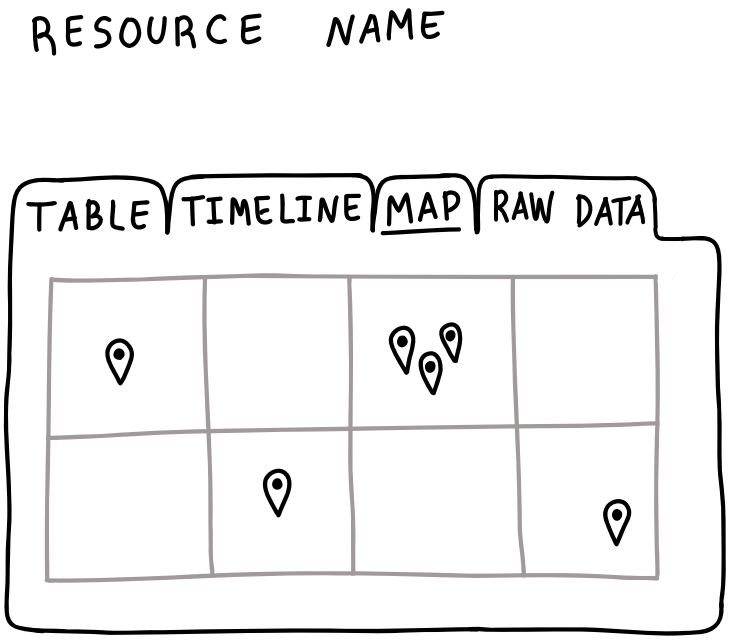
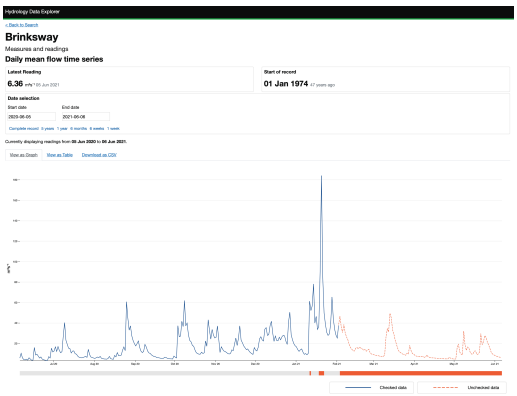
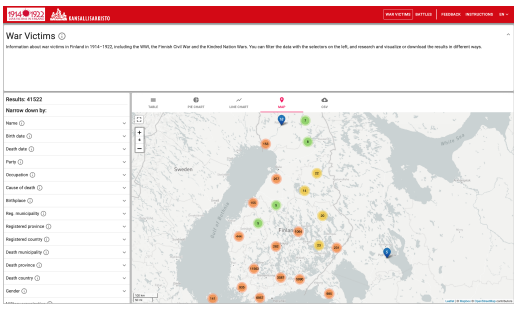


Figure 9.4: MULTI-VISUALISATION diagram.



(a) Hydrology Data Explorer [22].



(b) War Victims Portal [61].

Figure 9.5: MULTI-VISUALISATION examples.

9.3.3 Show Definitions

Context

Linked Data is still portrayed as data that will only be created and consumed by machines. Due to this, some resources and properties' names are not designed to be understandable by users and do not refer to the concept that they are representing. This hinders users in understanding the data that is presented to them. Other times, when the names are understandable and clear to humans, they refer to domain specific terms that may not be familiar to all, which also poses a hurdle for users understanding the data.

Problem

Users cannot interpret some of the names of the resources and properties of the data, and thus find difficult to understand the presented information.

Solution

Provide definitions of all resources and related properties present in the data.

When displaying a resource, set of resources, or search results list, the names of the related properties or resources themselves are not clear to all users, specially those not experts in the domain. Sometimes even the label of the resource or property is not clear and understandable.

Provide a detailed definition for all the resource and property names of the internal vocabularies, and as much as possible for used external vocabularies, by looking up URIs.

This will help the users to better understand the data, without having to do any additional research or investigation on the vocabularies used.

A diagram illustrating the solution described can be consulted in Figure 9.6. Concrete examples on how to apply this pattern are present in Figure 9.7.

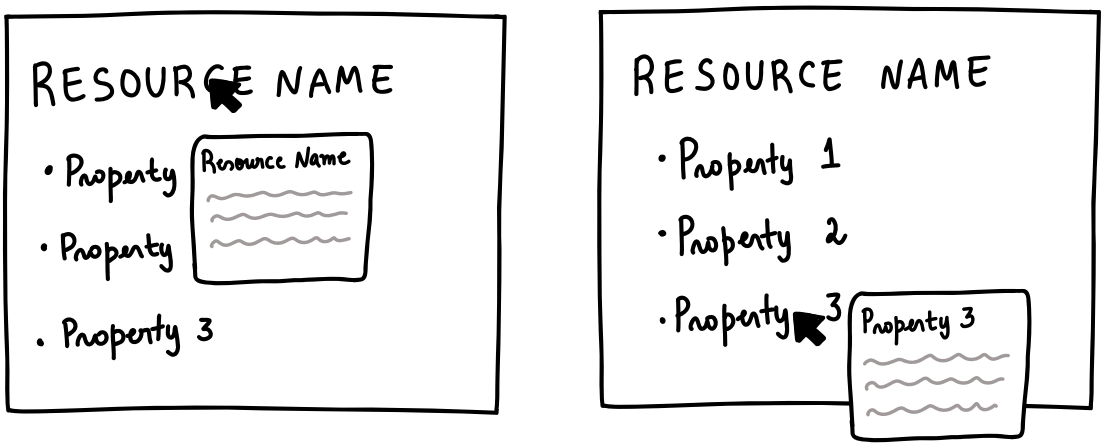
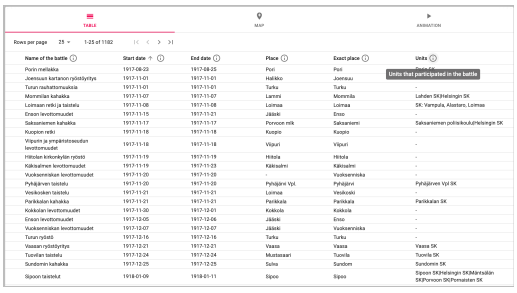
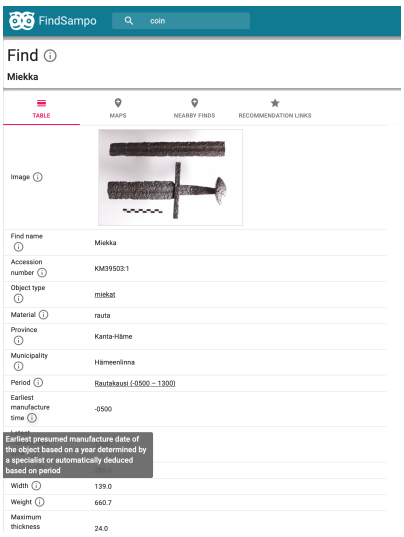


Figure 9.6: SHOW DEFINITIONS diagram.



(a) War Victim Portal [61].



(b) FindSampo [36].

Figure 9.7: SHOW DEFINITIONS examples.

9.3.4 Graph Visualisation

Context

Linked Data is still associated with graph visualisations due to its natural graph structure. However, most times graph visualisations are very dense and can be confusing for inexperienced and even experienced users. This is accentuated when the data collection is very large and interconnected. Nonetheless, in some cases graph visualisations can help the users understand and interpret the data.

Problem

Present data using graph visualisations that are usable and understandable by non-technical users and that allow them to take full advantage of the data represented.

Solution

Use graph visualisations for subsets of the data where the edges of the graph all have the same or similar meanings.

Applying graph visualisations to subsets of the data, such as family trees and friend networks, where all the edges have the same or very similar meanings, helps the users have a better overview of the data. Graph visualisations work best for these cases, because they display the information in a way that is similar to how the concepts relate in the real world.

How the graph is shown must also be taken into consideration to better its understanding. The edges must have a clear label for the name of the connection and overlapping must be minimised. The label of the resources and properties should be used instead of their names, and when possible SHOW DEFINITIONS (p. 80) must be provided.

In regards to the size of the graph, a full visualisation of all nodes and edges must be readable and understandable without any initial zooming in or out. If this is not the case, the subset might be too large to be visualised in a graph.

When there are different classes amongst the resources in the subset, different colours can be used to differentiate them. This helps the users perceive an immediate distinction between the different classes in the graph.

The different nodes in the graph, representing resources or concepts, must be presented with clickable links to the corresponding pages. This will allow the users to transverse the data and browse it with LINK BROWSING (p. 102).

A diagram illustrating the solution described can be consulted in Figure 9.8. Concrete examples on how to apply this pattern are present in Figure 9.9.

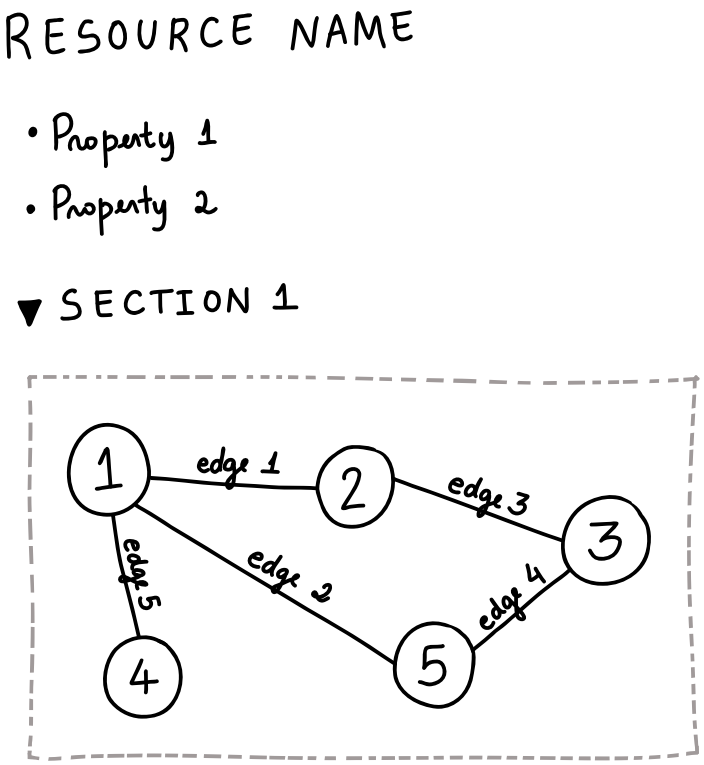


Figure 9.8: GRAPH VISUALISATION diagram.

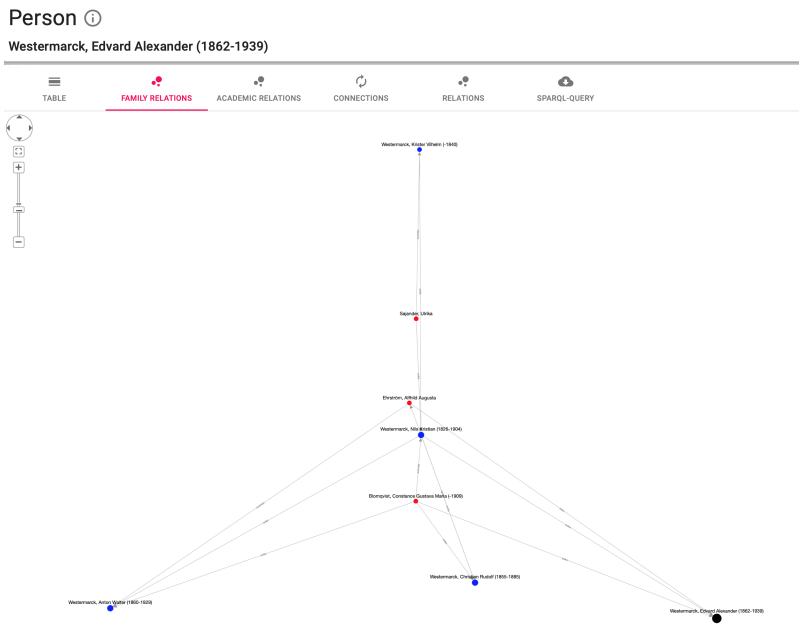


Figure 9.9: GRAPH VISUALISATION example - AkatemiaSampo [52].

9.3.5 Search Results Visualisation

Context

An application provides its users with features to create and perform search tasks, and implements a simple search approach like KEYWORD QUERIES (p. 96) in addition to more advanced options like ADVANCED SEARCH (p. 100).

Concepts represented in Linked Data often have a large amount of related information, and usually only a small set of that information is relevant in search tasks. These can depend on the search task itself, on the application's domain, and on the type of search result, or concept represented.

Problem

Provide users with an easy to understand list of search results, that aids their tasks and needs.

Solution

Define the types of search results for the data. Identify the most important information of each type. Present them in differentiable sections.

Analyse the data and select what terms, resources, or properties constitute its search result items. These can be based on the resources selected to define a TEMPLATE (p. 76) for, they will depend on the application domain and on its goals. When presenting the set of search results make clear which are part of each category, resources or properties, or different classes between the resources.

Present the search results to the user in a list, ordering them based on how well they match the search keywords and advanced search options. Allow the user to change the sorting based on different fields, such as by the associated dates, by ascending or descending order of numeric fields, or by how close or far they are to a given location.

If the list of search results is composed of items from different categories, for example that are instances of different classes, present them in easy differentiable sections. This will help the users to quickly find the information they are looking for.

For the visualisation of each search result, define a small set of the most important related information. It can be based on the TEMPLATE (p. 76) previously defined. When the search task includes advanced search options, make sure to include these in the most important information of each search result. Present each search result alongside its important information, directly in the search results list.

If the search results are composed of different types of data, like temporal or geographical, allow the user to choose from MULTI-VISUALISATION (p. 78).

A diagram illustrating the solution described can be consulted in Figure 9.10. Concrete examples on how to apply this pattern are present in Figure 9.11.

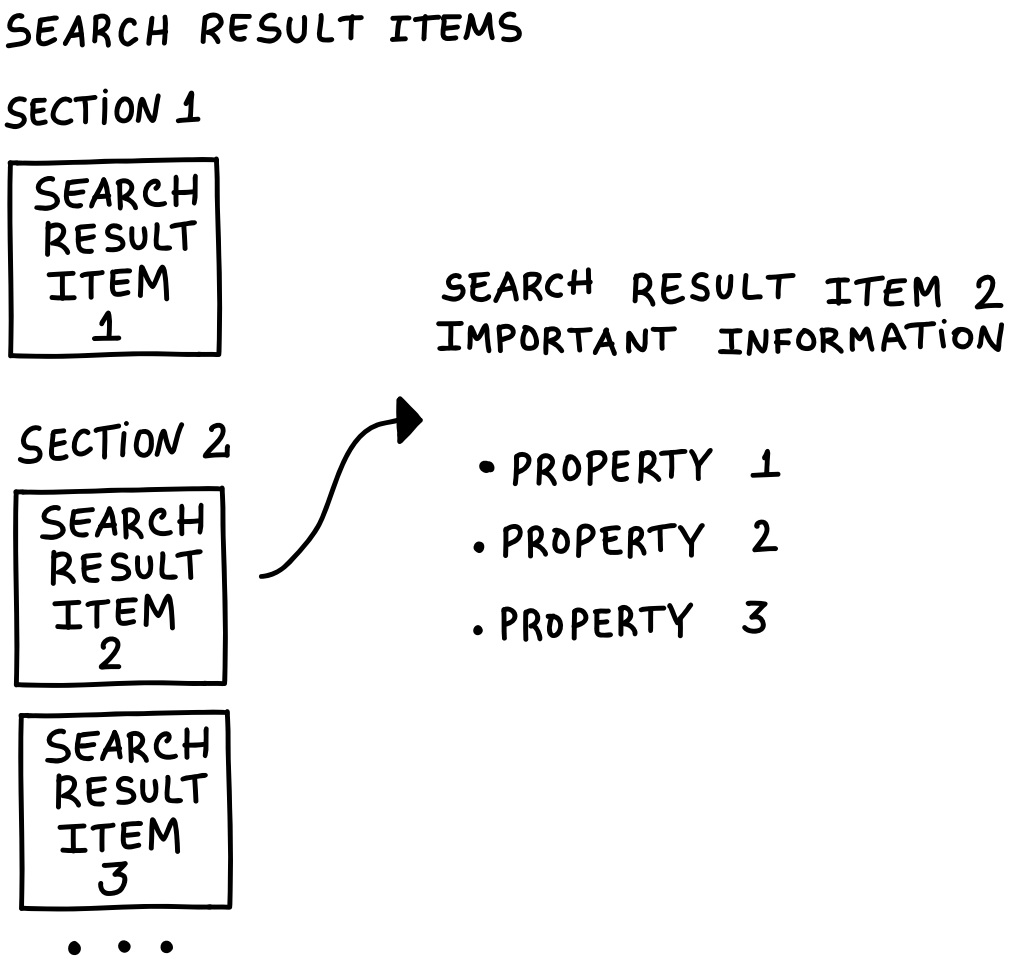
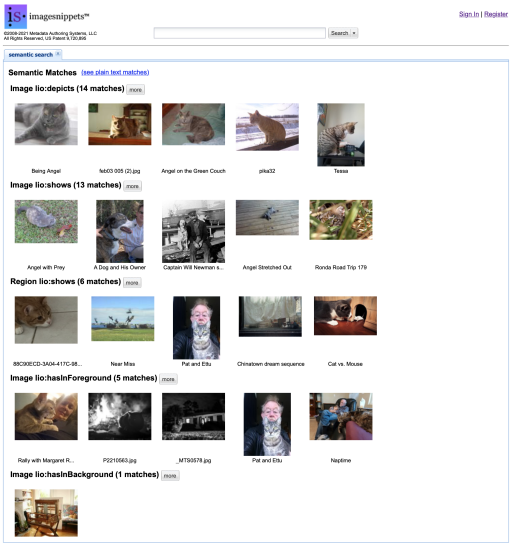
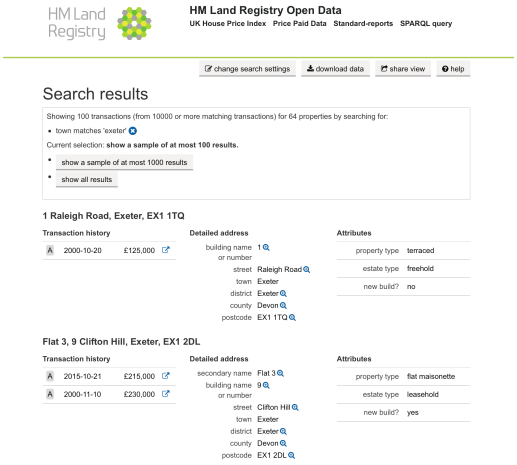


Figure 9.10: SEARCH RESULTS VISUALISATION diagram.



(a) ImageSnippets [81].



(b) HM Land Registry Open Data [27].

Figure 9.11: SEARCH RESULTS VISUALISATION examples.

9.3.6 Raw Data

Context

An application provides its users with MULTI-VISUALISATION (p. 78) to choose from when visualising a set of resources. The application is also targeted at users with some knowledge of Linked Data technologies.

Most of the times, data in RDF is not easy to understand by humans, even when they have experience working with and writing RDF. However, more traditional visualisations may not be detailed enough for some user tasks.

Problem

Allow the more experienced users to investigate the data further and deeper.

Solution

Provide the presented data in raw format.

In addition to other types of visualisations, have an option that allows the users to access the data in raw RDF format. This option provides the more experienced users with the possibility to investigate the data on a more deeper level than it is possible with the other visualisations. At the same time it does not hinder the experience of the inexperienced users, as other more accessible and traditional visualisation options are provided as well.

By providing a raw RDF visualisation of the data, always in addition to more accessible and understandable options, no assumption is made on the user experience and capabilities of understanding RDF. This will allow the users to consult this visualisation if preferred or needed for the user task at hand, enabling them to explore the data in whichever direction.

If the application's data is to be reused by other researchers and developers in their own projects, allow the users to DOWNLOAD DATA (p. 114).

A diagram illustrating the solution described can be consulted in Figure 9.12. Concrete examples on how to apply this pattern are present in Figure 9.13.

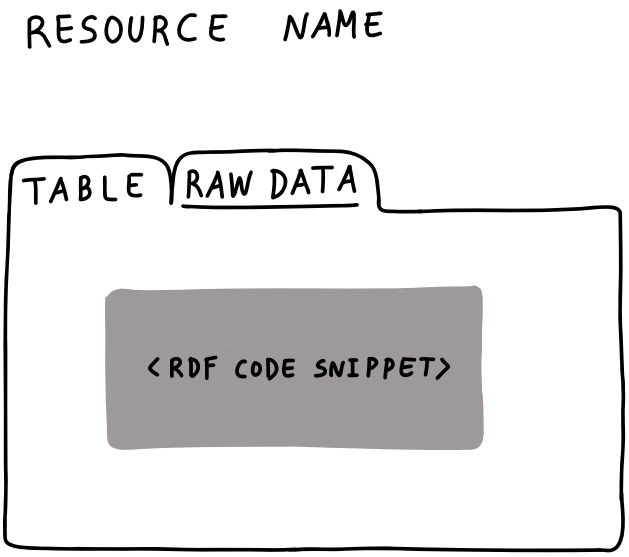


Figure 9.12: RAW DATA diagram.

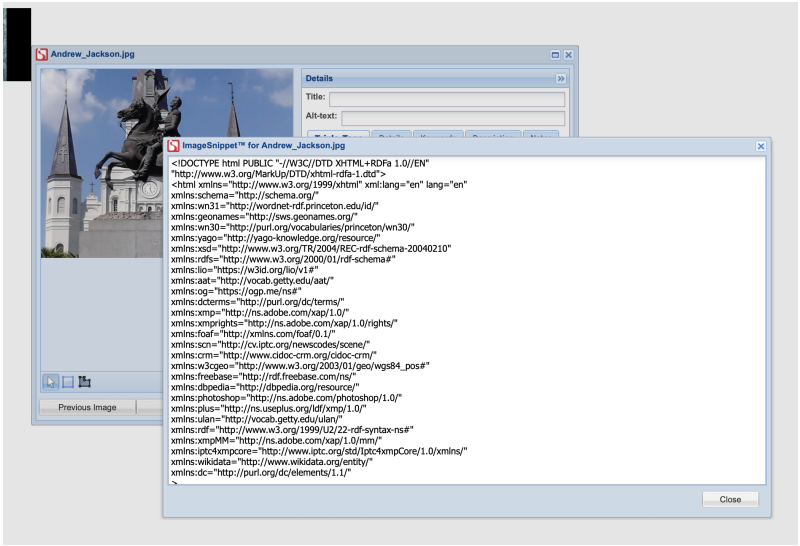


Figure 9.13: RAW DATA example - ImageSnippets [81].

9.3.7 Complete Information

Context

Linked Data is composed by a network of triples that are connected to each other and that together form information and knowledge capable of being understood by the users.

However, when applications do not aggregate the triples into information before presenting them, the users can feel lost while navigating the data. When triples are presented individually in a page, the users need to memorise the information that was previously presented and how they are connected, so that they can make the association that will allow them to come to the information, and then the needed knowledge.

Problem

When triples are presented individually, the users have difficulty in connecting the different information to be able to achieve the pieces of knowledge desired.

Solution

Aggregate small subsets of the data that form complete information and present them to the user.

According to the application's domain and goals, a study must be conducted on the most important and frequent user tasks. Then, an analysis on the information needed to complete each one of them must be performed. Lastly, the information pieces that are composed by the aggregation of more than one triple need to be identified.

Display to the users the triples or sets of triples when they form a complete piece of information, instead of single pieces of data that the users would have to mentally connect to be able to understand.

If the application is already using TEMPLATE (p. 76) to design and map the layout of the resources' pages, usually they are composed of one or more sections with other related resources that are connected to that resource's page through one property. However, based on the application's goals and domain it can be more beneficial for the users to visualise the information resulting from the aggregation of a relation from that resource to another through intermediary resources. If this is the case, that information must be presented to the user together in a way that allows them to consume it instantly. If not, these other related resources and information should be presented in another page, to which the user can navigate to from the original resource's page.

A diagram illustrating the solution described can be consulted in Figure 9.14. Concrete examples on how to apply this pattern are present in Figure 9.15.

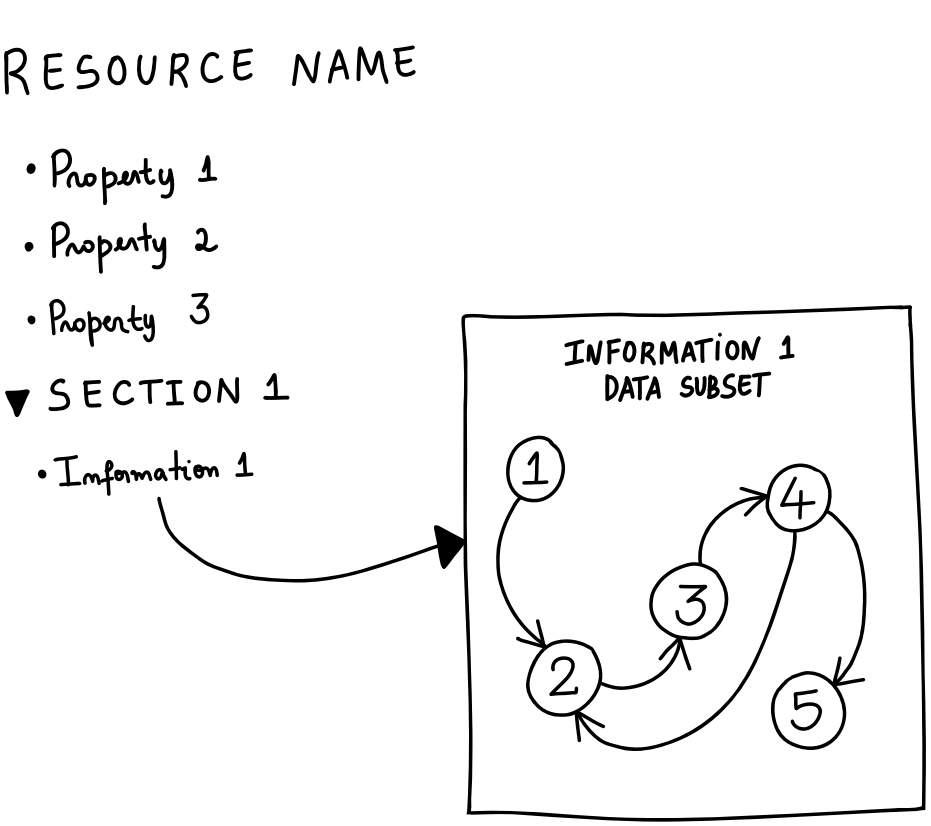
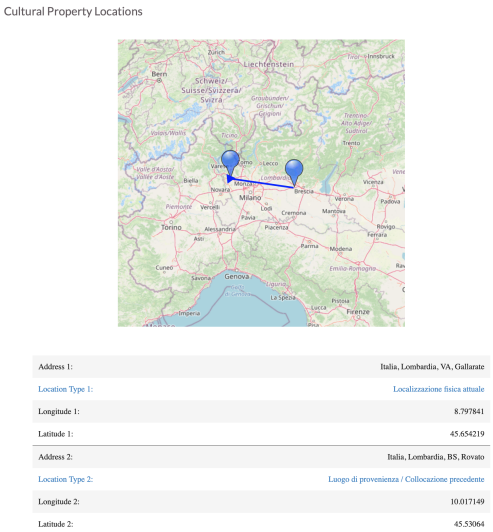
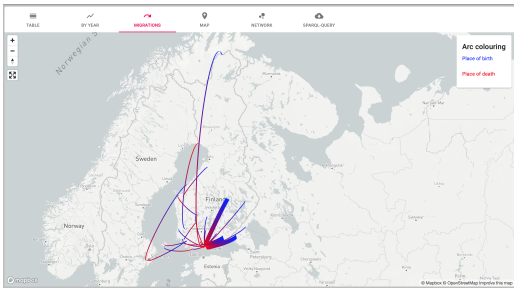


Figure 9.14: COMPLETE INFORMATION diagram.



(a) ODP Reactor [72].



(b) AkatemiaSampo [52].

Figure 9.15: COMPLETE INFORMATION examples.

9.3.8 Visual Information

Context

Concepts represented in Linked Data can have associated different types of data, such as textual and numeric data, or even temporal, geographical, and statistical data. Each one of these types of data has a visualisation technique that best suits them, and consequently aids the user to understand it and complete their tasks or fulfil their needs.

Problem

Display data to the user in a format that is easy to understand.

Solution

Map the data according to their type to visual information, using familiar visualisation techniques.

Display the data to the user using simple and familiar visualisation techniques that visually present the data according to their type and distinct characteristics. For numeric and textual data use lists and tables, for geographical data use maps, for temporal data use calendars and timelines, for statistical data use plots and tables. If the data represents concepts that are naturally related to each other in a network format, like family trees or friend networks, use GRAPH VISUALISATION (p. 82).

If the data presented contains some sort of media, like an audio, video, picture, or file, allow the user to consult it directly on the application. Provide a player for the audio or video, and embed the file or picture. This helps the user to more quickly understand what the presented data is, without having to take any additional steps to consult it, like downloading the file or opening a new window.

When the subset of the data the application is presenting to the user contains more than one data type, instead of choosing one visualisation, provide the user with MULTI-VISUALISATION (p. 78). By allowing the user to choose the best visualisation for them, no assumptions are made on what would be more useful for the vast majority of users.

A diagram illustrating the solution described can be consulted in Figure 9.16. Concrete examples on how to apply this pattern are present in Figure 9.17.

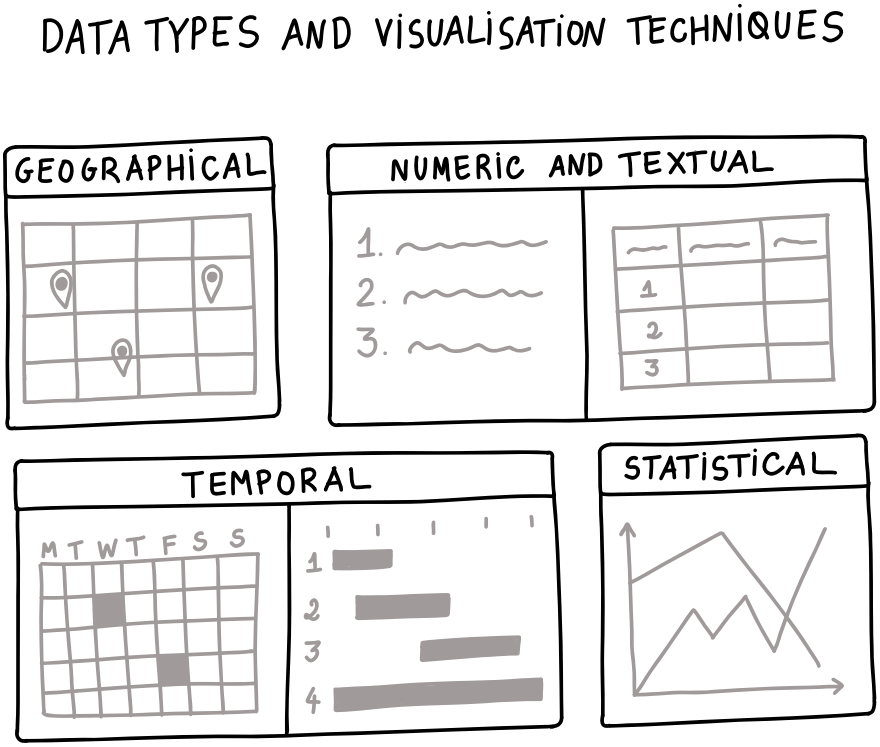
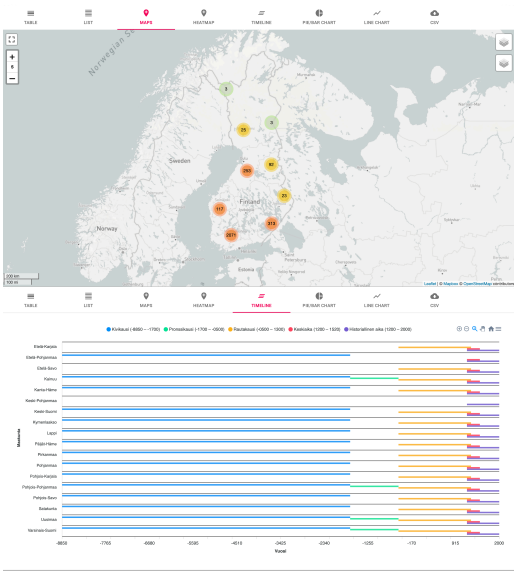
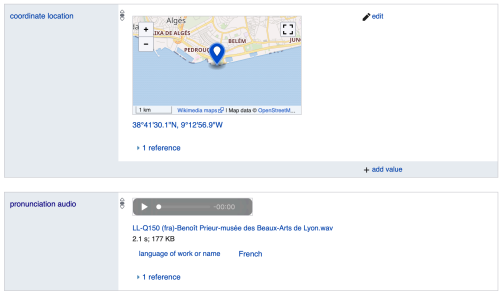


Figure 9.16: VISUAL INFORMATION diagram.



(a) FindSampo [36].



(b) Wikidata [80].

Figure 9.17: VISUAL INFORMATION examples.

9.3.9 Collection Overview

Context

Concepts, domains, and parts of domains represented in Linked Data are frequently modelled using a large number of classes, subclasses, and properties. Understanding and having a clear idea of how the concepts relate to each other, through their connections and hierarchy, can aid users in the comprehension of the data. This understanding will benefit users when navigating the application, in the construction of search tasks, and when authoring data. If the application's data is not restricted to a domain, and in part represents a subset of the web of data, this understanding becomes more difficult to provide.

Problem

When the users of an application are not experts, or don't have any previous knowledge on its domain, they can find difficult to understand the connections between the concepts or even the data that is presented to them. This can lead to more time spent on their tasks, or inability to complete them.

Solution

Provide a high-level overview of the data collection to the users.

This overview of the application data must show how the concepts of the domain relate to each other. The relations featured can include simple connections or hierarchy.

The visualisation of the overview must be simple and clear to be quickly understandable by a user with no knowledge of the domain. It can be a simple diagram or even a graph representation. If the application's domain and goals justify, the overview can just be composed of the 5 to 9 most important concepts. If so, the overview can be presented, for example in the navigation bar of the application, composed of clickable options with the names of the classes. If the subclasses list is not too extensive, they can also be included in the form of a dropdown list in the navigation bar options.

Regardless of the technique chosen to visualise the data collection overview, by clicking in any of the classes included the user should be redirected to a page where they can find a full list of the resources that are instances of the said class. From this page, the user should be allowed to initiate a FACETED BROWSING (p. 106) task, by filtering the instances based on their classes and connections.

When the application's data is not restricted to a certain domain and represents a large part of the web of data, it becomes impossible to provide a simple and clear visualisation for all users. An alternative to help the users situate a certain resource in the collection structure is by providing an HIERARCHICAL TREE (p. 94) of the resource.

A diagram illustrating the solution described can be consulted in Figure 9.18. Concrete examples on how to apply this pattern are present in Figure 9.19.

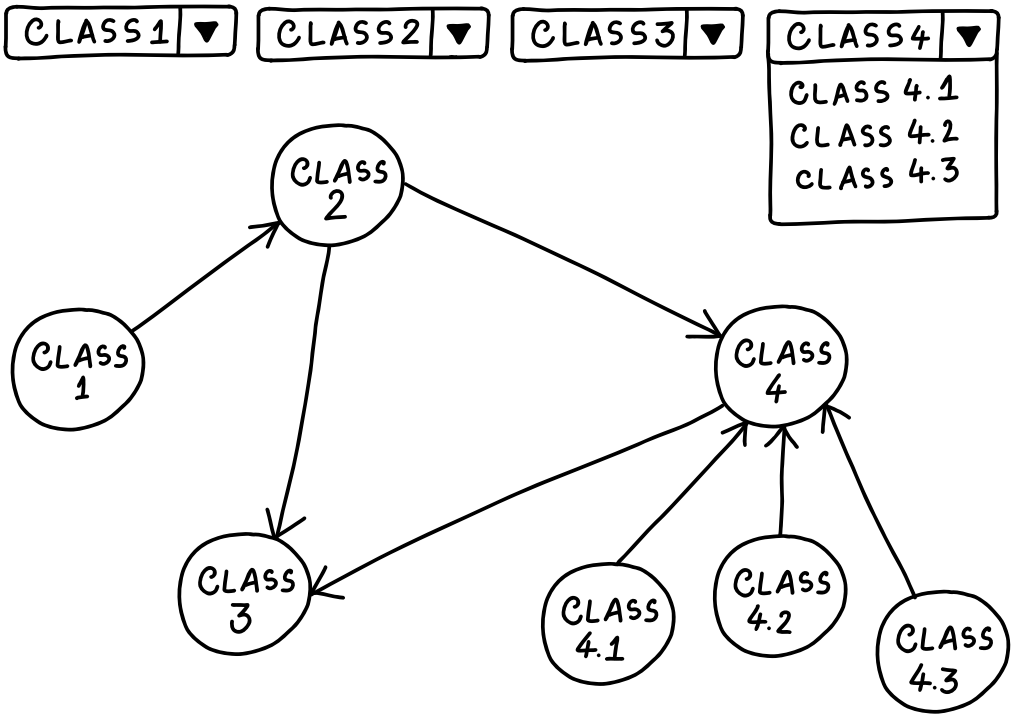
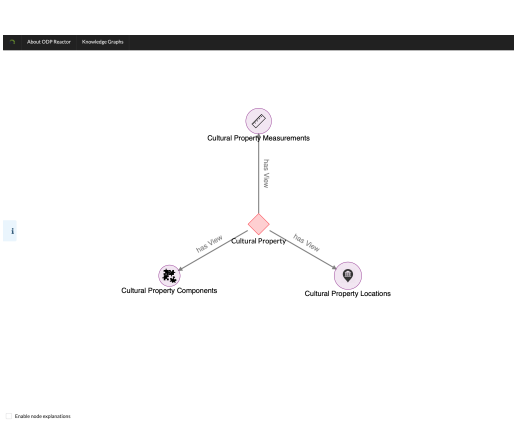
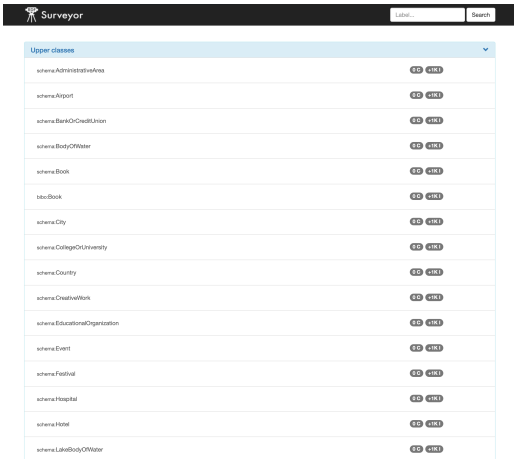


Figure 9.18: COLLECTION OVERVIEW diagram.



(a) ODP Reactor [72].



(b) RDF Surveyor [79].

Figure 9.19: COLLECTION OVERVIEW examples.

9.3.10 Hierarchical Tree

Context

Collections of data represented in Linked Data are usually composed of a large number of classes and subclasses. The natural structure of Linked Data creates complex hierarchical trees, that are hard to memorise and understand by users with none or little knowledge of the application's domain.

Problem

When the number of classes in a data collection is large, and their hierarchical organisation is complex, users with little knowledge of the domain find difficult to quickly and effectively understand the position of a resource in the collection.

Solution

Provide the hierarchical tree of a resource on its individual page.

In the page of each resource, provide a section containing the hierarchical tree of the resource. This will help the users to quickly situate the displayed resource in the data collection, and consequently aid the understanding of the data.

The visual representation of the hierarchical tree must be simple and clear, such as an indented list of all classes and subclasses. It must be provided in a section that is not immediately presented to the user, but can be opened and closed as needed, due to being additional information that will not be needed for all user tasks and needs. However, it is important to present the full hierarchical tree for the resource, even when the resource has multiple parents. It is also important to visually highlight the resource in the tree. This will ensure the hierarchical tree is clear about the class of the resource.

The classes present in the hierarchical tree must be clickable and should redirect the user to a page that provides a full list of the resources that are instances of that class. From this page, the users can initiate a FACETED BROWSING (p. 106) task, by filtering the instances based on their classes and connections.

A diagram illustrating the solution described can be consulted in Figure 9.20. Concrete examples on how to apply this pattern are present in Figure 9.21.

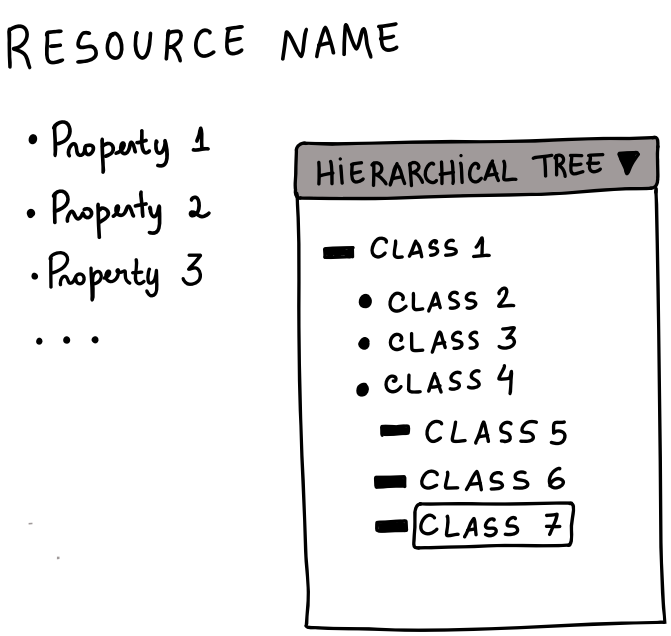
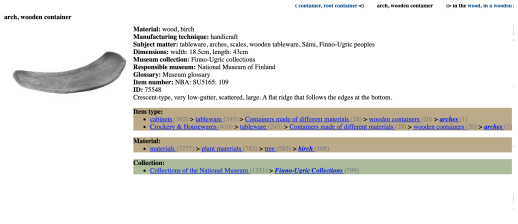
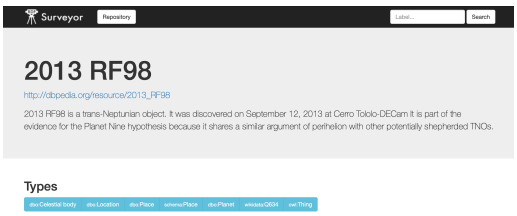


Figure 9.20: HIERARCHICAL TREE diagram.



(a) MuseumFinland [44].



(b) RDF Surveyor [79].

Figure 9.21: HIERARCHICAL TREE examples.

9.4 Searching Patterns

9.4.1 Keyword Queries

Context

The natural hierarchical and structured format of Linked Data opens the search and browsing solutions to novel and alternative possibilities. However, the ability to comfortably and quickly perform search tasks on applications is one of the most important features, as it allows the users to independently find data and complete their tasks.

Problem

Provide all users with a familiar and easy to use approach for searching the data.

Solution

Allow the users to perform search tasks using a keyword based approach.

Even though data represented in Linked Data allows for alternative and novel approaches for browsing and searching data, like faceted search systems, traditional and familiar approaches for the vast majority of users should be provided.

Keyword based queries are the most common search approach for the web user, as they are the main method used in all popular search engines. Thus, in addition to other novel and alternative query builders, search task methods, or browsing techniques provided in a Linked Data application, a keyword search feature must be always provided. This will ensure that a vast majority of users will immediately know and be comfortable performing search tasks and consequently, finding the intended data and completing their user tasks and needs. On how to present the search results list refer to SEARCH RESULTS VISUALISATION (p. 84).

The inputted keywords should be matched against the label and description of the resources, in addition to their name, as these fields often provide a more understandable explanation of what the resource represents.

If the application allows the users to perform and create more complex search tasks, such as the usage of filters on a diverse number of fields and sorting options, provide them with an ADVANCED SEARCH (p. 100). If the application is also targeted at users with previous experience of RDF and SPARQL provide the ability to create SPARQL QUERIES (p. 98). This will allow the users to create more complex queries if they need to, or to perform any previously written and saved queries.

A diagram illustrating the solution described can be consulted in Figure 9.22. Concrete examples on how to apply this pattern are present in Figure 9.23.

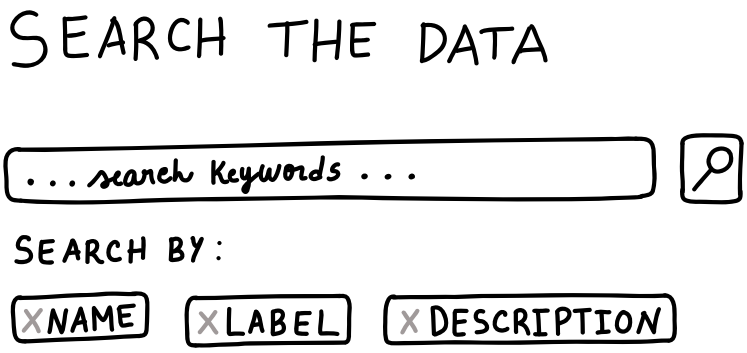
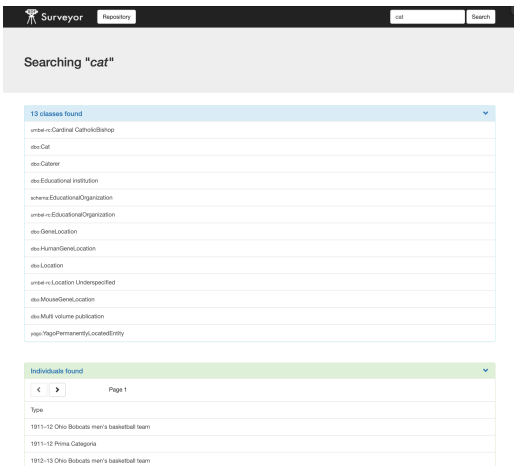
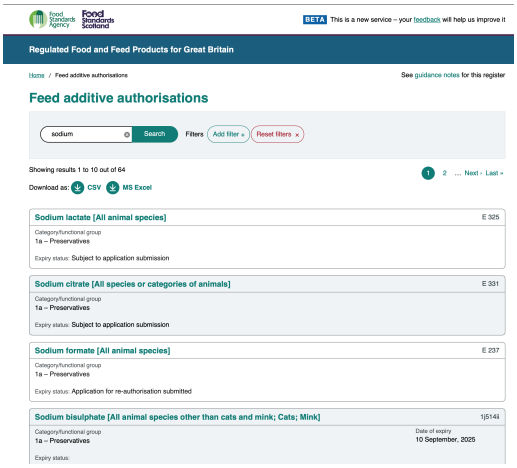


Figure 9.22: KEYWORD QUERIES diagram.



(a) RDF Surveyor [79].



(b) Register of regulated food and feed products for Great Britain [26].

Figure 9.23: KEYWORD QUERIES examples.

9.4.2 SPARQL Queries

Context

An application provides its users with search features and implements KEYWORD QUERIES (p. 96) or a similar simple search approach. The application targets users with RDF and SPARQL experience and wants to allow them to construct more complex and customisable queries.

Problem

When experienced users, with previous knowledge of RDF and SPARQL, want to perform complex search tasks, sometimes they may find the traditional approaches insufficient.

Solution

Provide users with a SPARQL query editor.

Provide the user with an option to create and edit queries in SPARQL, in addition to a simple keyword based approach and supplementary advanced search options. This will give the experienced users the possibility to create more complex and advanced queries that are impossible to accomplish with more traditional approaches.

The users should be able to copy and paste the SPARQL query code, in order to save and reuse it later, or use queries that were previously saved. Additionally, the SPARQL query should be addressable via a link, to allow the users to easily share the query and its results.

The application should provide a feature that allows the users to view and edit the corresponding SPARQL query of the search tasks performed using the keyword approach and the advanced search options. This allows the more experienced users to continue their search tasks when the traditional options reveal insufficient and SPARQL is more capable to perform the more complex queries.

A diagram illustrating the solution described can be consulted in Figure 9.24. Concrete examples on how to apply this pattern are present in Figure 9.25.

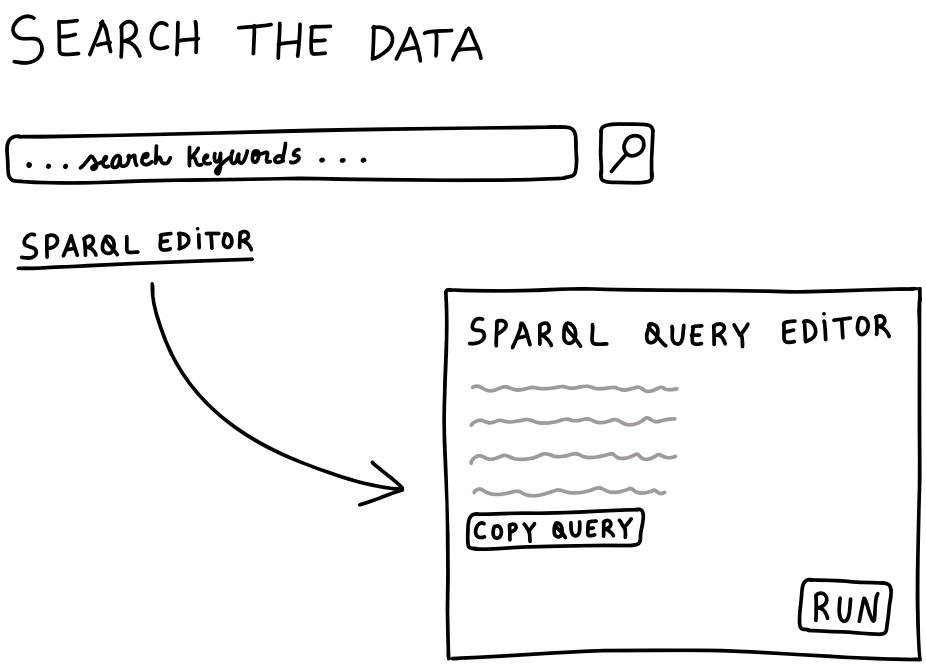
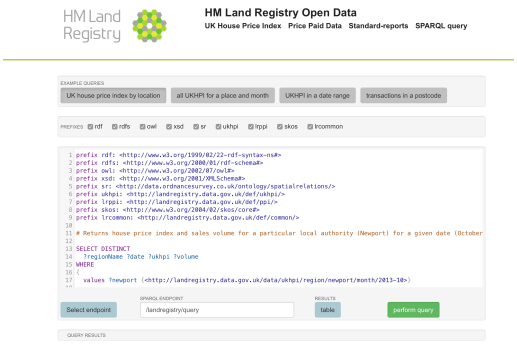


Figure 9.24: SPARQL QUERIES diagram.



(a) HM Land Registry Open Data [27].

(b) AkatemiaSampo [52].

Figure 9.25: SPARQL QUERIES examples.

9.4.3 Advanced Search

Context

An application provides its users with search features and implements KEYWORD QUERIES (p. 96) or a similar simple search approach. The application targets users with no knowledge of RDF or SPARQL and wants to allow them to create more complex queries.

Problem

Provide inexperienced users on RDF or SPARQL with methods that allow them to easily and quickly perform complex search tasks.

Solution

Additionally to a keyword based search, provide a set of advanced search options.

The set of advanced search options is composed of various filters on different properties and characteristics of the search results. As with other data presentation scenarios, it is important to match the type of the data to the way it is presented. Thus, the options can include a combination of more keyword queries on different fields of the resources, checkboxes to filter the classes of the results, dropdown lists when the field has a fixed set of options, sliders for filters on numeric fields, calendars to select dates or time intervals, and maps with area selection features to filter resources with geographical data.

The fields and properties where filters can be applied on depend on the application's domain and on its goals. An analysis on this regard should be conducted to select the filters that will be more important for the vast majority of users when creating more complex search tasks.

After the search task is performed and the search results are presented, it is important to keep the advanced options available to the user. This will allow the users to edit and tweak the search task without the need to restart from the beginning. On more on how to present search results refer to SEARCH RESULTS VISUALISATION (p. 84).

If the application also targets users experts on RDF or SPARQL, and they should have the ability to perform as complex queries as they need SPARQL QUERIES (p. 98) must be provided.

A diagram illustrating the solution described can be consulted in Figure 9.26. Concrete examples on how to apply this pattern are present in Figure 9.27.

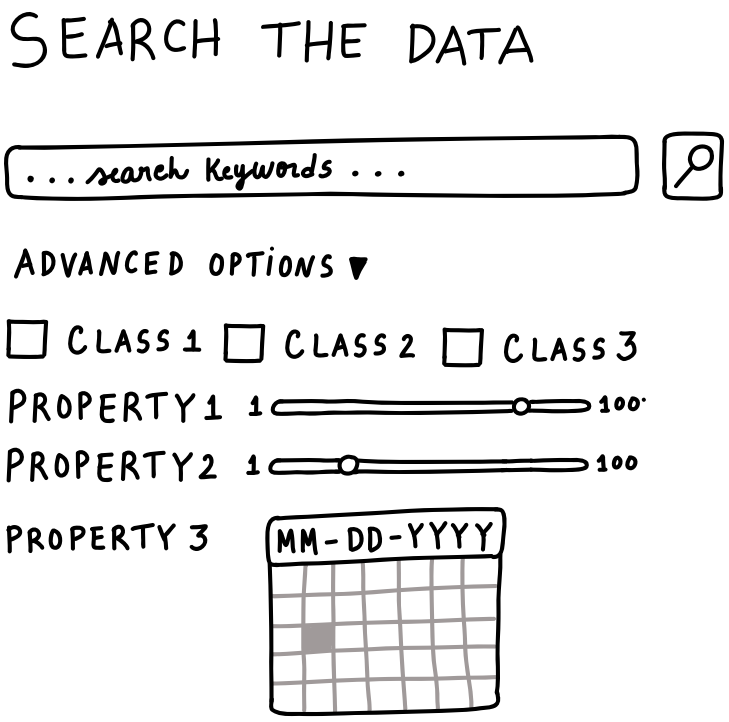
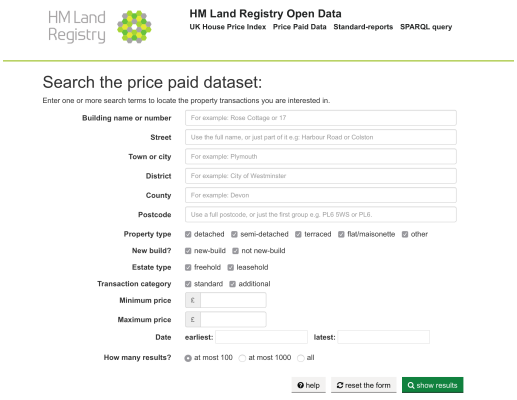
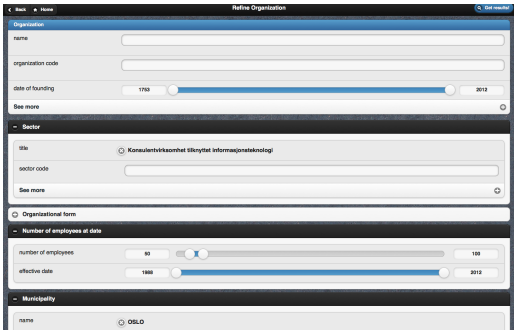


Figure 9.26: ADVANCED SEARCH diagram.



(a) HM Land Registry Open Data [27].



(b) PepeSearch [78].

Figure 9.27: ADVANCED SEARCH examples.

9.5 Browsing Patterns

9.5.1 Link Browsing

Context

Linked Data is composed of a large number of triples, that connect resources with primitive data and with other resources, forming a network of knowledge and information. These related resources can come from internal vocabularies to the application's data collection, or from external vocabularies from other applications and authors. Providing users with the means to browse the data in an application is an important feature. It is even more important when the application allows to visit resources from other vocabularies, allowing the users to browse a subset of the web of data.

Problem

Allow users to transverse the data and visit the pages of internal and external resources.

Solution

Provide hyperlinks for the related resources, to allow the users to navigate between them.

After using the TEMPLATE (p. 76) approach for the pages of the data resources, most likely the layout of the pages will have one or more sections containing related resources to the page's resource. Regardless of the visualisation technique or techniques chosen from VISUAL INFORMATION (p. 90) to display them, make sure that their name, label, or corresponding icon in a graph or map is a clickable link. This link must redirect the users to the corresponding resource's page, this will allow them to transverse the application's data easily and in a familiar manner. If the resources belong to an external vocabulary or collection still provide a link to their web page on another application, if one exists.

Whenever an application's data is too large and too interconnected, the users can feel lost in it after following several links and visiting different resources. When this happens employ BROWSE PATH (p. 104) to aid the users in their navigation through the data.

A diagram illustrating the solution described can be consulted in Figure 9.28. Concrete examples on how to apply this pattern are present in Figure 9.29.

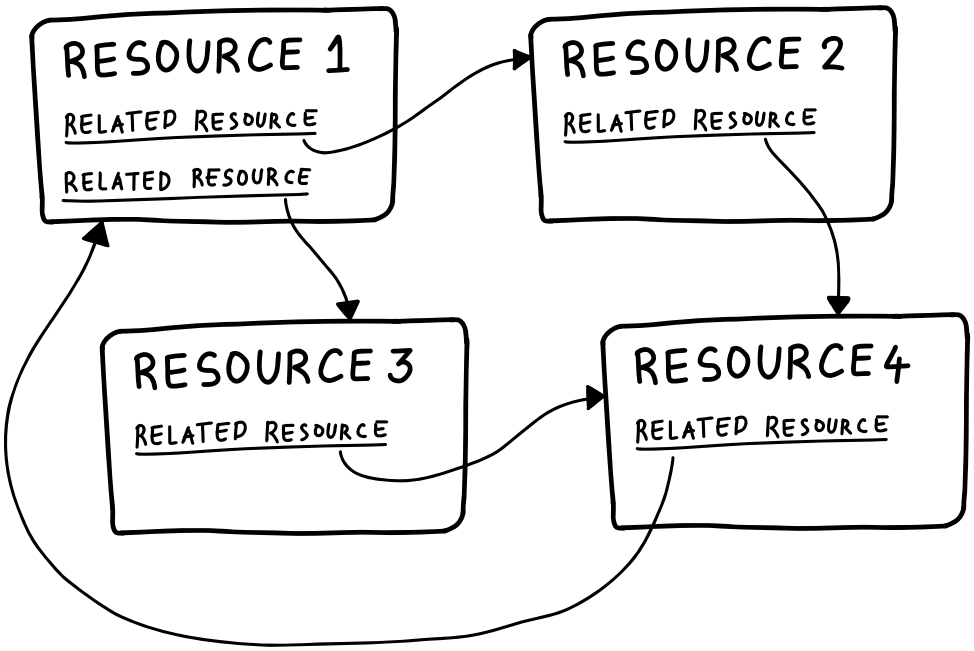
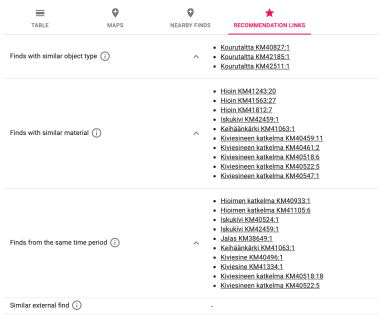
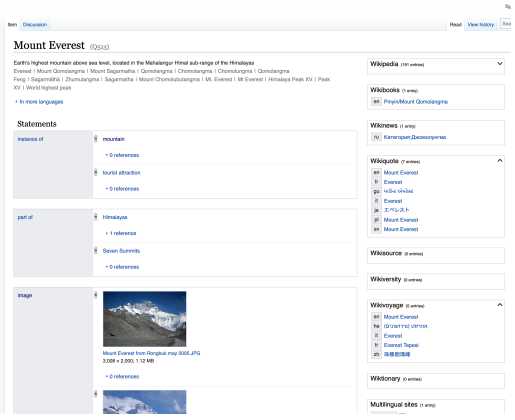


Figure 9.28: LINK BROWSING diagram.



(a) FindSampo [36].



(b) Wikidata [80].

Figure 9.29: LINK BROWSING examples.

9.5.2 Browse Path

Context

One of the characteristics of Linked Data is the inter-connectivity of the data and use of and linkage to other external vocabularies or datasets. Thus, a usual approach to allow users to browse and navigate through Linked Data is to provide hyperlinks to the internal and external related resources. This approach helps the users to easily navigate the data following the direction they want, or that corresponds to their user needs and tasks.

Problem

When navigating through Linked Data, the possible paths between resources are enormous. After following several hyperlinks and visiting many resources' pages, users lose track of where they started and the path that was taken.

Solution

Keep track of the user's path through the data and present it in a way that lets the user return to previous states or previously visited resources.

Presenting the previously visited resources by the user and the path taken while browsing the data will help them to quickly consult the pages already visited and make a decision about the direction in which to take their navigation, according to their needs or tasks.

Allowing the user to navigate to any of the previous states in their browsing path is crucial. It will aid the user when they decide that a different path needs to be taken, by providing a quick and easy way to go back to a previous state and follow a different direction. It will also be useful when users are trying to relate information about two different resources that are separated by relations to several other ones.

A diagram illustrating the solution described can be consulted in Figure 9.30. Concrete examples on how to apply this pattern are present in Figure 9.31.

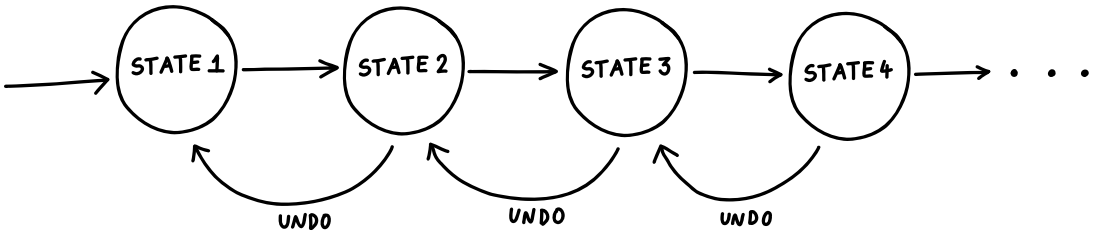


Figure 9.30: BROWSE PATH diagram.

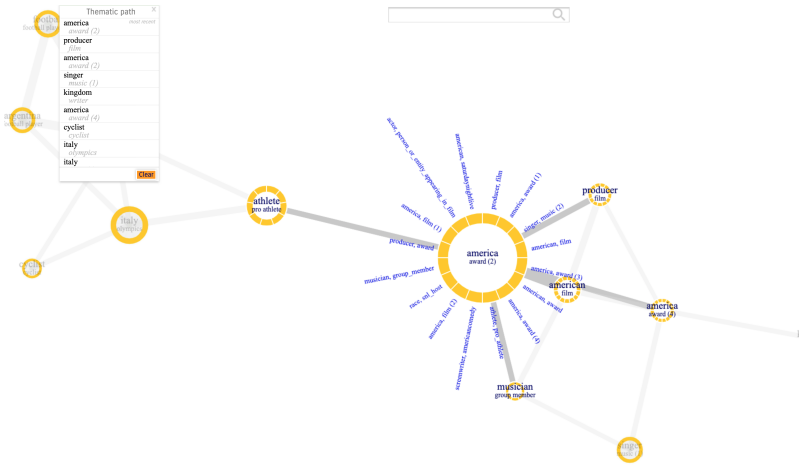


Figure 9.31: BROWSE PATH example - inWalk [13].

9.5.3 Faceted Browsing

Context

Linked Data can be navigated and explored in a variety of ways, such as creating search queries and following the links of the related resources of the search results. However, more simple, traditional, and familiar methods to the users often require some kind of initial search query and the selection of resources, before the users can truly browse the data. This may not be desirable for some user groups because it requires some previous knowledge of the data and the domain.

Problem

Provide the users with a feature that allows the browsing of the data collection without a previous search query.

Solution

Provide the user with a faceted browsing feature, based on the resources' classes.

Allow the users to filter the list of resources in the application, based on their classes and classes of related resources.

This method of navigating and browsing a data collection allows the user to explore without any previous knowledge, while at the same time contributes to their understanding of how the collection is structured.

Additionally to the list of resources that will get more and more filtered, the application should present to the user a list with filters for the resources classes, and a list of filters for the classes of related resources. The users should be able to toggle on and off each one of the filers, as needed. Each filter should contain the name of the class and if not active, the number of resources from the current list that match the filter. Any filter that combined with the ones already active results in an empty result list must be discarded and not presented to the user. This is one of the advantages of this browsing method, as it never leads the user to an empty result list.

Regarding the presentation of each resource in the list, it should be kept to as minimal and simple as possible, due to the fact that the lists are usually very large and extensive at the beginning of the browse. The name of each resource and the classes from the set of active filters that it belongs to must be presented in the list. By clicking on a resource name the users should be redirected to its corresponding page.

A diagram illustrating the solution described can be consulted in Figure 9.32. Concrete examples on how to apply this pattern are present in Figure 9.33.

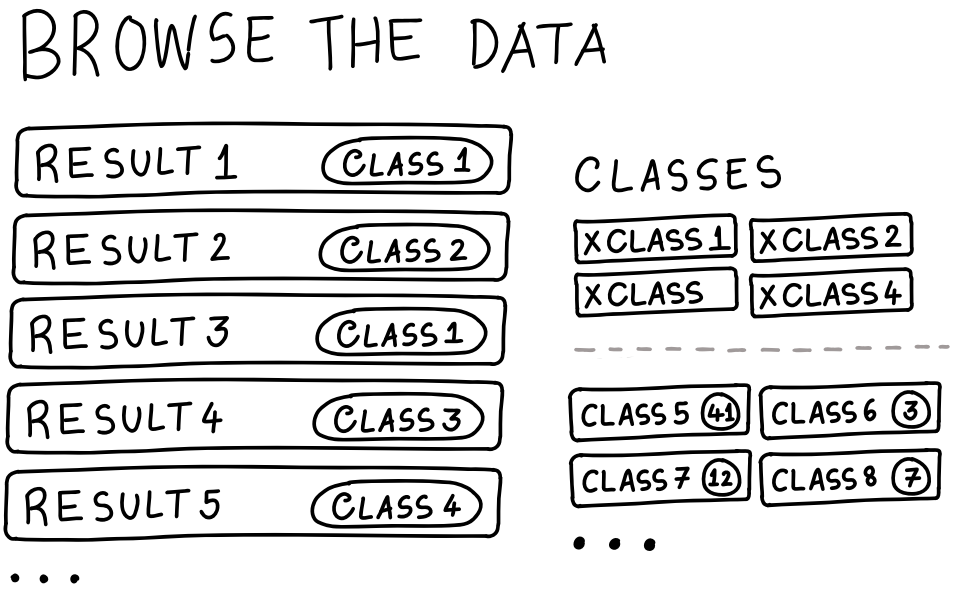
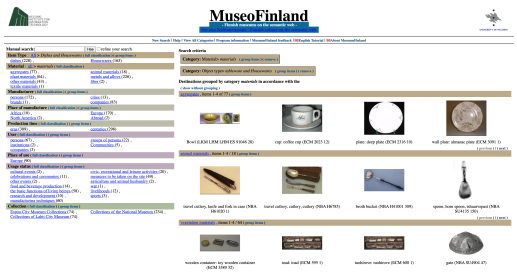
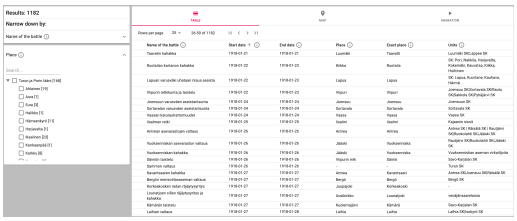


Figure 9.32: FACETED BROWSING diagram.



(a) MuseumFinland[44].



(b) War Victims Portal [61].

Figure 9.33: FACETED BROWSING examples.

9.6 Authoring Patterns

9.6.1 Multiple Input Boxes

Context

Raw RDF data is not easily comprehended by humans, especially users with no previous technical knowledge of Linked Data technologies. However, the ability to create and edit new data is a common feature in many Linked Data applications, and crucial to the development and growth of the web of data. If a Linked Data application provides authoring features to the users, the approach implemented must be accessible to non-expert users.

Problem

When users with no previous knowledge of Linked Data or RDF want to annotate resources or create and edit new triples, they often have difficulty in composing valid and semantically correct data.

Solution

Provide clear text areas for the user to input each part of the triple or triple set.

Regardless of whether an application gives the user the ability to create and edit a triple, annotation, or resource, each one is composed of several parts. Divide the triple or triples into their most basic components and provide the user with a single text input for each one. To the text inputs add a clear label of what should be entered in each one. If the label is not easily comprehensible by all users provide SHOW DEFINITIONS (p. 80).

The application must not allow the input of invalid information in any component of the triples. To aid the user in choosing valid options provide INPUT RECOMMENDATIONS (p. 110) in a dropdown list.

To further help the user to fill the input boxes use INPUT EXAMPLES (p. 112) with sample values for each one of them. These should be tailored to the domain, or resource created or edited.

Dividing the different components of the triples in different input boxes will hide from the users the technical aspects of RDF providing them with a familiar interface, similar to a traditional web form. At the same time it will give them the possibility to customise every possible component.

A diagram illustrating the solution described can be consulted in Figure 9.34. Concrete examples on how to apply this pattern are present in Figure 9.35.

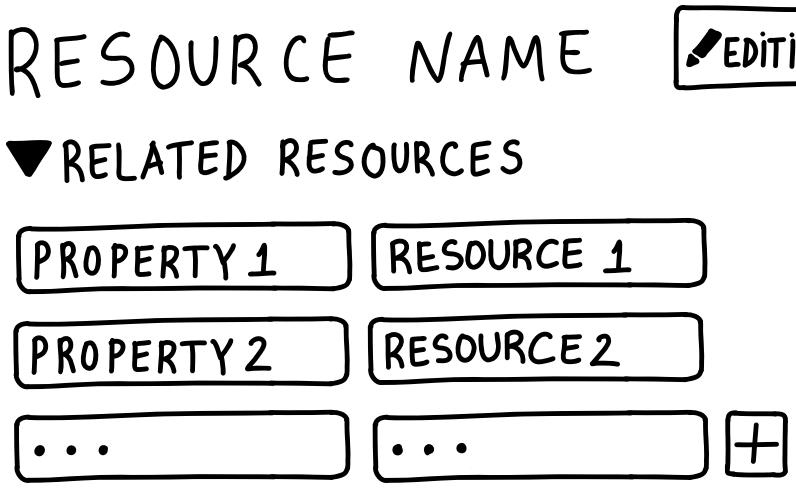
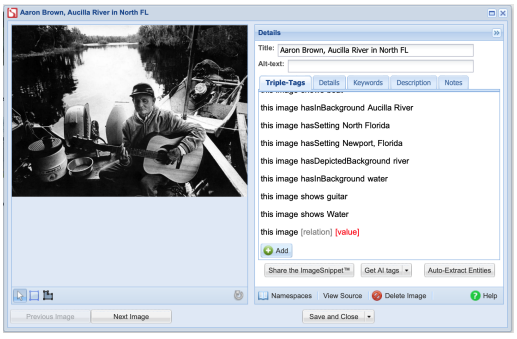
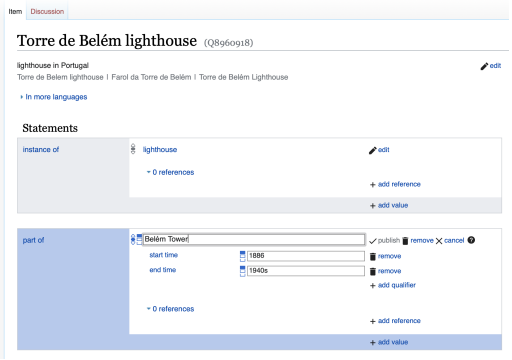


Figure 9.34: MULTIPLE INPUT BOXES diagram.



(a) ImageSnippets [81].



(b) Wikidata [80].

Figure 9.35: MULTIPLE INPUT BOXES examples.

9.6.2 Input Recommendations

Context

An application provides authoring features to the users and wants to make them more accessible to users with no technical experience of RDF or Linked Data. The application is already using MULTIPLE INPUT BOXES (p. 108) or other type of organisation for text inputs.

Problem

Even when individual input boxes with clear labels for each part of the triple to be edit or created are provided, users may sometimes have difficulty in choosing and entering valid options.

Solution

Provide a list of recommendations composed of only valid options for each one of the input boxes.

A limited number of recommendations should be presented to the user in a dropdown list or similar when the input box is selected. This selected number of recommendations can be the most relevant resources or properties of the domain, or the previously used ones by the user.

When the user starts to type on the input box the recommendations must be updated, changing to the search results of the matching resources or properties according to their labels and descriptions. This search can be performed on the internal knowledge base or on external others as well, depending on the domain and goals of the application. This result list must also be limited to a small number of recommendations, sorted by distance to the inputted word or relevance in the domain.

In both cases, a full list of the recommendations must be provided to the user when they choose to. In this full list, if possible additional information about the resources or properties should be provided, such as the description, to guide the users to make the best possible choice for them.

A diagram illustrating the solution described can be consulted in Figure 9.36. Concrete examples on how to apply this pattern are present in Figure 9.37.

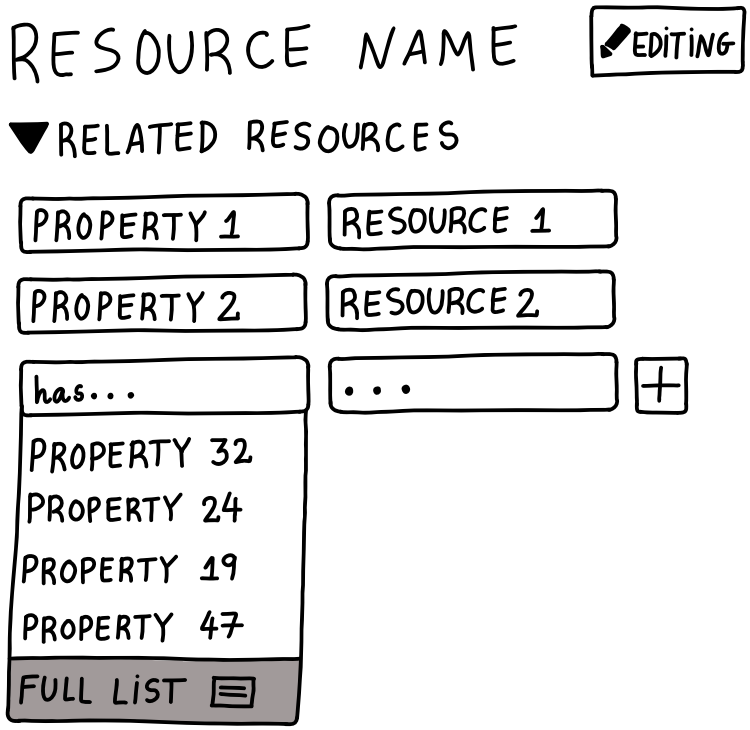
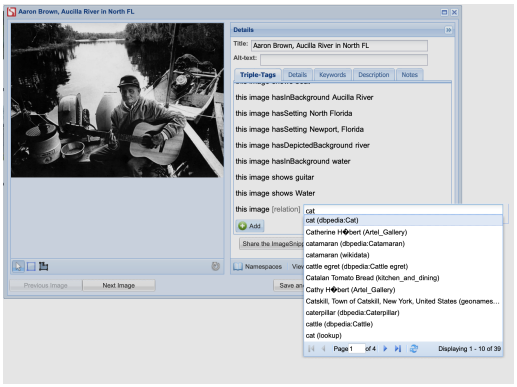
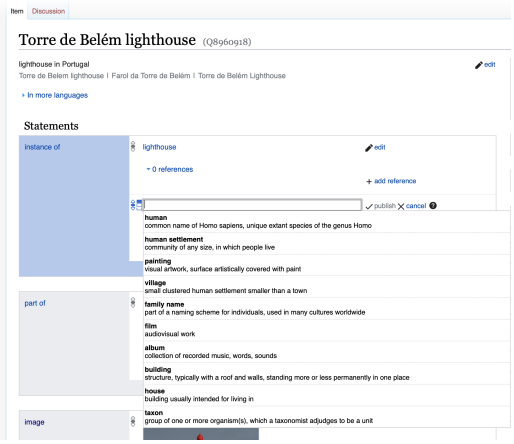


Figure 9.36: INPUT RECOMMENDATIONS diagram.



(a) ImageSnippets [81].



(b) Wikidata [80].

Figure 9.37: INPUT RECOMMENDATIONS examples.

9.6.3 Input Examples

Context

An application provides authoring features to the users and wants to make them more accessible to users with no technical experience with RDF or Linked Data.

Problem

When trying to create or edit data by filling out input boxes, users with less experience on the domain or with Linked Data encounter some difficulties in understanding what information should be inserted.

Solution

Provide examples of similar resources or triples when a user edits or creates data.

Present to the users examples of resources or triples similar to what the user is editing or creating. The examples can be composed of illustrative values tailored to the domain and resource or triple.

If the applications is also targeted at users with experience with RDF or that want to start to work with RDF, the examples can include the boilerplate to create the resource or triple at hand. Allow the users to copy and paste these code snippets.

Even when the triples and resources are divided in their different components and associated with an input box and a clear label, inexperienced users on the domain or with Linked Data may find hard understanding which information should be inserted where. Providing examples of similar resources tries to mitigate this difficulty as it allows the user to better understand how the information is structured, which kind of information are the resources composed of, and consequently how to input the information for the data that they want to create or edit.

A diagram illustrating the solution described can be consulted in Figure 9.38. Concrete examples on how to apply this pattern are present in Figure 9.39.

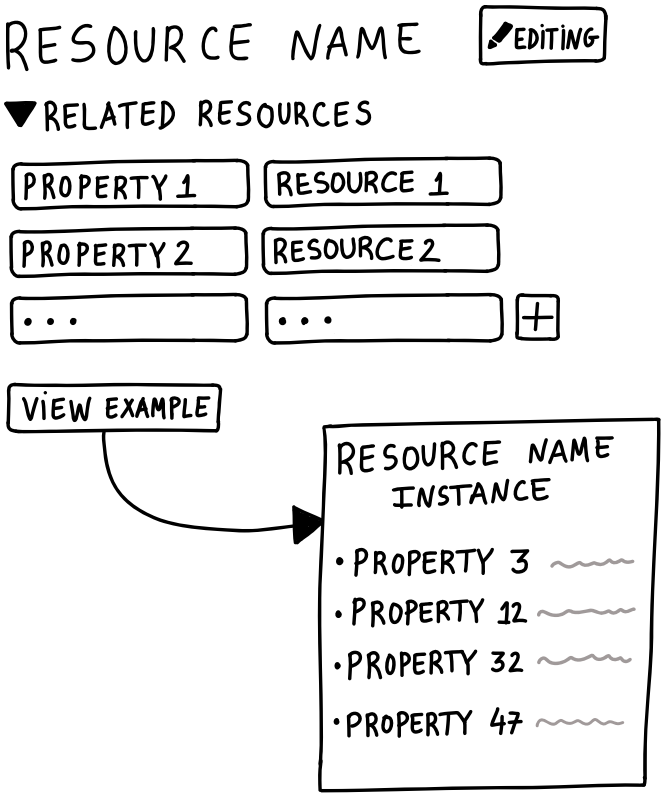


Figure 9.38: INPUT EXAMPLES diagram.

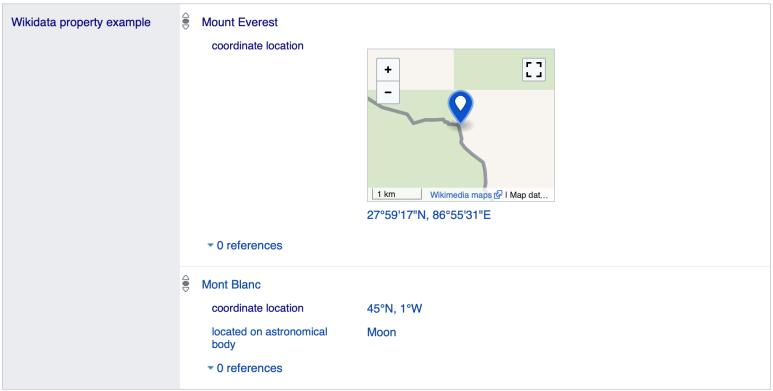


Figure 9.39: INPUT EXAMPLES example - Wikidata [80].

9.6.4 Download Data

Context

One of the first goals of the formalisation and usage of Linked Data was the motivation to share data and vocabularies and have them be used by other researchers, developers, and even companies.

An application's users are meant to be allowed to use its data for personal projects or other purposes.

Problem

Share the application's data with users that are interested in using it for their own projects.

Solution

Provide a download data button whenever a collection of data is presented to the user.

Whenever the user is presented with data of any kind, allow them to download that subset of data by providing a clear and visible download option. This download button can be added in scenarios like the following: the resources' pages, the search result lists, graphs, maps, tables, calendars, or other sets of related resources.

If the application wants to give the users the possibility to download the entirety of the data collection, provide a visually different download button. This specific download button should be included in a component that is present in all pages of the application, like the footer or header. It should also clearly indicate that it will result in the download of the complete data collection.

If possible provide the data collection or a subset of it in different file formats, presenting the user with the different options in a dropdown list, when the download button is selected.

A diagram illustrating the solution described can be consulted in Figure 9.40. Concrete examples on how to apply this pattern are present in Figure 9.41.

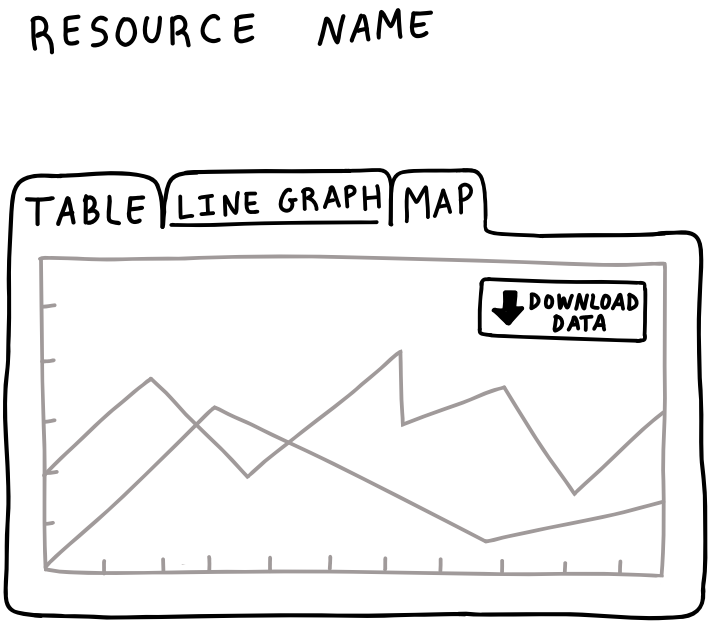
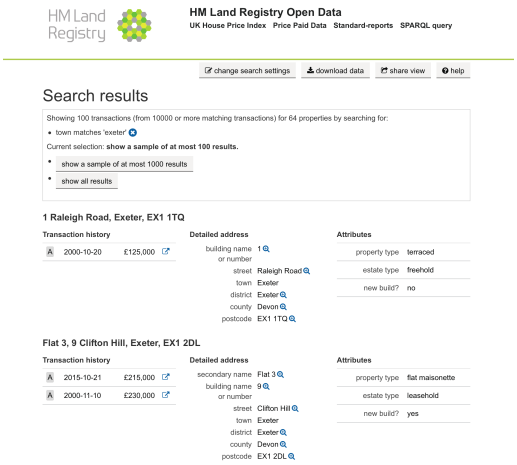
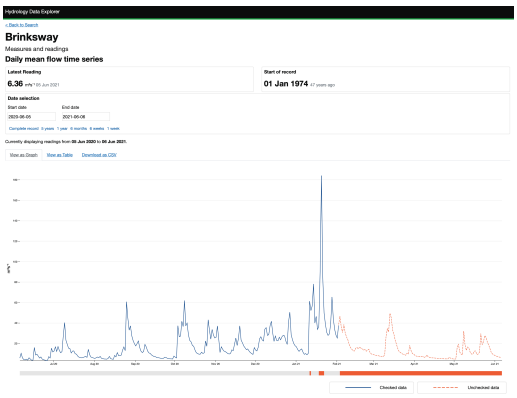


Figure 9.40: DOWNLOAD DATA diagram.



(a) HM Land Registry Open Data [27].



(b) Hydrology Data Explorer [22].

Figure 9.41: DOWNLOAD DATA examples.

Chapter 10

Patterns Evaluation

This chapter covers the methods followed to evaluate the pattern collection on user interaction with Linked Data and their results. In Section 10.1, we start out by conducting an analysis on the structure of the pattern collection, following the Pattern Classification method where the patterns are filed into categories, in order to identify weak and strong points. In Section 10.2, we describe the questionnaire conducted to survey the proposed patterns usage and adoption by the Linked Data community. This survey was based on a method to evaluate the patterns usage and quality, and identify which patterns need to be reviewed and revised.

10.1 Pattern Classification

Pattern classification is a method proposed by Seidel [70] that aims to identify possible strong and weak points in the topics covered by a pattern language or pattern collection. To perform this classification method, a set of rules or categories on how to classify the patterns is needed. Thus, the quality of the results of the classification depend on the quality of the chosen qualification system. Seidel recommends the use of a classification system that is grounded on scientific theory, or the use of a model and classifications derived from the literature. The results of this analysis are the distribution of the patterns on each category, and are more valuable when compared to existing classifications of related pattern languages or collections.

Based on the application of this pattern classification method provided by Seidel [70], we used as the categories for the classification of the patterns, the nine layers of the "Universal Model of User Interface" [4]. This classification system was chosen due to the fact that it has already been showed by Hübscher et al. [42] that it is capable of describing user interaction pattern languages, like Tidwell's [73] and van Welie and Traetteberg's [76]. The classification system based on the "Universal Model of a User Interface" is composed of 9 different categories or layers, divided in three tiers. The first tier is composed of the lowest levels of the user interface: conceptual model, task flow, and organisation model; that form the conceptual and organisational basis of the user experience. The second tier comprises the middle layers of the model: viewing and navigating, editing and manipulation, user assistance. These describe the interactive qualities, users actions,

and the system's reaction to them. Lastly, the final tier is composed of the presentation layers: layout, style, and text; that describe the specific visual and textual expressions of the interface. More detailed descriptions for each layer can be consulted on Table 10.1.

Layer	**Description**
The Conceptual Model	Describes the relationship between the interface and the outside world.
Task Flow	Concerns the manner in which users complete specific operations with the system and its technical environment.
Organisational Model	Describes how the system's content and functionalities are categorised and ordered in the information architecture.
Viewing and Navigating	Details the user interactions that allow the users to navigate the interface and affect the presentation of the content.
Editing and Manipulation	Contains the behaviours that result in permanent changes of the stored information.
User Assistance	Outlines the interface elements that inform the users of the system's activity and status, and that aid and educate the users.
Layout	Describes the design decisions on the placement and ordering of elements on the user interface.
Style	Details the emotion, tone, and visual vocabulary of the elements of the user interface.
Text	Contains all the written and language based elements of the interface.

Table 10.1: Universal Model of a User Interface Layers [4].

10.1.1 Results and Discussion

Seidel [70] recommends that the classification process is performed by someone familiar with the pattern collection to be analysed, thus the classification of the user interaction patterns for Linked Data was conducted by the author of the patterns. For each pattern, the author identified a matching category from the previously described categorisation system. The results from this classification can be consulted in Table 10.2. We can observe that 4 layers don't have patterns associated to them – the conceptual model, task flow, style, and text. From the categories with patterns associated, we have the viewing and navigating layer with 9 patterns, the user assistance and layout layers with 4 patterns each, the organisational model layer with 2 patterns, and the editing and manipulation layer with only 1 pattern. This analysis shows that all patterns from the user interaction for Linked Data pattern collection could be categorised into the categories of the chosen classification system, the "Universal Model of User Interface".

In order to perform a more in depth analysis on the results obtained from this pattern classification method, we included the results from another HCI pattern language, "Designing Interfaces:

Pattern	Layer
TEMPLATE (p. 76)	Layout
MULTI-VISUALISATION (p. 78)	Viewing and Navigation
SHOW DEFINITIONS (p. 80)	User Assistance
GRAPH VISUALISATION (p. 82)	Layout
SEARCH RESULTS VISUALISATION (p. 84)	Layout
RAW DATA (p. 86)	Layout
COMPLETE INFORMATION (p. 88)	Organisational Model
VISUAL INFORMATION (p. 90)	Organisational Model
COLLECTION OVERVIEW (p. 92)	Viewing and Navigation
HIERARCHICAL TREE (p. 94)	Viewing and Navigation
KEYWORD QUERIES (p. 96)	Viewing and Navigation
SPARQL QUERIES (p. 98)	Viewing and Navigation
ADVANCED SEARCH (p. 100)	Viewing and Navigation
LINK BROWSING (p. 102)	Viewing and Navigation
BROWSE PATH (p. 104)	User Assistance
FACETED BROWSING (p. 106)	Viewing and Navigation
MULTIPLE INPUT BOXES (p. 108)	Editing and Manipulation
INPUT RECOMMENDATIONS (p. 110)	User Assistance
INPUT EXAMPLES (p. 112)	User Assistance
DOWNLOAD DATA (p. 114)	Viewing and Navigation

Table 10.2: Pattern classification results.

Patterns for Effective Interaction Design" [73]. This pattern language was chosen because it is considered one of the most complete and important works regarding user interaction patterns. The distribution of the patterns across the different layers for both pattern collections is detailed in Table 10.3.

From this analysis we can point out that only one layer, the conceptional model, did not have any patterns associated from either pattern collections. The layers task flow, style, and text were only assigned to patterns of Tidwell's [73] collection. Viewing and navigation is the dominating layer across both pattern collections, with patterns filed in this category with a share of 39% and 45%. The pattern collection for user interaction with Linked Data is mainly focused on the layers viewing and navigation, user assistance, and layout. Comparing to Tidwell's [73] pattern language, we can conclude that the layers viewing and navigation, and organisational model play a similarly important role in both collections.

Even though the obtained results cannot be fully reliable due to the fact that only one rater was involved, the results still contain valuable information on the strong and weak points of the proposed pattern collection, compared to similar ones. The strong emphasis on patterns for viewing and navigation is not surprising, as for all Linked Data based applications, regardless of domain,

UI Layer	Tidwell [73]	User Interaction Patterns for Linked Data
1. The Conceptual Model	0.00%	0.00%
2. Task Flow	9.91%	0.00%
3. Organisational Model	13.51%	10.00%
4. Viewing and Navigation	38.74%	45.00%
5. Editing and Manipulation	9.91%	5.00%
6. User Assistance	5.41%	20.00%
7. Layout	10.81%	20.00%
8. Style	10.81%	0.00%
9. Text	0.90%	0.00%

Table 10.3: Pattern classification results comparison.

navigation features are crucial. Either regarding performing search tasks, or browsing the data in the application, these features pose new problems when Linked Data is used, due to the fact that everything is linked and to the difficulty in providing alternative approaches accessible to all users. Especially ones that are as powerful as the use of SPARQL queries, when searching data. In Linked Data based applications it is also important to provide the user with ways to customise the data visualisations according to their tasks and needs, due to the diversity of types of data present in Linked Data resources.

Following the viewing and navigation layer, we have two layers with the same percentage of patterns associated, user assistance and layout, both aspects are important when designing user interaction for Linked Data applications. One of the main goals of the proposed pattern collection is to gather good solutions that will aid the user when interacting with Linked Data, by providing information that will help them understand the data and its structure, that will aid them when creating and editing data, and when browsing the dataset. Thus, the moderately high percentage of patterns associated with this layer is in agreement with the pattern collections' goals. Regarding the layout layer, we can justify its percentage with the aim to help designers and developers make informed decisions when choosing what data to present and how to do it in a way that is accessible to all users of the application. This is an important aspect to cover in the pattern collection, because usually Linked Data is presented to the user with approaches that are not accessible to all and that pose user interaction problems.

Even though the organisational model layer has few problems associated to it, it has a comparatively similar percentage to the same layer for Tidwell's [73] collection. Many of the problems and aspects that would be covered by the patterns in this layer, would not emerge from the use of Linked Data, but from the domain and goals of the application. Solutions that are accessible to all users for problems regarding editing and manipulation are important to be covered by a pattern collection for Linked Data. However, the proposed collection only has 1 pattern associated to this layer. We can justify this with the difficulty to find solutions for this problem that are accessible to all users. Nevertheless, we believe it is a matter that needs to be explored more in depth in future

iterations of this pattern collection.

We can justify the lack of patterns associated with the conceptual model and task flow layers with the fact that the proposed pattern collection is focused on new user interaction patterns that arise with the use of Linked Data, regardless of the domain or technical environment of the applications. Patterns concerned with the style layer have not been considered, because their problems are unrelated to the use of Linked Data, and can be constrained by style guides or manuals of the company or organisation of the application. When problems related to the style layer come up in Linked Data based applications, the general solutions covered by collections like Tidwell's [73] can be applied. Similarly, problems regarding the text layer are unrelated to the use of Linked Data, thus the existing solutions can be used when these problems arise.

10.2 Pattern Usage Survey

To evaluate the patterns' adoption and usage amongst the Linked Data community, we conducted an online survey with researchers and developers from the field. Evaluating each pattern is important to understand if the problems and solutions documented are commonly encountered and used, and if the solutions are considered as good by the community.

In order to perform this evaluation, we based our questionnaire on the method proposed by Guerra and Fernandes [31]. The method proposed is composed by a 3 star rating system, that combines several characteristics of the patterns and pattern collection. The first star is attributed to a pattern if at least three known use cases are identified for it, the second and third stars are received by a pattern if it is considered a recurrent and good solution amongst experienced and inexperienced developers in the field. From this method, we conducted the usage and adoption questionnaire for experienced Linked Data developers and researchers.

Guerra and Fernandes outline a set of options that the participants of the questionnaire must choose between regarding their experience and opinion on the problem and solution documented by each pattern. These answer options are presented in Table 10.4, and allow the author of the patterns to understand if the participants have used the pattern or not, and if they consider it a good solution for the problem that it documents. Each response option is associated with an evaluation and with a number of points, that are used to calculate the final score for each pattern. Guerra and Fernandes state that for a pattern to receive an overall positive score and be recognised as a recurrent and good solution, it must obtain a score higher than half of the participants' total number.

10.2.1 Design and Execution

The questionnaire used for this online survey was designed according to the method by Guerra and Fernandes [31], previously detailed. Given that the participants were all volunteers, we took into account that their availability and willingness to spend time answering the survey was limited. Thus, instead of presenting the participants with a full version of each pattern, we decided to only include a summarised version, present in Appendix D, to minimise the total time of the

Answer Option	Evaluation	Points
I have already used this pattern structure exactly as it is documented.	Positive	+1
I have already used this pattern structure with some variations.	Positive	+1
I have not used this pattern structure yet because I did not need it.	Neutral	0
I have not used this pattern structure yet and I think it is not a good solution.	Negative	-1
I have not used this pattern structure yet and I think it could help to solve some problems that I faced.	Positive	+1
I am not sure if I used this pattern structure and I think it is a good solution.	Positive	+1
I am not sure if I used this pattern structure and I think it is a bad solution.	Negative	-1

Table 10.4: Pattern usage survey answer options [31].

survey, and consequently increase the number of responses. In addition to the 20 questions where the participants could report their opinion and usage of each pattern, we included an open text question for each pattern category, where the participants could leave any comments and thoughts on the patterns. The original questionnaire is available in Appendix E.

After designing the questionnaire we set it up using Google Forms and disseminated it via email to the Linked Data researchers interviewed in the pattern mining session, and additional mailing lists in the Linked Data community. Some of these mailing lists include the W3C's RDF-DEV Community Group[1] and Linked Open Data[2] mailing list, the Knowledge Graph Conference community on Slack[3], the Wikidata[4] mailing list, and the Europeana Tech[5] mailing list. The form was online from the 21st of May to the 14th of June, during which time we got 20 responses.

10.2.2 Survey Answers

From the 20 obtained answers to the pattern usage and adoption survey, we received from every participant a rating for almost every proposed pattern. Apart from the COLLECTION OVERVIEW, SPARQL QUERIES, INPUT RECOMMENDATIONS, DOWNLOAD DATA, MULTIPLE INPUT BOXES, and INPUT EXAMPLES patterns with 19 ratings, every pattern received a total of 20 ratings. The answers for each proposed pattern are combined in Figure 10.1.

We can point out from the obtained results that 13 out of the 20 proposed patterns have a 50 or higher percentage of participants reporting to have already used them, exactly as they are presented or with some variations. We can also highlight the LINK BROWSING pattern as the pattern with the highest percentage of participants stating to have used it, with 95%, and INPUT

[1] https://lists.w3.org/Archives/Public/semantic-web/
[2] https://lists.w3.org/Archives/Public/public-lod/
[3] https://www.knowledgegraph.tech
[4] https://lists.wikimedia.org/hyperkitty/list/wikidata@lists.wikimedia.org/
[5] https://www.europeana.eu/en

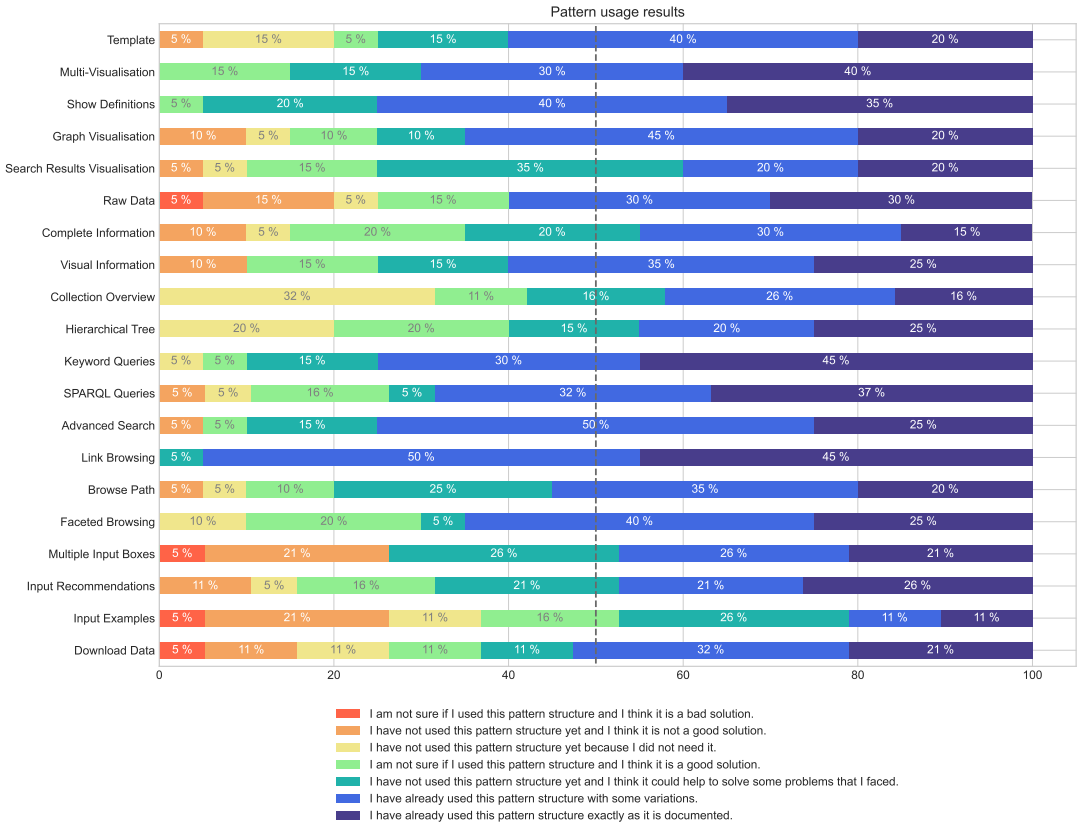


Figure 10.1: Pattern usage results.

EXAMPLES with the lowest percentage of about 20%. In regards to the answers that reported not using or not being sure if they have used the pattern, we can point out the SEARCH RESULTS VISUALISATION pattern with the highest percentage of participants considering it a good solution regardless, followed closely by INPUT EXAMPLES, COMPLETE INFORMATION and INPUT RECOMMENDATIONS patterns.

Regarding negative ratings of the patterns, where the participants reported not considering them a good solution, we can observe 7 out of the 20 patterns with no negative ratings. From the remaining 13 patterns, we can point out MULTIPLE INPUT BOXES, INPUT EXAMPLES, RAW DATA and DOWNLOAD DATA, with the highest percentage of participants considering them bad solutions. We can also observe that 13 out of the 20 patterns were reported as not being used due to the fact that they were not needed, highlighting the COLLECTION OVERVIEW and HIERARCHICAL TREE patterns as having the highest percentages.

From the answers obtained, we cannot observe a general preference for a category of proposed patterns over another, as the positive and negative ratings are distributed amongst the categories. However, we can point out that 3 out of the 4 patterns with the highest percentage of negative ratings belong to the authoring category, making it the category that was the least well received out of all four.

Regarding the comments by the survey participants, we can point out, related to the visualising patterns, a participant expressing the need for new visualisation techniques. In addition to the commonly used ones documented in the proposed patterns, this participant states that visualisation techniques that help users transcend complexity and improve contextual understanding are needed. We can also highlight an issue reported, when visualising Linked Data choosing between aggregation, where the resources are shown with their related information, or navigation, showing hyperlinks for the related resources. The participant states that the later results in a simple and consistent way of presenting the data, but can leave users lost when navigating. However, the former can result in overloading the user with information. This reported issue is one we try to tackle with the proposed pattern COMPLETE INFORMATION, but that we recognise it still needs special attention in a future iteration of the collection. Another participant pointed out that the visualisation of the dataset's metadata is important and should be covered by the collection, as it can help provide context and ease the interpretation of the data. Finally, a participant expressed their concern with the fact that graph visualisations don't scale very well, and can quickly become overwhelming to the users. This is something we covered and tried to tackle while writing the GRAPH VISUALISATION pattern.

Regarding the comments on the searching patterns, we can highlight one participant that states that while they usually use a variation of the SPARQL QUERIES pattern, they try to use instead augmented search techniques that tailor semantics to the persona and jobs the individual is trying to perform. We can also point out a problem reported by a participant, that should be considered in a future interaction of the pattern collection. This participant reports that in practice it is difficult to find the right balance of heuristic searching and producing too many results, as it would be useful to have the system match close results when a user misspells terms. However, this can greatly increase the number of results and consequently make hard for the user to find the right result.

For the browsing patterns, we can highlight two comments, one states that the participant found that keeping a browsing history does not help the user, according to their evaluations. The other comment states that while faceted browsing is a powerful feature that users tend to enjoy, it can cause a heavy load on the backend query engine.

Finally, regarding the authoring patterns, the majority of comments expressed the fact that users don't typically think of the underlying structure as triples, but rather in the domain language. For this reason, some participants tend to prefer to have the user interface do the work of translating between domain terms, that the user understands, and the underlying structure. We can also point one participant that stated the preference to encourage saving the Uniform Resource Identifiers instead of letting the user download the data.

10.2.3 Pattern Usage Results

After briefly analysing the unprocessed answers to the pattern usage survey, we followed the method outlined by Guerra and Fernandes to calculate the scores for each pattern resulting from the obtained answers. The final patterns' scores can be consulted on Table 10.5. Following the recommendation by Guerra and Fernandes [31] to consider a pattern score as positive if its absolute

number is larger than half of the total number of ratings, we can conclude that 18 out of the 20 proposed patterns have a positive score.

Looking at the results on Table 10.5, we can observe that the patterns MULTIPLE INPUT BOXES and INPUT EXAMPLES have a score lower than half of their total ratings, which means that it has an overall negative rating and needs to be revised. We can also point out that 3 patterns, MULTI-VISUALISATION, SHOW DEFINITIONS, and LINK BROWSING, have only received positive ratings, and obtained a perfect score of 20. In addition to the patterns with negative scores, we can also identify some other patterns that might benefit from a revision of their content or structure, namely 3 patterns, COLLECTION OVERVIEW, RAW DATA, and DOWNLOAD DATA, due to their very low score.

We can also highlight from the patterns' results that the four patterns with the lowest scores propose a solution where the internal structure of Linked Data is displayed to the user. Either by displaying data in RDF in the RAW DATA and DOWNLOAD DATA patterns or exposing the triple structure in the MULTIPLE INPUT BOXES and INPUT EXAMPLES patterns. This is one aspect that needs further attention on future iterations of the pattern collection, requiring revision of the already proposed graphs and an analysis on additional similar solutions.

Pattern	**Total**	**Positive**	**Neutral**	**Negative**	**Score**
MULTI-VISUALISATION (p. 78)	20	20	0	0	20
SHOW DEFINITIONS (p. 80)	20	20	0	0	20
LINK BROWSING (p. 102)	20	20	0	0	20
KEYWORD QUERIES (p. 96)	20	19	1	0	19
ADVANCED SEARCH (p. 100)	20	19	0	1	18
FACETED BROWSING (p. 106)	20	18	2	0	18
SEARCH RESULTS VISUALISATION (p. 84)	20	18	1	1	17
BROWSE PATH (p. 104)	20	18	1	1	17
VISUAL INFORMATION (p. 90)	20	18	0	2	16
HIERARCHICAL TREE (p. 94)	20	16	4	0	16
SPARQL QUERIES (p. 98)	19	17	1	1	16
TEMPLATE (p. 76)	20	16	3	1	15
GRAPH VISUALISATION (p. 82)	20	17	1	2	15
COMPLETE INFORMATION (p. 88)	20	17	1	2	15
INPUT RECOMMENDATIONS (p. 110)	19	16	1	2	14
COLLECTION OVERVIEW (p. 92)	19	13	6	0	13
RAW DATA (p. 86)	20	15	1	4	11
DOWNLOAD DATA (p. 114)	19	14	2	3	11
MULTIPLE INPUT BOXES (p. 108)	19	14	0	5	9
INPUT EXAMPLES (p. 112)	19	12	2	5	7

Table 10.5: Pattern classification results.

10.2.4 Conclusions

Upon the presentation of the results, we need to point out some threats to their validity that were identified. The participants of the pattern mining interviews were also included in this pattern evaluation, which can result in a tendency for more positive answers, as the proposed patterns were highly influenced by their reported experience and opinions. In addition, other participants from the Linked Data community were also included in the conducted evaluation. However, the total number of participant is small, and greatly justifies the need for a follow-up evaluation with a different pool of Linked Data researchers and developers. We also need to take into consideration the fact that only a summary of each pattern was shared with the participants for this evaluation. This can result in positive and negative misunderstandings of the patterns. Again, a more in depth and thorough evaluation needs to be conducted in the future, to further improve the proposed patterns and understand their possible adoption within the community.

All in all, the results from the pattern usage and adoption survey are positive, with 18 out of the 20 proposed user interaction patterns obtaining positive scores. However, the proposed patterns received a wide range of scores, and almost all need future revising and possibly restructuring. One of the next steps to improve the pattern collection is to understand why some of the patterns received negative or low scores. It is very important for future iterations of the patterns to know if their low score was impacted by the quality of the proposed solution, validity of the problem, lack of clearness in their written representation, or misunderstanding of the context. In order to come to this information, further evaluations need to be conducted, possibly in more open formats than the use of a questionnaire, such as one-on-one interviews or workshops.

Despite these limitations, they still indicate a positive reception of the proposed patterns by the Linked Data community. We believe the proposed pattern collection can be a valid and useful tool for developers and designers, regardless of experience, to improve the user interfaces of their Linked Data based applications and tools.

Chapter 11

Wikidata Practical Case

After proposing a pattern collection for user interaction with Linked Data and validating its structure, and the usage of the patterns and the quality of the solutions with the Linked Data community, we chose to complement its validation with an analysis of a well-known Linked Data based application. For this analysis, we selected Wikidata, a free knowledge base that integrates into a single location the Wikipedia's data, using Linked Data. Wikidata is an appropriate option for this analysis, due to its large user base, and due to the features provided to the users, encompassing the four categories of the proposed pattern collection, visualising, searching, browsing, authoring.

This chapter is divided in 3 sections, we start out by outlining the goals of this analysis and the questions we aim to answer in Section 11.1, then we present the outcome of this analysis by presenting the results in Section 11.2, and finally in Section 11.3 we briefly discuss the obtained results.

11.1 Goals

With this analysis we aim to answer RQ3, "How can the abstracted solutions (of the patterns) be applied to a specific context"?, with more detail in addition to what is presented in the examples section of each pattern. We will start by analysing Wikidata's [80] user interfaces and identifying which patterns are already implemented, and which patterns are not. Our goal is to present to developers and designers how the patterns can be applied to a specific context, thus we will describe and show how each pattern is implemented in the Wikidata's user interface. Additionally, we will present rough drafts of possible implementations for the patterns that are not used in the Wikidata's user interface. We will describe where the patterns can be applied, how they can be turned into a specific solution, and why the use of the pattern is or not justified in each specific context. Additionally, we will briefly discuss this experiment by highlighting any possible difficulties or problems that may come up during the process of applying the proposed pattern collection to this specific context. All in all, our intent is to show how the proposed patterns can be applied, separately and collectively, to better an application's user interface, and its users experience while using it.

11.2 Analysis Results

After analysing the different pages of Wikidata's user interface and exploring the various features provided to the users, we identified the patterns from the proposed collection that are implemented in the interface. From the proposed 20 user interaction patterns for Linked Data, we identified 11 patterns in the Wikidata's interfaces, 3 visualising patterns, 3 searching patterns, 1 browsing pattern, and 4 authoring patterns. We can point out that all the proposed searching and authoring patterns are present in the user interface, and that the visualising category is the one with the lowest percentage of patterns implemented, with 30%, followed by the browsing category with 33% of patterns.

The patterns present in Wikidata's user interfaces are listed next, along with where they can be encountered and how they were applied.

SHOW DEFINITIONS: Each resource is presented alongside its associated properties, in an individual page. When a user clicks on the name of each resource, they are redirected to that resource's page, where a description of the property is provided. Even though this implementation is not as straight forward and easy to access as when the definitions are provided when the user hovers the property's name, the meaning of every property is still provided without the user having to perform any external or additional search tasks. Additionally, the individual pages for the properties allow for more complete and extensive definitions. Wikidata's implementation of this pattern can be consulted in Figure 11.2.

SEARCH RESULTS VISUALISATION: After a user performs a search task, they are presented with a list containing the search results. Each search result is presented with its label, name, small description with the most relevant information, and number of statements and sitelinks. Sorting options are also available to the user, in order to customise the ordering according to their needs. In addition to this information, the search results could be divided in different sections according to their classes. Wikidata's implementation of this pattern can be consulted in Figure 11.3.

VISUAL INFORMATION: In a resource's page, the related information is presented according to its type, geographical information is presented using a map, photos are displayed without the need to download, and audio files can be played directly in the page. However, temporal data is presented in textual form, when it could be displayed using calendars or timelines when possible, and statistical data could be presented using plots. Wikidata's implementation of this pattern can be consulted in Figure 11.1.

KEYWORD QUERIES: The users are provided with the ability to perform simple keyword queries in the application's data. The user can also customise in which parts of the data the inserted keywords are search in. Wikidata's implementation of this pattern can be consulted in Figure 11.3.

SPARQL QUERIES: The users are provided with a SPARQL Query Service, where the users with SPARQL knowledge can create more complex queries. The user can quickly copy and paste queries into the editor, and examples are also provided. An additional feature where the users could visualise the SPARQL query of a performed keyword query could be implemented, to allow the users to transition from a keyword to SPARQL query more easily. Wikidata's implementation of this pattern can be consulted in Figure 11.4.

ADVANCED SEARCH: In addition to the provided keyword queries, users also have access to advanced options to add to their search tasks. Text areas and dropdown lists are provided for the additional filters to the search results. If other types of data were to be included in the advanced options, other types of filters could be added, such as sliders, calendars, and area selection on maps. Wikidata's implementation of this pattern can be consulted in Figure 11.3.

LINK BROWSING: In a resource's page all the related resources are provided with hyperlinks that the user can follow and visit each resource's page. If the related resource presented belongs to an external vocabulary, a link to its page could also be provided. Wikidata's implementation of this pattern can be consulted in Figure 11.1.

MULTIPLE INPUT BOXES: When creating or editing the information about a resource, the user is presented with text areas with clear labels for every piece of information that needs to be entered. Wikidata's implementation of this pattern can be consulted in Figure 11.1.

INPUT RECOMMENDATIONS: When creating and editing the information about a resource, the user is presented with valid recommendations when data is inputted in a text area. The recommendations are presented to the user when the text area is selected, and filtered as the user enters data. However, a full lookup of the valid options for that text area is not available. Wikidata's implementation of this pattern can be consulted in Figure 11.5.

INPUT EXAMPLES: In the individual pages of the properties, the user can consult a section with examples of resources that use that property and how they use it. Even though examples are not provided when the user is editing or creating, the property can be searched and examples consulted. If the examples were available to the user in the editing page, they could be consulted more easily. Wikidata's implementation of this pattern can be consulted in Figure 11.5.

DOWNLOAD DATA: After performing a search task using the SPARL query service, the user can download the search results. However, this is the only page where the user can directly download the data that is presented to them. Additionally, download data features could be added to resource pages and keyword search result pages. Wikidata's implementation of this pattern can be consulted in Figure 11.4.

Regarding the proposed patterns that are not implemented in Wikidata's user interface, we have identified 9 out of the 20 proposed patterns, 7 visualising patterns, and 2 browsing patterns.

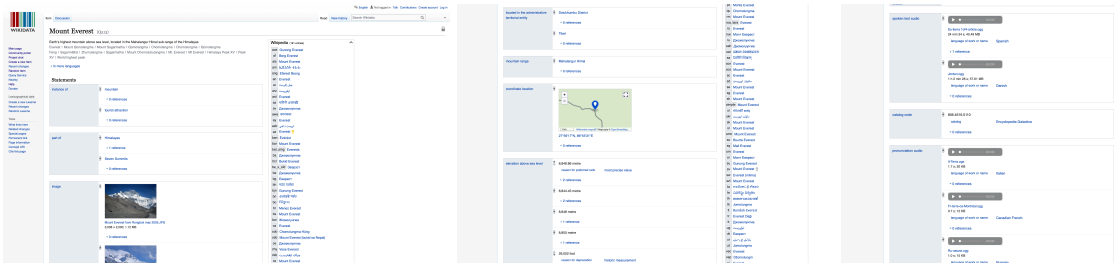


Figure 11.1: Wikidata resource page.

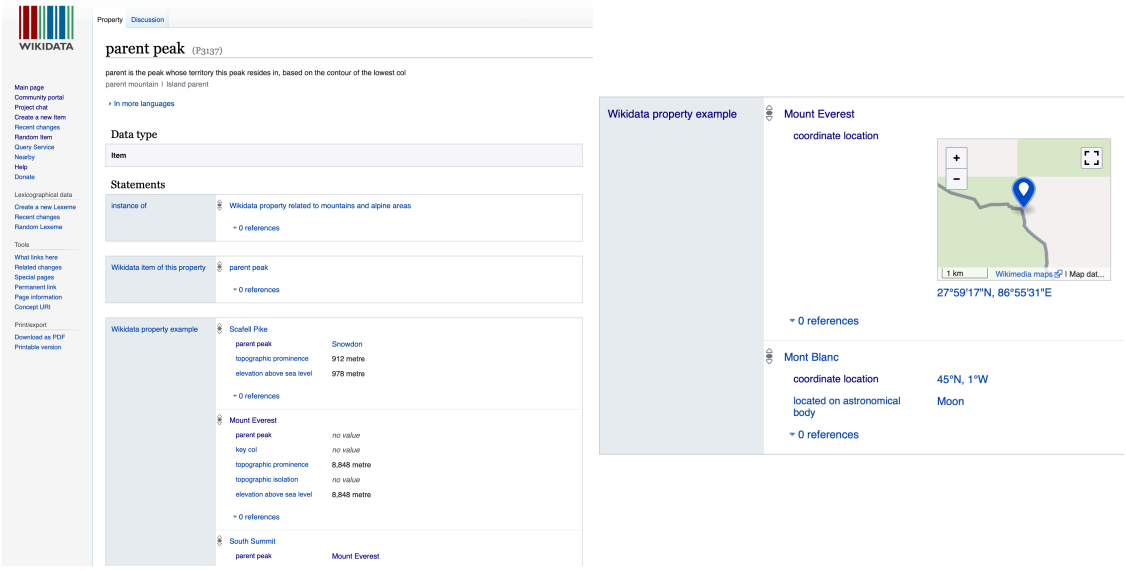


Figure 11.2: Wikidata property page.

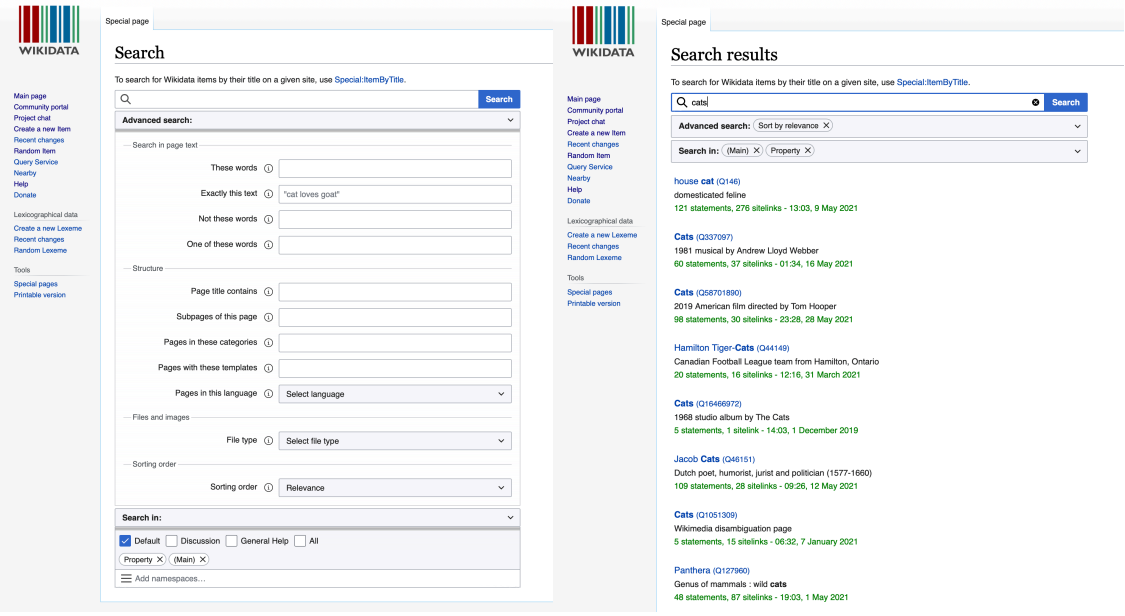


Figure 11.3: Wikidata search form and search results page.

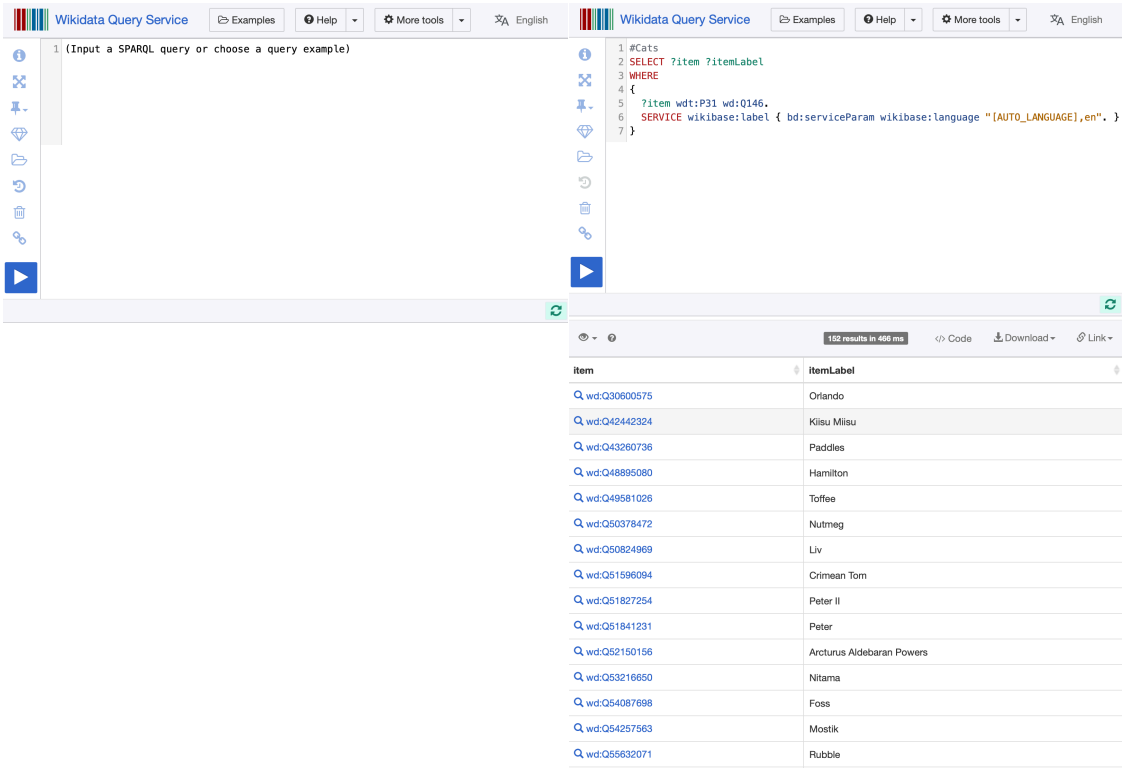


Figure 11.4: Wikidata SPARQL editor page.

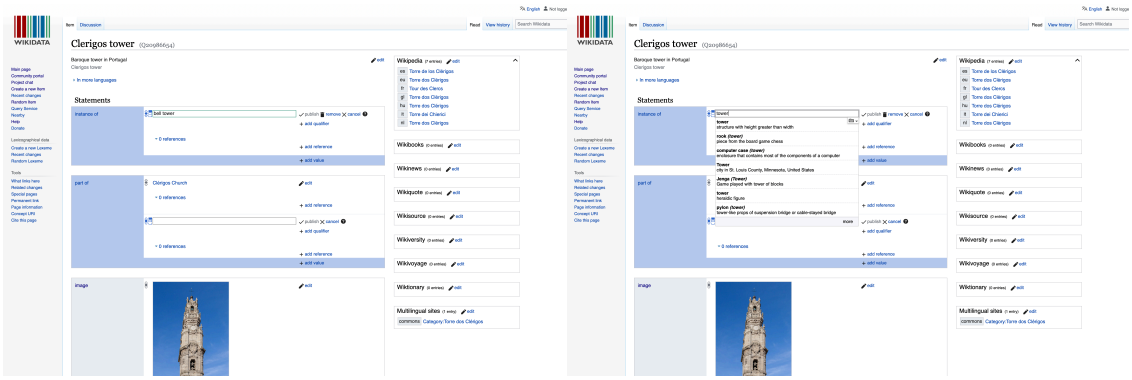


Figure 11.5: Wikidata edit resource page.

These are listed next, along with descriptions of why and where they can be applied, and drafts illustrating how they could be applied.

TEMPLATE: Wikidata's resources are generally composed of a large number of statements, containing related information and resources. These statements are also presented visually similar to each other on a single section. This is a case where the TEMPLATE pattern could help organise the display of the resource's information. When implementing this pattern, we should start by identifying which is the most important and relevant information about each resource, then defining an outline of how to order the information and which visualisation techniques to use. The remainder of the related resources and information should be placed in closed sections for the user to consult if needed. The implementation of this pattern will help the resource's page look more approachable to all users. A draft of a possible implementation of this pattern is illustrated in Figure 11.6.

MULTI-VISUALISATION: In the page of each resource, usually the different types of data are displayed with the visualisation technique that best suits them, implementing the VISUAL INFORMATION patterns. However, the user does not have the option to visualise all the statements associated to a data type together in a visualisation, as for example the ability to visualise all the geographical data associated in a map. The MULTI-VISUALISATION pattern could be applied by letting the user choose between visualising the statements in a list, as currently available, or in a map, graph, or calendar according to the different types of data. This provides the user different views on a resource's information, and a better overview of it.

GRAPH VISUALISATION: When displaying a resource's related information, Wikidata's user interface does not use graph-based visualisations for any type of data. This visualisation could be helpful when the resource has information about family trees or friend networks. This graph visualisation could be included in the set of options available when implementing the MULTI-VISUALISATION pattern.

RAW DATA: Similarly to the GRAPH VISUALISATION pattern, Wikidata also does not provide the user with the visualisation of raw data, when consulting a resource's page. The inclusion of raw data to the set of options available when implementing the MULTI-VISUALISATION pattern would allow the more experienced users to further inspect and explore the data, without the need to download it before.

COMPLETE INFORMATION: To our knowledge the Wikidata's user interface does not include any instance of an implementation of the COMPLETE INFORMATION pattern. Due to our lack of knowledge on the vocabulary and most important user tasks and related information, we are not capable of determining if there is a context in the Wikidata application that justifies the use of this pattern.

COLLECTION OVERVIEW: Wikidata does not provide the user with an overview of the data collection and structure of the vocabularies used. This may be due to the large number of classes and relationships encompassed in the application. However, the user would benefit from a set of the most relevant or most visited classes at the landing page of the application. This would allow the user to immediately know some of the information encompassed in the application, and would allow them to start browsing the data collection if a faceted browsing feature was also provided.

HIERARCHICAL TREE: When visiting a resource's page the user can consult the resource's class and subclasses. However, a full hierarchical tree is not provided, but could help the user better situate the resource in the data collection, and browse the neighbouring classes. A draft of a possible implementation of this pattern is illustrated in Figure 11.6.

BROWSE PATH: Wikidata's user interface allows the users to fully browse the data collection by following the links of the related resources, using the LINK BROWSING pattern. However, they could feel lost after visiting a large number of resources, and not remember what they have consulted. This is a case where the BROWSE PATH pattern could help the users when browsing the data. To implement this pattern, a breadcrumb style component could be added to the page, containing the list of resources the user has visited while using the application. The users should also be allowed to click on the resources and go back to their corresponding page. A draft of a possible implementation of this pattern is illustrated in Figure 11.6.

FACETED BROWSING: Even though Wikidata allows the users to perform search tasks and browse the dataset by LINK BROWSING, the application does not provide an option where the user can start browsing without a previous search task. The inclusion of the FACETED BROWSING could help overcome this problem. The implementation of this pattern in the Wikidata's user interface could pose some hurdles, due to the large number of classes and instances in the data collection. However, we believe they could be solved with the use of a pagination feature when presenting the list of instances, and a curation of the filters to be included.

11.3 Discussion

This analysis allowed us to understand how some of the proposed patterns are applied in the user interface of Wikidata, one of the most used and most important applications in the Linked Data field. We identified 11 out of 20 patterns in Wikidata's user interface, and identified the remaining 9 as possible to be implemented in order to better the users experience in the application. During this exercise, we found that a deep and thorough understanding of the vocabularies used is needed, as well as, the knowledge about which tasks are more frequently executed, and

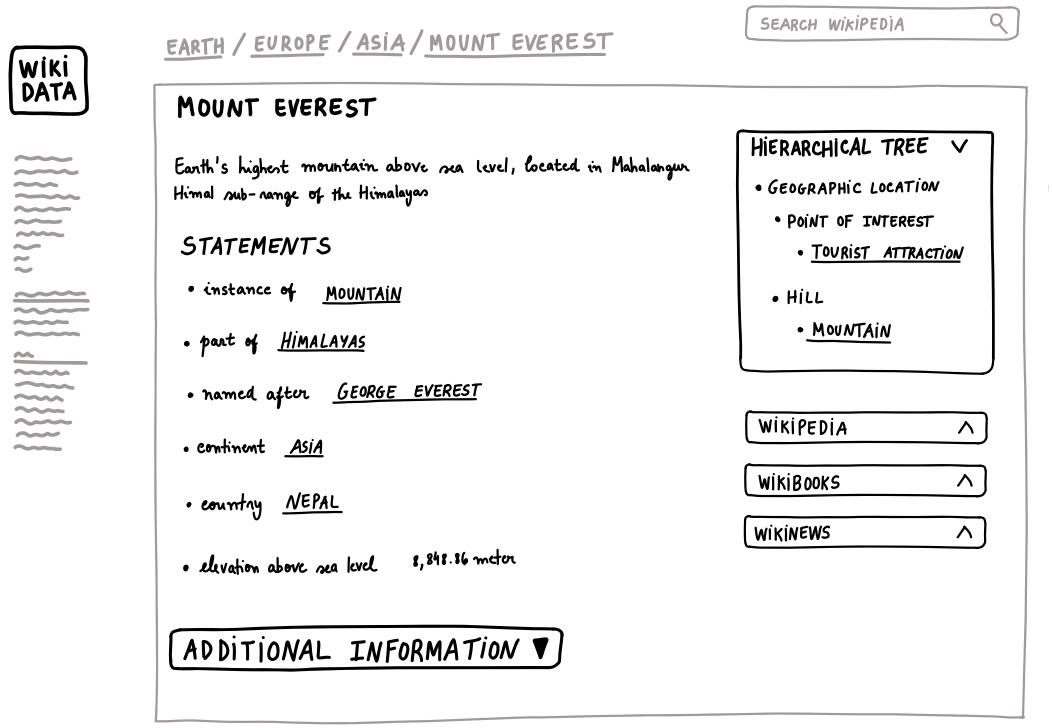


Figure 11.6: Draft of a possible implementation of the TEMPLATE, HIERARCHICAL TREE, and BROWSE PATH patterns in Wikidata's user interface.

which information is typically more important for the majority of users. This information is critical to understand if and how some patterns can be applied, such as the TEMPLATE, MULTI-VISUALISATION, COMPLETE INFORMATION, COLLECTION OVERVIEW, and FACETED BROWSING, where the implementation relies on choices of what information should be displayed, and how to do it. This was the only difficulty encountered in the execution of this analysis, where almost all the patterns proved able to be implemented regardless of vocabulary, or style choices of the application.

From this analysis, we can conclude that the proposed patterns are easily integrated and applied to existing user interfaces and Linked Data solutions, and that they work together to tackle a vast number of user interaction problems. Additionally, we were able to present to the user a more complete example of how the proposed abstract solutions can be applied together to a specific context.

Chapter 12

Conclusions and Future Work

With the rise of interest in Linked Data, we start to see more applications and platforms that use this data modeling paradigm. This interest is fueled by the benefits that semantically annotated and machine-readable information can have in many systems, like search engines and recommendation systems. However, Linked Data is still perceived as data only to be created and consumed by machines and robots, and not by humans as well. As a result Linked Data applications and tools often use the more traditional graph-based visualisations, that come with usability issues and are not familiar to the users, especially when associated with large and highly interconnected datasets. Thus, the web of data is still not commonly used by users without knowledge of Linked Data and previous experience with the related technologies. However, alternative visualisation techniques and user interaction approaches have been developed, studied, and evaluated in the literature since the beginning of the Semantic Web. Many of these less typical approaches have proven to be better solutions for inexperienced users regarding user interface usability.

One technique to formalise and document these user interaction techniques and best practices is in the form of patterns and pattern languages. While many pattern collections in the literature cover user interaction problems in more generic or specific contexts such as web, mobile, and social interfaces, there aren't works targeted at user interaction problems for Linked Data interfaces. We believe a pattern collection in this topic can support both experienced and inexperienced developers and designers in creating Linked Data applications that are intuitive and familiar to new users, motivating the use of the Semantic Web.

To collect and gain knowledge needed in the creation of the patterns, we started out by conducting a literature review on this topic to gather common approaches and solutions of user interaction design in Linked Data systems. We studied LD tools targeted at inexperienced users to comprehend which user interaction features, like visualisation techniques and search solutions, are more frequently provided. We can point out some recurring solutions, such as the use of faceted search interfaces and several data visualisation options when consulting a single or a group of resources. Surprisingly, graph-based visualisation was the least common approach encountered in the set of

works reviewed. This shows that researchers have started to deviate from the traditional visualisation techniques when developing for inexperienced users and that systematic documentation of these alternative solutions is necessary to broaden their use.

Next, we conducted an online questionnaire with members of the Linked Data community, to survey their experience with Linked Data, collect common problems while developing LD based applications, and to analyse their opinion on the usefulness of user interaction patterns for Linked Data. This survey allowed us to collect over 40 problems, with a clear focus on visualising data for the user interface problems reported. This study also proved the need for user interaction patterns, with over 90% of participants giving a positive rating to their usefulness.

Following the methods described in the literature on how to create a pattern language, we conducted pattern mining sessions from second-hand contributions, in the form of expert interviews. We executed 10 pattern mining interviews with Linked Data researchers and developers, whose contacts were collected with the help of the previously described online questionnaire. We consider these interviews as the most important phase in the development of the proposed pattern collection, as they allowed us to gather over 70 problems and solutions regarding user interfaces for Linked Data. The information collected proved very useful when deciding which problems to cover and which solutions to propose in the pattern collection.

The pattern collection proposed here is the combined result of the most common problems reported by members of the community in the conducted survey, the knowledge and experience gathered in the 10 pattern mining interviews with Linked Data researchers and developers, and the recurrent approaches collected through an extensive literature review of the solutions for user interaction with Linked Data. The proposed pattern collection is composed of 20 novel user interaction patterns for Linked Data, and is focused on four main aspects of user interaction, containing patterns concerning visualising, searching, browsing, and authoring problems. The patterns are presented using a traditional pattern format containing a context, problem, and solution, with the addition of a illustrative diagram and concrete examples, to improve their understanding. The proposed pattern collection allowed us to validate the defined hypothesis for this work by proving that the problems that arise when designing user interaction for Linked Data and their solutions can be abstracted in the form of patterns and converge into four different categories: visualising, searching, browsing, and authoring. The research questions outlined were also answered through the pattern collection. It encompasses the recurring problems for this topic, the techniques and strategies used to tacked them, and how they can be applied to a specific context. The answers for RQ1, RQ2, and RQ3 are presented, respectively, in the problem, solution, and examples section of each pattern.

Finally, to validate the proposed pattern collection we conducted a pattern classification method to identify and discuss its strong and weak points; a pattern usage survey to evaluate if the problems covered are commonly encountered, and if the solutions are considered good by the community; and an analysis of Wikidata's user interface, to identify which patterns are in use, which could be implemented, and to present a more thorough example of how the proposed collection can be applied. The pattern classification allowed us to identify an emphasis on patterns for viewing and

navigating, which is not surprising because visualisation and navigation features are crucial, regardless of domain, and pose new problems when Linked Data is used. We can also point out that patterns for editing and manipulating are lacking, what can be attributed to the difficulty in finding solutions that are accessible to all users. The results obtained in the pattern usage survey are positive overall, with 18 out of 20 patterns receiving a positive score, indicating a good reception of the patterns by the Linked Data community. This survey also allowed us to identify which patterns need revising and refining for future iterations. The analysis of Wikidata's user interface showed 11 out of the 20 patterns already present, and identified the remaining 9 as possible to be implemented. This analysis also allowed us to understand the importance of having knowledge about the vocabularies used on the applications, and about the most important information and frequent tasks for the users.

12.1 Main Contributions

The main contribution of this study is the proposed collection of 20 user interaction patterns for Linked Data. The topic of user interaction for Linked Data is one that is not frequently studied and that does not encompass any collection of documented guidelines or best practices. In this study we tackled this gap with the development of a pattern collection for this topic. Additionally, the conducted literature review on user interaction solutions for Linked Data is also a contribution of this work, as there is a lack of studies on this topic targeted at users with no previous experience of Linked Data technologies. We can also point out the information collected from the experience with Linked Data survey, the results of the pattern collection evaluation, and the findings of the Wikidata analysis as contributions of this work.

12.2 Future Work

The work presented here could be improved and continued in several ways. Firstly, additional evaluation sessions for the proposed pattern collection should be conducted, to understand why some patterns received less positive scores and how they can be improved. Increasing the number of participants for the evaluation sessions is also a concern to have in the future, in order to minimise any bias that might be present in the presented results. As discussed before, different formats of evaluation methods should also be taken into consideration in the future, to allow for the evaluation of the full version of the patterns. The participation in Writters' Workshops in PLoP conferences might also be useful to improve the pattern collection, as the opportunity to discuss the patterns with other experts in pattern writing is one of the best methods to revise and improve them.

Aside from future evaluation sessions to improve the proposed patterns, future work can also be outlined to expand the pattern collection. Additional pattern mining interviews can be conducted to gather even more solutions to user interaction problems that were not covered by the current version of the pattern collection, such as the ones already reported on the experience with

Linked Data survey and on the pattern usage survey. Further pattern mining sessions would also be beneficial to increase the number of developers and researchers involved, as to not have the patterns so highly influenced by the opinion of a small group of people.

The future goals for the proposed pattern collection include the improvement of the currently written patterns, and the addition of new ones to cover all the sub-problems encompassed in the design of user interaction for Linked Data, in order to come to a complete pattern language for this topic.

Appendix A

Experience with Linked Data Survey

The following pages describe the survey disseminated amongst Linked Data researchers and developers to assess their experience with Linked Data, detailed in Chapter 7. Section A.1 includes the original questionnaire.

A.1 Questionnaire

User interaction with Linked Data

This survey is carried out as an investigation tool in the context of a Master's dissertation on user interaction with Linked Data. The main goal is to create a set of design patterns to gather and document best practices in this topic.

Our goal is to collect information on the common problems of user interaction with Linked Data by asking the participants to share their experience with Linked Data/RDF/Semantic Web applications. Ultimately, we are looking for experienced users, developers or researchers, willing to participate in a more thorough interview about their experience and findings.

This survey is composed of a total of 15 questions (** 5min total **) divided into 3 sections. Section 1 comprises 7 questions about the participants' experience with Linked Data as a user and as a developer. Section 2, with 4 questions, surveys the participants' opinion on the importance and need fo design patterns for user interaction with Linked Data. Section 3, composed of 5 questions, aims to identify participants interest in participating in a more thorough interview.

Any questions or comments, please send them to:

Mariana Aguiar up201605904@fe.up.pt
Faculty of Engineering, University of Porto

Experience with Linked Data

Questions on your experience using or developing Linked Data applications

1. 1. Experience with Linked Data

Mark only one oval per row.

	0 - None	1 - Low	2	3	4	5 - High
Studying	◯	◯	◯	◯	◯	◯
Developing	◯	◯	◯	◯	◯	◯
Using	◯	◯	◯	◯	◯	◯

2. 2. Experience developing Linked Data applications

Mark only one oval per row.

	0 - None	1 - Low	2	3	4	5 - High
Modelling	◯	◯	◯	◯	◯	◯
User Interface	◯	◯	◯	◯	◯	◯
Backend	◯	◯	◯	◯	◯	◯

3. 3. What are some of the Linked Data projects/research studies you have been involved wit (using or developing)?

4. 4. As a user of Linked Data applications, how do you rate your usage frequency for each of the following tasks?

Mark only one oval per row.

	0 - None	1 - Low	2	3	4	5 - High
Visualising	◯	◯	◯	◯	◯	◯
Performing search tasks	◯	◯	◯	◯	◯	◯
Browsing and navigating	◯	◯	◯	◯	◯	◯
Creating and editing data	◯	◯	◯	◯	◯	◯

5. 5. As a user of Linked Data, how do you rate the frequency and importance of problems encountered for each of the following tasks?

Mark only one oval per row.

	0 - None	1 - Low	2	3	4	5 - High
Visualising	⬭	⬭	⬭	⬭	⬭	⬭
Performing search tasks	⬭	⬭	⬭	⬭	⬭	⬭
Browsing and navigating	⬭	⬭	⬭	⬭	⬭	⬭
Creating and editing data	⬭	⬭	⬭	⬭	⬭	⬭

6. 6. Which are some of the most common problems you have encountered when using Linke Data applications?

7. 7. Which Linked Data applications do you consider to be excellent examples in terms of supporting user interaction?

Importance of patterns

Questions on your opinion on the importance of design patterns for user interaction witl Linked Data.

8. 8. How do you rate the need for guidelines for designing user interaction with Linked Data

Mark only one oval.

	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	High

9. 9. Are you familiar with user interaction patterns or other pattern collections?

Mark only one oval.

- ◯ Yes
- ◯ No

10. 10. To what extent do you think patterns for user interaction with Linked Data would be useful for developers?

Mark only one oval.

	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	High

Contacts and comments

Questions on your interest in participating in an interview.
Collection of other contacts, mailing lists, projects, or studies in this topic

11. 11. Are you interested and available for a follow-up interview?

If Yes, don't forget to add your email below.

Mark only one oval.

- ◯ Yes
- ◯ No

12. 12. Name (only necessary if you are interested in participating in an interview)

13. 13. Email (only necessary if you are interested in participating in an interview)

14. 14. Do you have suggestions on other contacts, mailing lists, projects, or researchers that you think would be relevant to contact?

15. 15. Other comments?

This content is neither created nor endorsed by Google.

Google Forms

Appendix B

Pattern Mining Guide

The following pages describe the pattern mining interviews conducted with Linked Data researchers and developers, detailed in Chapter 8. Section B.1 includes the original interview guide.

B.1 Interview Guide

Interview Guide - Pattern Mining
User Interaction Patterns for Linked Data

Mariana Aguiar up201605904@fe.up.pt

March 2021

This interview guide was designed for the expert interviews conducted during the pattern mining phase to develop user interaction patterns for Linked Data. The interviews' goal is to gather and collect knowledge on researchers' and developers' experience with user interaction for Linked Data/RDF/Semantic Web applications.

In order to collect useful and clear information for the creation of patterns, we designed the questions to focus on the several pattern components: context, problem, solution, forces, and examples.

When possible, the interview should focus on the problems and corresponding solutions for inexperienced users. The end goal of the patterns is to promote the development of applications for everyone and consequently motivate the use of the Semantic Web.

The interview guide is composed of 3 parts and with an expected duration of 45 to 60 minutes. The first covers the participants' introductions, consent for recordings, interview and research context, and a brief overview of the data collected on the need and usefulness of patterns and guidelines for the topic at hand. The second part is composed of 6 main questions focused on the interviewee's experience with Linked Data as a user and as a developer and on the problems encountered. The third and final part of the interview is composed of 4 main questions covering the solutions for the problems mentioned previously; context, examples, and consequences of the solutions; and final tips and comments on the topic.

1 Introduction and Context

1. Introduction of the interviewer(s)
2. Introduction of the interviewee
3. Audio and video recording permission

4. Interview context
5. Presentation of preliminary survey results
    - Results on the need for guidelines for user interaction with Linked Data
    - Results on the usefulness of patterns for user interaction with Linked Data

2 Linked Data experience and user interaction problems

1. Can you talk a little about your opinion on the impact, future, preset, or past, of Linked Data on the web?
2. Can you tell me about how you started to work on or be interested in Linked Data?
    - How did that interest lead you to user interaction?
    - What are some of the projects or research studies that you have been involved with that you would like to highlight?
3. Do you use Linked Data based applications in your day to day life, for personal or work related tasks?
    - Can you describe your main interactions with these applications? What do you use them for?
4. Can you point out one or two user interaction problems in these applications?
    - Why do you think those approaches/features do not work?
    - What changes would you have implemented in that application if you had developed it?
5. Now in regards to your experience developing user interaction, do you usually have in mind inexperienced people (with no knowledge of Semantic Web technologies)?
    - How do you ensure that your applications will be easy to use by inexperienced users?
6. Can you point out some common problems you have faced when developing for inexperienced users? (or in general if interviewee has no experience developing for lay users)
    - From those problems you have mentioned, which two or three are the most important to you?

3 Solutions and examples

The following questions must be answered individually for each user interaction problem mentioned before. Depending on the reaming time, the interviewer should ask the following questions for one to three different problems highlighted by the interviewee.

1. What do you usually do to tackle them? What are your most recurrent approaches?
    - Can you recall other works or applications that use similar approaches or solutions?
        - If yes, what do these applications have in common with your own?
    - Can you think of other solutions to tackle the same problem?
        - If yes, what makes you choose one instead of the other?
2. Focusing again on your recurrent solution to a problem, can you describe the positive consequences of using that solution?
    - And negative?
3. Can you describe the negative consequences of not using that solution?
4. Do you have any general tips or commons for someone developing user interaction with Linked Data?

Appendix C

Patterns Problem Solution Summaries

Pattern	Problem	Solution	Category
TEMPLATE	When a resource's page presents a long list of visually similar triples containing information or related resources, users can feel overwhelmed at first glance and struggle to find the wanted information.	Define a set of the most important properties and related resources for each one of the resources in the data present in the application.	Visualising
MULTI-VISUALISATION	Finding and choosing the best visualisation technique for the users, that best suits the types of data and aids the user in their needs and tasks.	Provide the user with different types of visualisations for the data that is displayed.	
SHOW DEFINITIONS	Users cannot interpret some of the names of the resources and properties of the data, and thus find difficult to understand the presented information.	Provide definitions of all resources and related properties present in the data.	Visualising
GRAPH VISUALISATION	Present data using graph visualisations that are usable and understandable by non-technical users and that allow them to take full advantage of the data represented.	Use graph visualisations for subsets of the data where the edges of the graph all have the same or similar meanings.	Visualising
SEARCH RESULTS VISUALISATION	Provide users with an easy to understand list of search results, that aids their tasks and needs.	Define the types of search results for the data. Identify the most important information of each type. Present them in differentiable sections.	Visualising
RAW DATA	Allow the more experienced users to investigate the data further and deeper.	Provide the presented data in raw format.	Visualising
COMPLETE INFORMATION	When triples are presented individually, the users have difficulty in connecting the different information to be able to achieve the pieces of knowledge desired.	Aggregate small subsets of the data that form complete information and present them to the user.	Visualising
VISUAL INFORMATION	Display data to the user in a format that is easy to understand.	Map the data according to their type to visual information, using familiar visualisation techniques.	Visualising
COLLECTION OVERVIEW	When the users of an application are not experts, or don't have any previous knowledge on its domain, they can find difficult to understand the connections between the concepts or even the data that is presented to them. This can lead to more time spent on their tasks, or inability to complete them.	Provide a high-level overview of the data collection to the users.	Visualising
HIERARCHICAL TREE	When the number of classes in a data collection is large, and their hierarchical organisation is complex, users with little knowledge of the domain find difficult to quickly and effectively understand the position of a resource in the collection.	Provide the hierarchical tree of a resource on its individual page.	Visualising
KEYWORD QUERIES	Provide all users with a familiar and easy to use approach for searching the data.	Allow the users to perform search tasks using a keyword based approach.	Searching
SPARQL QUERIES	When experienced users, with previous knowledge of RDF and SPARQL, want to perform complex search tasks, sometimes they may find the traditional approaches insufficient.	Provide SPARQL experienced users with a SPARQL query editor.	Searching
ADVANCED SEARCH	Provide inexperienced users on RDF or SPARQL with methods that allow them to easily and quickly perform complex search tasks.	Additionally to a keyword based search, provide a set of advanced search options.	Searching
LINK BROWSING	Allow users to transverse the data and visit the pages of internal and external resources.	Provide hyperlinks for the related resources, to allow the users to navigate between them.	Browsing
BROWSE PATH	When navigating through Linked Data, the possible paths between resources are enormous. After following several hyperlinks and visiting many resources' pages, users lose track of where they started and the path that was taken.	Keep track of the user's path through the data and present it in a way that lets the user return to previous states or previously visited resources.	Browsing
FACETED BROWSING	Provide the users with a feature that allows the browsing of the data collection without a previous search query.	Provide the user with a faceted browsing feature, based on the resources' classes.	Browsing
MULTIPLE INPUT BOXES	When users with no previous knowledge of Linked Data or RDF want to annotate resources or create and edit new triples, they often have difficulty in composing valid and semantically correct data.	Provide clear text areas for the user to input each part of the triple or triple set.	Authoring
INPUT RECOMMENDATIONS	Even when individual input boxes with clear labels for each part of the triple to be edit or created are provided, users may sometimes have difficulty in choosing and entering valid options.	Provide a list of recommendations composed of only valid options for each one of the input boxes.	Authoring
INPUT EXAMPLES	When trying to create or edit data by filling out input boxes, users with less experience on the domain or with Linked Data encounter some difficulties in understanding what information should be inserted.	Provide examples of similar resources or triples when a user edits or creates data.	Authoring
DOWNLOAD DATA	Share the application's data with users that are interested in using it for their own projects.	Provide a download data button whenever a collection of data is presented to the user.	Authoring

Table C.1: Problems and solutions of the user interaction patterns for Linked Data.

Appendix D

Patterns Patlets

Pattern	Problem
TEMPLATE	When resources have a long list of properties and related resources, define a set of the most important ones and display them to the user in the resource's page. In addition, define an outline dictating how the properties and related resources must be displayed to the user, they should be divided in different sections according to the different kinds of information presented. Provide the remainder of the properties and resources in a collapsible section that the users can consult when needed.
MULTI-VISUALISATION	When resources or sets of resources have different types of data associated, provide the user with different types of visualisations for that data: tables, lists, maps, graphs, calendars, graphics, timelines. This will provide the users with the visualisation that best suits each type of data and a variety of user needs and tasks. In addition to whatever set of visualisations chosen, always provide a list or table to ensure accessibility of the data.
SHOW DEFINITIONS	When the resources' names are not understandable by humans or are unknown terms to not domain experts, provide definitions for all resources and related properties of your data. Provide the definitions for either internal vocabularies and external vocabularies, when possible.
GRAPH VISUALISATION	Only use graph visualisations for subsets of the data where the edges of the graph all have the same or similar meaning, like family trees and friend networks. If the entire graph is not understandable without zooming in or out, the subset might be too big to be visualised in a graph. Provide clear labels for the nodes and edges and minimise overlapping.
SEARCH RESULTS VISUALISATION	Define the items that constitute the types of search results for your data. Identify the most important information for each type. Present the search result items in a list in differentiable sections, each with the identified information. Allow the user to change the order in which the results are presented, based on the associated data, on numeric fields, or on how close or far they are to a given location. The option to customise the order to the results will allow the users to more quickly find what they are looking for, and better compare the results according to certain properties.
RAW DATA	When an application is targeted at users with some knowledge of Linked Data technologies, provide the presented data in raw format. This will allow the users to consult this information if needed, enabling them to explore the data in whichever direction and to validate their interpretations of it.
COMPLETE INFORMATION	When the information needed for a user task is a set of interconnected resources, previously aggregate those small subsets of data that form a complete piece of information and present it to the user.
VISUAL INFORMATION	Map the presented data to visual information according to their type, using familiar visualisation techniques. For numeric and textual data use lists and tables, for geographical data use maps, for temporal data use calendars and timelines, for statistical data use graphics. If your data represents concepts that are naturally related to themselves in a network format, like family trees or friend networks, use graph visualisations.
COLLECTION OVERVIEW	Provide a high-level overview of the data collection to the users, when the application is used by people with little to none experience on the domain. This will lead to a better understanding of the data, and consequently less time spent on tasks. When the application is not restricted to a domain, providing an understandable overview of the collection structure can be difficult. Opt for an alternative like resources' hierarchical trees instead.
HIERARCHICAL TREE	When the data collection is composed of a large number of classes and subclasses, in each resource's page provides its hierarchical tree. This will help the user situate the resource in the collection and gather more understanding of its overall structure.
KEYWORD QUERIES	When an application wants to provide all users with a familiar and easy to use approach to search the data collection, allow the users to perform search tasks using a keyword based approach. Due to the fact that keyword based queries are the most common search method, they should be provided in addition to more alternative and novel approaches, in order to be accessible to all users.
SPARQL QUERIES	When an application provides its users with search features and already implements keyword queries or a similar simple search approach, provide a SPARQL query editor. This will allow the more experienced users to perform complex search tasks without any restriction that may be imposed by the more traditional approaches. It should also be possible to view the corresponding SPARQL query for search tasks constructed with a keyword approach and additional advanced options.
ADVANCED SEARCH	When an application provides its users with search features and already implements keyword queries or a similar simple search approach, provide a set of advanced search options. These options should be composed of various filters on different properties and characteristics of the data, tailored to the application domain. Make sure to match the filter to the corresponding type of data, using checkboxes to filter the classes of the results, dropdown lists when the field has a fixed set of options, sliders for filters on numeric fields, calendars to select dates or time intervals, and maps with area selection features to filter resources with geographical data.
LINK BROWSING	When an application wants to provide users with means to browse the data, allow users to traverse it following links and visiting the pages of internal and external resources. When designing a resource's page include the names of any related resources as hyperlinks to the corresponding pages.
BROWSE PATH	When browsing through Linked Data following links or with a similar approach, the possible paths between resources are enormous, and users may lose track of where they started and the path that was taken. Keep track of the users' path through the data and present it in a way that lets the user return to previous states or previously visited resources.
FACETED BROWSING	More traditional and familiar browsing approaches require an initial search query. When an application is targeted at users with little to no domain knowledge this can pose a challenge to them. Additionally to other browsing approaches, provide the user with a faceted browsing feature, based on the resources' classes. It should allow the user to filter the full list of resources in the application, according to their classes and the classes of the related resources. This approach will allow the users to start browsing without any previous knowledge and at the same time will contribute to their understanding of the structure of the collection.
MULTIPLE INPUT BOXES	When an application wants to provide users with no knowledge of Linked Data technologies with methods that allow them to compose valid and semantically correct data, provide clear text areas for the user to input each part of the triple or triple set. The text areas should be provided for each component of the triples, subject, predicate, and object, and only accept valid options.
INPUT RECOMMENDATIONS	When an application provides inexperienced users with features to author data and implements an approach like multiple input boxes, provide a list of recommendations composed of only valid options for each one of the input boxes. A limited number of recommendations should be presented to the user in a dropdown list and it should be filtered as the user types in data. The initial limited list of recommendations can be composed of the most relevant resources or properties for the triple at hand, or the previously used ones by the user. A full list of the valid options should also be provided when needed.
INPUT EXAMPLES	When an application provides authoring features to users with no technical experience with RDF or Linked Data, they may encounter some difficulties in understanding what information should be inserted. Provide examples of similar resources or triples when a user edits or creates data. These examples can be composed of illustrative values tailored to the domain and to the resource or triple to be created or edited.
DOWNLOAD DATA	When an application's data is meant to be reused by other developers, researchers, or users, provide a download data button whenever a set of data is presented to the user. This download button can be added in scenarios like the following: the resources' pages, the search result lists, graphs, maps, tables, calendars, or other sets of related resources. If an application would like to give the users the possibility to download the entirety of the data collection, provide a visually different download button available in every page of the application.

Table D.1: Pattern collection patlets.

Appendix E

Pattern Usage Survey

The following pages describe the survey disseminated amongst Linked Data researchers and developers to assess the proposed user interaction patterns for Linked Data, detailed in Chapter 10. Section E.1 includes the original questionnaire.

E.1 Questionnaire

User Interaction Patterns for Linked Data - Adoption Survey

This survey is carried out as an investigation tool in the context of a Master's dissertation on user interaction with Linked Data. The main goal is to create a set of user interaction patterns to gather and document best practices in this topic.

After collecting experience of users, developers, and researchers with Linked Data, and conducting pattern mining interviews to gather information on the common problems of user interaction, we wrote a small collection of user interaction patterns with Linked Data. Our goal is to survey the adoption of these patterns and thoughts and opinions on them.

This survey is composed of a total of 24 questions (**20/25 min total**) divided into 4 sections.

Section 1 — 11 questions about visualising patterns.
Section 2 — 4 questions on searching patterns.
Section 3 — 4 questions about browsing patterns.
Section 4 — 5 questions on authoring patterns.

Any questions or comments, please send them to:

Mariana Aguiar up201605904@edu.fe.up.pt
Faculty of Engineering, University of Porto

Visualising

Questions on visualising user interaction patterns

1. 1. Template

When resources have a long list of properties and related resources, define a set of the most important ones and disp them to the user in the resource's page. In addition, define an outline dictating how the properties and related resourc must be displayed to the user, they should be divided in different sections according to the different kinds of informa presented. Provide the remainder of the properties and resources in a collapsible section that the users can consult when needed.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

2. 2. Multi-Visualisation

When resources or sets of resources have different types of data associated, provide the user with different types of visualisations for that data: tables, lists, maps, graphs, calendars, graphics, timelines. This will provide the users with the visualisation that best suits each type of data and a variety of user needs and tasks. In addition to whatever set o visualisations chosen, always provide a list or table to ensure accessibility of the data.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

3. 3. Show Definitions

When the resources' names are not understandable by humans or are unknown terms to not domain experts, provide definitions for all resources and related properties of your data. Provide the definitions for either internal vocabularies and external vocabularies, when possible.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

4. 4. Graph Visualisation

Only use graph visualisations for subsets of the data where the edges of the graph all have the same or similar mean like family trees and friend networks. If the entire graph is not understandable without zooming in or out, the subset might be too big to be visualised in a graph. Provide clear labels for the nodes and edges and minimise overlapping.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

5. 5. Search Results Visualisation

Define the items that constitute the types of search results for your data. Identify the most important information for each type. Present the search result items in a list in differentiable sections, each with the identified information. Allo the user to change the order in which the results are presented, based on the associated data, on numeric fields, or o how close or far they are to a given location. The option to customise the order to the results will allow the users to more quickly find what they are looking for, and better compare the results according to certain properties.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

6. 6. Raw Data

When an application is targeted at users with some knowledge of Linked Data technologies, provide the presented da in raw format. This will allow the users to consult this information if needed, enabling them to explore the data in whichever direction and to validate their interpretations of it.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

7. 7. Complete Information

When the information needed for a user task is a set of interconnected resources, previously aggregate those small subsets of data that form a complete piece of information and present it to the user.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

8. 8. Visual Information

Map the presented data to visual information according to their type, using familiar visualisation techniques. For numeric and textual data use lists and tables, for geographical data use maps, for temporal data use calendars and timelines, for statistical data use graphics. If your data represents concepts that are naturally related to themselves i network format, like family trees or friend networks, use graph visualisations.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

9. 9. Collection Overview

Provide a high-level overview of the data collection to the users, when the application is used by people with little to none experience on the domain. This will lead to a better understanding of the data, and consequently less time spen on tasks. When the application is not restricted to a domain, providing an understandable overview of the collection structure can be difficult. Opt for an alternative like resources' hierarchical trees instead.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

10. 10. Hierarchical Tree

When the data collection is composed of a large number of classes and subclasses, in each resource's page provid its hierarchical tree. This will help the user situate the resource in the collection and gather more understanding of i overall structure.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

11. 11. Notes and comments - Visualisation patterns

_____

_____

_____

_____

_____

Searching

Questions on searching user interaction patterns

12. 1. Keyword Queries

When an application wants to provide all users with a familiar and easy to use approach to search the data collectic allow the users to perform search tasks using a keyword based approach. Due to the fact that keyword based queri are the most common search method, they should be provided in addition to more alternative and novel approaches order to be accessible to all users.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

13. 2. SPARQL Queries

When an application provides its users with search features and already implements keyword queries or a similar simple search approach, provide a SPARQL query editor. This will allow the more experienced users to perform complex search tasks without any restriction that may be imposed by the more traditional approaches. It should als be possible to view the corresponding SPARQL query for search tasks constructed with a keyword approach and additional advanced options.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

14. 3. Advanced Search

When an application provides its users with search features and already implements keyword queries or a similar simple search approach, provide a set of advanced search options. These options should be composed of various filters on different properties and characteristics of the data, tailored to the application domain. Make sure to matcl the filter to the corresponding type of data, using checkboxes to filter the classes of the results, dropdown lists whe the field has a fixed set of options, sliders for filters on numeric fields, calendars to select dates or time intervals, ar maps with area selection features to filter resources with geographical data.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

15. 4. Notes and comments - Searching Patterns

Browsing

Questions on browsing user interaction patterns

16. 1. Link Browsing

When an application wants to provide users with means to browse the data, allow users to traverse it following links and visiting the pages of internal and external resources. When designing a resource's page include the names of a related resources as hyperlinks to the corresponding pages.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

17. 2. Browse Path

When browsing through Linked Data following links or with a similar approach, the possible paths between resource are enormous, and users may lose track of where they started and the path that was taken. Keep track of the users' path through the data and present it in a way that lets the user return to previous states or previously visited resour

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

18. 3. Faceted Browsing

More traditional and familiar browsing approaches require an initial search query. When an application is targeted a users with little to no domain knowledge this can pose a challenge to them. Additionally to other browsing approacl provide the user with a faceted browsing feature, based on the resources' classes. It should allow the user to filter t full list of resources in the application, according to their classes and the classes of the related resources. This approach will allow the users to start browsing without any previous knowledge and at the same time will contribut their understanding of the structure of the collection.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

19. 4. Notes and comments - Browsing Patterns

Authoring

Questions on authoring user interaction patterns

20. 1. Multiple Input Boxes

When an application wants to provide users with no knowledge of Linked Data technologies with methods that allov them to compose valid and semantically correct data, provide clear text areas for the user to input each part of the triple or triple set. The text areas should be provided for each component of the triples, subject, predicate, and obje and only accept valid options.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

21. 2. Input Recommendations

When an application provides inexperienced users with features to author data and implements an approach like multiple input boxes, provide a list of recommendations composed of only valid options for each one of the input boxes. A limited number of recommendations should be presented to the user in a dropdown list and it should be filtered as the user types in data. The initial limited list of recommendations can be composed of the most relevant resources or properties for the triple at hand, or the previously used ones by the user. A full list of the valid options should also be provided when needed.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

22. 3. Input Examples

When an application provides authoring features to users with no technical experience with RDF or Linked Data, the may encounter some difficulties in understanding what information should be inserted. Provide examples of similar resources or triples when a user edits or creates data. These examples can be composed of illustrative values tailo to the domain and to the resource or triple to be created or edited.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

23. 4. Download Data

When an application's data is meant to be reused by other developers, researchers, or users, provide a download da button whenever a set of data is presented to the user. This download button can be added in scenarios like the following: the resources' pages, the search result lists, graphs, maps, tables, calendars, or other sets of related resources. If an application would like to give the users the possibility to download the entirety of the data collectio provide a visually different download button available in every page of the application.

Mark only one oval.

- ◯ I have already used this pattern structure exactly as it is documented.
- ◯ I have already used this pattern structure with some variations.
- ◯ I have not used this pattern structure yet because I did not need it.
- ◯ I have not used this pattern structure yet and I think it is not a good solution.
- ◯ I have not used this pattern structure yet and I think it could help to solve some problems that I faced.
- ◯ I am not sure if I used this pattern structure and I think it is a good solution.
- ◯ I am not sure if I used this pattern structure and I think it is a bad solution.

24. 5. Notes and comments - Authoring Patterns

This content is neither created nor endorsed by Google.

Google Forms

References

[1] Yuma Akado, Sakurako Kogure, Alice Sasabe, Jei-Hee Hong, Keishi Saruwatari, and Takashi Iba. Five patterns for designing pattern mining workshops. In Claudius Link and Veli-Pekka Eloranta, editors, *Proceedings of the 20th European Conference on Pattern Languages of Programs, EuroPLoP 2015, Kaufbeuren, Germany, July 8-12, 2015*, pages 9:1–9:13. ACM, 2015. doi: 10.1145/2855321.2855331. URL https://doi.org/10.1145/2855321.2855331.

[2] Christopher Alexander. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, USA, 1977. ISBN 0-19-501919-9. URL https://archive.org/details/patternlanguage00chri/page/1216.

[3] Sören Auer, Sebastian Dietzold, and Thomas Riechert. Ontowiki - A tool for social, semantic collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, 2006. doi: 10.1007/11926078_53. URL https://doi.org/10.1007/11926078_53.

[4] Bob Baxley. Universal model of a user interface. In *Proceedings of the 2003 Conference on Designing for User Experiences*, DUX '03, page 1–14, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581137281. doi: 10.1145/997078.997090. URL https://doi.org/10.1145/997078.997090.

[5] Tim Berners-Lee. Linked data - design issues. Available at: https://www.w3.org/DesignIssues/LinkedData.html, Accessed last time in January 2021, July 2006.

[6] Tim Berners-Lee and Robert Cailliau. Worldwideweb: Proposal for a hypertext project. Available at: https://cds.cern.ch/record/2639699/files/Proposal_Nov-1990.pdf, Accessed last time in January 2021, May 1990.

[7] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, pages 28–37, May 2001. ISSN 00368733. doi: 10.1038/scientificamerican0501-34.

[8] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connoll, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction*, 2006. URL http://swui.semanticweb.org/swui06/papers/Berners-Lee/Berners-Lee.pdf.

[9] Nikos Bikakis and Timos K. Sellis. Exploration and visualization in the web of big linked data: A survey of the state of the art. In Themis Palpanas and Kostas Stefanidis, editors, *Proceedings of the Workshops of the EDBT/ICDT 2016 Joint Conference, EDBT/ICDT Workshops 2016, Bordeaux, France, March 15, 2016*, volume 1558 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL http://ceur-ws.org/Vol-1558/paper28.pdf.

[10] Jan O. Borchers. A pattern approach to interaction design. In Daniel Boyarski and Wendy A. Kellogg, editors, *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, Techniques, DIS 2000, New York City, NY, USA, August 17-19, 2000*, pages 369–378. ACM, 2000. doi: 10.1145/347642.347795. URL https://doi.org/10.1145/347642.347795.

[11] Dan Brickley and R.V. Guha. RDF schema 1.1. Available at: https://www.w3.org/TR/rdf-schema/, Accessed last time in January 2021, February 2014.

[12] Grégoire Burel, Amparo Elizabeth Cano, and Vitaveska Lanfranchi. Ozone browser: Augmenting the web with semantic overlays. In Sören Auer, Christian Bizer, and Gunnar Aastrand Grimnes, editors, *Proceedings of the Fifth Workshop on Scripting and Development for the Semantic Web , co-located with 6th European Semantic Web Conference, SFSW@ESWC 2009, Heraklion, Greece, May 31, 2009*, volume 449 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009. URL http://ceur-ws.org/Vol-449/Challenge1.pdf.

[13] Silvana Castano, Alfio Ferrara, and Stefano Montanelli. inwalk: Interactive and thematic walks inside the web of data. In Sihem Amer-Yahia, Vassilis Christophides, Anastasios Kementsietsidis, Minos N. Garofalakis, Stratos Idreos, and Vincent Leroy, editors, *Proceedings of the 17th International Conference on Extending Database Technology, EDBT 2014, Athens, Greece, March 24-28, 2014*, pages 628–631. OpenProceedings.org, 2014. doi: 10.5441/002/edbt.2014.60. URL https://doi.org/10.5441/002/edbt.2014.60.

[14] James O. Coplien and Neil B. Harrison. *Organizational Patterns of Agile Software Development*. Prentice-Hall, Inc., USA, 2004. ISBN 0131467409.

[15] Christian Crumlish and Erin Malone. *Designing Social Interfaces - Principles, Patterns, and Practices for Improving the User Experience*. O'Reilly, 2009. ISBN 978-0-596-15492-9. URL http://www.oreilly.de/catalog/9780596154929/index.html.

[16] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011. doi: 10.3233/SW-2011-0037. URL https://doi.org/10.3233/SW-2011-0037.

[17] John Davies and Richard Weeks. Quizrdf: Search technology for the semantic web. In *37th Hawaii International Conference on System Sciences (HICSS-37 2004), CD-ROM / Abstracts Proceedings, 5-8 January 2004, Big Island, HI, USA*. IEEE Computer Society, 2004. doi: 10.1109/HICSS.2004.1265293. URL https://doi.org/10.1109/HICSS.2004.1265293.

[18] Stephen Davies, Jesse Hatfield, Chris Donaher, and Jessica Zeitz. User interface design considerations for linked data authoring environments. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *Proceedings of the WWW2010 Workshop on Linked Data on the Web, LDOW 2010, Raleigh, USA, April 27, 2010*, volume 628 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010. URL http://ceur-ws.org/Vol-628/ldow2010_paper17.pdf.

[19] David E. Delano. *Patterns Mining*. Cambridge University Press, USA, 1998. ISBN 0521648181.

[20] Federico Desimoni and Laura Po. Empirical evaluation of linked data visualization tools. *Future Gener. Comput. Syst.*, 112:258–282, 2020. doi: 10.1016/j.future.2020.05.038. URL https://doi.org/10.1016/j.future.2020.05.038.

[21] Leigh Dodds and Ian Davis. Linked data patterns: A pattern catalogue for modelling, publishing, and consuming linked data. Available at: https://patterns.dataincubator.org/book/, Accessed last time in January 2021, May 2012.

[22] Department for Environment Food & Rural Affairs. Hydrology data explorer. Available at: http://environment.data.gov.uk/hydrology/landing, Accessed last time in June 2021, 2021.

[23] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1 edition, 1994. ISBN 0201633612. URL http://www.amazon.com/Design-Patterns-Elements-Reusable-Object-Oriented/dp/0201633612/ref=ntt_at_ep_dpi_1.

[24] Nicholas Gibbins, Stephen Harris, and Monica Schraefel. Applying mspace interfaces to the semantic web. In *World Wide Web Conference 2004 (01/01/04)*, 2003. URL https://eprints.soton.ac.uk/258639/.

[25] Hugh Glaser, Harith Alani, Les Carr, Sam Chapman, Fabio Ciravegna, Alexiei Dingli, Nicholas Gibbins, Stephen Harris, Monica M. C. Schraefel, and Nigel Shadbolt. CS aktivespace: Building a semantic web application. In Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *The Semantic Web: Research and Applications, First European Semantic Web Symposium, ESWS 2004, Heraklion, Crete, Greece, May 10-12, 2004, Proceedings*, volume 3053 of *Lecture Notes in Computer Science*, pages 417–432. Springer, 2004. doi: 10.1007/978-3-540-25956-5_29. URL https://doi.org/10.1007/978-3-540-25956-5_29.

[26] UK Government. Regulated food and feed products for great britain. Available at: https://data.food.gov.uk/regulated-products/, Accessed last time in June 2021, 2021.

[27] UK Government. Hm land registry open data. Available at: https://landregistry.data.gov.uk, Accessed last time in June 2021, 2021.

[28] Ontology Engineering Group. Portal de datos bibliográficos de la biblioteca nacional de españa. Available at: https://datos.bne.es/inicio.html, Accessed last time in June 2021, 2021.

[29] RDF Working Group. Resource description framework (RDF). Available at: https://www.w3.org/RDF/, Accessed last time in January 2021, February 2014.

[30] The W3C SPARQL Working Group. SPARQL 1.1 overview. Available at: https://www.w3.org/TR/sparql11-overview/, Accessed last time in January 2021, March 2008.

[31] Eduardo M. Guerra and Clovis T. Fernandes. An evaluation process for pattern languages. In Rebecca Wirfs-Brock and Uirá Kulesza, editors, *Proceedings of the 8th Latin American Conference on Pattern Languages of Programs, SugarLoafPLoP 2010, Salvador, Bahia, Brazil, September 23-27, 2010*, pages 18:1–18:11. ACM, 2010. doi: 10.1145/2581507.2581525. URL https://doi.org/10.1145/2581507.2581525.

[32] Greg Guest, Kathleen MacQueen, and Emily Namey. *Applied Thematic Analysis*. Thousand Oaks, California, 2012. URL https://www.doi.org/10.4135/9781483384436.

[33] Florian Haag, Steffen Lohmann, Steffen Bold, and Thomas Ertl. Visual SPARQL querying based on extended filter/flow graphs. In Paolo Paolini and Franca Garzotto, editors, *International Working Conference on Advanced Visual Interfaces, AVI 2014, Como, Italy, May 27-29, 2014*, pages 305–312. ACM, 2014. doi: 10.1145/2598153.2598185. URL https://doi.org/10.1145/2598153.2598185.

[34] Peter Haase, Michael Schmidt, and Andreas Schwarte. The information workbench as a self-service platform for linked data applications. In Olaf Hartig, Andreas Harth, and Juan F. Sequeda, editors, *Proceedings of the Second International Workshop on Consuming Linked Data (COLD2011), Bonn, Germany, October 23, 2011*, volume 782 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011. URL http://ceur-ws.org/Vol-782/HaaseEtAl_COLD2011.pdf.

[35] Neil B. Harrison. The language of shepherding: A pattern language for shepherds and sheep. *Pattern Languages of Program Design*, 5:507–530, 1999.

[36] Pejam Hassanzadeh, Eero Hyvönen, Esko Ikkala, Jouni Tuominen, Suzie Thomas, Anna Wessman, and Ville Rohiola. Findsampo platform for reporting and studying archaeological finds using citizen science. In Alessandro Adamou, Enrico Daga, and Albert Meroño-Peñuela, editors, *Proceedings of the Third Workshop on Humanities in the Semantic Web (WHiSe 2020) co-located with 15th Extended Semantic Web Conference (ESWC 2020), Heraklion, Greece, June 2, 2020 (online)*, volume 2695 of *CEUR Workshop Proceedings*, pages 33–40. CEUR-WS.org, 2020. URL http://ceur-ws.org/Vol-2695/paper4.pdf.

[37] Marc Hassenzahl and Noam Tractinsky. User experience - a research agenda. *Behav. Inf. Technol.*, 25(2):91–97, 2006. doi: 10.1080/01449290500330331. URL https://doi.org/10.1080/01449290500330331.

[38] Marti Hearst. Design recommendations for hierarchical faceted search interfaces. In *Proc. SIGIR 2006, Workshop on Faceted Search*, pages 26–30, August 2006.

[39] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011. doi: 10.2200/S00334ED1V01Y201102WBE001. URL https://doi.org/10.2200/S00334ED1V01Y201102WBE001.

[40] Philipp Heim, Sebastian Hellmann, Jens Lehmann, Steffen Lohmann, and Timo Stegemann. Relfinder: Revealing relationships in RDF knowledge bases. In Tat-Seng Chua, Yiannis Kompatsiaris, Bernard Mérialdo, Werner Haas, Georg Thallinger, and Werner Bailer, editors, *Semantic Multimedia, 4th International Conference on Semantic and Digital Media Technologies, SAMT 2009, Graz, Austria, December 2-4, 2009, Proceedings*, volume 5887 of *Lecture Notes in Computer Science*, pages 182–187. Springer, 2009. doi: 10.1007/

978-3-642-10543-2_21. URL https://doi.org/10.1007/978-3-642-10543-2_21.

[41] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph. OWL 2 web ontology language primer (second edition). Available at: https://www.w3.org/TR/2012/REC-owl2-primer-20121211/, Accessed last time in January 2021, December 2012.

[42] Christian Hübscher, Stefan L. Pauwels, Sandra P. Roth, Javier A. Bargas-Avila, and Klaus Opwis. The organization of interaction design pattern languages alongside the design process. *Interact. Comput.*, 23(3):189–201, 2011. doi: 10.1016/j.intcom.2011.02.009. URL https://doi.org/10.1016/j.intcom.2011.02.009.

[43] David Huynh, Stefano Mazzocchi, and David R. Karger. Piggy bank: Experience the semantic web inside your web browser. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings*, volume 3729 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 2005. doi: 10.1007/11574620_31. URL https://doi.org/10.1007/11574620_31.

[44] Eero Hyvönen, Eetu Mäkelä, Mirva Salminen, Arttu Valo, Kim Viljanen, Samppa Saarela, Miikka Junnila, and Suvi Kettula. Museumfinland - finnish museums on the semantic web. *J. Web Semant.*, 3(2-3):224–241, 2005. doi: 10.1016/j.websem.2005.05.008. URL https://doi.org/10.1016/j.websem.2005.05.008.

[45] Takashi Iba and Taichi Isaku. Holistic pattern-mining patterns a pattern language for pattern mining on a holistic approach. In *Proceedings of the 19th Conference on Pattern Languages of Programs, PLoP 2012*, 2012.

[46] Esko Ikkala, Eero Hyvönen, Heikki Rantala, and Mikko Koho. Sampo-ui: A full stack javascript framework for developing semantic portal user interfaces. *Semantic Web–Interoperability, Usability, Applicability*, 2020.

[47] Akrivi Katifori, Constantin Halatsis, George Lepouras, Costas Vassilakis, and Eugenia G. Giannopoulou. Ontology visualization methods - a survey. *ACM Comput. Surv.*, 39(4):10, 2007. doi: 10.1145/1287620.1287621. URL https://doi.org/10.1145/1287620.1287621.

[48] Carsten Keßler, Mathieu d'Aquin, and Stefan Dietze. Linked data for science and education. *Semant. Web*, 4(1):1–2, January 2013. ISSN 1570-0844.

[49] Jakub Klímek, Petr Skoda, and Martin Necaský. Survey of tools for linked data consumption. *Semantic Web*, 10(4):665–720, 2019. doi: 10.3233/SW-180316. URL https://doi.org/10.3233/SW-180316.

[50] Sari A. Laakso. User interface design patterns. Available at: https://www.cs.helsinki.fi/u/salaakso/patterns/, Accessed last time in January 2021, September 2003.

[51] Bongshin Lee, Cynthia Sims Parr, Catherine Plaisant, Benjamin B. Bederson, Vladislav Daniel Veksler, Wayne D. Gray, and Christopher Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Trans. Vis. Comput. Graph.*, 12(6):

1414–1426, 2006. doi: 10.1109/TVCG.2006.106. URL https://doi.org/10.1109/TVCG.2006.106.

[52] Petri Leskinen and Eero Hyvönen. Linked open data service about historical finnish academic people in 1640-1899. In Sanita Reinsone, Inguna Skadina, Anda Baklane, and Janis Daugavietis, editors, *Proceedings of the Digital Humanities in the Nordic Countries 5th Conference, Riga, Latvia, October 21-23, 2020*, volume 2612 of *CEUR Workshop Proceedings*, pages 284–292. CEUR-WS.org, 2020. URL http://ceur-ws.org/Vol-2612/short14.pdf.

[53] Nicolas Marie and Fabien L. Gandon. Survey of linked data based exploration systems. In Dhavalkumar Thakker, Daniel Schwabe, Kouji Kozaki, Roberto García, Chris Dijkshoorn, and Riichiro Mizoguchi, editors, *Proceedings of the 3rd International Workshop on Intelligent Exploration of Semantic Data (IESD 2014) co-located with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Italy, October 20, 2014*, volume 1279 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014. URL http://ceur-ws.org/Vol-1279/iesd14_8.pdf.

[54] Gerard Meszaros and Jim Doble. A pattern language for pattern writing. *Pattern Languages of Program Design*, 3:529–574, 1997.

[55] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., USA, 2002. ISBN 9780465067107.

[56] Tim O'Reilly. What is web 2.0. Available at: https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html, Accessed last time in February 2021, September 2005.

[57] Kara Pernice. User interviews: How, when, and why to conduct them. Available at: https://www.nngroup.com/articles/user-interviews/, Accessed last time in May 2021, October 2018.

[58] Laura Po, Nikos Bikakis, Federico Desimoni, and George Papastefanatos. *Linked Data Visualization: Techniques, Tools, and Big Data*. Morgan & Claypool Publishers LLC, March 2020. doi: 10.2200/s00967ed1v01y201911wbe019.

[59] Eric Prud'hommeaux and Andy Seaborne. SPARQL query language for rdf. Available at: https://www.w3.org/TR/rdf-sparql-query/, Accessed last time in January 2021, January 2008.

[60] Dennis Quan, David Huynh, and David R. Karger. Haystack: A platform for authoring end user semantic web applications. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, volume 2870 of *Lecture Notes in Computer Science*, pages 738–753. Springer, 2003. doi: 10.1007/978-3-540-39718-2_47. URL https://doi.org/10.1007/978-3-540-39718-2_47.

[61] Heikki Rantala, Ilkka Jokipii, Mikko Koho, Esko Ikkala, Jouni Tuominen, and Eero Hyvönen. Building a linked open data portal of war victims in finland 1914-1922. In Sanita Reinsone, Inguna Skadina, Anda Baklane, and Janis Daugavietis, editors, *Proceedings of the Digital Humanities in the Nordic Countries 5th Conference, Riga, Latvia, October 21-23, 2020*, volume 2612 of *CEUR Workshop Proceedings*, pages 310–317. CEUR-WS.org, 2020. URL http://ceur-ws.org/Vol-2612/short17.pdf.

[62] Maria Rosala. How to analyze qualitative data from UX research: Thematic analysis. Available at: https://www.nngroup.com/articles/thematic-analysis/, Accessed last time in May 2021, September 2019.

[63] Maria Rosala. Writing an effective guide for a UX interview. Available at: https://www.nngroup.com/articles/interview-guide/, Accessed last time in May 2021, February 2021.

[64] Alessio De Santo and Adrian Holzer. Interacting with linked data: A survey from the SIGCHI perspective. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, Honolulu, HI, USA, April 25-30, 2020*, pages 1–12. ACM, 2020. doi: 10.1145/3334480.3382909. URL https://doi.org/10.1145/3334480.3382909.

[65] Alice Sasabe, Tomoki Kaneko, Kaho Takahashi, and Takashi Iba. Pattern mining patterns: A search for the seeds of patterns. In *Proceedings of the 23rd Conference on Pattern Languages of Programs*, PLoP '16, USA, 2016. The Hillside Group.

[66] Guus Schreiber and Yves Raimond. RDF 1.1 primer. Available at: https://www.w3.org/TR/rdf11-primer/, Accessed last time in January 2021, June 2014.

[67] Bill Scott and Theresa Neil. *Designing Web Interfaces - Principles and Patterns for Rich Interactions*. O'Reilly, 2009. ISBN 978-0-596-51625-3. URL http://www.oreilly.de/catalog/9780596516253/index.html.

[68] Ahmed Seffah. The evolution of design patterns in hci: From pattern languages to pattern-oriented design. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems*, PEICS '10, page 4–9, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450302463. doi: 10.1145/1824749.1824751. URL https://doi.org/10.1145/1824749.1824751.

[69] Niels Seidel. Making web video accessible: interaction design patterns for assistive video learning environments. In Claudius Link and Veli-Pekka Eloranta, editors, *Proceedings of the 20th European Conference on Pattern Languages of Programs, EuroPLoP 2015, Kaufbeuren, Germany, July 8-12, 2015*, pages 17:1–17:16. ACM, 2015. doi: 10.1145/2855321.2855339. URL https://doi.org/10.1145/2855321.2855339.

[70] Niels Seidel. Empirical evaluation methods for pattern languages: Sketches, classification, and network analysis. In *Proceedings of the 22nd European Conference on Pattern Languages of Programs, EuroPLoP 2017, Irsee, Germany, July 12-16, 2017*, pages 13:1–13:24. ACM, 2017. doi: 10.1145/3147704.3147719. URL https://doi.org/10.1145/3147704.3147719.

[71] Nigel Shadbolt and Kieron O'Hara. Linked data in government. *IEEE Internet Computing*, 17(4):72–77, July 2013. ISSN 1089-7801. doi: 10.1109/MIC.2013.72. URL https://doi.org/10.1109/MIC.2013.72.

[72] STLab. Odp reactor. Available at: https://arco.istc.cnr.it/test-odpreactor/datasets, Accessed last time in June 2021, 2021.

[73] Jenifer Tidwell. *Designing interfaces - patterns for effective interaction design (2. ed.)*. O'Reilly, 2011. ISBN 978-1-449-37970-4. URL http://www.oreilly.de/catalog/9781449379704/index.html.

[74] Anders Toxboe. User interface design patterns. Available at: http://ui-patterns.com, Accessed last time in January 2021, 2021.

[75] Giovanni Tummarello, Christian Morbidoni, and Michele Nucci. Enabling semantic web communities with dbin: An overview. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 of *Lecture Notes in Computer Science*, pages 943–950. Springer, 2006. doi: 10.1007/11926078_69. URL https://doi.org/10.1007/11926078_69.

[76] Martijn van Welie and Hallvard Traetteberg. Interaction patterns in user interfaces. In *7th Pattern Languages of Programs Conference (PLoP 2000).*, 2000. PLoP 2000 ; Conference date: 01-01-2000 Through 01-01-2000.

[77] Martijn van Welie and Gerrit C. van der Veer. Pattern languages in interaction design. In Matthias Rauterberg, Marino Menozzi, and Janet Wesson, editors, *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*. IOS Press, 2003.

[78] Guillermo Vega-Gorgojo, Martin Giese, Simen Heggestøyl, Ahmet Soylu, and Arild Waaler. Pepesearch: Semantic data for the masses. *PLOS ONE*, 11(3):1–12, 03 2016. doi: 10.1371/journal.pone.0151573. URL https://doi.org/10.1371/journal.pone.0151573.

[79] Guillermo Vega-Gorgojo, Laura Slaughter, Bjørn Marius von Zernichow, Nikolay Nikolov, and Dumitru Roman. Linked data exploration with RDF surveyor. *IEEE Access*, 7:172199–172213, 2019. doi: 10.1109/ACCESS.2019.2956345. URL https://doi.org/10.1109/ACCESS.2019.2956345.

[80] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, 2014. doi: 10.1145/2629489. URL https://doi.org/10.1145/2629489.

[81] Margaret Warren and Patrick J. Hayes. Bounding ambiguity: Experiences with an image annotation system. In Lora Aroyo, Anca Dumitrache, Praveen Paritosh, Alexander J. Quinn, Chris Welty, Alessandro Checco, Gianluca Demartini, Ujwal Gadiraju, and Cristina Sarasua, editors, *Proceedings of the 1st Workshop on Subjectivity, Ambiguity and Disagreement in Crowdsourcing, and Short Paper Proceedings of the 1st Workshop on Disentangling the Relation Between Crowdsourcing and Bias Management (SAD 2018 and CrowdBias 2018) co-located the 6th AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2018), Zürich, Switzerland, July 5, 2018*, volume 2276 of *CEUR Workshop Proceedings*, pages 41–54. CEUR-WS.org, 2018. URL http://ceur-ws.org/Vol-2276/paper5.pdf.

[82] Tim Wellhausen and Andreas Fiesser. How to write a pattern?: a rough guide for first-time pattern authors. In Andreas Fießer and Paris Avgeriou, editors, *16th European Conference on Pattern Languages of Programs, EuroPLoP 2011, Irsee, Germany,July 13-17,*

2011, page 5. ACM, 2011. doi: 10.1145/2396716.2396721. URL https://doi.org/10.1145/2396716.2396721.

[83] Yahoo. Yahoo design pattern library. Available at: https://perma.cc/RAZ6-N4XM, Accessed last time in January 2021, January 2015.