Charles University in Prague

Faculty of Mathematics and Physics

DOCTORAL THESIS

RNDr. Tomáš Knap

Towards Trustworthy Linked Data Integration and Consumption

Department of Software Engineering

Supervisor of the doctoral thesis:	RNDr. Irena Holubová, PhD.
Study programme:	Computer Science
Specialization:	Software Systems

Prague 2013

I would like to thank all those who supported me in my doctoral studies and in the work on my thesis. In the first place I really appreciate the help and advices received from my supervisor Irena Holubová. I am grateful for her numerous corrections and comments.

Secondly, I would like to thank to students who participated on the development of the ODCleanStore tool, without their effort, it would take much longer to accomplish a useable tool for managing and querying Linked Data. In particular, I would like to thank to Jan Michelfeit, with whom I was collaborating on the important parts of the data fusion algorithm and who did an excellent job regarding the implementation of the data querying service in ODCleanStore. I would like to also thank to Martin Nečaský, Jakub Klímek, and other my colleagues at the XML and Web Engineering Research Group for valuable comments and feedbacks on ODCleanStore.

Thirdly, I would like to also thank to Andre Freitas for collaboration on the W3P provenance model for the Web and Sean O'Riain and Edward Curry for numerous comments regarding the proper research work and amazing research stay in Digital Enterprise Research Institute in Galway.

I would like to also thank to Vojtěch Svátek for being able to participate on the EU ICT FP7 project (LOD2). And undoubtedly, I must also express my thanks to all the anonymous reviewers of my papers for helpful remarks and ideas.

My thanks also go to institutions that provided financial support for my research work. During my doctoral studies, my work was partially supported by GAUK 3110, EU ICT FP7 under No.257943 (LOD2 project), the Czech Science Foundation (GACR), grant numbers P202/10/0573 and 201/09/H057, and by SVV-2011-263312 (2011), SVV-2012-265312 (2012), and SVV-2013-267312 (2013). Special thanks also go to the Technology Agency of the Czech Republic (TACR, project TA02010182) for supporting the application of the ODCleanStore tool to the domain of legislation documents.

Last but not least, I am very thankful to Veronika who provided me unlimited support during my doctoral studies.

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In .... date .... signature of the author

Title:
Towards Trustworthy Linked Data Integration and Consumption

Author:
RNDr. Tomáš Knap

Department:
Department of Software Engineering

Supervisor:
RNDr. Irena Holubová, PhD., Department of Software Engineering

Abstract:
We are now finally at a point when datasets based upon open standards are being published on an increasing basis by a variety of Web communities, governmental initiatives, and various companies. Linked Data offers information consumers a level of information integration and aggregation agility that has up to now not been possible. Consumers can now "mashup" and readily integrate information for use in a myriad of alternative end uses. Indiscriminate addition of information can, however, come with inherent problems, such as the provision of poor quality, inaccurate, irrelevant or fraudulent information. All will come with associated costs of the consumed data which will negatively affect data consumer's benefit and Linked Data applications usage and uptake.

In this thesis, we address these issues by proposing ODCleanStore, a Linked Data management and querying tool able to provide data consumers with Linked Data, which is cleansed, properly linked, integrated, and trustworthy according to consumer's subjective requirements. Trustworthiness of data means that the data has associated data provenance, which satisfies the consumer's requirements, has certain data quality required by the data consumer, and is provided by trustworthy agents. We propose in the thesis a novel data fusion component for ODCleanStore which solves conflicts among the consumed heterogenous data and supplements the integrated data with justified quality scores and provenance metadata. Furthermore, to enable expressing and tracking of data provenance on the Web, we propose a novel provenance model for the Web – W3P. We also discuss trust models and their usability to compute trustworthiness of agents in social networks; a factor which contributes to the trustworthiness of consumed Linked Data. The ODCleanStore tool is available under an open license and is planned to be used by the Agile Knowledge Engineering and Semantic Web (AKSW) research group at the University of Lepzig, by the Department of Computer Science, Systems and Communication at the University of Milan-Bicocca, by the Semantic Web Company, Austria, and at the `http://opendata.cz` portal.

Keywords:
Linked Data consumption, Linked Data integration, data provenance on the Web, trust in social networks

Název práce:
Integrace a konzumace důvěryhodných Linked Data

Autor:
RNDr. Tomáš Knap

Katedra:
Katedra Softwarového Inženýrství

Vedoucí disertační práce:
RNDr. Irena Holubová, PhD., Katedra Softwarového Inženýrství

Abstrakt:

V posledních letech celá řada jedinců a společností (od univerzitních výzkumníků, přes soukromé společnosti, až po vládní úřady) začíná publikovat na Webu svá data s využitím otevřených standardů. Linked Data nabízí konzumentům dat úroveň agregace a integrace dat, která nebyla až doposud možná. Uživatelé si nyní mohou sestavit potřebná data zcela dle svých požadavků. S tímto procesem výběru dostupných dat z široké palety zdrojů souvisí řada problémů, zejména nízká kvalita dat, nepřesnost dat, nízká relevance dat pro účely daného konzumenta a také data, která jsou záměrně pozměněna. Všechny tyto faktory pak přináší další náklady související s použitím dat, které negativně ovlivňují přínos principů Linked Data pro jejich uživatele a rozšíření Linked Data aplikací.

V této práci proto navrhujeme ODCleanStore, nástroj pro správu a dotazování Linked Data, který je schopen poskytovat konzumentům dat Linked Data, která jsou pročištěná, prolinkovaná a důvěryhodná dle daných subjektivních požadavků konzumenta. Důvěryhodnost dat znamená, že tato data jsou asociovaná s informací o jejich původu, který odpovídá požadavkům konzumenta, dále tato data mají určitou kvalitu požadovanou konzumentem a také jsou poskytována důvěryhodnými uživateli. V této práci navrhujeme vlastní komponentu pro integraci dat implementovanou do nástroje ODCleanStore, která řeší konflikty mezi heterogenními daty a doplňuje takto integrovaná data informacemi o kvalitě dat a původu těchto dat. Abychom umožnili vyjádřit informace o původu dat na Webu, navrhujeme v práci vlastní model pro zachycení původu dat na Webu – W3P. V práci také diskutujeme důvěryhodnostní modely a jejich užitečnost pro výpočet důvěryhodnosti agentů (poskytovatelů a konzumentů Linked Data) v sociálních sítích. Nástroj ODCleanStore je k dispozi pod otevřenou licencí a je plánováno jeho nasazení ve výzkumné skupině Agile Knowledge Engineering and Semantic Web (AKSW) na Lipské univerzitě, na univerzitě v Miláně, pro zákazníky společnosti Semantic Web Company v Rakousku a v rámci projektu `http://opendata.cz`.

Klíčová slova:
konzumace Linked Data, integrace Linked Data, původ dat na Webu, důvěryhodnost v sociálních sítích

Contents

List of Figures

List of Tables

Listings

1. Introduction

All over the world governments and various organizations are connecting to the uprising trend of publishing governmental data as open data[1]; open data is original non-aggregated machine readable data which is freely available to everyone, anytime, and for whatever purpose. As a result, citizens paying taxes to the government are able to see and analyze the performance of the government by observing the raw data or using third-party applications visualizing and analyzing the data; companies can use the data to run their business. The most important shift from the current governmental practice of publishing the data to publishing data as open data is to: (1) release the data imprisoned in the documents (e.g., PDF files, Excel spreadsheets) and databases (e.g., relational databases), not being publicly available at all, (2) provide the data in the machine readable formats, not as PDF files, scanned PNG documents, or HTML documents.

Nevertheless, cannot we do more than just opening the data to simplify their exploration and creation of applications on top of them? If globally unique identifiers were used for resources (e.g., companies, public contracts) in the form of HTTP URIs [12], data about these resources could be published on these URIs. Data consumers could then use the current Web infrastructure to obtain relevant information about any resource by simply dereferencing its HTTP URI. Furthermore, if the open data was represented using RDF data model [120], such data (1) would be machine readable by the applications consuming the data, even if the applications do not know the particular data schema, (2) would have formal semantics, and (3) the schema behind the data would not need to be built upfront and fixed – it may evolve as the data evolves. In RDF data model, every information is expressed in a form of RDF triples recalling simple sentences, e.g., a sentence "John works in CompanyX" will form a triple consisting of a subject "John", a predicate "works in", and an object "CompanyX" [2].

Finally, using RDF data model and taking into account globally unique HTTP URIs, data from various sources may be easily linked together and, thus, a huge web of interconnected data can be created. As a consequence, data consumers can then operate on top of a single data space. Such idea described is precisely the idea of the *Linked Data* [19] approach. To illustrate creation of links between data, suppose that one dataset contains data about all Czech public contracts (e.g., a national portal of public contracts, `http://isvzus.cz`) and another dataset contains data about Czech business entities (e.g., a national business registry, `http://or.justice.cz`). As a result, to express that a public contract (from the first dataset) is realized by a certain contracting authority, the buyer of the contract (from the second dataset), we may introduce an RDF triple, where the particular public contract is the subject of the RDF triple, the particular business entity is the the object of the RDF triple, and the predicate expresses the nature of the relation, i.e., "being a contracting authority".

The Linked Data approach, introduced in 2006 by Tim Berners-Lee, inventor of the Web, refers to a set of best practices for exposing, sharing, and connecting

[1] `http://opendatahandbook.org/en/`

[2] The subject, predicate, and object in the given RDF triple should be HTTP URIs, but they are abbreviated for clarity.

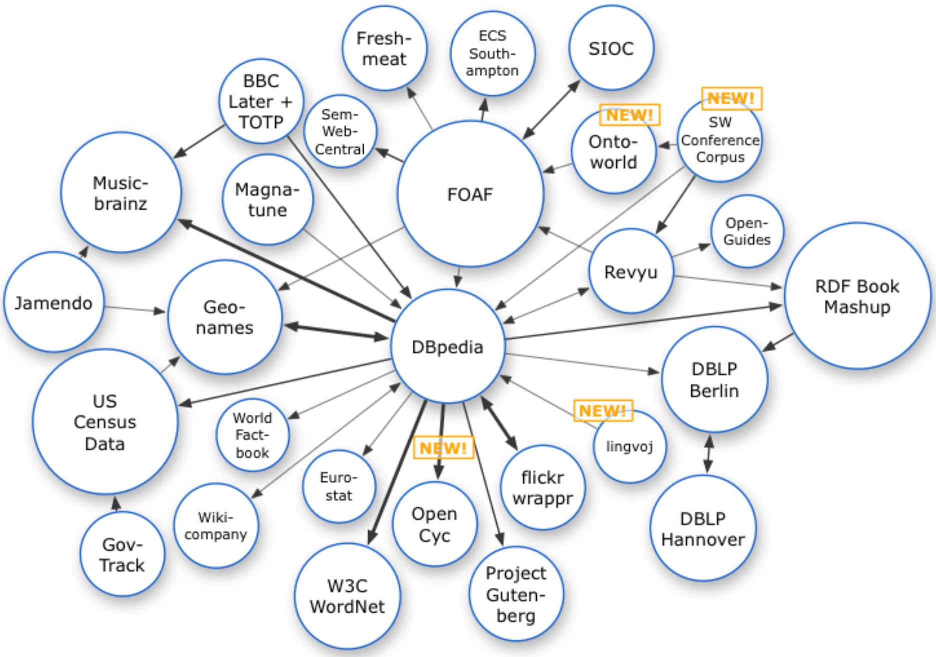


Figure 1.1: The Linking Open Data cloud diagram (November 2007)
(Source: http://richard.cyganiak.de/2007/10/lod/)

structured data on the Web [19]. The basic principle of Linked Data is to use HTTP URIs as names for things (resources), so that people can use current Web architecture to look up them and retrieve information on these resources. Furthermore, Linked Data approach is using RDF data model (RDF triples) to express facts about the resources, which allows humans as well as machines to browse and use the data space.

A significant effort in adoption and application of the Linked Data principles has been done in the Linked Open Data W3C SWEO Community Project (LOD)[3], supported by the W3C Semantic Web Education and Outreach Group[4]. The original and ongoing aim of that project is to extract data available under open licenses from wide variety of data sources (relational databases, XML native stores, RDF silos, (X)HTML pages, RSS feeds, etc.), link them together, and publish them on the Web as *Linked Open Data*, i.e., Linked Data published under an open license. From the evolution of the Linked Open Data cloud (see Figures 1.1, 1.2, and 1.3), it is obvious that the number of data (RDF triples) is growing exponentially since 2007, when the idea was sparked. Hundreds of datasets of varying size[5], covering wide range of domains, contain hundreds of billions of triples available to be consumed.

[3] Linking Open Data W3C Community Project, http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[4] Semantic Web Education and Outreach Group. http://www.w3.org/2001/sw/sweo/

[5] Every cirle represents a dataset with the number of triples proportionally corresponding with the size of the circle, see http://richard.cyganiak.de/2007/10/lod/#details. Compare, e.g., "DBpedia" dataset representing the machine readable version of all infoboxes from English version of Wikipedia and the dataset "Amsterdam Museum" describing cultural heritage objects related to the city of Amsterdam.

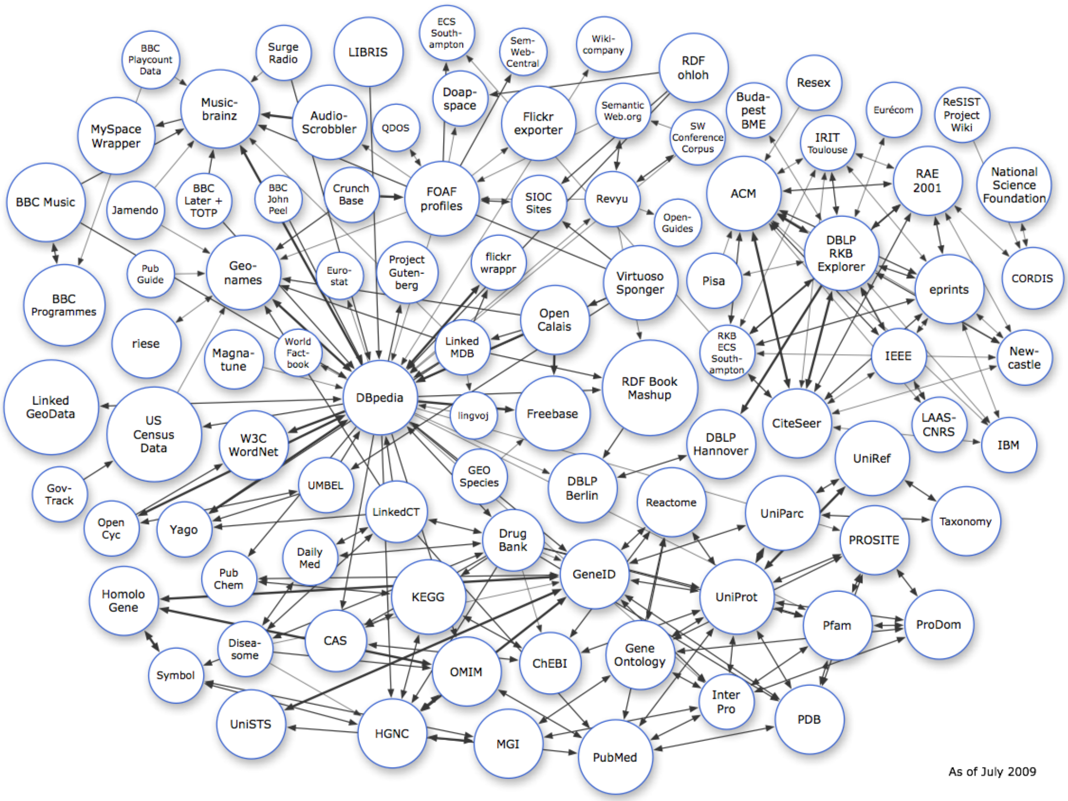


Figure 1.2: The Linking Open Data cloud diagram (July 2009)
(Source: `http://richard.cyganiak.de/2007/10/lod/`)

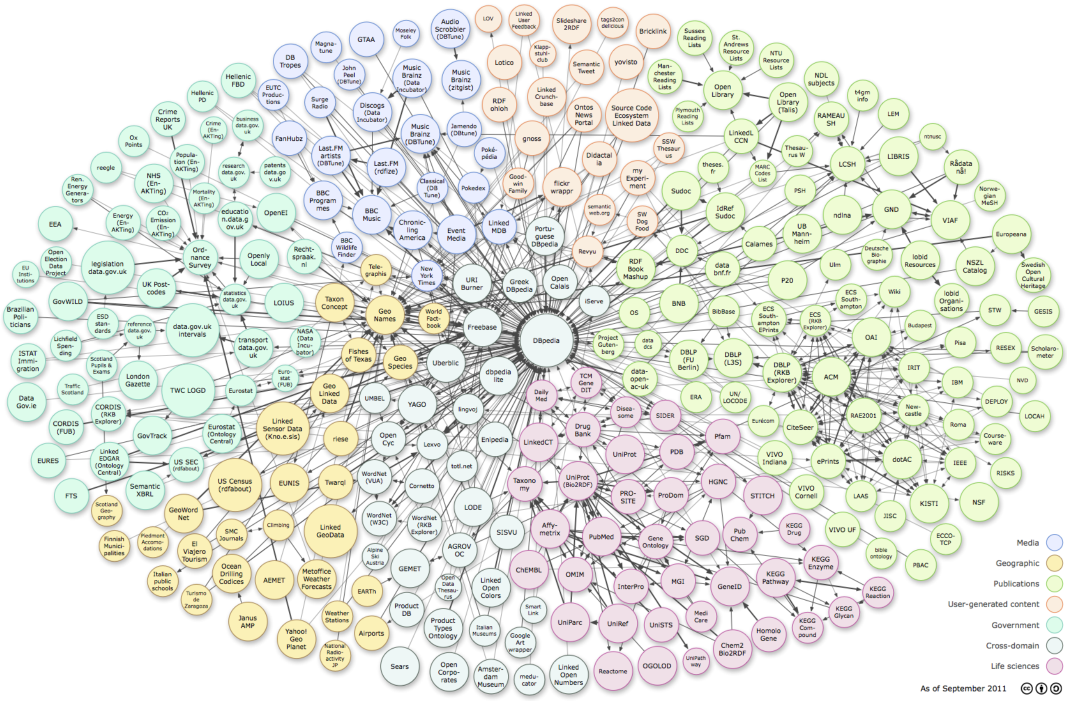


Figure 1.3: The Linking Open Data cloud diagram (September 2011)
(Source: `http://richard.cyganiak.de/2007/10/lod/`)

In the early stage of the LOD project, project participants were primarily university researchers and small companies; however, very soon, companies like the BBC, Thomson Reuters, and the Library of Congress revealed the power of Linked Data and joined this effort. In 2009, the government of the United States and of the United Kingdom realized the importance of publishing government data on the Web using Linked Open Data and committed towards this direction [6].

The most important shift from open data to Linked Open Data is the ability to reference and reuse uniformly resources in different machine readable documents. As a result, the web of data (Linked Open Data approach) is more powerful than the web of machine readable documents (open data approach), because the valuable raw data is integrated at the data level, not at the document or application level as in the open data approach. Therefore, Linked Open Data approach significantly decreases the human and financial resources needed to build Linked Data applications.

The Linked Data approach for publishing data is also slowly gaining momentum in the Czech Republic. As a part of the Open Government Partnership[7], the Czech Government committed to opening the governmental data trapped in the hidden silos, Excel spreadsheets, XML, and (X)HTML files according to open data principles. To realize that, the methodology for publishing data as open data was created for the Government of the Czech Republic [8]. The machine readability of open data enables the consequent automated transformation of open data to Linked Open Data and creation of the Czech governmental Linked Open Data cloud, which will be connected to the LOD cloud.

1.1 Problem Motivation

The advent of Linked Data [19] accelerates the evolution of the Web into an exponentially growing information space (as depicted in Figures 1.1, 1.2, and 1.3) where the unprecedented volume of data will offer information consumers a level of information integration and aggregation agility that has up to now not been possible. Consumers can now "mashup" and readily integrate information for use in a myriad of alternative end uses. Indiscriminate addition of information can, however, come with inherent problems, such as the provision of poor quality, inaccurate, not-interlinked, irrelevant or fraudulent information. All will come with associated costs of the data integration and consumption which will ultimately affect data consumer's benefit and Linked Data applications usage and uptake.

Scenario 1.1. Assume that Alice, an investigative journalist of the serious newspaper PragueNews, is writing an article about the motivation of the individual Prague council's members to support certain (rather suspicious) public contract C. For that purpose she would like to use the Czech governmental Linked Open Data cloud, which contains (1) datasets with public contracts obtained from several sources (e.g., national portal `http://isvzus.cz` or web pages of Prague municipality), (2) council members' voting from the past two years available in

[6] `http://www.whitehouse.gov/the_press_office/Transparency_and_Open_Government` and `http://bit.ly/15U15vk`

[7] `http://www.opengovpartnership.org/`

[8] `http://bit.ly/14XwVoz` (in Czech)

two datasets extracted by two different non-profit organizations, (3) dataset with the Prague municipal budget and payments made published by the Prague municipality, (4) opinions of experts about public contract C and other public contracts realized in the past, and (5) articles about the council members' activities published in the past by PragueNews and other newspapers; all exposed as Linked Data.

Using her Linked Data browser, Alice may start by searching for contract C, e.g., by typing the title of C. The Linked Data browser will display the list of relevant contracts and Alice selects C. Consequently, the browser automatically dereferences the URI behind C and, as a result, the browser obtains details about C together with the links to other data sources and displays the resulting data in a useful way. Alice may browse the data space by clicking on the links and letting the browser to dereference all the new URIs as described. Using the browser, Alice may easily find out which payments made were associated with the realization of contract C, what the experts think about contract C, which council members were voting for contract C and what were the reasons for that; she may further read articles relating to past activities of council members.

As a result, Linked (Open) Data sounds like the holy grail of the investigative journalists and citizens, but the following Problems P1 – P7 has to be addressed, so that Alice may use the Linked Data cloud efficiently.

P1: Data Linkage

The Linked Data, Alice is browsing, will be poorly linked, because data publishers, e.g., Czech governmental bodies, newspapers, or non-profit organizations, may not be aware of all datasets already published as Linked Data; thus, the data publishers do not link the data properly to other data sources available as Linked Data. As a result, the resulting descriptions of resources (e.g., of public contract C) will not be as rich (as useful) as they might have been and Alice would have to spend some time to find relevant resources for her article not being linked to contract C.

P2: Data Cleansing and Transformation

Currently, most of the Linked Data are still obtained from open data by automatic extraction and conversion to Linked Data. Such conversion may be executed by universities, non-profit organizations, or Linked Data application developers, who did not find the required resource and would like to help (or would not mind helping) the Linked Data community. Nevertheless, every such automated conversion yields in erroneous data, which has to corrected manually by Alice. Such converted data may be also poorly linked (see Problem P1).

P3: Data Integration

When integrating data from multiple sources, data conflicts may emerge (different sources may claim, e.g., different estimated prices of public contract C). Such conflicts have to be solved by Alice manually. But the data integration is not just about solving data conflicts, but also about deduplication of resources (Alice has to manually reveal that two different HTTP URIs, one hosted by the national portal of public contracts, the other one by the Prague municipality, represent the same public contract C, but published by different public bodies) and alignment of data schemas (e.g., different sources may employ different RDF predicates holding values for the estimated prices of contracts). Thus, to use integrated Linked Data, Alice has to firstly manually deduplicate resources[9] and align the schemas used; after that, Alice has to manually resolve the conflicts, e.g., she may decide to use the estimated price for C claimed by the national portal of public contracts, or she may compute an average value for the estimated price of C claimed by all available sources.

P4: Data Provenance

Even if the data integration could be automated, questions regarding the data provenance or lineage of the integrated data emerge – where is the integrated value (e.g., the estimated price of contract C) coming from, which agent (e.g., journalist, or governmental officer) was responsible for generation/transformation of that data, from which primary data source the data was derived, which sources support the claimed values, etc.

Furthermore, Bob, a colleague of Alice, may have (1) manually cleansed the data (to address Problem P2), (2) created certain links (to address Problem P1), and (3) published the adjusted data back to the Linked Open Data cloud, so that the whole Linked Data community may benefit. Such approach of data republishing is called *pay-as-you-go approach*, a typical approach when publishing data as Linked Data [19]. However, Alice should always know which data was published by official public bodies and which were republished by other journalists, because she considers official governmental data as more trustworthy. She should also know which consumed data was created by Cyril, who is considered by her as a trustworthy data publisher, and which data was created by Dave, who is not considered by her as a trustworthy publisher.

Furthermore, the reader of the Alice's article should be able to see the data provenance of the article and, e.g., be able to fetch the original primary sources quoted in the article. If the data provenance of quoted primary sources was missing or could not be provided together with the Alice's article, the credibility of such article would be low and PragueNews would lose their readers. Therefore, Alice either has to consume data together with their data provenance or she has to supplement the consumed and reused data with appropriate provenance information.

P5: Data Quality

Since one of the basic tenets of the Web is the AAA slogan — Anyone can say Anything about Any topic [3] – and because of the pay-as-you-go approach, which implies that the information in the Linked Data cloud may be refined by lots of Linked Data contributors (similarly as in case of wikipedia articles), the data Alice will browse and examine can be of various quality, articles may be differently accurate, recent, etc. To that end, Alice has to manually examine and evaluate the data quality dimensions important for her task, such as the

[9] Deduplication is a special case of linkage, where the semantics of the link is "two resources are equivalent".

accuracy and timeliness of the consumed data. Such Alice's manual effort has two drawbacks – it is time consuming, and she has to be an expert in the given data domain in order to be able to evaluate the quality of the data effectively.

P6: Trustworthiness of Agents

As a journalist, Alice has to know to which extent the data she is using or quoting in her article is *trustworthy.* Data provenance (discussed in Problem P4), representing the context of the consumed data, and data quality (discussed in Problem P5) analyzing the content of the consumed data, are important inputs for Alice when deciding about data trustworthiness.

Nevertheless, creation of a comprehensive quality assessment module which may be applied to any kind of data sources with excellent results is a tough goal – the examined data source may be too tiny for any data quality analysis or it may describe a rather specific domain, on which the quality assessment metric does not behave reasonably [7]. Therefore, in many cases, Alice has to rely solely on the data provenance of the consumed data. However, in these cases, Alice may not be able judge the trustworthiness of agents (e.g., publishers) in the provenance records, because she does not have any experience with them; she may not be able judge the trustworthiness of procedures responsible for data creation or publication, because she does not have any experience with the agents responsible for the execution of these procedures [85]. Therefore, the question is, how Alice should judge the trustworthiness of the data, if data provenance availability or data quality assessment does not help her much.

The solution might be (1) to maintain a social network of agents being associated with the published data or procedures publishing such data and (2) to define a trust metric computing (estimating) trustworthiness of agents – to which extend Alice should trust agents in the social network.

P7: Trustworthy Linked Data Consumption

Alice should be provided not only with the data itself, but also with an evidence of the data trustworthiness – data provenance, data quality, trustworthiness of agents. Furthermore, Alice has to have a way how to express certain subjective requirements on the consumed data, so that the consumed data is trustworthy for the task at her hand. For example, she may need to express that she would like to consume only data with certain accuracy, certain provenance records behind (e.g., coming only from certain sources), and for which the trustworthiness of agents – data providers – is high enough. Such requirements, Alice has, should be automatically enforced as the data is prepared for her.

1.2 Scope of the Thesis

Problems P1 – P7 outline the general scope of the thesis – how to provide the Linked Data consumers (such as Alice) with Linked Data, which is *cleansed*, properly *linked*, *integrated*, and *trustworthy* according the consumer's subjective and situation specific requirements. Trustworthiness of data means that the data has associated *provenance*, which satisfies the consumer's requirements, has certain *data quality* required by the data consumer, and is provided by *trustworthy agents*.

We observe the problem from the Linked Data consumers' perspective and suppose the explorative nature of the data consumption – the consumer starts from a certain point in the data space and browses the space. We do not discuss how the consumer may subscribe to the upfront defined portions of the Linked Data cloud.

Section 1.3 describes how the thesis addresses Problems P1 – P7 outlined. Some of them are covered deeply, introducing new methods of approaching the problem, some of them reuse already existing solutions or require further work and are covered only marginally.

1.3 Proposed Approach

From Problems P1 – P7 outlined, it is obvious that a certain tool is needed which would automate the tasks done manually by Alice in P1 – P7 and will allow the trustworthy Linked Data consumption and further usage, e.g., by the Alice's browser. Such tool should ensure that Alice will be provided with Linked Data already being cleansed (P1), linked (P2), integrated (P3), and trustworthy (P4, P5, P6, and P7). Such tool would increase Alice's experience with Linked Data, it would lower the costs of writing her articles substantially, and further accelerate the Linked Data usage and update. Such tool is proposed in the thesis and is called *ODCleanStore*.

ODCleanStore is a part of a framework (called *LDI framework* henceforward) consisting of (1) a *data extraction module* for extracting non-RDF data and its conversion to RDF data format, (2) a *data processing module* processing extracted RDF data and creating curated – cleansed, linked, quality assessed, and transformed – Linked Data, (3) a *query execution module* allowing data consumers to query the curated data and obtain integrated and trustworthy Linked Data for their particular needs, and (4) a *data visualization and analysis module* allowing data consumer to visualize and analyse the consumed Linked Data. The overall picture of the LDI framework is presented in Figure 1.4; the modules mentioned are depicted by light blue boxes.

The data extraction and data visualization and analysis modules are out of the scope of the thesis, the first one is developed as the Strigil project[10], the second one as the Payola project[11]. The scope of the ODCleanStore tool is presented in Figure 1.4 by the dark green box – ODCleanStore covers the data processing and data querying modules.

Data Processing Module

The incoming RDF *data feeds* are processed by *data processing pipelines* in ODCleanStore. Each pipeline successively executes a defined (and customizable) set of data processing units (*transformers*) ensuring that the feed is curated – cleansed, enriched with new data, arbitrarily transformed (addresses Problem

[10] http://strigil.sourceforge.net

[11] http://payola.github.com/Payola/

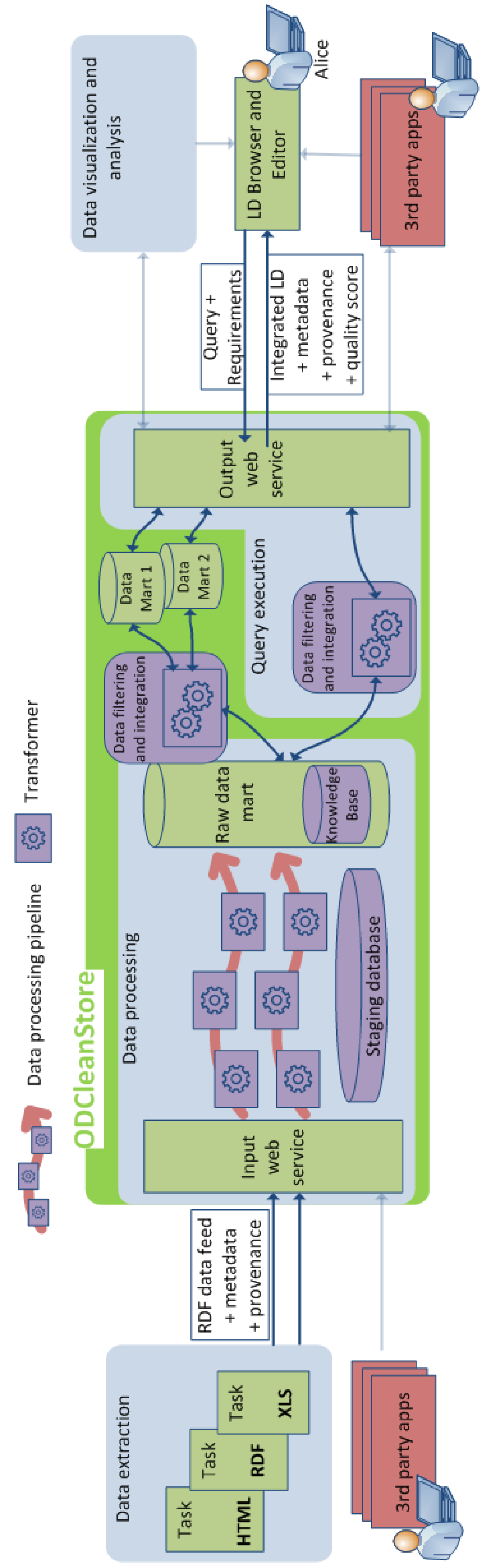


Figure 1.4: ODCleanStore and LDI framework

P2), resources in the feed are deduplicated and linked to already existing resources in the *raw data mart* (P1, P3), and the quality score of the feed is assessed (P5). When the pipeline finishes, the curated RDF data feed is populated to the raw data mart together with any auxiliary data and metadata created during the pipeline execution, such as links to other resources or metadata about the feed's quality. Such data in the raw data mart is ready for any further data querying.

Query Execution Module

Query execution module contains an *output web service*, to which data consumers may send their *queries* and, optionally, their set of *requirements* on the resulting data. As a result on their query, they obtain integrated Linked Data, which are trustworthy w.r.t. the query specific requirements.

Important components of the query execution module are the *data filtering component*, which addresses Problem P7 by enforcing specific requirements of data consumers (if there are some requirements accompanying the query), and *data integration component*, which addresses Problems P3 and P5 by integrating the resulting data and computing the quality of the integrated data. The data integration component also addresses Problem P7 by allowing the customization of the data integration and, as a result, a customization of the integrated quality computation.

As depicted in Figure 1.4, the data filtering and integration modules may be executed either at the query time or when the specialized *data marts* are prepared. Thus, data consumers of LDI may directly query the prepared data marts with pre-filtered and pre-integrated data (based on the default requirements) or use the raw data mart with cleansed, linked but not yet filtered or integrated data and let ODCleanStore to filter and integrate the data on-the-fly according to their requirements. The first approach favors the query performance, the latter one favors the customization of the consumed data. In the thesis, we focus on the online on-the-fly data filtering and integration as the consumers' queries arrive, which enables data consumers (e.g., journalists) to customize the consumed data and receive only relevant and trustworthy data for the particular task. Hereafter in the thesis, if talking about data querying, we mean querying of the raw data mart.

1.3.1 Data Linkage (P1, P3)

The important aspect of Linked Data is to create links between resources, e.g., to express that a city (a resource) is located in the given country (a resource), or that a contracting authority is responsible for a public contract. Such links are crucial to provide dense web of data graph, so that Alice does not need to create links manually as part of the browsing of the data space (Problem P1). Furthermore, such links also supplement data integration efforts sketched in Section 1.3.4, thus, data linkage also contributes to Problem P3.

Every data processing pipeline may contain a special transformer, *linker*, which addresses data linkage by automatically linking resources in the processed RDF data feeds against resources in the raw data mart. Every such linker is driven by a group of policies describing the conditions for creating links between resources. Linkers internally employ Silk, a tool described in Section 2.4, for the creation of links, and Silk-LSL language for describing the conditions for links creation. Policies for the linker are created by domain experts, who know the target domain (e.g., public procurement).

1.3.2 Data Normalization and Transformation (P2)

The data processing pipeline may contain a special transformer, a *data normalizer*, which addresses Problem P2 by automatically cleansing or transforming RDF data feeds by employing SPARQL Update language [67], which is a powerful and easy to use language standardized by W3C. Data normalizers are prepared by domain experts.

Complex cleansing and data transformation tasks which cannot be achieved by employing SPARQL Update queries, i.e., cannot be achieve by any data normalizer, can be targeted by implementing a *custom transformer* for ODCleanStore. In that case, domain experts have to cooperated with programmers to provide the custom transformer addressing the particular instance of Problem P2.

1.3.3 Data Quality (P5)

The ability to assess the *information quality* (*IQ*) presents one of the most important aspects of the information integration on the Web and will play a fundamental role in the continued adoption of Linked Data principles [141, 60]. IQ is usually described in different works by a series of *IQ dimensions* which represent a set of desirable characteristics for an information resource [112, 162, 7, 141]. Wang & Strong [162] present an extensive survey of IQ dimensions, based on the results of the questionnaire given to the panel of human subjects; papers [112, 1] cover IQ dimensions for the Web.

Since the information quality is a broad research topic, ODCleanStore could either (1) focus on implementation of one selected IQ dimension which could be trained to automatically assess the quality of certain types of consumed data (i.e., data from certain domains) or (2) focus on a general approach to assess the data quality but with the help of domain experts, who have to configure the quality assessment. We decided to address Problem P5 by the latter approach. Every data processing pipeline may contain a special transformer, *quality assessor*, which assigns a quality score to the processed RDF data feed based on the compliance of the feed with a group of *QA policies* prepared by domain experts. In Section 3.1.3, we describe the construction of such policies.

1.3.4 Data Integration (P3)

The needs for data integration is motivated by Problem P3. Basic steps of the data integration are – *duplicate detection* (to detect duplicated resources), *schema mapping* (to align schemas), and *data fusion* (to fuse the data and resolve the conflicts) [23]. In Chapter 4, we briefly discuss the duplicate detection and schema mapping approaches; duplicate detection uses the outputs of linkers on the data processing pipelines and the schema mapping uses manually configured mappings between ontologies stored in the internal *knowledge base* of ODCleanStore. Furthermore, we focus in detail on the third step of the data integration – data fusion. We propose a novel data fusion algorithm and discuss how the algorithm resolves the conflicts, computes quality of the integrated RDF triples, and can be customized.

The data fusion algorithm also contributes to Problem P5, because the resulting quality score computed for every integrated RDF triple is based on: (1) the quality scores produced by the quality assessors on the pipelines processing the data feeds contributing to that integrated triple, (2) the differences between the conflicting values of the triples being integrated and (3) the agreement among the triples being integrated on the resulting integrated triple's value. We discuss the quality computation in more detail in Chapter 4.

Data consumers may also customize the data integration, e.g., by specifying the desired conflict handling strategy executed by the data fusion algorithm; customization options are described in Chapter 4.

1.3.5 Data Provenance (P4)

Provenance or lineage of data provides the necessary contextualization for Alice, the information consumer, to analyze the trustworthiness of the information [136, 60, 83]. Without having provenance records associated with the data Alice is working with, she cannot find out who claimed the information, when, which process was behind the creation of that information; she cannot properly quote the source of the data in her article. Based on that, she cannot establish trust in the data; readers cannot establish trust in Alice's article.

To express and track provenance information on the Web, a suitable provenance model must exist. In Chapter 5, we discuss that current provenance models are not suitable for expressing provenance of data on the Web [12] and, as a result, we propose a novel provenance model W3P. Such provenance model allows (1) data publishers to express different dimensions of data provenance for the data submitted to ODCleanStore and (2) data consumers (such as Alice) to effectively examine provenance records behind the consumed data, so that they can decide on the data trustworthiness. Since the data without provenance records cannot be used in a serious business scenario [45], all credible data producer are motivated to provide provenance records. The W3P provenance model is also built in a way that consumers' requirements on data provenance may be easily expressed.

1.3.6 Trustworthy Agents (P6)

Social networks are recognized as a valuable source of information [85], e.g., to obtain requirements on the data of other data consumers or to observe data published by others; however, they can be full of malicious agents as well [76]. Therefore, the aspect of *trust* of an agent (Alice) willing to depend on another agent (Bob) in the social network is of crucial importance.

Social networks allow the transfer of trust from agents behind the data (e.g., data publishers) to the data itself [70]; if Alice trusts Bob, such information can be also used to judge the trustworthiness of the data published by Bob. As a result, trust in social networks may be used as a justification for data consumers why the provided data is trustworthy.

[12] At least at the time the provenance model W3P was proposed, see Section 7.3.

To judge the trustworthiness of Bob (the journalist, data publisher) Alice does not know (Problem P6), she may rely on other agents (journalists, public bodies, citizens) in the social network who can judge the trustworthiness of Bob; such agents are called *recommenders.* If she cannot estimate trustworthiness of these recommenders, she has to rely on other recommenders who are able to judge trustworthiness of these recommenders, etc. If we consider every such "ability of agent u to judge the trustworthiness of another agent v" as a (direct) *trust relation* between u to v, oriented, we can define a *social trust network* as a directed oriented graph with vertices being the agents and edges being such trust relations. Various *trust metrics*, often relying on transitivity of trust relations, can be then employed to estimate trustworthiness of agents not having a direct trust relation in the social trust network.

Nevertheless, lots of papers defining the trust metrics estimating trustworthiness of agents in social trust networks, such as [74, 148, 123], ignore the extent to which trust is situation-aware – it is *domain* and *task specific.* Someone who may be trusted for financial advices may not be trusted for film recommendations. Furthermore, trust is often, e.g. [76, 171, 148, 123], comprehended as a "black box" and indivisible concept. Since trust is so complex concept [92], semantics of the estimated trust relying on transitivity of social trust relations is ambiguous, as illustrated in Chapter 6.

To address these issues, in Chapter 6, we comprehend trust as a concept formed by the set of underlying *trusting beliefs* [57, 126]. Trust relations are never quantified directly, but they are derived from the quantifications of the beliefs forming trust. By deriving trust from its beliefs – the simpler and more intuitive concepts, the confusion of social network's members *what trust actually is* is minimized. To that end, in Chapter 6, we (1) extend social trust network to a *social trust beliefs network*, (2) elaborate which beliefs (such as honesty, competence, or experience) may support the quantification of trust relations, and (3) survey current approaches (*trust metrics*) for estimating trust between two agents not having a direct trust relation between them and discuss the suitability of these metrics for computing trust in the social trust beliefs network.

Chapter 6 is driven by a different motivational scenario introduced in Section 6.1; in Chapter 7, we discuss how the results of Chapter 6 contribute to addressing Problem P6.

1.3.7 Trustworthy Linked Data Consumption (P7)

To provide trustworthy Linked Data to data consumers, the query execution module of ODCleanStore has to provide consumers with an *evidence* of the data trustworthiness regarding (1) data provenance, (2) quality of the data, and (3) trustworthiness of agents providing the data.

To fully address the trustworthy linked data consumption problem (Problem P7), the data consumers should be also allowed to specify certain types of *requirements* they would like to enforce for the given query. The types of requirements correspond with the types of evidences introduced, i.e., requirements on (1) data provenance, (2) quality of the data, and (3) trustworthiness of agents. As a result, the query execution module of ODCleanStore should be able to automatically enforce these requirements as the response on the query is prepared.

The data filtering component should serve as the main module for applying these requirements, data integration component is also influenced by the consumers' requirements.

The types of requirements (1) – (3) introduced are supported by the results of paper [20], where they define a trust policy as "a subjective procedure used for evaluating the trustworthiness of information in a specific situation" and they distinguish three types of such policies: context-based, content-based, and rating-based. These types correspond directly with our types of requirements (1) – (3).

In the next paragraphs we discuss for each type of evidence to which extent ODCleanStore is able to provide such evidence. We also discuss for each such evidence how consumers of ODCleanStore may express their requirements and to which extent ODCleanStore is able to enforce them.

Data Provenance

Evidence. The resulting integrated data produced by the query execution module of ODCleanStore is supplemented with provenance information associated with the data feeds contributing to the resulting integrated triples as discussed in Section 3.2 and in Section 5.8. As a result, Alice can manually browse the provenance information of the consumed data and get the evidence for the trustworthiness of such data. The W3P provenance model described in Chapter 5 allows to efficiently express provenance information behind the data feeds submitted to ODCleanStore and, consequently, provides such provenance information as a provenance evidence to data consumers.
Consumers' Requirements. In Section 5.9, we describe how consumers can define their own situation-specific requirements in the form of *provenance policies*. The provenance policies are capable of filtering certain data sources and preferring others due to certain aspects in the data provenance records associated with these sources. We describe how these provenance policies can be (1) constructed by data consumers and (2) automatically enforced (applied) as part of the data filtering component in ODCleanStore. Furthermore, the design of W3P provenance model substantially influences the efficiency of the provenance policies' enforcement.

Data Quality

Evidence. ODCleanStore supplements the resulting integrated data with quality scores, which are computed based on (1) the results of the quality assessors and quality aggregators on the data processing pipelines, and (2) the integrated quality scores computed as part of the data integration component. Every quality score is also justified by the list of QA policies which were applied, as part of the quality assessor, to the data sources the integrated data originates from. As a result, Alice may observe the quality score of any integrated triple, observe the data sources which contributed to the integrated triple, and also see the list of QA policies justifying the quality score.
Consumers' Requirements. ODCleanStore provides data consumers with the possibility to define certain requirements on the data fusion algorithm, e.g., a consumer may specify the conflict handling policies driving the data fusion or certain aspects of the integrated quality score computation. These requirements and their enforcement are discussed in Chapter 4. No data quality requirements

are enforced in the data filtering component; this is a future work, which will be provided together with a set of more advanced quality assessors assessing the individual quality scores for every information quality dimension.

Trustworthy Agents

Since Problem P6 was not addressed for the social network of agents behind ODCleanStore, any provision of evidence or support for consumers' requirements on trustworthy agents is a future work, sketched in Chapter 7.

Linked Data Browser for Alice

The implementation of the Linked Data browser Alice may use to consume the trustworthy linked data is out of the scope of the thesis. We implemented only prototype HTML interfaces (see Figures 3.3, 3.4, and 3.5), which allows to interact with the query execution module (output web service) of ODCleanStore; these interfaces are part of the ODCleanStore's distribution package. In Chapter 7, we discuss how the Linked Data browser should support Alice when specifying her provenance requirements on the consumed data.

1.4 Research Questions

The focal theory of this thesis is motivated mainly by Problems P3, P4, P5, P6, and P7, but also contributes to Problems P1 and P2. Main research questions being addressed in the thesis are as follows.

Q1 How should the data be automatically cleansed, linked, and quality assessed in ODCleanStore?

Q2 How should the data conflicts be resolved during the data fusion in ODCleanStore?

Q3 How should the quality of the integrated triples be computed?

Q4 How could a consumer adjust the data fusion and integrated quality computation?

Q5 How should the provenance information be expressed for data on the Web?

Q6 How should the trust be computed in social trust networks?

Q7 How should be the consumers' provenance requirements on the trustworthy data expressed and enforced by ODCleanStore?

Table 1.1 summarizes the research questions; every research question is supplemented with the problems motivating it, the chapters of the thesis where the question is targeted, and relevant author's publications for that question. The list of all relevant publications for the thesis is introduced in Section 1.6.

Table 1.1: Research questions

Question	**Problem**	**Chapters**	**Author's Publications**
Q1 How should the data be automatically cleansed, linked, and quality assessed in ODCleanStore?	P1, P2, P5	3	[111, 105], [128, 109]
Q2 How should the data conflicts be resolved during data fusion in ODCleanStore?	P3	4	[106, 128]
Q3 How should the quality of the integrated triples be computed?	P3	4	[106, 128]
Q4 How could a consumer adjust the data fusion and integrated quality computation?	P3, P7	4	[106, 128]
Q5 How should the provenance information be expressed for data on the Web?	P4	5	[60]
Q6 How should be the consumers' provenance requirements on the trustworthy data expressed and enforced by ODCleanStore?	P4, P7	5	[103]
Q7 How should the trust be computed in social trust networks?	P6, P7	6	[107, 110, 108], [101, 102, 64]

1.5 Main Contributions

Main contributions of the thesis are as follows, every contribution is associated with one or more questions from Section 1.4.

C1 ODCleanStore tool – a Linked Data management tool, which allows data cleansing, linking, quality assessment, and allows to directly apply the research conducted w.r.t. questions Q2 – Q7 (associated with Q1).

C2 A data fusion algorithm, integrated data quality computation, and the data fusion customization (associated with Q2, Q3, and Q4).

C3 Definition of a general provenance model W3P for expressing provenance information on the Web; expressing and enforcing provenance requirements in ODCleanStore (associated with Q5 and Q6).

C4 Trust model and a metric for computing trust in social trust beliefs networks (associated with Q7).

1.6 Relevant Published Work

The list of author's publications relevant for the thesis (associated with Questions Q1 – Q7) is as follows.

Journal Articles

- A. Freitas, T. Knap, S. O'Riain, and E. Curry. W3P: Building an OPM based provenance model for the Web. *Future Generation Comp. Syst.*, 27(6):766–774, 2011, ISSN: 0167-739X, IF: 1.978, 5-Year IF: 1.594[13]

Full Conference Papers

- T. Knap, J. Michelfeit, and M. Nečaský. Linked Open Data Aggregation: Conflict Resolution and Aggregate Quality. In *COMPSAC Workshops*, pages 106–111, Izmir, Turkey, 2012. IEEE Computer Society

- T. Knap, M. Nečaský, and M. Svoboda. A Framework for Storing and Providing Aggregated Governmental Linked Open Data. In *Proceedings of Joint International Conference on Electronic Government and the Information Systems Perspective, and Electronic Democracy (EGOVIS/EDEM)*, pages 264–270, Vienna, Austria, 2012. Springer

- T. Knap and I. Mlýnková. Revealing Beliefs Influencing Trust between Members of the Czech Informatics Community. In *Proceedings of the 3rd International Conference on Social Informatics (SocInfo)*, pages 226–239, Singapore, 2011. Springer

- T. Knap and I. Mlýnková. Quality Assessment Social Networks: A Novel Approach for Assessing the Quality of Information on the Web. Proceedings of the 8th International Workshop on Quality in Databases of VLDB '10: 36th International Conference on Very Large Data Bases, 2010. `http://www.vldb2010.org/proceedings/files/vldb_2010_workshop/QDB_2010/Paper1_Knap_Mlynkova.pdf`, Retrieved 07/03/2013

- T. Knap and I. Mlýnková. Web Quality Assessment Model: Trust in QA Social Networks. In *Proceedings of 8th International Conference on Ubiquitous Intelligence and Computing (UIC)*, pages 252–266, Banff, Canada, 2011. Springer

- T. Knap and I. Mlýnková. Towards Topic-based Trust in Social Networks. In *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing (UIC)*, pages 635–649, Xi'an, China, 2010. Springer

- T. Knap. Provenance Policies for Subjective Filtering of the Aggregated Linked Data. In *Proceedings of the 5th International Conference on Advances in Databases, Knowledge, and Data Applications*, DBKDA'13, pages 95–99, Seville, Spain, 2013. IARIA

[13] Source: Thomson Reuters Journal Citation Report 2012

Demo & Positional Papers, Posters

- T. Knap, J. Michelfeit, J. Daniel, P. Jerman, D. Rychnovský, T. Soukup, and M. Nečaský. ODCleanStore: A Framework for Managing and Providing Integrated Linked Data on the Web. In *Proceedings of 13th International Conference on Web Information Systems Engineering (WISE)*, pages 815–816, Paphos, Cyprus, 2012. Springer

- J. Michelfeit and T. Knap. Linked Data Fusion in ODCleanStore. In *Proceedings of the 11th International Semantic Web Conference (Posters & Demos)*, Boston, USA, 2012. CEUR-WS.org

- T. Knap. Trusting Beliefs: A Different Way to Comprehend Trust in Social Networks. 8th Extended Semantic Web Conference (ESWC), Positional paper, Greece, 2011. `http://www.ksi.mff.cuni.cz/~knap/publications/2011/eswc-pp.pdf`, Retrieved 07/03/2013

- T. Knap. Trusting Beliefs: A Way to Comprehend Trust between Members of the Czech Informatics Community. Extended Semantic Web Conference (ESWC) Summer School, Poster, Greece, 2011. `http://www.ksi.mff.cuni.cz/~knap/publications/2011/eswc-school-poster.pdf`, Retrieved 07/03/2013

- J. Galgonek, T. Knap, M. Kruliš, and M. Nečaský. SMILE - A Framework for Semantic Applications. In *Proceedings of OTM 2010 Workshops*, pages 53–54, Crete, Greece, 2010. Springer

- T. Knap, J. Klímek, J. Mynarz, M. Nečaský, and J. Stárka. OpenGov - Towards More Transparent Public Contracts. Indian-summer school on Linked Data (ISSLOD), Poster, Germany, 2011. `http://www.ksi.mff.cuni.cz/~knap/publications/2011/isslod-poster.pdf`, Retrieved 07/03/2013

1.7 Thesis Outline

In Chapter 2 we provide introduction to Linked Data terminology and technology. Chapter 3 presents the ODCleanStore tool, the concept of data processing pipelines and query execution module of ODCleanStore; further major contributions of the thesis tackle particular parts of ODCleanStore. In Chapter 4 we describe the novel data fusion algorithm of the data integration component in ODCleanStore. Chapter 5 describes the novel data provenance model (W3P) for the Web and how ODCleanStore deals with data provenance. Furthermore, Chapter 5 also discusses how data consumers may specify their requirements on the provenance information; such requirements are enforced by the data filtering module of ODCleanStore. Chapter 6 introduces the trust model and metric for computing trust in social networks. In Chapter 7, we summarize the lessons learned, provide conclusions, and outline future work.

2. Linked Data

In Section 2.1, we start with further motivation for the importance of machine readability of data and describe the principles of Linked Data on an illustrative scenario. In Section 2.2, we describe Resource Description Framework (RDF), its data model, and RDF serializations; furthermore, we introduce RDF Schema and Web Ontology Language – languages for describing vocabularies on the Web – and present briefly the ontology for public contracts domain. In Section 2.3, we describe SPARQL – the query and update language for RDF data. Finally, in Section 2.4, we introduce two tools, Virtuoso and Silk, which are used in ODCleanStore to process Linked Data.

2.1 Introduction to Linked Data

Scenario 2.1. Suppose that Alice from Scenario 1.1 is writing another article about the budget proposal of the German city of Berlin for the year 2013. To write the article, she is using her favorite editor. For writing the article, she needs (1) the name of the current mayor of Berlin to track the responsibility for the budget and (2) the latest measured population of Berlin (together with the date when it was measured) to compare the Berlin's budget proposal per capita with the Prague's budget proposal.

The question is whether it is possible for her favorite editor to obtain such information automatically from the Web and include it to her article. First, let us assume there is no Linked Open Data cloud. In that case, there are couple of web pages holding the population of Berlin – e.g., the official Berlin's web portal[14] and Wikipedia[15] – and couple of web pages holding the name of the current mayor of Berlin[16]. Nevertheless, to automatically extract facts from these kinds of pages, the editor would have to know upfront the structure of these pages, e.g., that the population of Berlin is on the 15th row in the 1st table within the element `DIV` with attribute `id = bomain_content` at the given page of the official Berlin web portal[17]. Furthermore, if the layout of that web page changes, the rules for extracting the population of Berlin have to be updated, which brings heavy burden to the application developers/configurators to keep up the step with the changes of the web pages' layouts. Obviously, this approach is hardly maintainable with the increasing number of web pages and Alice's editor cannot work this way. The approach can be improved by various machine learning and statistical methods; however, in this case, the recall of the facts Alice may automatically use in her article is increased only at the expense of the precision of the extracted data.

The problem is that contents of these web pages are not *machine readable* – the editor cannot understand that the particular value within the particular web page represents the name of the Berlin's mayor or its population. The Linked

[14] `http://www.berlin.de/berlin-im-ueberblick/zahlenfakten/index.en.html`

[15] `http://en.wikipedia.org/wiki/Berlin`

[16] E.g. `http://www.berlin.de/rbmskzl/rbm/lebenslauf/curriculum.html` and `http://en.wikipedia.org/wiki/Berlin`

[17] `http://www.berlin.de/berlin-im-ueberblick/zahlenfakten/index.en.html`

Data approach presents, together with the Linked Open Data cloud including the required information about Berlin, the solution for Scenario 2.1.

2.1.1 Linked Data Approach

The Linked Data [19] approach refers to a set of best practices for exposing, sharing, and connecting structured data on the Web. The term Linked Data was introduced in 2006 by Tim Berners-Lee, inventor of the Web, who outlined four principles for Linked Data on the Web[18]:

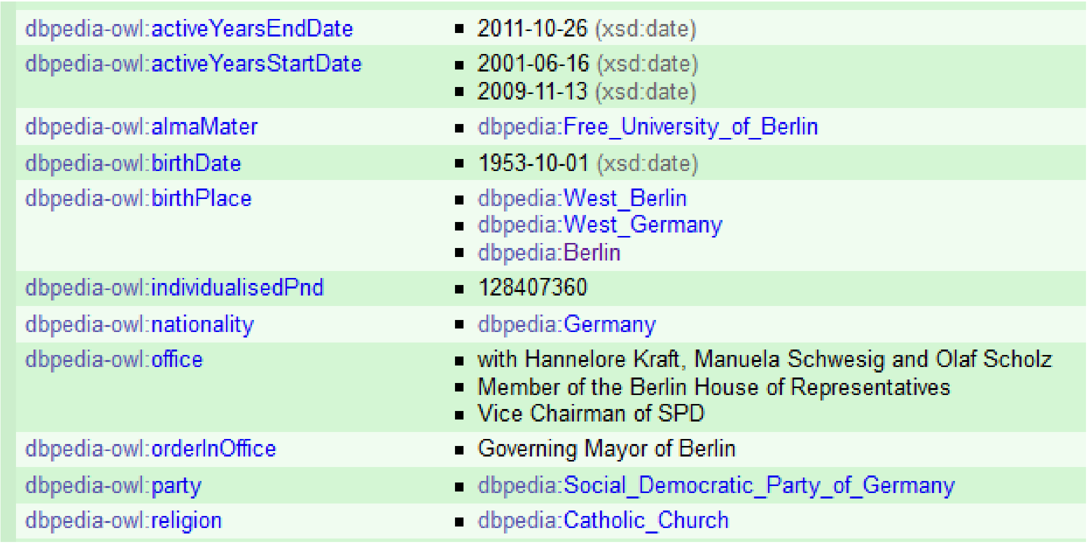
1. Use URIs as names for things (resources).
2. Use HTTP URIs so that data consumers can look up those names.
3. When someone looks up an URI, provide useful information, using RDF and SPARQL related standards.
4. Include links to other URIs, so that data consumers can discover more things (resources).

Let us explain these principles using Scenario 2.1. The current mayor of Berlin would need, according to Principles 1 and 2, an HTTP URI, e.g., `<http://dbpedia.org/resource/Klaus_Wowereit>`. Such URI should be dereferenceable – if put to the browser, the page containing details about Klaus Wowereit should be displayed according to Principle 3. Since we selected an existing URI for the current mayor of Berlin, the URI is dereferenceable and the resulting RDF triples are displayed in Figure 2.1 (namespace prefix `dbpedia-owl:` is associated with namespace `http://dbpedia.org/ontology/`, namespace prefix `dbpedia:` is associated with namespace `http://dbpedia.org/resource/`). Figure 2.1 also depicts links to other resources according to Principle 4, such as the link to Klaus Wowereit's alma mater – the resource `dbpedia:Free_University_of_Berlin`.

We have to distinguish between *non-information* and *information* resources. Whereas a non-information resource represents the particular *real-world object*, such as the particular mayor of Berlin in Scenario 2.1, the information resource represents the document, page, image or any other *representation* of that object. "As a rule of thumb, all *real-world objects* that exist outside of the Web are non-information resources."[19] Obviously, one real-world object may have two or more representations, one might be the (X)HTML page about that object for humans and the second one might be RDF data model serialization for machines. The provision of the proper representation is implemented by the content negotiation within the HTTP protocol. Therefore, if the resource `dbpedia:Klaus_Wowereit` is dereferenced by a web browser, the resulting resource is `<http://dbpedia.org/page/Klaus_Wowereit>` representing the human readable version of the RDF data about Klaus Wowereit (Figure 2.1).

[18] Berners-Lee, T. Linked Data – Design Issues.
`http://www.w3.org/DesignIssues/LinkedData.html`

[19] `http://www4.wiwiss.fu-berlin.de/bizer/pub/linkeddatatutorial/`



dbpedia-owl:activeYearsEndDate	▪ 2011-10-26 (xsd:date)
dbpedia-owl:activeYearsStartDate	▪ 2001-06-16 (xsd:date) ▪ 2009-11-13 (xsd:date)
dbpedia-owl:almaMater	▪ dbpedia:Free_University_of_Berlin
dbpedia-owl:birthDate	▪ 1953-10-01 (xsd:date)
dbpedia-owl:birthPlace	▪ dbpedia:West_Berlin ▪ dbpedia:West_Germany ▪ dbpedia:Berlin
dbpedia-owl:individualisedPnd	▪ 128407360
dbpedia-owl:nationality	▪ dbpedia:Germany
dbpedia-owl:office	▪ with Hannelore Kraft, Manuela Schwesig and Olaf Scholz ▪ Member of the Berlin House of Representatives ▪ Vice Chairman of SPD
dbpedia-owl:orderInOffice	▪ Governing Mayor of Berlin
dbpedia-owl:party	▪ dbpedia:Social_Democratic_Party_of_Germany
dbpedia-owl:religion	▪ dbpedia:Catholic_Church

Figure 2.1: Excerpt of a DBpedia page
(Source: `http://dbpedia.org/resource/Klaus_Wowereit`)

According to Principles 1 – 4, the key technologies Linked Data lies on are - URIs (a unique identification of resources), HTTP (a simple and universal mechanism for retrieving data), and RDF (a generic data model for expressing facts about resources and linking resources together). By employing HTTP protocol, Linked Data directly builds on the general architecture of the Web. The RDF data model and other related standards are described in more detail in Section 2.2. To start with, it is important to know that the RDF data model is a generic model where every fact is expressed in a form of an RDF triple consisting of a subject, a predicate (or a property), and an object of that triple; the triple can be read as a sentence, consisting of a subject, a predicate, and an object – e.g., the triple `(dbpedia:Berlin, dbpedia-owl:leader, dbpedia:Klaus_Wowereit)` can be read as "Berlin has a leader (mayor) Klaus Wowereit". Subjects and predicates are identified by HTTP URIs. Objects may or may not use HTTP URIs; if they do not use them, they represent literal values, such as the population of Berlin. Such literal values are not called resources, thus, they do not violate the Linked Data principles, however, they cannot be dereferenced. Therefore, from the Linked Data perspective, objects identified by HTTP URIs (resources) are of utmost importance – these resources provide the only mean how to reach literal values.

Further, let us suppose that Alice's favorite editor (Scenario 2.1) is able to process Linked Data. RDF data model is self-descriptive, Alice's RDF-aware editor always knows what is a subject, predicate, and object of every RDF triple, even without understanding the semantics behind the URIs in the triple. For example, this is not true for the XML data model, where even simple facts, e.g., that a book has an author with the given name, can be represented by using various structural patterns [20].

However, to be able to select from dozens of triples about the city of Berlin the right one containing the population of Berlin, the RDF-aware editor has to

[20] `http://www.w3.org/DesignIssues/RDF-XML.html`

know that there is a predicate `dbpedia-owl:populationTotal` with a subject being of type `dbpedia-owl:PopulatedPlace` and an object being an integer holding the population of the subject – the city of Berlin. Such information about the domains and ranges of the predicates is described by the RDF vocabulary definition language, RDF Schema [31]. The advantage of the RDF representation of the Berlin's population is that it is not anyhow connected with the actual position of that value at a web page. Therefore, as long as the RDF data is available and the Alice's editor is aware of the predicates holding populations of cities and names of mayors, the automated extraction of such information from the Linked Open Data cloud and its automated usage in the Alice's article is possible.

2.1.2 Web of Data Graph

Let us recall the LOD Project with the datasets as depicted in Figure 1.3. If we consider every fact (a triple) being an edge in a graph leading from its subject to its object and being typed according to the predicate of that triple, we will get a giant graph containing billions of vertices and edges, connecting together hundreds of datasets – we will call such a graph a *web of data graph.* With such a number of triples, new tools for storing, querying, linking the data, etc., have to emerge.

2.2 Resource Description Framework (RDF)

The main technologies Linked Data is built on are HTTP, URI, RDF, and SPARQL. The technologies HTTP and URI were already discussed briefly in Section 2.1.1, RDF is described in this section, SPARQL in Section 2.3.

The Resource Description Framework (RDF) [87] is a framework for representing information on the Web [99]. RDF may be used to "publish structured information on the Web, and exchange information between web-based information systems"[16].

RDF is designed to (1) have a simple data model behind which is easily processable and manipulatable by applications, (2) have formal semantics (with a rigorously defined notion of entailment providing a basis for well founded deductions), (3) use extensible vocabularies (data models), (4) be independent of any specific serialization, and (5) allow anyone to make statements about any resource [99]. By satisfying these requirements, "RDF aims to be employed as lingua franca, capable of moderating between other data models that are used on the Web" [16], i.e., it is suitable for representing integrated information coming from multiple heterogeneous sources with different data schemas. Therefore, RDF suits well the situation Alice has to deal with in Scenario 1.1.

2.2.1 RDF Data Model

RDF data model is a simple graph-based data model. In RDF data models, all objects of interest are called *resources.* Resources have *properties* or *predicates.* Each property has a *property type* and a *property value.* Property values may be atomic (e.g., strings or numbers) or references to other resources, which in turn may have their own properties. Information about resources is represented in the form of *triples* [16].

RDF Triples

Every *triple* consists of a *subject*, a *predicate* and an *object*. Every subject represents the resource the information is about, predicate represents the property of that resource, and object represents the value of that property, i.e., a triple is "a statement of a relationship between the things" [99].

Definition 2.1. Suppose an infinite set $\mathcal{U}$ of URI resources, an infinite set $\mathcal{B}$ of blank nodes, and an infinite set $\mathcal{L}$ of literals; the sets $\mathcal{U}$, $\mathcal{B}$, an $\mathcal{L}$ are pairwise disjoint. Set $\mathcal{Z}$ denotes an infinite set of all *RDF nodes*, i.e., $\mathcal{Z} = \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}$. Then, a triple $(s, p, o) \in (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$ is called an *RDF triple* (or simply *triple*); s, p, o are the subject, the predicate, and the object of the triple, respectively. Subjects, predicates, and objects of RDF triples are *RDF nodes*.

In Definition 2.1, we distinguish three types of RDF nodes – URI resources, literals, and blank nodes. URI resources are nodes identified by a globally unique identifier, an HTTP URI, which satisfies the Linked Data principles and whose syntax follows the one described in [12]. URI resources may be used to identify any object of an interest – it could be a real-world object (a non-information resource) or its representation (an information resource). The HTTP URI should be dereferenceable; when it is dereferenced, a proper representation of the resource should be provided based on the HTTP content negotiation [91, 86] – e.g., HTML representation for humans, and the RDF serialization (Section 2.2.2) for web applications.

Blank nodes can be used in one or more RDF triples to identify a resource [16]. They do not have identifiers which are visible behind the scope of a set of RDF triples (called RDF graph, see Definition 2.2), i.e., they are unique only w.r.t. the given set of triples. They may be used to associate two or more object values with a single subject represented by a blank node; e.g., a node representing a price of the product may be represented as a blank node, such a blank node is associated with two object values, one holding the actual price of the product, the second one the currency in which the price is expressed. It is always up to the data creator to decide which data should be represented by URI resources and which by resources represented as blank nodes.

Literals represent values of the properties, such as a number, a text, or a date. Literals may be plain or typed. A plain literal is just a string, optionally supplemented with a language tag, identifying the language of the string [99]. Typed literals contain the string value of the literal and its type, typically defined by XML Schema datatypes specification [14].

Example 2.1. A triple `(x:Alice, foaf:knows, y:Bob)` is an example of a triple corresponding with a sentence "Alice knows Bob". The subject is URI resource `x:Alice`, the predicate is URI resource `foaf:knows`, the object is URI resource `y:Bob`. The expressions `x:`, `foaf:`, `y:` are particular namespace prefixes [99], which represent abbreviations for the full HTTP URIs, e.g., `foaf:knows` is an abbreviation for `http://xmlns.com/foaf/0.1/`. In Figure 2.1, we can see 16 triples.

RDF Graphs

Definition 2.2. An *RDF graph* H is a set of RDF triples, i.e., $H \subset (\mathcal{U} \cup \mathcal{B}) \times \mathcal{U} \times (\mathcal{U} \cup \mathcal{B} \cup \mathcal{L})$.

RDF graphs can be represented as node and arc diagrams [87]. Informally, "in this notation, a triple is represented by a node for the subject, a node for the object, and an arc for the predicate, directed from the subject node to the object node"[16].

Definition 2.3. A diagram $D_H = (V, E)$ is a directed graph, where H is an RDF graph, $V = \{z \in \mathcal{Z} \mid \exists x, y \in \mathcal{Z} \wedge ((z, x, y) \in H \vee (x, y, z) \in H)\}$, and $E \subseteq V \times V$, $E = \{(s, o) \mid s, o \in V \wedge \exists p \in \mathcal{Z} \wedge (s, p, o) \in H\}$.

Named Graphs and Quads

Named Graph data model [38], a simple extension of the RDF data model, allows naming of RDF graphs, which is useful for efficient representation of the descriptive and provenance metadata behind RDF graphs [16]. Thesis [16] discusses why the existing approach using RDF reification[21] is not sufficient for descriptive and provenance metadata representations.

Definition 2.4. A *named graph* $G \in \mathcal{G}$ is a pair (u, H), where H is an RDF graph and $u \in \mathcal{U}$ is a name for H; $\mathcal{G}$ is an infinite set of all named graphs. We say that a triple $(s, p, o) \in G \iff (s, p, o) \in H$; $s, p, o \in \mathcal{Z}$.

Definition 2.5. Suppose a named graph $G = (u, H)$. Let us introduce a *quad* $(s, p, o, u) \in \mathcal{Q}$, as an abbreviated expression for a triple $(s, p, o) \in (u, H)$; $\mathcal{Q}$ is an infinite set of all quads; $s, p, o \in \mathcal{Z}$.

Definition 2.6. Let us define a *quad template* $T \in \mathcal{P}(\mathcal{Q})$ as a quad which may contain an asterix character, '*', at the position of subject, predicate, object, or graph. Thus, every quad template T represents a set of quads having an arbitrary value at the position of the asterix in the template. For example, quad template $T = (*, *, *, g)$ represents the set of quads $\{(s', p', o', g') \mid \exists s', p', o' \in \mathcal{Z} \wedge \exists g' \in \mathcal{G} \wedge g' = g\}$.

Definition 2.7. Suppose an RDF node $z \in \mathcal{Z}$. Further suppose a set of quads $Q \subseteq \mathcal{Q}$. Let us define a function $nodeIn : \mathcal{Z} \times \mathcal{P}(\mathcal{Q}) \rightarrow \{true, false\}$, s.t. $nodeIn(z, Q) = true \iff \exists a, b \in \mathcal{Z} \wedge \exists c \in \mathcal{G} \wedge ((z, a, b, c) \in Q \vee \exists (a, z, b, c) \in Q \vee \exists (a, b, z, c) \in Q)$. Furthermore, let us define a set $Z_Q \subseteq \mathcal{Z}$ as a set of RDF nodes used in the quads Q, i.e., $Z_Q = \{z \in \mathcal{Z} \mid nodeIn(z, Q)\}$.

Definition 2.8. Suppose a graph $g \in \mathcal{G}$. Further suppose a set of quads $Q \subseteq \mathcal{Q}$. Let us define a function $graphIn : \mathcal{G} \times \mathcal{P}(\mathcal{Q}) \rightarrow \{true, false\}$, s.t. $graphIn(g, Q) = true \iff \exists x, y, z \in \mathcal{Z} \wedge (x, y, z, g) \in Q$. Furthermore, let us define a set $G_Q \subseteq \mathcal{G}$ as a set of named graph used in the quads Q, i.e., $G_Q = \{g \in \mathcal{G} \mid graphIn(g, Q)\}$.

[21] `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/#section-Reification`

2.2.2 RDF Serializations

"To facilitate the interchange of RDF data between information systems, a concrete serialization syntax is needed" [16]. The RDF/XML Syntax Specification [9] describes the serialization of RDF to XML. Various other serializations exist, such as the plain text serializations *N-Triples* [78], *N3* [11], and *Turtle* [10]. Turtle serialization extends the N-Triples notation with the selected features from N3, but excludes those features from N3, which require extension of the RDF data model [16].

In the rest of the thesis, the Turtle serialization is used for the serialization of all RDF data examples. Listing 1 shows a sample set of RDF triples serialized according to the Turtle syntax. As may be observed, RDF nodes of every RDF triple are serialized in the order of subject, predicate, and object; every triple is finished by a dot; URIs are enclosed with brackets. The Turtle syntax also supports `@prefix` directive, which allows declaration of a short prefix name for a long prefix repeatedly used in URIs (Lines 1 – 4) [10]. Literals are enclosed with quotations marks, optionally supplemented with a language tag (Line 7) or datatype (Line 9). "Two shortcuts are provided to combine several triples: A semicolon introduces another property of the same subject. A comma introduces another object with the same property and subject" [16]; the semicolon shortcut is introduced in Lines 6, 7 and 8.

```
1 @prefix pc: <http://purl.org/procurement/public-contracts#> .
2 @prefix dc: <http://purl.org/dc/terms/> .
3 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
4 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
5
6 <http://abc.com/pc/1> rdf:type pc:PublicContract ;
7   dc:title "Buying 100 cars for the police"@en ;
8   pc:contractingAuthority <http://abc.com/buyer/1> ;
9   pc:numberOfTenders "3"^^xsd:integer .
```

Listing 1: A public contract in Turtle RDF syntax

In [17], a TriG syntax is introduced, which extends the Turtle serialization with the possibility to express named graphs. Listing 2 is based on Listing 1, but all the triples about the public contract are contained within the named graph `<http://abc.com/ng/1>`. Furthermore, Listing 2 contains one more contract `<http://abc.com/pc/2>`.

```
@prefix pc: <http://purl.org/procurement/public-contracts#> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://abc.com/ng/1> {
    <http://abc.com/pc/1> rdf:type pc:PublicContract ;
        dc:title "Buying 100 cars for the police"@en ;
        pc:contractingAuthority <http://abc.com/buyer/1> ;
        pc:numberOfTenders "3"^^xsd:integer .

    <http://abc.com/pc/2> rdf:type pc:PublicContract ;
        dc:title "Road Construction"@en ;
        pc:contractingAuthority <http://abc.com/buyer/2> ;
        pc:numberOfTenders "1"^^xsd:integer .
}
```

Listing 2: A public contract in TriG RDF syntax

2.2.3 RDF Schema

The RDF vocabulary definition language RDF Schema (RDFS) [31] is a language describing groups of related resources (classes of resources) and the relationships between these classes. Two basic classes of RDFS are: `rdfs:Class` and `rdf:Property`; `rdfs:Class` is a class of resources that are RDF classes, `rdf:Property` is the class of properties of RDF classes [16]. The `rdf:type` property may be used to state that a resource is an instance of the class [31]; in Turtle syntax, `rdf:type` property is abbreviated as `a`.

RDFS vocabulary may be also used to describe inheritance relations between classes and properties. Property `rdfs:subClassOf` may be used to express that all instances of one class (the subject of the property `rdfs:subClassOf`) are also instances of another, more generic class (the object of the property `rdfs:subClassOf`); e.g., `foaf:Person` is a subclass of `foaf:Agent` [32]. Property `rdfs:subPropertyOf` may be used to express that all resources related by one property (the subject of the property `rdfs:subPropertyOf`) are also related by another, more generic property (the object of the property `rdfs:subPropertyOf`); e.g., property `w3po:wasCreatedBy`, representing that an artifact (document) was created by an agent, is a subproperty of a more generic property `w3po:wasAssociatedWith`, representing that an artifact was associated with an agent [100].

RDF Schema language primitives (including `rdfs:Class` and `rdf:Property`) are defined in two namespaces: namespace `http://www.w3.org/2000/01/rdf-schema#` which is conventionally associated with namespace prefix `rdfs:`, and namespace `http://www.w3.org/1999/02/22-rdf-syntax-ns#` associated with `rdf:` [31].

```
@prefix pc: <http://purl.org/procurement/public-contracts#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix gr: <http://purl.org/goodrelations/v1#> .

pc:PublicContract a rdfs:Class ;
    rdfs:label "Public contract"@en .

pc:contractingAuthority a rdf:Property ;
    rdfs:label "Contracting authority"@en ;
    rdfs:comment "Institution which issues a public contract,
        receives tenders to the contract and chooses a suitable
         supplier on the base of the conditions given by the
        contract. Cardinality 0..1"@en ;
    rdfs:domain pc:PublicContract ;
    rdfs:range gr:BusinessEntity.

pc:numberOfTenders a rdf:Property ;
    rdfs:label "Number of tenders received"@en ;
    rdfs:comment "Property for number of tenders received.
        Cardinality 0..1"@en ;
    rdfs:domain pc:PublicContract ;
    rdfs:range xsd:nonNegativeInteger .
```

Listing 3: Simplified public contracts ontology

Listing 3 shows an excerpt of the simplified public contracts ontology describing classes and properties using RDFS and being relevant for the domain of public procurement. The excerpt defines a class `pc:PublicContract` to represent the public contract concept and three properties: property `dc:title` relating a public contract with its title (a literal with the English language tag), property `pc:contractingAuthority` relating a public contract with the contracting authority, the buyer (an instance of a class `gr:BusinessEntity`), and property pc:numberOfTenders relating a contract with a typed literal holding the number of tenders for the contract. Property dc:title and class gr:BusinessEntity are defined in external vocabularies [49, 122]. Listing 1 shows the particular public contract being an instance of the class pc:PublicContract.

Unlike typical object oriented programming languages such as Java, where a class is defined "in terms of the properties its instances may have, the RDF vocabulary description language describes properties in terms of the classes of resource to which they apply" [31]; the predicate rdfs:domain of RDFS describes the domain of a property, the predicate rdfs:range describes the range of a property.

2.2.4 Web Ontology Language and Ontologies

RDFS is relatively limited in terms of the expressivity, therefore, applications requiring a more expressive ontology language should use the Web Ontology Language (OWL) [125]. OWL extends RDFS with additional modeling primitives. OWL primitives are defined in namespace http://www.w3.org/2002/07/owl#, conventionally associated with namespace prefix owl:.

OWL can describe more detailed characteristics of properties – it may express that a certain property is an inverse property of another property (using predicate owl:inverseOf), or that a certain property is equivalent to another property (owl:equivalentProperty). OWL can also formulate cardinality constraints on properties by using the predicates owl:minCardinality and owl:maxCardinality. Furthermore, OWL can also define constraints on the values of certain properties (predicates owl:allValuesFrom, owl:someValuesFrom). For further details about OWL, the reader is referred to [125].

Public Contracts Ontology (PCO)

To illustrate an example of OWL ontology, let us briefly present the Public Contracts Ontology (PCO)[22], an ontology for expressing details about public contracts, such as buyers of the contracts, tenders associated with the contracts, suppliers of these tenders, or prices of the contracts [104]. Figure 2.2 provides a brief overview of the classes and properties available in PCO. The preferred prefix for PCO is pc:, which stands for http://purl.org/procurement/public-contracts#. PCO ontology primitives are used in further examples in the thesis.

2.3 SPARQL

SPARQL is a family of W3C standards; the most important are SPARQL 1.1 Query Language [66], describing a declarative query language for RDF data, and SPARQL 1.1 Update [67], describing declarative language for specifying and executing updates to RDF data.

[22] http://purl.org/procurement/public-contracts#

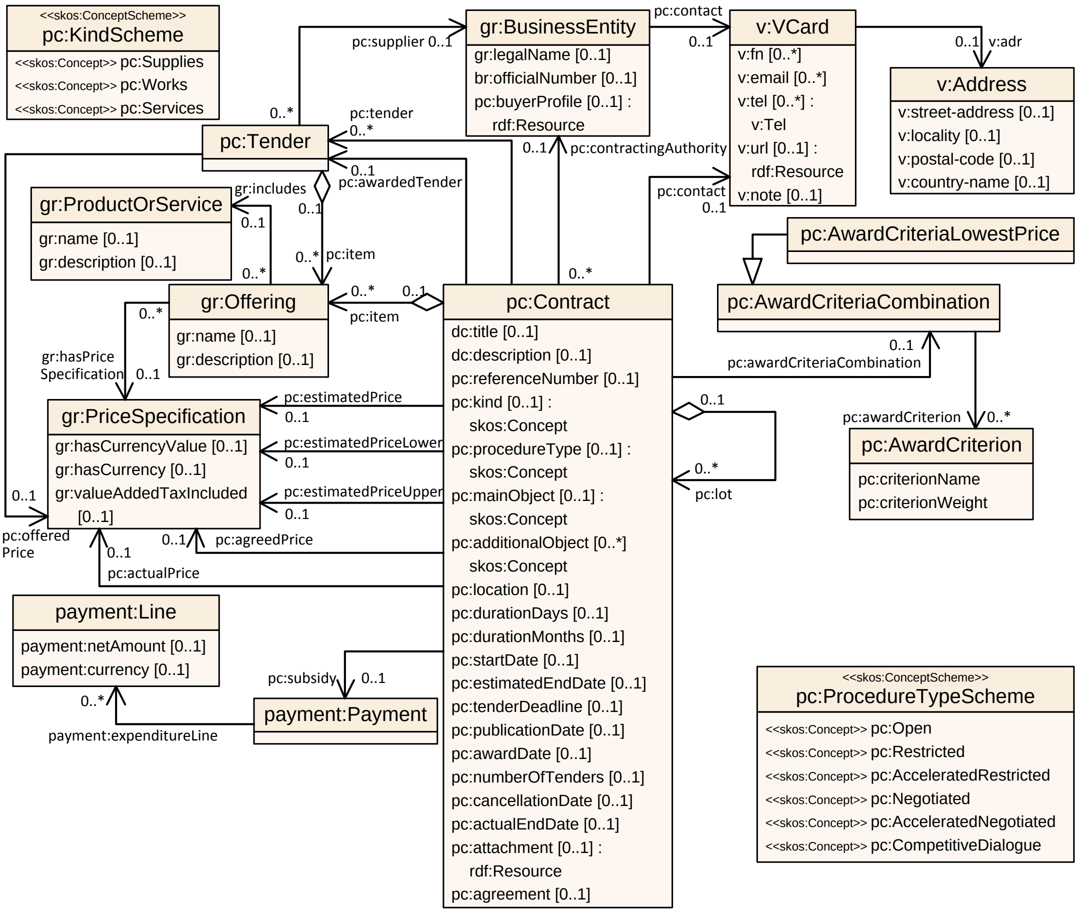


Figure 2.2: UML diagram of Public Contracts Ontology

Table 2.1: SPARQL 1.1. query response

title	**number**
“Buying 100 cars for the police”@en	“3”^^xsd:integer
“Road Construction”@en	“1”^^xsd:integer

2.3.1 SPARQL Query Language

```
PREFIX pc: <http://purl.org/procurement/public-contracts#>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?title ?number
FROM <http://abc.com/ng/1>
WHERE {
   ?s a pc:Contract ;
      dc:title ?title ;
      pc:numberOfTenders ?number .
}
```

Listing 4: Sample SPARQL 1.1. query

Listing 4 contains a sample SPARQL query, which selects the title of the public contract (i.e., the value of the predicate `dc:title`) and the number of tenders for that contract (i.e., the value of the predicate `pc:numberOfTenders`). If the query is executed against data described by Listing 2, the result of the query contains two solution (variable bindings) for the variables *?title* and *?number* as depicted in Table 2.1. Lines 1 – 3 of Listing 4 contain prefix declarations to introduce short prefixes which may be used in the query instead of the long prefixes of certain URIs.

Triple and Query Patterns

Lines 8 – 10 of Listing 4 contain three *triple patterns* of the query written using Turtle serialization syntax (see Section 2.2.2). A triple pattern is a basic building block of each query; it is a triple, which may contain variables at the position of a subject, a predicate, or an object; these variables start with a question mark, e.g., *?title* in Listing 4.

Lines 7 – 11 of Listing 4 define a *query pattern*, containing a WHERE clause and a *graph pattern.* SPARQL supports different types of graph patterns [66]. In the particular example above (Listing 4), the query pattern is formed by a *group graph pattern* (bounded by the curly brackets in Lines 7 and 11), which further contains a *basic graph pattern* formed by a set of triple patterns (Lines 8 – 10).

The basic idea of a SPARQL query execution is the *pattern matching.* A query pattern in Listing 4 consisting of a group graph pattern, which itself contains one basic graph pattern, can be matched against the RDF data in the database, if all the variables in the query pattern (i.e., *?title*, *?number*, *?s*) can be bound to particular values. A result of such binding is called a *solution.* The query may result in more solutions. Solutions may be further constrained by posing conditions on variable values, such as by adding FILTER[23] clause to the basic graph pattern.

Type of Query

Line 5 of Listing 4 defines the type of the query being executed – *SELECT*, *DESCRIBE*, *CONSTRUCT*, or *ASK*, which influences the result of the SPARQL query. The result of a SELECT query is a sequence of solutions, i.e., sets of variable bindings. The result of an ASK query is true, if there is at least one solution; otherwise false. The result of a DESCRIBE query is an RDF graph with data about the resource being described, e.g., DESCRIBE `dbpedia:Prague` gives as a result the set of RDF triples containing data about `<http://dbpedia.org/resource/Prague>`; the particular structure of the resulting data is query processor dependent. The result of a CONSTRUCT query is an RDF graph constructed from the given template – graph pattern with variables from the query pattern[24].

RDF Datasets

The named graphs data model described in Section 2.2 has been adopted with a slight modification as the data model underlying the SPARQL query language [16]. SPARQL introduces the term RDF dataset as follows.

Definition 2.9. An *RDF dataset* is a set $\{H, (u_1, H_1), (u_2, H_2), \ldots, (u_n, H_n)\}$, where H and each H_i are RDF graphs, and each u_i is an URI. All u_i are distinct. H is called the default graph; $\{(u_i, H_i)\}$, $1 \le i \le n$, is the set of named graphs.

[23] `http://www.w3.org/TR/sparql11-query/#rFilter`

[24] `http://www.w3.org/TR/sparql11-query/#construct`

The default graph allows the named graphs functionality of SPARQL to be optional and provides backward compatibility with RDF data models not supporting named graphs [16]. Line 6 of Listing 4 specifies the RDF dataset being queried, i.e., the graph `<http://abc.com/ng/1>`.

2.3.2 SPARQL Update Language

SPARQL 1.1 Update provides new types of queries, so that (1) RDF graphs can be created, cleared, dropped, copied, or moved and (2) RDF triples can be inserted into RDF graphs, deleted from RDF graphs, or updated. Listing 5 provides an example of SPARQL Update query, which renames all people with the given name "Bill" to "William" [25].

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>

DELETE { ?person foaf:givenName ‘Bill’ }
INSERT { ?person foaf:givenName ‘William’ }
WHERE {
  ?person foaf:givenName ‘Bill’
}
```

Listing 5: Sample SPARQL 1.1 Update query

2.4 Linked Data Tools

LOD2[26] technology stack[27] defines a set of tools usable for various operations with Linked Data. This section describes only Linked Data tools used in further chapters of the thesis.

Virtuoso

Virtuoso[28], a component of the LOD2 Stack, is a platform for data management, access, and integration. It includes a native RDF repository for storing, managing and querying RDF data.

Silk

Another component of the LOD2 stack is Silk[29], a popular tool for discovering and creating links in the Web of Data by interlinking previously disconnected RDF data sources [22]. The resources to be interlinked are described by the Silk-LSL language[30]. This flexible approach takes advantage of the richly structured RDF data. The descriptions of resources are used as inputs for any comparison; if the similarity of these descriptions is high enough, new link of the desired type is created (with certain probability).

[25] The example is taken from [67].

[26] `http://lod2.eu`

[27] `http://stack.lod2.eu/`

[28] `http://virtuoso.openlinksw.com/`

[29] `http://www4.wiwiss.fu-berlin.de/bizer/silk/`

[30] `http://www.assembla.com/wiki/show/silk/Link_Specification_Language`

2.5 Summary

In this section, we further motivated the needs for Linked Data. We described in detail two important building blocks of the Linked Data approach: (1) Resource Description Framework, its data model, serializations, and languages RDFS and OWL for describing vocabularies (ontologies), and (2) SPARQL, a family of W3C standards, which defines the languages for querying and updating RDF data. Finally, Virtuoso and Silk, Linked Data tools internally used by ODCleanStore, were briefly described.

3. ODCleanStore

ODCleanStore is a tool for Linked Data management. The goal of ODCleanStore is to enable RDF data processing and consequent data querying providing data consumers with trustworthy data, which may be customized according to their needs. This chapter may be comprehended as a base for other chapters particularizing the data integration and trustworthy data consumption. Figure 3.1 recalls the general idea of ODCleanStore.

ODCleanStore comprises of an *engine* running the server part of ODCleanStore and a graphical administration web interface to manage, monitor, and debug the server part. The interface of the engine consists of two web services – (1) an *input web service* that accepts new RDF data feeds and queues them for the processing in the *staging database* and (2) an *output web service* for querying the curated data in the *raw data mart.* Data processing module, i.e. the execution of data processing pipelines, is described in more detail in Section 3.1; query execution module is described in Section 3.2. Data integration component, executed as a part of the query execution module, is described in Chapter 4. Data filtering component, executed as a part of the query execution module and enforcing the consumers' provenance requirements is described in Section 5.9. The engine operates on top of two Virtuoso database instances:

- staging database (RDF cache) – Data submitted to ODCleanStore by the input web service is stored and queued there until being processed by the data processing pipeline.

- raw data mart (clean database) – Raw data mart contains data already being successfully processed by data processing pipelines. At the end of a pipeline processing data are copied to the raw data mart. Data in the raw data mart is used for querying.

ODCleanStore is developed at the Charles University in Prague, Faculty of Mathematics and Physics, as a part of the OpenData.cz initiative and the LOD2 FP7 project; it is published as a free software under the Apache License 2.0. For the full documentation of ODCleanStore, the reader is referred to the attached DVD or the project website[31].

3.1 Data Processing Module

In this section we describe the data processing in ODCleanStore and focus on the presentation of the predefined transformers being available.

The data processing flow in ODCleanStore is as follows. The input web service consumes an RDF *data feed* and stores it to the staging database. The RDF data feed is a set of named graphs including the *data graph* (the main named graph of the feed) and named graphs holding descriptive and provenance metadata. Based on the pipeline identifier within the feed's descriptive metadata, ODCleanStore engine launches the particular *data processing pipeline* containing an execution of the sequence of data transformers (data processing units), which may normalize

[31] `http://sourceforge.net/p/odcleanstore/`

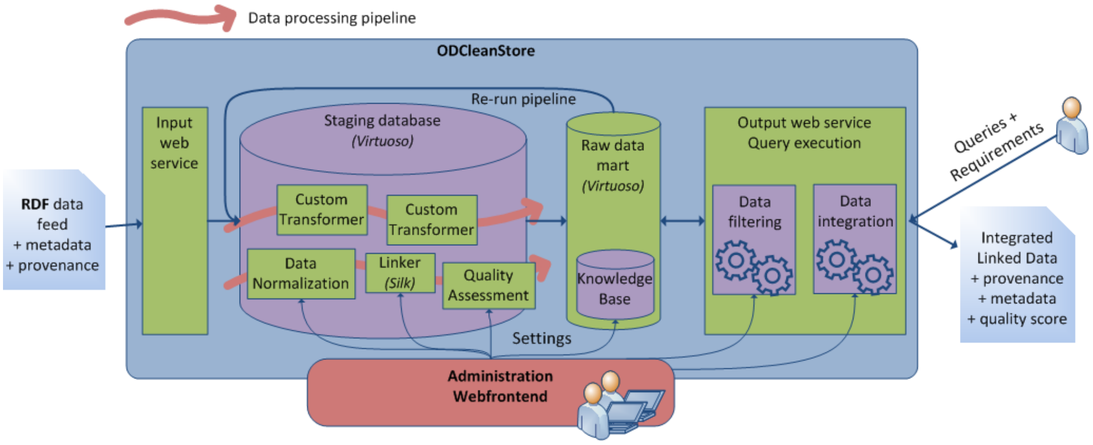


Figure 3.1: ODCleanStore – an overview

the data, deduplicate the data against or link the data to the data in the raw data mart, assess the quality of the data, or execute an arbitrary data transformation. The transformers may supplement the descriptive and provenance metadata in the feed and add auxiliary named graphs to the data feed, e.g., graphs containing the generated links to other resources or auxiliary data for the next transformer on the pipeline. After successfully executing all the transformers on the given pipeline, the data feed is stored to the raw data mart containing:

- the transformed data graph
- the descriptive metadata of the data feed optionally supplemented with further metadata provided by the transformers on the pipeline, such as the quality score written by the quality assessor transformer
- the provenance metadata of the data feed
- a set of auxiliary named graphs provided by the transformers on the pipeline, such as the `owl:sameAs` links generated by the transformer

Pipeline, the core concept of the data processing module, may process not only (1) the new incoming data accepted by the input web service, but also (2) data feeds already being stored in the raw data mart, which needs further refinement. In the latter case, if a certain data feed should be re-processed by the pipeline, a copy of that feed is created in the staging database, the appropriate pipeline is launched with the copied feed as the input and the resulting feed (after successful completion of the pipeline) replaces the original feed in the raw data mart. Although Virtuoso does not fully support transactions over RDF data, pipeline data processing in ODCleanStore is implemented in a way that data is kept consistent (See [129] for details).

Raw data mart in ODCleanStore may be populated on a regular basis with the data coming from one source; in that case, every such feed should be considered as an update of the previous feed. In ODCleanStore, if two data feeds are coming from one source, being submitted by the same agent, and has the same update tag [129], then the newer feed (data graph) is considered as an *update* of the previous one.

Transformer Interfaces

Every transformer – a data processing unit – is a pluggable Java class implementing the `Transformer` interface shown in Listing 6 and satisfying the further conditions defined in [129].

```
  public interface Transformer {
    void transformGraph(TransformedGraph feed,
        TransformationContext context)
      throws TransformerException;

    void shutdown() throws TransformerException;
}
```

Listing 6: ODCleanStore – transformer interface

The actual data processing is implemented in `transformGraph()` method. All required information is passed in its arguments, one with the reference to the staging database pointing to the feed it should transform, the second one with the context of the feed (e.g., the pipeline processing the feed). Data between transformers are not passed in memory, but rather stored in the staging database and only URIs of the feeds and connection credentials for accessing the staging database are given to the transformer. This minimizes the need for complicated interfaces for data passing, makes it easier to work with large data, let the transformer to choose its own method for accessing the database and give it the full power of SPARQL query and update languages [129].

Predefined transformers

ODCleanStore is shipped with few types of transformers predefined for the most common data management operations:

- *Data normalizer* transformer (see Section 3.1.1)
- *Linker* transformer (Section 3.1.2)
- *Quality assessor* transformer (Section 3.1.3)
- *Quality aggregator* transformer (Section 3.1.4)

Instances of the predefined transformers are configured via groups of data normalization, linkage, and quality assessment policies, respectively. Each instance of a predefined transformer can accept multiple groups of policies. As a result, it is possible to assign all interrelated policies to a transformer instance while it is still possible (1) to avoid duplication of policies in different groups and (2) to keep the policies logically relating in one group. Groups of policies may be shared among pipelines.

Apart from that, if more specialized transformers are needed, they can be developed and used. For example, one can write a custom transformer implementing the defined transformer interface and checking against the Czech Business Register whether data about companies holds valid identification numbers; if not, identification numbers may be taken from the Czech Business Register as it is considered to be an authoritative source of information about Czech companies.

Transformer Class and Transformer Instance

It is important to distinguish between a *transformer class* and a *transformer instance*. By a transformer class, we mean the Java class which implements the transformer interface and is registered in ODCleanStore. A transformer instance is an assignment of a transformer class to a pipeline, different instances of the same class may have different configurations and also different groups of policies assigned to them. For example, the quality assessor transformer class is registered in ODCleanStore by default. The user can create two pipelines and assign the quality assessor transformer class to each of them, thus creating two transformer instances. If it is obvious from the context whether we mean transformer (its class) or transformer (its instance), we use just the term transformer.

3.1.1 Data Normalizer

Data normalizer is a built-in transformer class, aimed to be applied early in the whole data processing pipeline to simplify work of other transformers. Its main goal is to correct errors in the graphs of the incoming data feeds and to remove inconsistencies in the data w.r.t. the ontology describing the data. Data normalizer can also execute transformation tasks. For example, a data normalizer may correct typos in the names of months, convert dates to a given format, adjust identification numbers of public contracts so that they satisfy certain prescriptions, change names of predicates, etc. More complex data cleansing and data transformation tasks, which cannot be addressed by data normalizers, must be targeted in a separated (custom) transformer class.

The list of sample cleansing and transformation scenarios doable by the data normalizer and relevant for the public procurement domain (data described by the Public Contracts Ontology introduced in Section 2.2.4) is as follows:

1. The value of the predicate `gr:hasCurrencyValue` (holding the price of the contract) is a valid float number (e.g., “1276,00” is converted to “1276.00”).

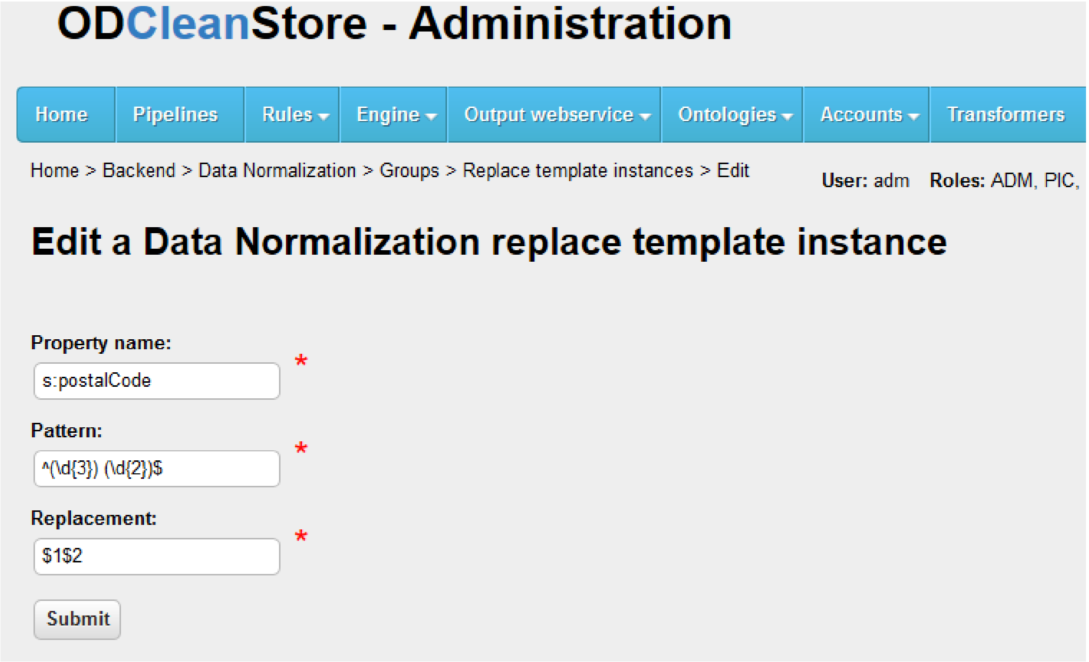
2. Postal codes (values of the predicate `vcard:postal-code`) in addresses are unified, e.g., 143 06 is converted to 14306.

3. Abbreviations of the company names are expanded, well-known typos corrected, extra spaces in the company names removed, street names expanded (e.g., “Neuburg Str.” → “Neuburg Street”).

4. Data about public contracts are enriched with new predicates/classes (e.g., predicate `pc:numberOfTenders` may be computed from the data by counting the number (`pc:Tender`) instances associated with the given public contract).

5. Outdated properties from older versions of the Public Contracts Ontology are renamed.

The functionality of the particular instance of the data normalizer is driven by the groups of data normalization policies assigned to that instance. Each data normalization policy is a sequence of policy fragments. Each rule fragment has its type – *INSERT*, *DELETE*, or *MODIFY* – and its body depending on its


ODCleanStore - Administration
Home
Pipelines
Rules
Engine
Output webservice
Ontologies
Accounts
Transformers
Home > Backend > Data Normalization > Groups > Replace template instances > Edit
User: adm
Roles: ADM, PIC,
Submit

Edit a Data Normalization replace template instance

Property name: s:postalCode *

Pattern: ^(\d{3}) (\d{2})$ *

Replacement: $1$2 *

Figure 3.2: Data normalization policies' settings in ODCleanStore

type and being prescribed by the grammar for the proper SPARQL construct – `INSERT`[32], `DELETE`[33], or `MODIFY`[34]. Policy fragments are applied one by one; based on the type of the policy fragment, certain data is inserted, deleted or modified and the result is immediately visible in the staging database. A sample policy fragment renaming predicate `pc:oldProp` is:

```
DELETE {?s pc:oldProp ?o} INSERT {?s pc:newProp ?o} WHERE {GRAPH
                  $$graph$$ {?s pc:oldProp ?o}},
```

Due to incomplete support for SPARQL 1.1 in Virtuoso, in particular, because of the missing `BIND(`*expression* `AS` *var*`)`[35], it was necessary to allow use of subqueries for data manipulation and transformation by introducing `$$graph$$` macro, which is replaced before the SPARQL execution with the name of the data graph being currently processed by the data processing pipeline.

SPARQL Update is powerful enough to remove, insert, or modify certain RDF data. Still, it is easy to be learned and is standardized by W3C. Data normalization policies may be either directly written in SPARQL or they may be specified using the predefined HTML templates in the administration web interface of ODCleanStore for common data normalization operations, such as "to replace the value of a property with another value"; the filled HTML templates are internally converted to the SPARQL rule fragments. Figure 3.2 shows the instantiation of the HTML template for Cleansing and transformation scenario 2.

Data normalizers address Problem P2 by employing SPARQL Update queries for automated data normalization. Complex cleansing and data transformation

[32] `http://www.w3.org/TR/sparql11-query/#rInsertClause`

[33] `http://www.w3.org/TR/sparql11-query/#rDeleteClause`

[34] `http://www.w3.org/TR/sparql11-query/#rModify`

[35] `http://www.w3.org/TR/sparql11-query/#bind`

which cannot be achieved by employing progression of SPARQL normalization policy fragments can be targeted by an instance of a generic custom transformer.

ODCleanStore supports generation of data normalization policies from ontologies behind the data, therefore, certain aspects of the incoming data may be checked by automatically prepared policies [129].

3.1.2 Linker

Linker transformer is a built-in transformer class. The main purpose of this component is to interlink URIs which represent the same real-world entities by generating `owl:sameAs` links. It can be also used for creating other types of links between differently related URIs. The Silk framework is used as the linking engine. Sets of linkage policies for the engine are written in Silk-LSL. The generated links are stored in an auxiliary graph of the data feed, so that the generated links can be recomputed and regenerated any time. Linker always links the processed data feed against the raw data mart and can also link the data feed against itself. The list of sample linkage scenarios relevant for the public procurement domain is as follows:

1. Creating `owl:sameAs` links between business entities (companies) having the same identification numbers but different URIs.

2. Creating `owl:sameAs` links between the same cities represented by different URIs.

3. Linking contracts to CPV codes[36] via the predicate `pc:mainObject`.

4. Linking contracts to the NUTS region codes[37] via the predicate `pc:location`.

Silk accompanies every generated link with the probability that the link is correct according to the Silk policy used. Pipeline administrator can set a threshold for these probabilities; if the probability of link correctness is higher than the threshold, the link is created.

Linker addresses Problem P1 by providing ways how to declaratively describe links between two resources. It also supports P3 by creating `owl:sameAs` links for deduplication of entities. The details about the linker transformer may be found in [129].

Related Work

We do not intend in our thesis to conduct any research in the area of record linkage, but rely on the research actively undertaken by other communities. Most of the research in this area is dealing with the precision and recall of the created links, which might be high for certain domains but low for the others. Paper [90] improves the performance of Silk without sacrificing recall by introducing multidimensional index which reduces the number of comparisons Silk needs to finish its computation.

[36] `http://simap.europa.eu/codes-and-nomenclatures/codes-cpv/codes-cpv_en.htm`

[37] `http://epp.eurostat.ec.europa.eu/portal/page/portal/nuts_nomenclature/introduction`

The ability to actively learn certain linkage policies based on the already created supervised set of links is also researched these days. Paper [89] introduces a way, how Silk can learn policies based on a manually supplied set of reference links.

Silk also provides a graphical user interface, Silk Workbench, which supports the user with the ability to manage linkage policies, view results, and also give a feedback to Silk regarding the correctness of the created links. As a result, Silk can learn from good and bad samples of generated links. ODCleanStore enables to import the Silk policies from or export the Silk policies to Silk Workbench.

Apart from Silk, there is a tool called Limes, a link discovery framework for metric spaces[38]. Such tool provides similar functionalities as Silk. According to [144], Limes outperforms Silk in large-scale matching tasks; however, it is restricted to a metric space. A result, semi-metrics, e.g., JaroWinkler [163], cannot be used. Furthermore, Limes does not posses any user interface where users can give feedback to the correctness of the generated links. In the future, ODCleanStore can support more linking tools, not just Silk.

3.1.3 Quality Assessor

Quality assessor is a special transformer class, which can assign a named graph quality score to the processed data feed based on its data graph compliance with a certain group of quality assessment (QA) policies prepared by domain experts.

The list of sample QA policies, the public procurement data should adhere to and the quality assessor should enforce, is as follows:

1. The date held by the predicate `pc:awardDate` is later than the date held by the predicate `pc:publicationDate`.
2. The predicate `pc:referenceNumber` contains a value satisfying the given regular expression – it starts with “0_”.
3. The predicate `pc:actualPrice` exists.
4. At least one contact person of the contracting authority responsible for the given contract is available.
5. The summary of the contract award criteria weights is 100%.

The functionality of the particular instance of the quality assessor is driven by the groups of QA policies assigned to that instance. Each QA policy is defined as follows. A QA policy $p \in P_{QA}$ is a tuple $(cond, weight)$, where $cond \in C$ and $weight = w(p)$, where $w : P_{QA} \rightarrow [0, 1]$ quantifies the weight of the policy p. P_{QA} is the infinite set of all QA policies. Set C is a set of all valid `GroupGraphPatterns`[39] within the `WHERE` clause[40] of a SPARQL `SELECT` query, optionally followed by a *solution modifier*[41] containing expressions like `GROUP BY`, `HAVING`, or `ORDER BY` [66]. We do not use SPARQL `ASK` queries, which would be sufficient in terms of the

[38] http://aksw.org/Projects/LIMES.html

[39] http://www.w3.org/TR/rdf-sparql-query/#rGroupGraphPattern

[40] http://www.w3.org/TR/sparql11-query/#rWhereClause

[41] http://www.w3.org/TR/sparql11-query/#rSolutionModifier

resulting answer, but which do not allow solution modifiers, such as `GROUP BY`. The following condition $cond \in C$ addresses QA Policy 2:

```
{ ?s pc:referenceNumber ?n.  FILTER regex (?n, "^0_") }
```

Policies may also be automatically generated from ontologies. Table 3.1 lists sample QA policies generated from ontologies. These QA policies are automatically used together with other QA policies based on the `rdf:type` property in the data feed.

QA Policy Application

A QA policy $p = (cond, weight) \in P^a_{QA}$ can be successfully applied on the data graph $g \in \mathcal{G}$ if and only if the SPARQL query `SELECT * FROM NAMED` g `WHERE` $\{cond\}$ returns some solutions; $P^a_{QA} \subseteq P_{QA}$, P^a_{QA} denotes the list of policies defined by the quality assessor a. The successful application of a policy is expressed as $a_{QA}(p, g) = true$; otherwise, if the policy was not successfully applied, $a_{QA}(p, g) = false$; $a_{QA} : P_{QA} \times \mathcal{G} \rightarrow \{true, \ false\}$.

Let us introduce a function $s_{ng} : \mathcal{G} \rightarrow [0, 1]$ in Formula 3.1, s.t. $s_{ng}(g)$ computes the *named graph quality score* of the data graph $g \in \mathcal{G}$ and is based on the weights of the QA policies P^a_{QA} successfully applied to g, i.e.:

$$s_{ng}(g) = \beta \cdot \prod_{\{p \in P^a_{QA} | a_{QA}(p,g) = true\}} (1 - w(p)) \tag{3.1}$$

Quality assessor uses negative policies, thus, if the QA policy was successfully applied, there is some problem with the data and the named graph quality score is decreased; the higher the weight of policy p, the more the named graph quality score is decreased. The default score, β, is typically equal to 1.

The score of the graph, $s_{ng}(g)$, computed by a *quality aggregator* transformer, is stored as a value of the predicate `odcs:score` in the descriptive metadata graph of the data feed containing g.

Related Work

Information Quality is usually described in different works by a series of *IQ dimensions* which represent a set of desirable characteristics for an information resource [112, 162, 7, 141]. Wang & Strong [162] present an extensive survey of IQ dimensions, based on the results of the questionnaire given to the panel of human subjects; papers [112, 1] cover IQ dimensions for the Web.

ODCleanStore currently checks whether the data graphs satisfy certain QA policies, which may check consistency, completeness, or accuracy of the data graphs. In the future, the QA transformer will assess not a single score, but a

Table 3.1: Example of QA policies generated from ontologies

Type of property	**Constraint checked**
`owl:FunctionalProperty`	$[x, y_1], [x, y_2] \in p \rightarrow y_1 = y_2$
`owl:InverseFunctionalProperty`	$[x_1, y], [x_2, y] \in p \rightarrow x_1 = x_2$
`skos:ConceptScheme` [130]	$[x, y] \in p \rightarrow y \in p.hasTopConcept$

vector of quality scores, each score corresponding to evaluation of one IQ dimension. It will also take into account outputs of data normalizers and linkers, not just the incoming data graphs.

3.1.4 Quality Aggregator

Quality aggregator is a transformer class, which aggregates the named graph quality scores of the currently processed data feed and all the data feeds stored in the raw data mart and published by the same entity as the currently processed data feed. In other words, quality aggregator computes a *quality score of the publisher*.

Suppose that $pub(g)$ denotes the publisher (e.g., `http://isvzus.cz`) of the named graph $g \in \mathcal{G}$ specified as a value of the descriptive metadata predicate `odcs:publishedBy`. Suppose that $PU = \{pub(g) \mid g \in \mathcal{G}\}$ is the set of all publishers. A score of the publisher $u \in PU$ is computed using a function $s_{pu} : PU \to [0, 1]$ as depicted in Formula 3.2, i.e., as the weighted average of the quality scores s_{ng} of named graphs published by u weighted by the number of triples in these graphs.

$$s_{pu}(u) = \frac{\sum_{\{g \in \mathcal{G} | pub(g)=u\}} |g| \cdot s_{ng}(g)}{\sum_{\{g \in \mathcal{G} | pub(g)=u\}} |g|} \tag{3.2}$$

The score of the publisher, s_{pu}, computed as part of *quality aggregator* transformer, is stored in a general space of the raw data mart as a value of the predicate `odcs:publisherScore`.

3.2 Query Execution

Data processing fills the raw data mart with the curated data. The query execution module runs on top of the raw data mart. Data consumers may query the raw data mart and, as a result, they receive relevant and trustworthy Linked Data being integrated and filtered according to their needs. We describe the data integration component in Chapter 4. Section 5.9 discuses in more detail the data filtering component applying provenance requirements of data consumers. In the further explanation of the query execution module in this section, we suppose that *no* data filtering is required and that the data integration runs with the default settings.

3.2.1 Types of Supported Queries

A data consumer can query the raw data mart through the output web service of ODCleanStore. The output web service is a REST web service which can be accessed using both GET and POST HTTP methods equivalently. The output web service supports *URI*, *keyword*, *named graph* and *metadata* queries:

- URI query lets the data consumer to specify URI of the resource, he is interested in; as a result, he gets the integrated view on the data about that resources, i.e., the result of the query will contain the integrated facts involved in the raw data mart about the given resource regardless of the data graph the facts are originating from.

- Keyword query allows the consumer to specify a set of keywords; as a result, triples involving literals containing all these keywords are integrated and presented to the data consumer.
- Named graph query allows the consumer to retrieve all the data from a certain data graph.
- Metadata query allows the consumer to get descriptive and provenance metadata of a selected data graph. Furthermore, the metadata query also returns the list of QA policies applied to that data graph by quality assessors.

3.2.2 Query Execution Motivational Scenario

In this section, we present a motivational scenario, which shows how data consumers may use the types of supported queries, described in Section 3.2.1, to obtain data they are interested in.

Scenario 3.1. Suppose that the raw data mart of ODCleanStore contains data about the German city of Berlin coming from multiple LOD cloud sources – DBpedia[42], GeoNames[43], and Freebase[44]. Further, suppose that Alice, a data consumer and journalist, is writing a short history of the city of Berlin in Germany, for which she requires data about Berlin.

Suppose that Alice does not know the URI of the city of Berlin. Therefore, she submits the keyword "Berlin" to the query execution module of ODCleanStore, i.e., invokes the keyword query. As a result, the list of relevant triples is returned as depicted in Figure 3.3.

Alice may select the URI resource representing the city of Berlin, i.e., `dbpedia:Berlin`. By selecting that resource, she invokes the underlying URI query. As a response, she receives all the information ODCleanStore knows about `dbpedia:Berlin` integrated from all the available sources. An excerpt of the HTML result on the URI query `dbpedia:Berlin` is in Figure 3.4.

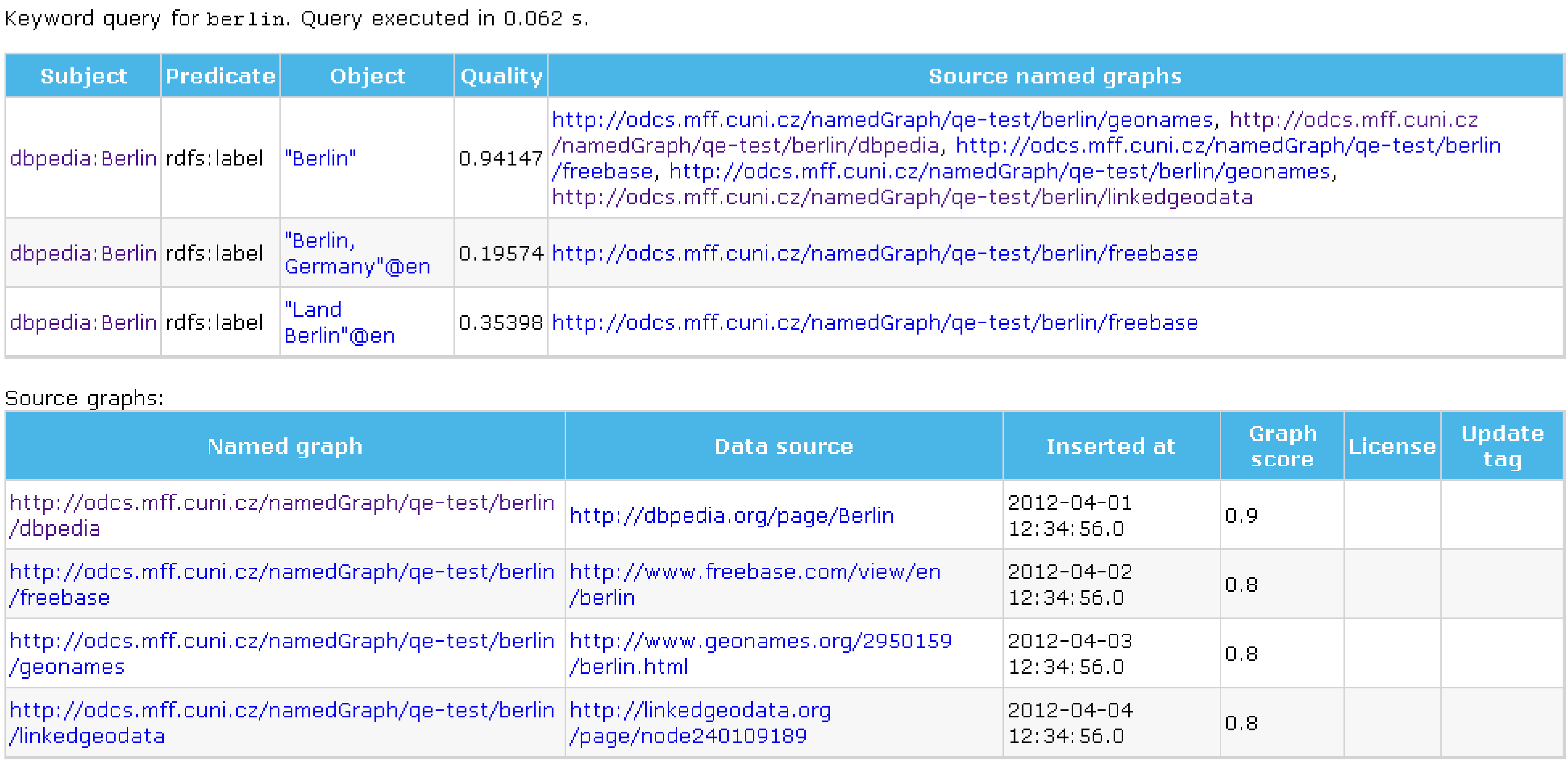
Alice may browse the integrated view on the data about Berlin (Figure 3.4), she may click on any URI resource to invoke further URI queries to ODCleanStore. Alice may also examine the metadata of the integrated triples to see source graphs – data graphs from which the particular integrated triple originates (see column "Source named graphs" in Figure 3.4) – and the quality score of the integrated triples (see column "Quality").

If Alice is interested in the original data from a certain source graph, e.g., `<http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia>`, she may click on the source graph in the column "Source named graphs" (see Figure 3.4) and, as a result, named graph query is invoked and Alice is presented with all the triples contained in the selected source graph (see Figure 3.5).

[42] `http://dbpedia.org`

[43] `http://www.geonames.org/`

[44] `http://www.freebase.com/`



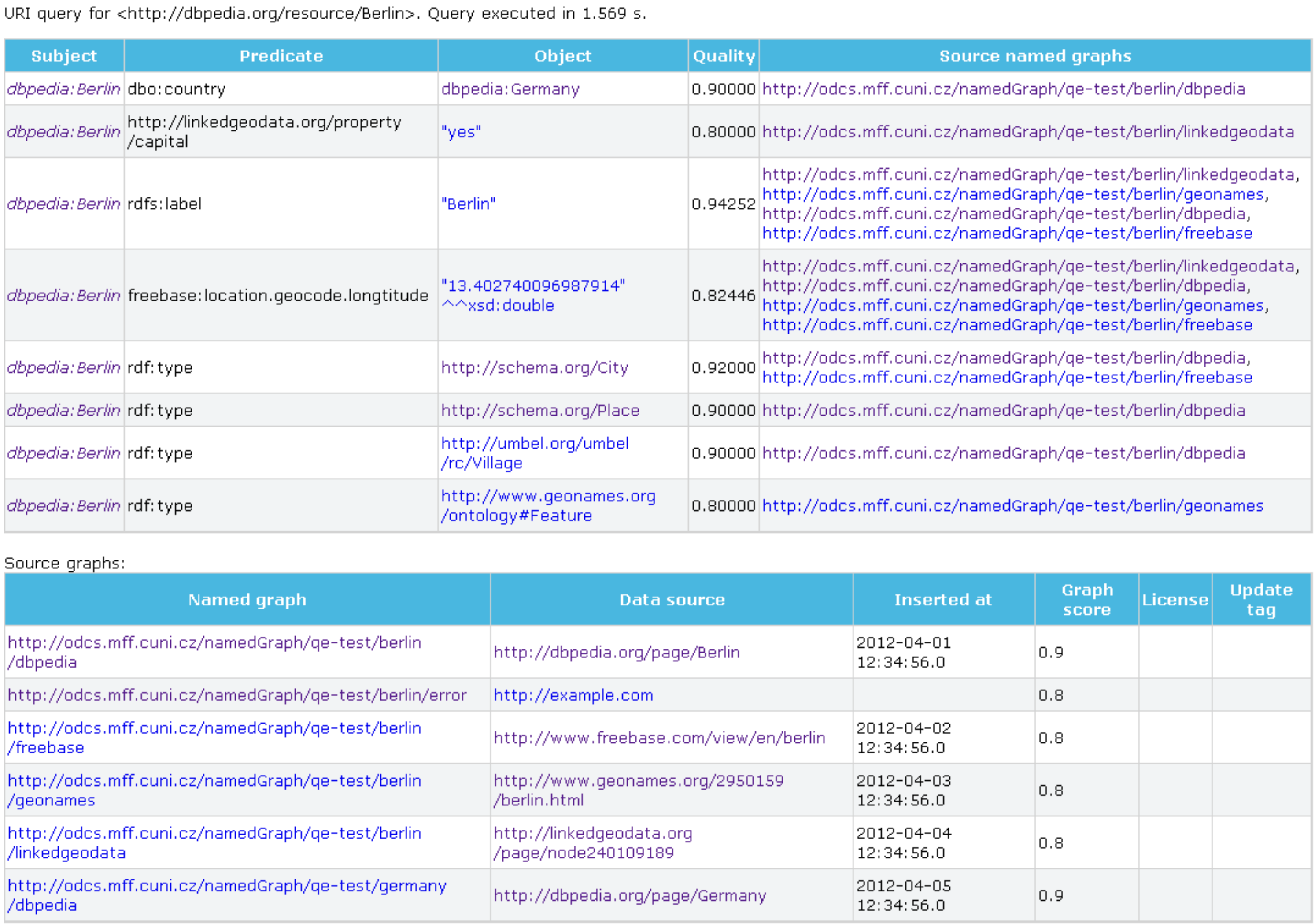
Keyword query for `berlin`. Query executed in 0.062 s.

Subject	Predicate	Object	Quality	Source named graphs
dbpedia:Berlin	rdfs:label	"Berlin"	0.94147	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/linkedgeodata
dbpedia:Berlin	rdfs:label	"Berlin, Germany"@en	0.19574	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase
dbpedia:Berlin	rdfs:label	"Land Berlin"@en	0.35398	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase

Source graphs:

Named graph	Data source	Inserted at	Graph score	License	Update tag
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia	http://dbpedia.org/page/Berlin	2012-04-01 12:34:56.0	0.9		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase	http://www.freebase.com/view/en/berlin	2012-04-02 12:34:56.0	0.8		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames	http://www.geonames.org/2950159/berlin.html	2012-04-03 12:34:56.0	0.8		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/linkedgeodata	http://linkedgeodata.org/page/node240109189	2012-04-04 12:34:56.0	0.8		

Figure 3.3: Example of the keyword query response in HTML



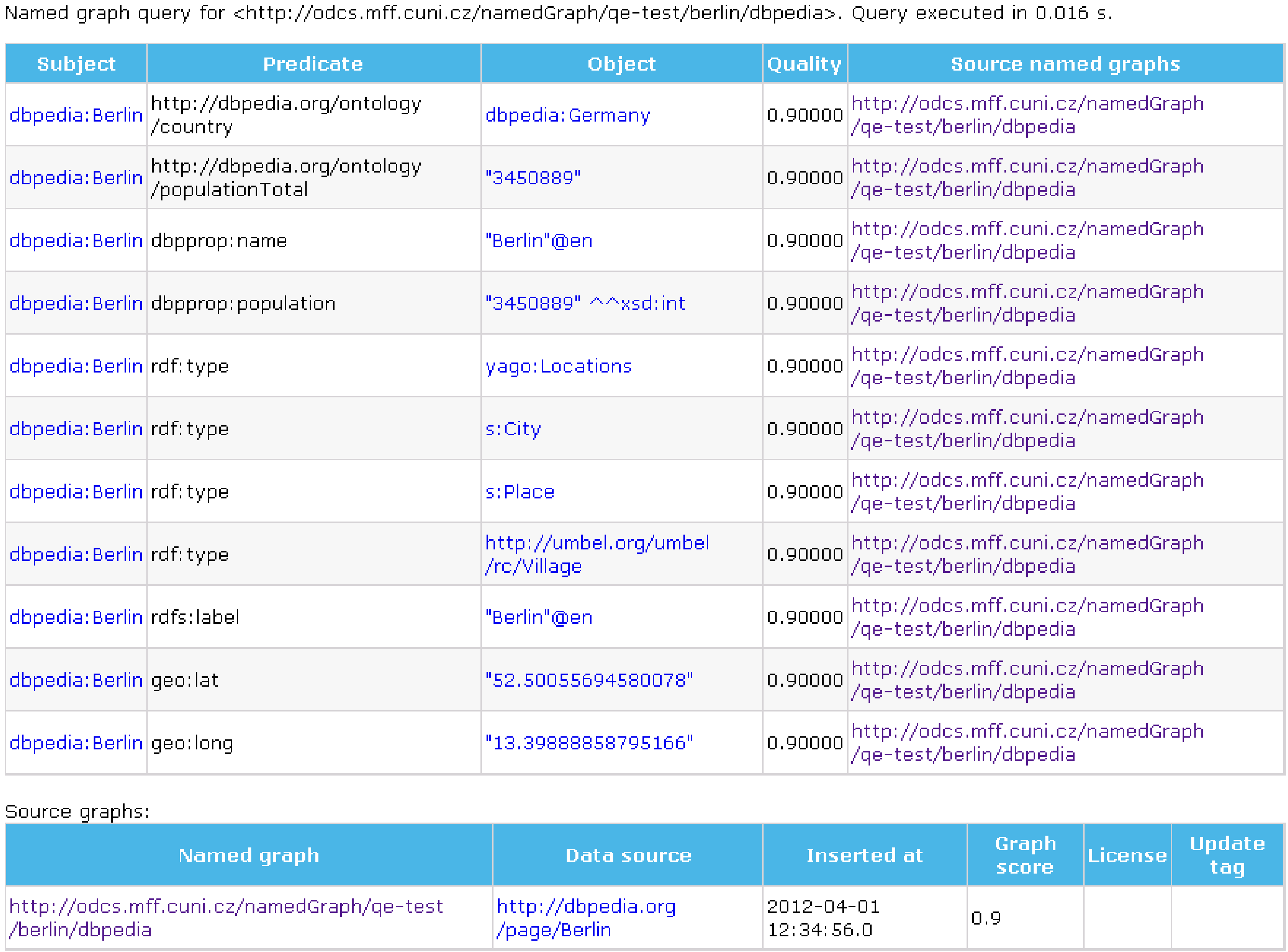
URI query for <http://dbpedia.org/resource/Berlin>. Query executed in 1.569 s.

Subject	Predicate	Object	Quality	Source named graphs
dbpedia:Berlin	dbo:country	dbpedia:Germany	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	http://linkedgeodata.org/property/capital	"yes"	0.80000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/linkedgeodata
dbpedia:Berlin	rdfs:label	"Berlin"	0.94252	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/linkedgeodata, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase
dbpedia:Berlin	freebase:location.geocode.longtitude	"13.402740096987914"^^xsd:double	0.82446	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/linkedgeodata, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase
dbpedia:Berlin	rdf:type	http://schema.org/City	0.92000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia, http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase
dbpedia:Berlin	rdf:type	http://schema.org/Place	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdf:type	http://umbel.org/umbel/rc/Village	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdf:type	http://www.geonames.org/ontology#Feature	0.80000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames

Source graphs:

Named graph	Data source	Inserted at	Graph score	License	Update tag
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia	http://dbpedia.org/page/Berlin	2012-04-01 12:34:56.0	0.9		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/error	http://example.com		0.8		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/freebase	http://www.freebase.com/view/en/berlin	2012-04-02 12:34:56.0	0.8		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/geonames	http://www.geonames.org/2950159/berlin.html	2012-04-03 12:34:56.0	0.8		
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/linkedgeodata	http://linkedgeodata.org/page/node240109189	2012-04-04 12:34:56.0	0.8		
http://odcs.mff.cuni.cz/namedGraph/qe-test/germany/dbpedia	http://dbpedia.org/page/Germany	2012-04-05 12:34:56.0	0.9		

Figure 3.4: Example of the URI query response in HTML



Named graph query for <http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia>. Query executed in 0.016 s.

Subject	Predicate	Object	Quality	Source named graphs
dbpedia:Berlin	http://dbpedia.org/ontology/country	dbpedia:Germany	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	http://dbpedia.org/ontology/populationTotal	"3450889"	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	dbpprop:name	"Berlin"@en	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	dbpprop:population	"3450889" ^^xsd:int	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdf:type	yago:Locations	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdf:type	s:City	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdf:type	s:Place	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdf:type	http://umbel.org/umbel/rc/Village	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	rdfs:label	"Berlin"@en	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	geo:lat	"52.50055694580078"	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia
dbpedia:Berlin	geo:long	"13.39888858795166"	0.90000	http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia

Source graphs:

Named graph	Data source	Inserted at	Graph score	License	Update tag
http://odcs.mff.cuni.cz/namedGraph/qe-test/berlin/dbpedia	http://dbpedia.org/page/Berlin	2012-04-01 12:34:56.0	0.9		

Figure 3.5: Example of the named graph query response in HTML

Furthermore, each source graph in Figure 3.4 is accompanied with its metadata, such as when the source graph was inserted to ODCleanStore (see column "Inserted at" in Figure 3.4), its named graph quality score s_{ng} (see column "Graph score"), and from which original (primary) source it was obtained (see column "Data source"). Furthermore, Alice may click on any source graph in the bottom table of Figure 3.4; as a result, metadata query is invoked and all available descriptive and provenance metadata about the selected graph are presented to Alice.

The interface presented in Figures 3.3, 3.4, and 3.5 is a part of the standard ODCleanStore distribution. However, it is meant as a prototype interface to test the capabilities of the query execution module of ODCleanStore, not as a solid user interface which the journalists, such as Alice, may directly use. Alice would need nicer interface, which further hides technical details of Linked Data (such as URIs), provides the capability of adjusting the visualization of the returned data, hiding data, etc.

3.2.3 Query Format

Table 3.2 gives details on the query format of the supported types of queries – it lists parameters that can be used with the URI, keyword, named graph, and metadata queries. The `uri` parameter is required for URI, named graph, and metadata queries, `kw` parameter is required for keyword query. The parameter `format` holds the desired format of the response, which might be HTML, TriG, or RDF/XML. In Section 4.8, we will extend the query format to incorporate data consumer's requirements on the data integration; in Section 5.9, we suggest how the query format can be extended to support data provenance requirements.

Name	**Description**	**Possible values**	**Default value**
`uri`	searched URI; *used for URI, named graph, and metadata queries*	*string*	*N/A*
`kw`	searched keyword(s); *used for keyword query*	*string*	*N/A*
`format`	format of the response	`html`, `trig`, `rdfxml`	`html`

Table 3.2: URI, keyword, and named graph query parameters

URI, Named Graph, and Metadata Queries Query

The value of the `URI` parameter must be either a full valid URI, or a prefixed name. Available prefixes, which may be used in the queries, are managed in the administration frontend of ODCleanStore [129].

Keyword Query

The kw parameter can contain one or more keywords separated by whitespaces. If a keyword itself contains spaces, it may be enclosed in double quotes. The output web service looks for literals that contain all of the keywords. Keywords can also contain the * wildcard, but they must begin with at least four non-wildcard characters. The output web service also looks for an exact match of the entire kw value (i.e., without any division to keywords). If the kw value is a number, then numeric typed literals will also match; if the kw value is formatted as xsd:dateTime[45], then xsd:dateTime typed literals will also match.

3.2.4 Result Format

The parameter format in Table 3.2 holds the desired format of the response, which might be HTML, TriG, or RDF/XML. Since the main purpose of the output web service is to be used by the applications, such as the editor Alice is using to browse the raw data mart, the HTML resulting format is mainly for illustration purposes. RDF/XML format and TriG format are both used for machines/applications consuming the responses on the queries. However, RDF/XML does not support quads, thus, descriptive and provenance metadata is removed from RDF/XML representation. In the further description of the queries' result, we focus on the description of the richest format, TriG; the detailed description of the rest of the formats can be found in [129].

URI and Keyword Queries

The result of the query execution module contains integrated triples returned as a response to the URI or keyword query and descriptive and provenance metadata for the sources of the integrated triples. The URIs in the response are replaced with the relevant labels, which may be specified in the administration interface of ODCleanStore [129]. An HTML example of the result on the URI query dbpedia:Berlin is in Figure 3.4, an HTML example of the result on the keyword query is in Figure 3.3. Listing 7 contains an excerpt of the corresponding TriG result for the same query. Such result contains:

- integrated triples returned as a response to the query, each one placed in a unique named graph, called *triple named graph* (see Lines 11 – 14 and Lines 16 – 19)
- metadata of the integrated triples associated with the triple named graphs (Lines 24 – 33), i.e., their integrated quality scores (held by the value of the predicate odcs:quality, Lines 26 and 32) and source graphs (held by the value of the predicate odcs:sourceGraph, Lines 27, 28, and 33)
- metadata of source graphs (Lines 35 – 48), i.e., source named graphs quality scores, s_{ng}, produced by quality aggregators (held by the predicate odcs:score, Lines 37 and 46), quality scores of the publishers (odcs:publisherScore, Line 42), the publishers of the source graphs (prov:publishedBy,

[45] http://www.w3.org/TR/xmlschema-2/#dateTime-lexical-representation

Line 40), when the source graphs were inserted to ODCleanStore (`prov:insertedAt`, Lines 38 and 47), where the data was extracted from (`prov:source`, Lines 39 and 48), and the licenses of the sources (`dc:license`, Line 41)

- metadata about the query response itself – a title (`dc:title`, Line 52), date (`dc:date`, Line 53), number of result triples (`odcs:totalResults`, Line 54), the query – the value of the `uri` parameter (`odcs:query`, Line 55), and links to each resulting integrated triple (`odcs:result`, Lines 56 and 57)

```
1  @prefix odcs: <http://opendata.cz/infrastructure/odcleanstore/>
       .
2  @prefix odcsRes: <http://opendata.cz/infrastructure/
      odcleanstore/query/results/> .
3  @prefix odcsData: <http://opendata.cz/infrastructure/
      odcleanstore/data/> .
4  @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
6  @prefix dc: <http://purl.org/dc/terms/> .
7  @prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
8  @prefix prov: <http://purl.org/provenance#> .
9  @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
10
11 # Resulting data
12 odcsRes:1 {
13   dbpedia:Berlin rdfs:label "Berlin"@en .
14 }
15
16 odcsRes:2 {
17   dbpedia:Berlin dbpedia-owl:populationTotal
18     "3420768"^^xsd:int .
19 }
20
21 #Descriptive metadata of the query
22 <http://opendata.cz/infrastructure/odcleanstore/query/metadata>
       {
23
24   #Descriptive metadata of the integrated triple
25   odcsRes:1
26     odcs:quality 0.92 ;
27     odcs:sourceGraph odcsData:e0cdc9d7-e2d8-4bde ;
28     odcs:sourceGraph odcsData:b68e21f7-363f-4bfd .
29
30   #Descriptive metadata of the integrated triple
31   odcsRes:2
32     odcs:quality 0.8966325468133597 ;
33     odcs:sourceGraph odcsData:b68e21f7-363f-4bfd .
34
35   #Descriptive metadata of the source graph
36   odcsData:e0cdc9d7-e2d8-4bde
37     odcs:score 0.9 ;
38     prov:insertedAt "2012-04-01 12:34:56.0"^^xsd:dateTime ;
39     prov:source <http://dbpedia.org/page/Berlin> ;
40     prov:publishedBy <http://dbpedia.org/> ;
41     dc:license <http://creativecommons.org/licenses/by-sa/3.0/>
           ;
42     odcs:publisherScore 0.9 .
43
44  #Descriptive metadata of the source graph
45   odcsData:b68e21f7-363f-4bfd
46     odcs:score 0.8 ;
47     prov:insertedAt "2012-04-04 12:34:56.0"^^xsd:dateTime ;
48     prov:source <http://linkedgeodata.org/page/node240109189> .
49
50   <http://ld.opendata.cz:8087/uri?uri=...>
51     a odcs:QueryResponse ;
52     dc:title "URI search: http://dbpedia.org/resource/Berlin" ;
53     dc:date "2012-08-01T10:20:30+01:00" ;
54     odcs:totalResults 2 ;
55     odcs:query "http://dbpedia.org/resource/Berlin" ;
56      odcs:result odcsRes:1 ;
57      odcs:result odcsRes:2 .
58  }
```

Listing 7: Example of URI or keyword query response in TriG RDF syntax

Named Graph Queries

A named graph query selects all triples stored in the given data graph. The format of the results for the named graph query is exactly the same as for URI or keyword queries. The only difference is that labels for URI resources in the result are not retrieved (unless they are contained in the named graph) and data integration considers only the selected data graph. An HTML example of the named graph query result is in Figure 3.5.

Metadata Queries

A metadata query selects all the descriptive and provenance metadata of the desired data graph. The metadata includes the description of the QA policies applied to the data graph as part of a quality assessor on the data processing pipeline; such description justifies the quality score of that data graph. Apart from that, the result also contains metadata of the query. Listing 8 contains an excerpt of the TriG result on the metadata query for the named graph `<http://opendata.cz/infrastructure/odcleanstore/data/e0cdc9d7-e2d8-4bde>` being a source graph for integrated data in Line 27 of Listing 7.

Listing 8 contains metadata about graph `<http://opendata.cz/infrastructure/odcleanstore/data/e0cdc9d7-e2d8-4bde>` (Lines 12 – 18) and about the query response itself (Lines 36 – 42), both types of these metadata are already described in Section 3.2.4.

Apart from that, Listing 8 also contains information about violated QA policies (Lines 21 – 34), including also basic characteristics of the violated policies, such as their description (`dc:description`). Finally, Listing 8 involves a set of provenance data (Lines 44 – 47) being attached to the graph (Line 18); in this particular example, there is only one illustrative provenance triple in Line 47; however, provenance records can contain hundreds of triples. These provenance data should be expressed using the W3P provenance model for the Web introduced in Chapter 5.

```
1   @prefix odcs: <http://opendata.cz/infrastructure/odcleanstore/>
        .
2   @prefix odcsData: <http://opendata.cz/infrastructure/
        odcleanstore/data/> .
3   @prefix prov: <http://purl.org/provenance#> .
4   @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
5   @prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
6   @prefix dc: <http://purl.org/dc/terms/> .
7   @prefix w3po: <http://purl.org/provenance/w3p/w3po#> .
8   @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
9
10  #Descriptive metadata of the query
11  <http://opendata.cz/infrastructure/odcleanstore/query/metadata>
        {
12    #Descriptive metadata of the source graph
13    odcsData:e0cdc9d7-e2d8-4bde
14      prov:insertedAt "2012-04-01 12:34:56.0"^^xsd:dateTime ;
15      prov:source <http://dbpedia.org/page/Berlin> ;
```

```
    dc:license <http://creativecommons.org/licenses/by-sa/3.0/> ;
    prov:publishedBy <http://dbpedia.org/> ;
    odcs:provenanceMetadataGraph <http://opendata.cz/infrastructure/odcleanstore/provenanceMetadata/e0cdc9d7-e2d8-4bde> ;

    odcs:score 0.72 ;
    odcs:violatedQARule <http://opendata.cz/infrastructure/odcleanstore/QARule/10> ;
    odcs:violatedQARule <http://opendata.cz/infrastructure/odcleanstore/QARule/20> .

  #Descriptive metadata of the QA policy application
  <http://opendata.cz/infrastructure/odcleanstore/QARule/10>
    a odcs:QARule ;
    odcs:coefficient 0.8 ;
    dc:description "Procedure type ambiguous" .

  #Descriptive metadata of the QA policy application
  <http://opendata.cz/infrastructure/odcleanstore/QARule/20>
    a odcs:QARule ;
    odcs:coefficient 0.9 ;
    dc:description "Procurement contact person missing" .

  <http://ld.opendata.cz:8087/namedGraph?uri=...>
    a odcs:QueryResponse ;
    dc:title "Metadata for named graph:
      http://opendata.cz/infrastructure/odcleanstore/data/e0cdc9d7-e2d8-4bde" ;
    dc:date "2012-08-01T10:20:30+01:00" ;
    odcs:query "http://opendata.cz/infrastructure/odcleanstore/data/e0cdc9d7-e2d8-4bde" .
}

#Provenance metadata of the source graph
<http://opendata.cz/infrastructure/odcleanstore/provenanceMetadata/e0cdc9d7-e2d8-4bde> {
  odcsData:e0cdc9d7-e2d8-4bde w3po:isValidFrom "2012-08-01T10:20:30+01:00"^^xsd:dateTime .
}
```

Listing 8: Example of metadata query response in TriG RDF syntax

3.3 Related Work

Section 3.3.1 describes the frameworks and tools which are also intended to process RDF data and has overlapping functionality with ODCleanStore. Section 3.3.2 describes tools which may use the input web service of ODCleanStore to send RDF data to ODCleanStore. Section 3.3.3 describes tools which may use output web service of ODCleanStore to visualize/analyze the data.

3.3.1 RDF Frameworks and Tools

Linked Data Integration Framework

Linked Data Integration Framework[46] (LDIF) is an open-source Linked Data integration framework that can be used by Linked Data applications to transform Web data and normalize URIs while keeping track of data provenance. The framework consists of a scheduler, data import and an integration component with a set of pluggable modules.

[46] `http://www4.wiwiss.fu-berlin.de/bizer/ldif/`

LDIF components encompass the whole process from data import and processing to integration and quality assessment. We use some of LDIF components internally in ODCleanStore (Silk). The main difference is that LDIF is a framework other applications can build on, while ODCleanStore is a ready-to-use solution that can be easily deployed and managed via a web interface. Furthermore, LDIF solves the data integration offline as the data arrives to LDIF. This is a major difference from us, because we fuse the data (solve the conflicts) during a query time. Differences in quality assessment and data integration with Sieve, a part of the LDIF framework, are described in Chapter 4. LDIF also supports provenance tracking [47].

Linked Data Manager

Linked Data Manager (LDM)[48] is a Java based Linked (Open) Data Management suite to schedule and monitor required Extract-Transform-Load jobs for web-based Linked Open Data portals as well as for sustainable data management and data integration usage.

LDM data processing pipeline is similar to the data processing pipeline in ODCleanStore. Nevertheless, LDM does not provide (1) any permanent storage, (2) any data integration capabilities, (3) any transformers for linking resources, and (4) any direct access to the data – it does not provide query capabilities. Thus, LDM can be used to send data to ODCleanStore and access it from here.

3.3.2 RDF Data Producing Tools

Strigil implements a web scraper and document extractor that produces RDF data and submits such data to ODCleanStore.

D2R Server[49] is a tool for publishing relational databases as Linked Data. Such tool can be used to provide input data to ODCleanStore.

There are tools for manual data cleansing, such as Google Refine[50] with an RDF extension[51], which may be used as an alternative to custom transformers in ODCleanStore. Such tools, after doing certain cleansing, may export data in RDF and submit it to the input web service of ODCleanStore, which may use such data for further cleansing, quality assessment, linking, and data integration.

Apache Stanbol[52] provides a set of reusable components for semantic content management; it is not itself a content management system, but serves as an extension of the existing content management system, which connects to Apache Stanbol via RESTful web services. Apache Stanbol uses (1) *Entityhub* for caching and managing index of various external repositories, such as DBPedia, (2) *Contenthub*, which stores the whole documents in the original format together with their RDF metadata provided by the *Enhancer* and (3) *FactStore* which stores N-ary relations between entitites. Raw data mart in ODCleanStore does not store documents in the original format, nor N-ary facts; ODCleanStore stores

[47] See Figure 2 of `http://static.lod2.eu/Deliverables/deliverable-4.3.2.pdf`.

[48] `http://www.semantic-web.at/linked-data-manager`

[49] `http://d2rq.org/d2r-server`

[50] `http://code.google.com/p/google-refine/`

[51] `http://refine.deri.ie/`

[52] `http://stanbol.apache.org/`

RDF quads together with their descriptive and provenance metadata. The most common use case for Apache Stanbol is extraction of information implemented by Stanbol Enhancer. Such extracted information can be directly cleaned-up and reconciled using Stanbol Entityhub together with Google Refine. Data extracted and converted to RDF format by Stanbol can be inserted to ODCleanStore to further curate the data and provide integrated views on the data, customizable according to the consumer's needs.

3.3.3 RDF Data Consuming Tools

Payola is a tool for visualizing RDF data. It can consume data via SPARQL queries executed against the selected RDF stores. The raw data mart of ODCleanStore can serve for that purpose, but data integration is not available for SPARQL queries.

Linked Data browsers, such as tabulator or disco may use the output web service to display the integrated views on the selected URI queries [53].

3.4 Summary

In this section we presented an overview of the ODCleanStore tool, involving two important modules – a data processing and query execution. The query execution module further involves data integration and data filtering components. The data integration component is introduced in Chapter 4. The data filtering component enforcing the consumers' provenance requirements is described in Section 5.9.

Regarding the data processing module, we described various types of transformers – data normalizer, custom transfomer, linker, quality assessor, and quality aggregator – which play important roles in addressing Problems P1, P2, P3, and P5 outlined in Section 1.1. Transformers on data processing pipelines also provide further inputs to the query execution module, e.g., quality assessors produce quality scores.

The query execution module provides an interface for the data consumer to query the data in the raw data mart and obtain trustworthy and integrated Linked Data. We describe on the illustrative examples the interface provided by ODCleanStore to data consumers, the four types of supported queries, and the format of responses on these queries.

Relevant Author's Publications

The content of this chapter is covered by papers describing the ODCleanStore tool [111, 105]. Paper [128] demonstrates that the query execution module including the data integration component was implemented in ODCleanStore and is running. The mechanism of QA policies is motivated by paper [109].

[53] See `http://bit.ly/f1lzLS` for the list of Linked Data browsers.

Main Contributions

The main contribution of this chapter is the ODCleanStore tool as a whole, which allows to directly apply the research conducted in further chapters; in particular, it provides the query execution module, which (1) incorporates data integration and data fusion components, (2) is able to supplement data provided to data consumers with descriptive and provenance metadata, and (3) provides a practical user interface (Linked Data browser) to observe the results of the query execution module. Further contributions of ODCleanStore are as follows:

- a data processing pipeline for automated cleansing, linking, and quality assessment (addressing Problems P1, P2, P5)
- support for user specific pipelines, custom transformers (not just cleaners, linkers, and quality assessors), which may be easily added (supporting Problems P1, P2, P3, and P5)
- query execution module provides the resulting data in various RDF serializations, thus, being easily used by the web applications consuming Linked data, such as the Linked Data browser Alice is using
- an administration interface for setting up a pipeline, monitoring the pipeline's execution, debugging the pipeline, managing the transformers available, managing policies for transformers, and managing the query execution module (supporting P1, P2, P3, P5, and P7)

4. Data Integration in ODCleanStore

The query execution module of ODCleanStore, described in Section 3.2, runs on top of the raw data mart. Data consumers may query the raw data mart and, as a result, they receive relevant and trustworthy Linked Data being integrated and filtered according to their needs. This chapter starts by discussing and motivating the data integration component in ODCleanStore (Sections 4.1 and 4.2), executed as a part of the query execution module in ODCleanStore. Consequently, we focus on the data fusion step of data integration.

In particular, we propose a novel data fusion algorithm in Section 4.3, which is implemented in ODCleanStore. The two important phases of the algorithm are described in more detail in Sections 4.4 and 4.5. In Section 4.6, we discuss the time complexity of the algorithm. In Section 4.7, the proposed algorithm is evaluated. The data fusion algorithm may be customized by a data consumer as discussed in Section 4.8.

4.1 Motivational Scenario

Suppose Scenario 3.1. Furthermore, suppose that the identifiers for the resource Berlin are `dbpedia:Berlin` (representing the resource Berlin in the DBpedia source), `<http://sws.geonames.org/2950159/>` (representing the resource Berlin in the Geonames source), `<http://rdf.freebase.com/ns/en.berlin>` (representing the resource Berlin in the Freebase source).

Suppose that Alice already obtained one URI of the city of Berlin, e.g., `dbpedia:Berlin`, according to the procedure described in Section 3.2.2. When she invokes the URI query for that URI, the response (depicted in Figure 3.4) should contain an integrated view on the data about Berlin. Creation of such integrated view requires several steps.

Firstly, the integration algorithm has to find out that the meaning of various predicates is the same. For example, in one source, latitude of the city is expressed as a value of the predicate `geo:lat`, but another source uses the predicate `fb:location.geocode.latitude`. This problem is further referenced as *schema mapping*.

Secondly, the integration algorithm has to find out that the URI for Berlin used in the consumer's URI query represents the same "Berlin" as the two other URIs: `<http://sws.geonames.org/2950159/>` in Geonames and `<http://rdf.freebase.com/ns/en.berlin>` in Freebase. This problem is referenced as *duplicate detection*.

Thirdly, the data conflicts which emerge as a result of entity deduplication and schema mapping have to be solved. For example, after merging three representations of "Berlin" originating from three different sources, one gets three predicates holding latitude and three predicates holding longitude of the city of Berlin. This step of data integration, called *data fusion*, depends on the conflict handling strategy the consumer prefers. Data consumer may decide to receive all the distinct values for certain predicates, regardless of the conflictness of such values. Alternatively, the consumer may decide to receive only triples with conflict-free object values (for the given subject and property). But in most cases, the consumer wants to resolve the conflicts according to certain policies. In that case, different conflict resolution policies may be suitable for different predicates – the consumer may want to compute the average value for the values of the properties `geo:long` and `geo:lat` holding latitude and longitude of "Berlin", select the best value (with the highest integrated quality) for `rdfs:label` of the city, or select an arbitrary value from the values of the property `dbprop:population`; namespace prefix `geo:` is associated with namespace `http://www.w3.org/2003/01/geo/wgs84_pos#`, namespace prefix `dbprop:` is associated with namespace `http://dbpedia.org/property/`. Furthermore, for all the integrated triples returned as a result of the query, the consumer would like to see the quality of such integrated triples – *integrated quality score*. Such integrated triple's quality should be influenced by all the object values, which contributed to the selected/computed object value of the integrated triple.

4.2 Definition of the Problem

Bleiholder and Naumann [23] present overview and classification of different ways of integrating data and also provide survey of the data integration systems both from academia and industry. According to [23], the data integration consists of three main steps: *schema mapping* (detection of equivalent schema elements in different sources), *duplicate detection* (detection of equivalent resources) and *data fusion*. All these steps were motivated in the scenario above and are discussed in more detail further. We also discuss how ODCleanStore addresses these steps.

Data integration in general can be solved at the design time (while filling up the raw data mart) or query time. Lots of works deal with data integration during the design time, e.g., [48, 51]. Since different data integration strategies are worth for different data consumers and also for different situations at their hands, ODCleanStore allows to integrate the data during query execution in order to take into account consumer's requirements on the data.

In Sections 4.2.1 and 4.2.2, we describe how ODCleanStore supports the first two steps of the data integration – schema mapping and duplicate detection. In Section 4.3 and the rest of the chapter, we discuss the data fusion part for which we propose a novel data fusion algorithm.

4.2.1 Schema Mapping

The response prepared for Alice would be the more beneficial, the more sources agree on the common set of vocabularies (data models) used across various data sources holding information about "Berlin". Unfortunately, there is no way how to define a common set of vocabularies in the environment of the Web, where anybody can say anything about anything, so the ex post schema mappings are necessary.

The schema mapping step of data integration is about detection and creation of mappings between terms (e.g., predicates) from different vocabularies (data schemas) expressing certain relationships between these terms; one of the common relationships is the equivalence of two predicates. Schema mapping is crucial for any data integration activities, it allows more concise descriptions of resources [23]. Schema mapping might be considered as a special type of linking activity (Problem P1), where the links are created between the vocabulary terms. Therefore, Silk can be used for creating mappings not only between resources in data graphs but also between vocabulary terms. Furthermore, paper [21] presents the R2R framework for mapping terms on the Web based on the policies described by the R2R Mapping Language.

The data integration component of ODCleanStore handles the problem of schema mappings by enabling to create manual `owl:sameAs` links between ontology terms (classes, properties) in the knowledge base (see Figure 3.1). For example, an equivalence between properties `geo:lat` and `fb:location.geocode.latitude` may be expressed by creating a triple (`geo:lat`, `owl:sameAs`, `fb:location.geocode.latitude`) in the knowledge base of ODCleanStore. In the future, we will incorporate a support for writing Silk or R2R policies for that purpose.

4.2.2 Duplicate Detection

Duplicate detection is a special case of the linking activity (Problem P1) and covers an important category of links which express equivalence between resources. Thus, these links enable to deduplicate different HTTP URIs representing the same resource. For example, there might exist two different URIs for the city of Berlin, each of the URI is associated with different facts (triples) about Berlin. By knowing that these two URIs represent the same resource (the same city of Berlin), we can get a merged set of triples associated with one reconciled URI representing the city of Berlin.

The duplicate detection is solved in ODCleanStore by adding appropriate linkers as transformer instances on the data processing pipelines; as a result, triples with the `owl:sameAs` predicate deduplicating two resources (the subject and the object of that triple) are created. For the data from the motivational scenario in Section 4.1, we can define a linker creating `owl:sameAs` links between the different URIs representing the same city of Berlin based on the similarity of the city's name and the geographical location of the city.

4.2.3 Data Fusion

Since the same resource can be described by various sources (data graphs), conflicts may arise when integrating data about certain resource (e.g., a city of Berlin) coming from multiple sources. To solve this, ODCleanStore allows to solve data conflicts as a part of the data fusion algorithm.

In the rest of this chapter, we detail the problem of data fusion and propose and evaluate a novel data fusion algorithm. The data fusion algorithm supposes that proper mappings between ontologies (results of the schema mapping step) and the links between resources representing the same entities (results of the duplicate detection step) are available to the data fusion algorithm in the form of RDF triples.

4.3 Data Fusion Algorithm

Before outlining the data fusion algorithm and describing its inputs and outputs, let us introduce the needed terminology.

Definition 4.1. Suppose two quads $q_1 = (s_1, p_1, o_1, g_1)$ and $q_2 = (s_2, p_2, o_2, g_2)$, $q_1, q_2 \in \mathcal{Q}$, where s_1, p_1, o_1, s_2, p_2, $o_2 \in \mathcal{Z}$, g_1, $g_2 \in \mathcal{G}$. We say that two quads q_1 and q_2 are *α-equivalent*, i.e., $q_1 \equiv_\alpha q_2$, if and only if $s_1 = s_2$ and $p_1 = p_2$. Let us introduce an equivalence class $Q_{s,p}$, $s, p \in \mathcal{Z}$, holding all α-equivalent quads having the subject s and predicate p, i.e., $Q_{s,p} = [(s, p, o, g)] = \{(s', p', o', g') \quad s = s' \wedge p = p'\}$, $o, s', p', o' \in \mathcal{Z}$, $g, g' \in \mathcal{G}$.

Definition 4.2. Suppose two quads $q_1 = (s_1, p_1, o_1, g_1)$ and $q_2 = (s_2, p_2, o_2, g_2)$, $q_1, q_2 \in \mathcal{Q}$, where s_1, p_1, o_1, s_2, p_2, $o_2 \in \mathcal{Z}$, g_1, $g_2 \in \mathcal{G}$. Quads q_1 and q_2 are *duplicate quads* if $q_1 \equiv_\alpha q_2 \wedge o_1 = o_2 \wedge g_1 = g_2$.

Definition 4.3. Suppose two quads $q_1 = (s_1, p_1, o_1, g_1)$ and $q_2 = (s_2, p_2, o_2, g_2)$, $q_1, q_2 \in \mathcal{Q}$, where s_1, p_1, o_1, s_2, p_2, $o_2 \in \mathcal{Z}$, g_1, $g_2 \in \mathcal{G}$. Quads q_1 and q_2 are *supporting quads* if $q_1 \equiv_\alpha q_2 \wedge o_1 = o_2 \wedge g_1 \neq g_2$.

When fusing data from several source named graphs (sources), the data fusion algorithm has to deal with *data conflicts*, which occurs when two α-equivalent quads have inconsistent object values [166].

Definition 4.4. Suppose two quads $q_1 = (s_1, p_1, o_1, g_1)$ and $q_2 = (s_2, p_2, o_2, g_2)$, $q_1, q_2 \in \mathcal{Q}$, where s_1, p_1, o_1, s_2, p_2, $o_2 \in \mathcal{Z}$, g_1, $g_2 \in \mathcal{G}$. Quads q_1 and q_2 are *o-conflicting quads* if $q_1 \equiv_\alpha q_2 \wedge o_1 \neq o_2$. The object values o_1 and o_2 are called *o-conflicting values.*

The *s-conflicting quads*, having inconsistent subject values, may be defined symmetrically to o-conflicting quads. The data fusion algorithm deals with o-conflicting quads, but it may be trivially extended to support also s-conflicting quads. Further in the text, if talking about *conflicting* quads or *conflicting* values, we mean o-conflicting quads or o-conflicting values.

Bleiholder and Naumann [23] distinguish two kinds of data conflicts: (a) uncertainty about the object of the triple, caused by missing information; and (b) contradictions, caused by different object values. In our case, we do not consider conflicts between a value and no value, we simply use that value if available. The uncertainty of the value is reflected when computing quality score of the data graph and, as a result, also when computing quality of the integrated triples.

Survey in [23] identified three basic conflict handling strategies:

- *conflict resolution strategy* – data conflicts are resolved according to the set of *conflict resolution policies*

- *conflict ignorance strategy* – data conflicts are tolerated, all conflicting quads are returned

- *conflict avoidance strategy* – if the data conflict occurs, all the conflicting quads are removed from the resulting data

4.3.1 Conflict Resolution Policies

The conflict resolution policies drive the resolution of conflicts for the conflict resolution strategy. The considered conflict resolution policies are based on commonly used policies in database conflict resolution strategies [8]. We distinguish two types of conflict resolution policies – *deciding* and *mediating*. Each application of the conflict resolution policy either selects one value or more values from the conflicting values (a deciding conflict resolution policy) or computes new value (a mediating conflict resolution policy) as depicted:

- Deciding conflict resolution policies:
  - ANY, MIN, MAX, SHORTEST, LONGEST – an arbitrary, minimum, maximum, shortest, or longest value is selected from the set of conflicting values.
  - BEST – the value with the highest integrated quality (see Section 4.5.2) is selected. If more values have the same quality, the latest value is used. If both have the same time stamp and integrated quality, an arbitrary one is selected.
  - LATEST – the value with the newest time is selected. If more values have the same newest time, we select the one with higher integrated quality (see Section 4.5.2). If more values have the same quality and time stamp, an arbitrary value is selected.
- Mediating conflict resolution policies:
  - AVG, MEDIAN, CONCAT – the average, median, or concatenation of conflicting values is computed.

4.3.2 Conflict Handling Policies

Conflict resolution policies in Section 4.3.1 are relevant only for the conflict handling strategy. Since conflict ignorance and avoidance strategies may be also implemented as certain policies, we decided to introduce the term *conflict handling policies*, which involves all the conflict resolution policies (introduced in Section 4.3.1) and further involves policies:

- ALL – all values are selected from the set of conflicting values. The conflicting values are grouped by the same subject, predicate, and object. Motivated by the conflict ignorance strategy.
- AVOID – if there is a non-conflicting value, that value is selected. Otherwise, no value is selected. Motivated by the conflict avoidance strategy.

Let us define a set P_{CH} containing all the types of conflict handling policies introduced. The conflict handling policy may be specified either on the global level or per predicate level. If defined on the per predicate level for a predicate, such conflict handling policy has precedence over the global conflict handling policy for that predicate.

The value selected or computed by the application of the conflict handling policy is called an *integrated value*, the whole triple/quad with the integrated value is called *integrated triple/quad*. Every integrated quad is associated with the *integrated quality score* quantifying the quality of the integrated (object) value.

We distinguish *quality-free* and *quality-aware* policies. A quality-free policy does not require integrated quality score to select/compute the integrated value; quality-aware policies are the policies which are not quality-free. Quality-aware policies are the policies BEST and LATEST. All the other policies are quality-free.

4.3.3 Data Fusion Settings

Data fusion settings define the set of selected conflict handling policies for different predicates. Furthermore, these settings define the desired *data fusion error strategy*; if the conflict handling policy cannot be applied to a value (e.g., an average of a string literal, or when applying MEDIAN conflict handling policy to a mix of numeric and date values.), the quad with that value may be either discarded (data fusion error strategy *IGNORE*), or included in the output but not integrated (data fusion error strategy *RETURN_ALL*). Data fusion settings also set the *multivalue* parameter either on the global level for the whole algorithm or per predicate; this parameter is discussed in Section 4.5.2. As depicted in Section 4.8, data fusion settings may be specified together with the URI or keyword query introduced in Section 3.2.4.

4.3.4 Input/Output of the Data Fusion Algorithm

The input of the data fusion algorithm is formed by:

- a collection $Q_x \subseteq \mathcal{Q}$ of quads to be integrated; such collection is fetched from the raw data mart based on the particular URI or keyword query – for an URI query, where $x \in U$ is the value of the `uri` parameter of the query, $Q_x = (x, *, *, *) \cup (*, *, x, *)$; for a keyword query with a sequence of keywords x, $Q_x = (*, *, l, *)$, where literal $l \in \mathcal{L}$ contains all the keywords from x

- results of the duplicate detection step of data integration – a collection $Q_{links} \subseteq \mathcal{Q}$ of quads containing `owl:sameAs` links between resources occurring in the quads Q_x and other resources in the raw data mart, i.e.,
$Q_{links} = \{(s, \texttt{owl:sameAs}, o, g) \in \mathcal{Q} \mid nodeIn(s, Q_x) \vee nodeIn(o, Q_x)\}$

- results of the schema mapping step of data integration in ODCleanStore – a collection $Q_{mapps} \subseteq \mathcal{Q}$ of quads containing `owl:sameAs` links between resources (ontology primitives) occurring in the quads Q_x and other resource in the knowledge base of ODCleanStore, i.e.,
$Q_{mapps} = \{(s, \texttt{owl:sameAs}, o, g) \in \mathcal{Q} \mid nodeIn(s, Q_x) \vee nodeIn(o, Q_x)\}$

- set $S_{ng} = \{s_{ng}(g) \mid g \in G_{Q_x}\}$ of relevant data graphs' quality scores (see Section 3.1.3)

- set $S_{pu} = \{s_{pu}(pub(g)) \mid g \in G_{Q_x}\}$ of relevant publishers' quality scores (see Section 3.1.4)
- data fusion settings as described in Section 4.3.3

The output of the data fusion algorithm is a collection of integrated quads, $Q_x^I \subseteq \mathcal{Q}$, enriched for each integrated quad with the integrated quality score of the quad and source graphs contributing to the quad.

4.3.5 Algorithm Outline

As depicted in Algorithm 1, the data fusion algorithm has two phases:

- *Implicit data fusion*, which does not depend on the chosen data fusion settings and quality scores S_{ng} and S_{pu}; it prepares input data for the *explicit data fusion* phase, so that the conflict handling policies may be applied and integrated quality can be computed independently on the chosen resource URIs or overlapping older versions of data (Line 2).
- *Explicit data fusion*, which fuses every α-equivalence class $Q_{s,p}$ using the conflict handling policy defined for the predicate p and computes the integrated quads Q_x^I (Lines 3 – 5).

Algorithm 1 Data fusion algorithm

```
Output: $dataFusion(Q_x, Q_{links}, Q_{mapps}, S_{ng}, S_{pu}, FS)$
1: $Q_x^I \leftarrow \emptyset$
2: $Q_x/\equiv_\alpha \leftarrow implicitDataFusion(Q_x, Q_{links}, Q_{mapps})$
3: for each $Q_{s,p} \in Q_x/\equiv_\alpha$ do
4:     $Q_x^I \leftarrow Q_x^I \cup explicitDataFusion(Q_{s,p}, S_{ng}, S_{pu}, FS)$
5: end for
6: return $Q_x^I$
```

4.4 Implicit Data Fusion

Before discussing the implicit data fusion phase of the data fusion algorithm, let us introduce an auxiliary graph $SA = (Z_{Q'_x}, E_{sameAs})$, where $Q'_x = Q_x \cup Q_{links} \cup Q_{mapps}$, vertices are the RDF nodes $Z_{Q'_x}$ and edges are formed by `owl:sameAs` links between these vertices, i.e., $E_{sameAs} = \{(s, o) \subseteq Z_{Q'_x} \times Z_{Q'_x} \mid \exists g \in G_{Q'_x} \wedge (s, \texttt{owl:sameAs}, o, g) \in Q'_x\}$.

Algorithm 2 describes the implicit data fusion phase. In Line 1, the auxiliary graph $SA = (Z_{Q'_x}, E_{sameAs})$ is constructed. In Line 2, set $\mathcal{C}$ of weakly connected components of SA is built. Consequently, in Lines 3 – 5, each URI resource $z \in Z_{Q_x}$ is replaced with a canonical URI, $uri(C)$, being a single URI representant for the component $C \in \mathcal{C}$, $z \in C$. In Lines 6 – 8, the duplicated quads are eliminated. In Lines 9 – 13, if certain graph g_2 is an update of another graph g_1, denoted as $g_2 \propto g_1$, and both g_1 and g_2 are in G_{Q_x}, we remove all quads originating from the outdated graph g_1. Finally, in Line 14, quads Q_x are grouped

Algorithm 2 Implicit data fusion

```
Output: $implicitDataFusion(Q_x, Q_{links}, Q_{mapps})$
construct graph $SA = (Z_{Q'_x}, E_{sameAs})$
$\mathcal{C} \leftarrow weaklyConnectedComponents(SA)$
for each $z \in Z_{Q_x} \wedge z \in \mathcal{U}$ do
    $z \leftarrow uri(C)$, where $C \in \mathcal{C} \wedge z \in C$
end for
while $\exists q_1, q_2 \in Q_x$, s.t. $q_1, q_2$ are duplicate quads do
    $Q_x \leftarrow Q_x \setminus \{q_2\}$
end while
for each $g_1 \in G_{Q_x}$ do
    if $\exists g_2 \in G_{Q_x} \wedge g_2 \propto g_1$ then
        $Q_x \leftarrow Q_x \setminus (*, *, *, g_1)$
    end if
end for
return $Q_x/\equiv_\alpha$
```

to α-equivalence classes, i.e., the quotient set $Q_x/\equiv_\alpha$ is created and returned as a result of the implicit data fusion phase.

Regarding the time complexity, (1) the grouping of quads Q_x into α-equivalence classes and removing duplicate quads requires sorting in $\mathcal{O}(|Q_x| \log |Q_x|)$, assuming comparison of two RDF nodes in constant time, and (2) the set of weekly connected components is found in linear time w.r.t. $|E_{sameAs}| + |Z_{Q'_x}|$. Therefore, the time complexity of the implicit data fusion phase (Algorithm 2) is $\mathcal{O}(|Q_x| \log |Q_x| + |E_{sameAs}|)$.

4.5 Explicit Data Fusion

As depicted in Lines 3 – 5 of Algorithm 1, explicit data fusion is executed for each α-equivalence class $Q_{s,p} \in Q_x/\equiv_\alpha$ separately, the set $Q_x/\equiv_\alpha$ is the output of the implicit data fusion phase.

Let us denote a sequence $\widetilde{Q_{s,p}} = \{q_1, \ldots, q_i, \ldots, q_n\}$, $q_i = (s, p, o_i, g_i)$, $i \in \{1, \ldots, n\}$, as a sequence of quads containing an ordered collection of quads from $Q_{s,p}$, i.e., $\forall q_i, q_i \in \widetilde{Q_{s,p}} \iff q_i \in Q_{s,p}$; the exact ordering of quads $\widetilde{Q_{s,p}}$ is not important, but has to be fixed. Let us further denote sequence $V = \{v_1, \ldots, v_n\}$ as the ordered collection of all object values o_i, i.e., $v_i = o_i$, $\forall i \in \{1, \ldots, n\}$.

For every single explicit data fusion run, subject s and predicate p are fixed. As a result, the context of the single explicit data fusion run is given by the collection of object values $v_i \in V$ and graphs $g_i \in G_{Q_x}$ these objects (object values) originate from. In further description of the single explicit data fusion run, subject s, predicate p, v_i and g_i, $i \in \{1, \ldots, n\}$, are always as defined here. Steps 1 – 4 of one such explicit data fusion run are as follows:

1) Apply the conflict handling policy relevant to $Q_{s,p}$.

2) Compute the integrated quality qs for the integrated value(s) from Step 1.

3) Apply the conflict handling policy again to the value(s) from Step 2 (only in case of quality-aware policies).

4) Compose resulting integrated quads; every such resulting quad is supplemented with (1) the integrated quality score quantified by qs and (2) source graphs contributing to the computation of the integrated (object) value of the quad.

4.5.1 Step 1: Conflict Handling Policy Application

In Step 1, the *conflict handling policy* relevant for the α-equivalent quads $Q_{s,p}$ is applied to the ordered collection V.

Definition 4.5. A conflict handling policy $po \in P_{CH}$ is *relevant* for the α-equivalent quads $Q_{s,p}$ if and only if the data fusion settings either associate policy po with the given predicate p, or, if such association is not available, po is the global conflict handling policy.

Let us introduce a sequence A holding values V selected or computed as a result of the relevant conflict handling policy application; $0 \leq |A| \leq |V|$. If $|A| = 0$ (it may happen for the conflict handling policy AVOID), the explicit data fusion run ends. In further description of the algorithm, we suppose that $|A| > 0$.

In case of all quality-free conflict handling policies, the application works as is described in Sections 4.3.1 and 4.3.2. For the quality-aware conflict handling policy BEST, the policy application in Step 1 selects all conflicting values and the final value is selected during the policy application in Step 3. In case of the quality-aware conflict handling policy LATEST, the application in Step 1 selects all the values with the latest (newest) time (it can be more values) and the final value is selected during the application of the policy in Step 3. The reason for the two rounds of the quality-aware conflict handling policies' applications is that the integrated quality, being computed in Step 2, is not available in Step 1. The data fusion algorithm could have started by computing the integrated quality for all values V first and then could have applied all the policies at once; however, that approach would be inefficient for the majority of conflict handling policies, because lots of the computed integrated quality scores (all except one) would have been thrown away.

4.5.2 Step 2: Integrated Quality Computation

In Step 2, we compute the quality score of integrated values $v \in A$ according to function $qs : A \rightarrow [0, 1]$; $qs(v)$ is an important input to Step 3, where the quality-aware policy is applied.

Suppose that the set of graphs that agree on value $v \in A$ is denoted as $agree(v) = \{g_i \mid v_i = v\}$. Let us introduce the function $s : G \rightarrow [0, 1]$ computing the quality score of the data graph g as a weighted average of scores $s_{ng}(g)$ (output of the quality assessor transformer) and $s_{pu}(g)$ (output of the quality aggregator transformer), where $\gamma \in [0, 1]$ is a parameter of the algorithm set by the data integration component.

$$s(g) = \gamma \cdot s_{ng}(g) + (1 - \gamma) \cdot s_{pu}(pub(g)) \tag{4.1}$$

Multiple real-world cases and the motivational scenario in Section 4.1 lead us to three factors of the integrated quality computation: (1) quality scores $s(g_i)$ of the source graphs $g_i \in G_{Q_x}$, (2) conflicting values – the difference between value v and other conflicting values from V and (3) confirmation by multiple sources – the size of $agree(v)$. These three factors are further accompanied by constraints that should be satisfied by function qs. The most important constraints are:

1. If $|V| = 1$, then $qs(v) = s(g_1)$.

2. If there is a source graph g claiming a value v which is not conflicting with any other value $v' \in V$, $qs(v) \geq s(g)$.

3. $qs(v)$ is increasing with the increasing quality scores of source graphs (quantified by s) the value v was selected from or calculated from.

4. $qs(v)$ is decreasing with difference of other values $v_i \in V$, taking their quality scores $s(g_i)$ into consideration.

5. If multiple source graphs agree on the same value, the quality of the integrated quad should be increased.

6. If k source graphs with $s(g_i) = 1$ (the highest quality score) claim a value completely different from value v, then $qs(v)$ should be decreased approximately k times. If these source graphs have lower quality scores $s(g_i)$, the decrease of $qs(v)$ is lower.

In the next sections, we will discuss these three factors of the integrated quality computation and, consequently, the formula for qs incorporating all these factors and the constraint outlined is proposed.

First Quality Factor – Quality Scores

First, we calculate integrated quality $qs_1(v)$, $qs_1 : A \to [0, 1]$, based on the quality scores s of the graphs the value $v \in A$ originates from. In Step 1, the value v may have been calculated from all the sources (in case of conflict handling policies AVG, MEDIAN, CONCAT), or it may have been selected from one named graphs containing the quad (s, p, v, g_i) (in case of other conflict handling policies). For computing $qs_1(v)$, in the former case, we use formula (a), in the latter case, we use formula (b) of Formula 4.2. Formula 4.2 was designed with Constraint 2 (on page 74) in mind.

$$qs_1(v) = \begin{cases} \operatorname{avg}\{s(g) \mid g \in \{g_1, \ldots, g_n\}\} & \text{(a)} \\ \max\{s(g) \mid g \in agree(v)\} & \text{(b)} \end{cases} \tag{4.2}$$

Second Quality Factor – Conflicting Values

In the second step, we compute integrated quality $qs_2(v)$, $qs_2 : A \to [0,1]$, based on $qs_1(v)$ and differences of conflicting values V. For that, we use a metric $d : U \times U \to [0,1]$ satisfying $d(v,v) = 0$. We use $d(x,y) = |(x-y) \ / \ \mathrm{avg}(x,y)|$ in case of numeric literals, normalized Levenshtein distance [116] in case of string literals, difference divided by a configurable maximum value in case of dates, and $d(x,y) = 1$, where $x \neq y$, for URI resources and nodes of different types.

Decreasing the quality of the integrated quads based on the conflicting values is not the right solution in all situations. For example, the predicate `rdf:type` often has (for the same subject) multiple valid object values which are not conflicting. Therefore, data fusion settings may contain the *multivalue* parameter for each predicate p describing the desired behavior, i.e., whether the quality $qs_2(v)$ should be decreased for each other conflicting value or not. Let us introduce function $multi(p) \in \{true, false\}$ expressing the setting of the multivalue parameter for predicate p.

Depending on the setting of the multivalue parameter, $qs_2(v)$ is computed for $v \in A$ as described in Formula 4.3. If $multi(p) = false$ and the set V contains conflicting values different from v, $qs_2(v)$ is reduced increasingly with the value of metric d and the quality score s of the source graph the conflicting value originates from (Constraint 3 on page 74).

$$qs_2(v) = \begin{cases} qs_1(v) \cdot \left(1 - \frac{\sum_{i=1}^{n} s(g_i) d(v,v_i)}{\sum_{i=1}^{n} s(g_i)}\right) & \text{if } multi(p) = false \\ qs_1(v) & \text{if } multi(p) = true \end{cases} \tag{4.3}$$

Third Quality Factor: Confirmation by Multiple Sources

Intuitively, if multiple different sources agree on a single value, such value should have higher quality score than each of the sources individually. We reflect this in the final phase of the integrated quality computation, $qs_3(v)$, depicted in Formula 4.4; $qs_3 : A \to [0,1]$, $C \in \mathbb{N}$ is a constant.

$$qs_3(v) = qs_2(v) + (1 - qs_2(v)) \cdot \min\left(\frac{-qs_1(v) + \sum_{g \in agree(v)} s(g)}{C}, 1\right) \tag{4.4}$$

Formula for Integrated Quality $qs(v)$

Not all the factors above make sense for all the conflict handling policies. In case of the conflict handling policies CONCAT and AVOID, the integrated quality $qs(v)$ is computed as $qs(v) \equiv qs_1(v)$, because $|A| = 1$. In case of conflict handling policies AVG and MEDIAN, $qs(v) \equiv qs_2(v)$, in order to take into account the distribution of values. Otherwise, quality score of the integrated value v is $qs(v) \equiv qs_3(v)$, taking into account all the three factors presented and satisfying Constraints 1 – 6 on page 74.

The time complexity of the integrated quality computation for a fixed $v \in A$ is $\mathcal{O}(|V| \cdot D)$; D is the complexity of the distance metric evaluation. This gives us the overall complexity of $\mathcal{O}(|V|^2 \cdot D)$ for ALL and BEST conflict handling policies, $\mathcal{O}(|V| \log |V| + |V| \cdot D$ for MEDIAN and $\mathcal{O}(|V| \cdot D)$ for other conflict handling policies.

4.5.3 Step 3: Quality-aware Conflict Handling Policy Application

In Step 3, the quality-aware policies are applied once again, using the results of integrated quality computation (Step 2). In case of quality-free policies, we set $A' = A$ and skip this step.

In case of the conflict handling policy BEST, we set: $A' = \{v\}$, where $v \in A \wedge \nexists v' \in A, v' \neq v$, s.t. $qs(v') > qs(v) \vee ((qs(v') = qs(v)) \wedge (t(v') > t(v)))$; $t(x)$ denotes the time when the data graph containing the triple with (object) value x was inserted to ODCleanStore. In case of the conflict handling policy LATEST, we set $A' = \{v\}$, where $v \in A \wedge \nexists v' \in A \wedge v' \neq v$, s.t. $qs(v') > qs(v)$.

4.5.4 Step 4: Resulting Integrated Quads Composition

Step 4 composes resulting integrated quad(s), $Q^I_{s,p}$; every such resulting quad is supplemented with (1) the integrated quality score quantified by qs and (2) source graphs contributing to the computation of the integrated (object) value of the quad. Therefore, every resulting integrated quad contains:

- quad (s, p, a, g_a), where $a \in A'$ is the integrated value, g_a is the new unique *triple named graph* (see Section 3.2.4)
- triple (g_a, `odcs:quality`, $qs(a)$) holding the integrated quality $qs(a)$ of integrated value a
- triples (g_a, `odcs:sourceGraph`, g_i) holding source graphs contributing to integrated value a

Listing 7 shows in Lines 11 – 33 the output of the data fusion algorithm, which is created by Steps 4 executed for each explicit data fusion run.

4.6 Time Complexity

In this section, we discuss the time complexity of the data fusion algorithm. Let $N = |Q_x|$ be the total number of input quads to the data fusion algorithm, $S = |E_{sameAs}|$ the number of `owl:sameAs` links (it is the number of edges in graph SA), $|G_{Q_x}|$ the number of data graphs participating in the data fusion, ($|G_{Q_x}| \leq N$). Let $CQ = \{cq_1, cq_2, \ldots, cq_K\}$ be the quotient set $Q_x/{\equiv_\alpha}$, $K \in \mathbb{N}$, and $n_i = |cq_i|$ the size of i-th α-equivalence class. D is the complexity of distance metric evaluation; distance metric is evaluated in linear time for strings (modified Levenshtein distance) and in constant time for other cases.

The time complexity of the implicit data fusion phase is $\mathcal{O}(N \log N + S)$ as described in Section 4.4. Regarding the explicit data fusion phase, conflict handling policies are applied with the time complexities given in Section 4.5.2. To sum up, the time complexity of the data fusion algorithm is as follows:

- In case of ALL and BEST conflict handling policies:

$$\mathcal{O}\left(N \log N + S + \sum_{i=1}^{K} \left(n_i^2 D\right)\right) \tag{4.5}$$

- For conflict handling policies other than ALL and BEST:

$$\mathcal{O}\left(N \log N + S + \sum_{i=1}^{K} \left(n_i D\right)\right) \tag{4.6}$$

In the worst case, when $K = 1$ and $G_{Q_x} = N$, this gives us the time complexity of the data fusion algorithm as follows:

- In case of ALL and BEST conflict handling policies:

$$\mathcal{O}\left(N^2 D + S\right) \tag{4.7}$$

- For conflict handling policies other than ALL and BEST :

$$\mathcal{O}\left(N \log N + ND + S\right) \tag{4.8}$$

4.7 Experiments

In this section we provide experiments of the data fusion algorithm. Firstly, we provide an illustrative example of the data fusion algorithm use. Since the data fusion is meant to be executed at the query time, secondly, we evaluate the practical feasibility of the data fusion algorithm in terms of reasonable response times. Thirdly, we discuss to which extent the data integration, and in particular the data fusion, contributes to more complete, concise, and consistent data.

4.7.1 Data Fusion in Motivational Scenario

Let us start with an illustrative example of the data fusion algorithm execution. Suppose that there is data in the raw data mart about Berlin coming from multiple sources as introduced in the motivational scenario in Section 4.1; apart from that, we intentionally added one more data source called DBPediaError which states that `geo:lat` of Berlin is approximately 13 degrees (instead of approximately 52 degrees).

Suppose we set up the following data fusion settings: we choose conflict handling policy AVG for `geo:long`, BEST for `rdfs:label`, and ALL for `dbprop:population`. We suppose that the quality score $s(g)$ is 0.9 for DBpedia source graph containing information about Berlin and 0.8 for other source graphs with information about Berlin. The *multivalue* parameter of the data fusion settings is set to true for `rdfs:label` and `rdf:type` predicates.

When we request the integrated view on the data about Berlin, Table 4.1 gives the results of the data fusion algorithm. The integrated quality of `dbprop:population` is decreased, because there were different conflicting values; the quality of label “Berlin” is very high (even higher than the score s of the DBpedia

Table 4.1: Data fusion illustrative example

Predicate	Value	Source	Quality
`rdfs:label`	Berlin	all	0.963
`rdf:type`	`dbpedia-owl:City`	DBpedia	0.900
`dbprop:population`	3450889	DBpedia	0.897
`dbprop:population`	3426354	GeoNames	0.797
`geo:long`	13.4074	all	0.833
`geo:lat`	42.7402	all	0.497

Table 4.2: Data fusion algorithm's execution times for DBpedia evaluation

Triples	Integration	Multivalue	Time
100,000	ALL	no	1.75 s
100,000	ANY	no	1.02 s
100,000	ALL	yes	1.01 s
100,000	CONCAT	yes	0.96 s
100,000	ANY	yes	0.83 s

source graph), because all source graphs agree on it; the quality of `geo:lat` is significantly lower, because of the introduced erroneous value. The data for this illustrative example are part of the standard ODCleanStore distribution, which may be downloaded from `http://sourceforge.net/p/odcleanstore/`.

4.7.2 DBpedia Evaluation

The data fusion algorithm is executed on-the-fly during query execution. Therefore, the practical feasibility (performance) of the data fusion algorithm has to be evaluated. We evaluated the algorithm's performance on the dataset consisting of data from DBpedia infoboxes[54]. In order to simulate conflicts in the data, we have generated random `owl:sameAs` links between subjects sharing a common property and ran the fusion algorithm on a subset of 100,000 randomly selected triples from the dataset. This subset gave us 2,500 sets of conflicting triples comprising 37,500 triples altogether. The median of number of conflicts was 27 triples. In Table 4.2, we measured how long the process ran for various settings on average [55].

The number of conflicting values will be typically small – small hundreds of α-equivalence classes with sizes in tens of values for a typical URI query [56]. Therefore, our experiment gives us very reasonable times per α-equivalence class and has demonstrated that data fusion algorithm is indeed fast enough to work in real-world settings.

[54] `http://wiki.dbpedia.org/Downloads32#h72-1`

[55] Hardware configuration: Intel Core2 Duo 2x2.4 Hz, 3 GB RAM

[56] Taking into account data available in DBPedia.

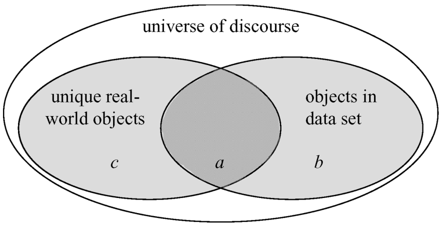


Figure 4.1: Measuring completeness and conciseness in analogy to precision and recall, letters a, b, c denotes the number of objects in the corresponding region [23]

4.7.3 Contribution to Completeness, Conciseness, and Consistency of Data

In this section, we demonstrate the impact of the data integration (and in particular data fusion) process by comparing the completeness, conciseness, and consistency of the original datasets and the integrated dataset. We focus on the data fusion algorithm and the discussion of extensional completeness, and consistency.

Definition of completeness, conciseness, and consistency

According to Bleiholder and Naumann [23], data integration should increase quality of the consumed data along three dimensions: completeness, conciseness and consistency; they distinguish extensional (data level) and intensional (schema level) completeness and conciseness.

Dataset is *complete* if it contains all the necessary predicates/objects (resources) for the task at hand. In particular, *intensional completeness* ($compl_i(d)$) of the dataset d is measured as the proportion of unique properties in d to the set of all available unique properties in the universe; *extensional completeness* ($compl(d)$) is measured as the proportion of unique objects in the dataset d to the set of all available unique objects in the universe, i.e., $compl(d) = \frac{a}{a+c}$; a, c as illustrated in Figure 4.1.

Dataset is *concise* if it does not contain redundant properties/objects (i.e., two equivalent properties/objects with different identifiers). Thus, *intensional conciseness* ($conci_i(d)$) of the dataset d is defined as the proportion of unique properties in d to all properties in the d; *extensional conciseness* ($conci(d)$) is the proportion of unique objects in the dataset d to all objects in d, i.e., $conci(d) = \frac{a}{a+b}$ according to Figure 4.1.

Dataset d is *consistent* if it is free of data conflicts, therefore, consistency, $consist(d)$, can be defined as a proportion of objects in the dataset d without any conflicting values among their properties to the set of all objects in d.

Datasets

As the set D of testing datasets, we employed English (EN), German (DE), and Polish (PL) dumps of DBpedia 3.7[57]; in particular we used Ontology Infobox Types and Ontology Infobox Properties feeds, which were loaded to ODCleanStore.

[57] http://wiki.dbpedia.org/Downloads37

Table 4.3: Unique occurrences of the selected properties P in D

Property	**DE**	**PL**	**EN**
`dbpedia-owl:numberOfEmployees`	1326	16	9792
`dbpedia-owl:formationDate`	462	20	2154
`dbpedia-owl:revenue`	179	9	6460

On top of these datasets, we decided to demonstrate the completeness, conciseness, and consistency with respect to the set P of three selected properties – `dbpedia-owl:numberOfEmployees`, `dbpedia-owl:formationDate`, `dbpedia-owl:revenue` – describing the business entities (instances of the class `dbpedia-owl:Company`); the first property holds the number of employees of a company, the second one holds the date when a company was formed, and the third one holds the revenue of a company. We selected the set of these properties, because all of them are common properties of business entities and all of them are not multivalue properties; for multivalue properties, the $consist(d)$, for certain dataset d, would have to be defined differently.

If the business entity uses the same identifier (URI) in two or more datasets (EN, DE, PL), then that entity is shared by more datasets. Nevertheless, we do not deduplicate business entities in the dataset, so if the URIs of two business entities differ, then we assume they really represent different entities. The total number of unique business entities (instances of the class `dbpedia-owl:Company`) in all examined datasets is 42,357. The number of all business entities in the given datasets is 6,774 for DE, 2,312 for PL, and 40,132 for EN; the number of business entities defined only in the given datasets (and not in the others) is 1,853 for DE, 314 for PL, and 34,536 for EN. Table 4.3 shows, for the given dataset $d \in D$ and property $p \in P$, the number of business entities from d for which p exists and, at the same time, p does not exist for the same entity in a different dataset. Table 4.3 illustrates that all datasets contain certain properties not involved in other datasets.

Completeness, Conciseness, and Consistency Applied to D

If we apply the dimensions above (considered properties are the properties from P, considered objects are the particular resources – business entities – in datasets D), we find out that:

- $compl_i(d) = 100\%, \forall d \in D$, because all the datasets use all three observed properties from P
- $compl(DE) = \frac{6774}{42357} = 15.99\%$, $compl(PL) = 5.46\%$, $compl(EN) = 94.75\%$
- $conci_i(d) = 100\%, \forall d \in D$, because all the datasets utilize the same properties from P, which are not redundant
- $conci(d) = 100\%, \forall d \in D$, because we suppose that if the identifiers are different they represent different entities (i.e., there are no redundant resources)
- $consist(DE) = 98.32\%$, $consist(PL) = 99.78\%$, $consist(EN) = 99.53\%$

Table 4.4: Completeness/consistency of properties P in D

Property	DE	PL	EN
`dbpedia-owl:numberOfEmployees`	42.93/96.49%	4.58/95.28%	28.41/99.15%
`dbpedia-owl:formationDate`	9.82/99.1%	2.6/100%	5.92/98.4%
`dbpedia-owl:revenue`	5.85/98.48%	1.34/100%	16.67/99.22%

Table 4.5: Completeness of properties P in the integrated dataset

Property	DI	$\mathbf{DI}_{\mathbf{\vert DE}}$	$\mathbf{DI}_{\mathbf{\vert PL}}$	$\mathbf{DI}_{\mathbf{\vert EN}}$
`dbpedia-owl:numberOfEmployees`	30.1%	51.86%	32.7%	29.84%
`dbpedia-owl:formationDate`	6.76%	13.79%	14.49%	6.57%
`dbpedia-owl:revenue`	16.24%	29.7%	19.85%	16.92%

As we can see, the dimensions not providing 100% results are extensional completeness, and consistency. We will examine these in more detail further.

Completeness and Consistency for Properties in P

Similarly as in [127], to provide better insight into the impact of the different conflict handling strategies on the extensional completeness and consistency of the resulting dataset, we introduce (extensional) completeness and consistency of the dataset $d \in D$ restricted to the particular property $p \in P$ as:

$$compl(p,d) = \frac{|unique\ objects\ with\ p\ in\ d|}{|all\ unique\ objects\ in\ d|} \tag{4.9}$$

$$consist(p,d) = \frac{|objects\ without\ conflicts\ for\ p\ in\ d|}{|all\ unique\ objects\ with\ p\ in\ d|} \tag{4.10}$$

Table 4.4 depicts $compl(p,d)$ and $consist(p,d)$ for all the selected properties $p \in P$ and all datasets $d \in D$. Furthermore, Table 4.5 depicts in the first column $compl(p,DI)$ for the integrated dataset DI (resulting from the application of the data fusion algorithm to all business entities using the conflict resolution strategy, i.e., any conflict resolution policy); $compl(p,DI)$ and $consist(p,DI)$ is restricted in the 2., 3., and 4. column of Table 4.5 to the business entities originally introduced in the datasets DE, PL, or EN, respectively (denoted as $DI_{|d}$, $d \in D$).

Results & Discussion

From Tables 4.4 and 4.5, we see that in case of data fusion with an arbitrary conflict resolution policy the completeness of all the properties in the integrated dataset DI is increased. If we compare the completeness $compl(p,DI_{|d})$, for $p \in P$, $d \in D$, we can see that for smaller datasets PL and DE, the increase in completeness is in tens to hundreds of percents of the original completeness $compl(p,d)$ in Table 4.4. Therefore, a consumer of the PL or DE dataset can profit substantially from the data fusion having several times higher completeness for all properties from P. Even the consumer of the dataset EN (concentrating the biggest amount of data from the beginning) profits from the data fusion – the completeness $compl($dbpedia-owl:formationDate$, EN)$ is increased by more than 10%. Data fusion with any conflict resolution policy also guarantees consistency ($consist(p, DI) = 100\%$), because all conflicts are resolved (we suppose that no errors occur during the conflict resolution, e.g., because of incorrect conflict resolution policy usage).

In case of the conflict handling strategy (conflict handling policy AVOID), $consist(p, DI)$ is the same as for conflict resolution strategy; however, completeness is lower ($compl($dbpedia-owl:numberOfEmployees, $DI_{|EN}) = 25.59\%$, $compl($dbpedia:formationDate, $DI_{|EN}) = 5.93\%$, and $compl($dbpedia-owl:revenue, $DI_{|EN}) = 16.22\%$), because the conflicting properties are removed (for comparison with $compl(*, EN)$, see the last column of Table 4.5).

In case of the conflict ignorance strategy (conflict handling policy ALL), completeness is the same as for the arbitrary conflict resolution policy, however, consistency is reduced ($consist($dbpedia-owl:numberOfEmployees$, DI_{|EN}) = 88.53\%$, $consist($dbpedia-owl:formationDate$, DI_{|EN}) = 97.14\%$, and $consist($dbpedia-owl:revenue, $DI_{|EN}) = 99.2\%$), because lots of new conflicts arise (compare with the consistency of the datasets in Table 4.4).

Summary

In terms of the consistency, conflict resolution policies and conflict handling policy AVOID guarantee 100% consistency of the integrated dataset DI, more than the original datasets $d \in D$. Conflict handling policy ALL provides lower consistency of DI, even lower than the individual original datasets $d \in D$.

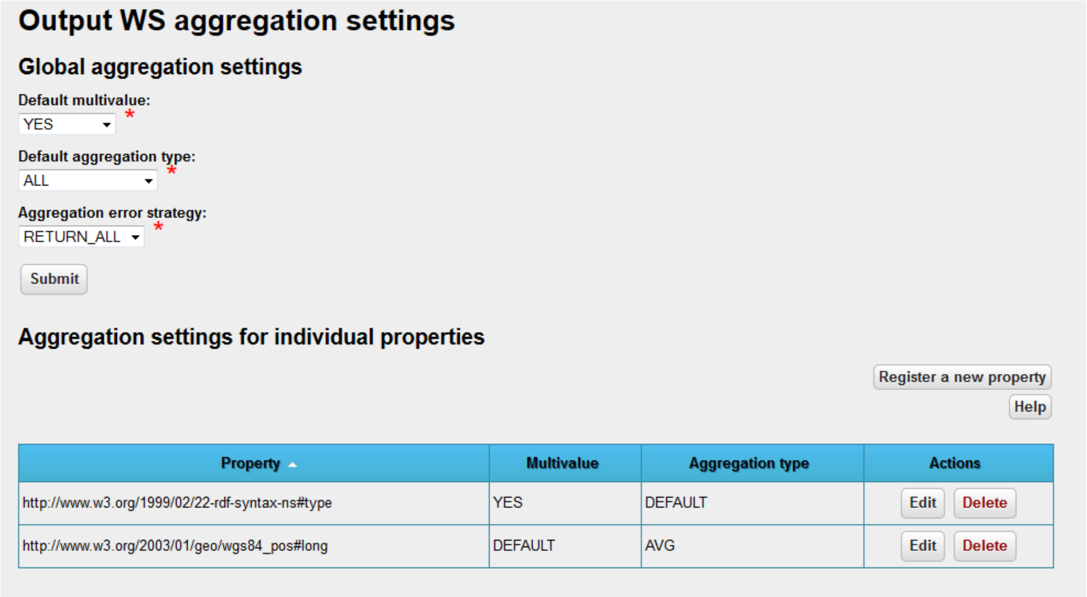
In terms of the completeness, conflict resolution policies and conflict handling policy ALL provide the same completeness of the integrated dataset DI, which is better than the completeness of the original datasets $d \in D$. Conflict handling policy AVOID may provide completeness of DI even lower than the individual datasets $d \in D$ had.

Finally, $conci(DI)$, $conci_i(DI)$, and $compl_i(DI)$ of the integrated dataset DI are equal to 100%, because the integration does not break that dimensions.

4.8 Customizing Data Fusion Algorithm

Default data fusion settings should be specified by administrators via administration interface of ODCleanStore, see Figure 4.2. Apart from that, every data consumer (or the application he is using) may customize the data fusion settings per individual URI or keyword query. Such customization overrides the default data fusion settings. Table 4.6 presents further parameters which may be specified when constructing URI and keyword queries for ODCleanStore; Table 4.6 extends the basic parameters of ODCleanStore queries introduced in Table 3.2.

The parameters `aggr` and `paggr[`*`property`*`]` support the string values directly corresponding with the abbreviations of conflict handling policies introduced in Sections 4.3.2 and 4.3.1. In the current distribution of ODCleanStore, we do not support the conflict handling policy `AVOID`. The parameters `multivalue` and `pmultivalue[`*`property`*`]` define the multivalue settings on the global and per predicate level. The parameter `es` defines data fusion error strategy.



Output WS aggregation settings

Global aggregation settings

Default multivalue: YES *

Default aggregation type: ALL *

Aggregation error strategy: RETURN_ALL *

Aggregation settings for individual properties

Property	Multivalue	Aggregation type	Actions
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	YES	DEFAULT	Edit Delete
http://www.w3.org/2003/01/geo/wgs84_pos#long	DEFAULT	AVG	Edit Delete

Figure 4.2: Data fusion settings in ODCleanStore

4.9 Related Work

4.9.1 RDF Data Fusion Tools

To the best of our knowledge, there is just one another Linked Data fusion software – Sieve – currently under development [127]. Sieve is a part of the Linked Data Integration Framework [151], see Section 3.3, it adds quality assessment and data fusion capabilities to the LDIF framework. Sieve uses customizable scoring functions to output data quality descriptors. Based on these quality descriptors (and/or optionally other descriptors), Sieve can use configurable *FusionFunctions* to clean the data according to task-specific requirements.

Sieve offers functionality similar to our Data Fusion component; however, the purpose of Sieve in LDIF is different - it fuses data while being stored to the raw data mart and not during execution of queries, thus, provides no data fusion customization during data querying. This may be suitable when the desired data are known in advance, but it is not sufficient for open Web environments, where every consumer has different requirements on the integrated data; even the same consumer may have different requirements on the data depending on the task at his hand. Furthermore, ODCleanStore provides quality for each result statement, but Sieve computes quality only for whole named graphs.

4.9.2 Other Data Integration Systems

Aurora [165] is an integration system of heterogenous data residing in relational and object-oriented databases, i.e., deals with non-RDF data; its query model [166] enriches the SQL SELECT by enabling to define attribute conflict resolution functions (e.g., `age[ANY]` means that any attribute value for the attribute `age` is used in the query) and record conflict resolution functions, which deal with key attributes of the records. ODCleanStore offers more built-in conflict handling policies, on the other hand, Aurora allows user defined attribute aggregation

Name	Description	Possible values	Default value
`es`	a data fusion error strategy – handling of values for which data fusion fails	`IGNORE`, `RETURN_ALL`	`RETURN_ALL`
`aggr`	a global conflict handling policy	*string*	`ALL`
`multivalue`	a global multivalue setting	0, 1	0
`paggr` `[`*property*`]`	a conflict handling policy for the given property; example: `paggr[rdfs:label]=ANY`	*string*	*N/A*
`pmultivalue` `[`*property*`]`	a multivalue setting for the given property; example: `pmultivalue[rdf:type]=1`	0, 1	*N/A*

Table 4.6: Data fusion settings introduced in the URI or keyword query

functions; in ODCleanStore, record conflicts are either discovered by linkers, or there is no record conflict at all. Furthermore, the model enables to define for the query a conflict tolerant strategy – *HighConfidence*, *RandomEvidence*, and *PossibleAtAll*, specifying whether all attribute values, randomly chosen attribute value, or any attribute value, respectively, must satisfy the given query's `WHERE` condition. We currently do not solve any conflicts within `WHERE` conditions, because the data are requested by using URI or keyword queries, we do not support data fusion for SPARQL queries.

Fusionplex [139] is a system for integrating heterogeneous information sources residing in relational databases. Consumers can specify a conflict resolution function based on feature-based resolution (e.g., the probability of accuracy, availability, or cost) and content-based resolution, supporting elimination functions (MAX) or fusion functions (ANY, AVERAGE). The features of Fusionplex (e.g., accuracy, availability) represent data quality dimensions, which will be covered by ODCleanStore in the future. Similar to ODCleanStore, Fusionplex clusters the integrated data to *polytuples* representing different versions of the same information (database row).

OO_{RA} [118] is an object oriented global data model that can accommodate instance heterogeneities of attributes from local data models. The model enables to specify thresholds and conflict resolution functions for attribute level conflicts; if a conflict satisfies the threshold (e.g., the edit distance between two strings, values of the attribute, is at maximum 1), the conflict is tolerable and the preferred resolution function deduces the returned value; NULL is returned otherwise. In ODCleanStore, unless specifying the conflict handling policy AVOID, all conflicts are tolerable and we always try to return a value to the data consumer; the threshold approach is quite restrictive in the Web environment where two conflicting values can have very often high distance.

4.10 Summary

In this section we described and motivated data integration component of ODCleanStore. In particular, we presented a novel data fusion algorithm, which was implemented in ODCleanStore. We particularized important aspects of the algorithm – (1) solving conflicts among data, (2) computing integrated quality, and (3) being customizable by the data consumers.

We demonstrated that the data fusion algorithm is fast enough to work in real world settings – a request with 2500 α-equivalence classes of quads, i.e., more than a typical request "Give me all information about the resource X" triggers, has response time 1.75s for the most complicated conflict handling policy ALL.

We discussed for the given scenario (Section 4.7.3) the contribution of the data fusion algorithm to completeness, conciseness, and consistency of the datasets. For smaller datasets, such as Polish and German DBpedia dumps, the increase in completeness of the integrated dataset is in tens to hundreds of percents of the original datasets' completeness. Conflict resolution policies also guarantee the consistency of the integrated data, which is not guarantied by the original datasets.

Relevant Author's Publications

The content of this chapter is mainly covered by paper [106] describing the novel data fusion algorithm. The demo associated with paper [128] demonstrates that the data fusion algorithm was implemented in ODCleanStore, is running, and may be customized.

Main Contributions

The novel data fusion algorithm implemented in ODCleanStore presents one of the main contributions of the thesis. The detailed contributions of the data fusion algorithm are as follows:

- The data fusion algorithm supports the typical conflict handling strategies [58].
- Every resulting integrated quad is supplemented with (1) the integrated quality score and (2) source graphs contributing to the computation of the integrated (object) value of the quad.
- The data fusion algorithm is customizable – conflict handling policies may be customized on the global and per predicate level, a multivalue flag may be set on the global or per predicate level, a data fusion error strategy may be selected.

[58] Except of the conflict handling strategy AVOID, which was not implemented in the current version of ODCleanStore

5. W3P: Provenance Model for the Web

The previous chapters described how the data coming from the Web should be processed and integrated. In this chapter, we describe how the provenance data about such data should be expressed.

Section 5.1 presents motivational scenario, which should illustrate how data provenance helps Alice, the data consumer from Scenario 1.1, to decide which data is worth using. Further sections introduce the definition of provenance, description of the current provenance research focus, and how the data provenance needs for expressing provenance of the data on the Web differ from other domains' needs.

The core part of this chapter, Section 5.5, elaborates the requirements for the provenance model for the Web. Consequently, Section 5.6 builds the provenance model for the Web (called W3P), which should be used for expressing and tracking provenance behind data on the Web. In Section 5.7, a case study applying W3P provenance model to one of the use cases (introduced in Section 5.5) is presented.

In Section 5.8, we discuss how ODCleanStore would benefit from the W3P provenance model. Afterwards, in Section 5.9, we discuss the concept of provenance policies realizing the subjective consumers' provenance requirements on the consumed data; the provenance polices are intended to be enforced in the data filtering component of ODCleanStore.

5.1 Motivational Scenario

This motivational scenario illustrates how data provenance helps Alice, the data consumer from the Scenario 1.1, to decide which data is worth using. The Linked Data presented to Alice from the Czech Linked Open Data cloud poses many non-trivial questions she has to face before she may use such data, e.g.:

Q1 What is the origin of the given expert opinion about certain public contract? Is the origin trustworthy?

Q2 Is the expert opinion provided by `http://wearetheexperts.com` original or derived from another source? If derived, which source is quoted? How many organizations/people support that opinion?

Q3 When was the article "We are constructing the most expensive highways in Europe" published?

Q4 Which of the contradicting values for the estimated price of the contract should be preferred?

Q5 Does Alice has the rights to reuse the images published by "TopPragueNews"?

Provenance or lineage of the data helps Alice significantly to answer all Questions Q1 – Q5, because provenance provides the necessary contextualization for the information consumer to analyze the quality of the information at hand [136, 60, 83].

Furthermore, provenance allows for transfer of trust from entities behind the resources to the information in the resources [70]. As a result, if provenance information relates a certain article to a certain author who is known and trusted by Alice, she may also trust the content of that article.

To further clarify the substantial importance of provenance for the information consumer, we point out to an experiment introduced by Pinheiro da Silva et al. [45]. In this experiment, the scientists were trying to identify and explain imperfections of a set of maps. The results show that around 80% of scientists correctly identified the imperfections of the maps when they know data provenance of these maps. Without any provenance information at hand, the same scientists were able to identify only around 10% of all map imperfections.

5.2 Provenance – Definition of the Term

According to the Oxford English Dictionary[59], provenance or lineage is defined as "(i) the fact of coming from some particular source or quarter; origin, derivation. (ii) the history or pedigree of a work of art, manuscript, rare book, etc.; concretely, a record of the ultimate derivation and passage of an item through its various owners." In other words, provenance should be comprehended either as the source or derivation of an object or as a record of such derivation.

In data processing systems, Moureau [136] presented a generic "Provenance as a Process" definition – "the provenance of a piece of data is the process that led to that piece of data". Moureau [136] also summarized other definitions describing provenance in different contexts – such as in the area of relational databases or workflow systems. The definitions in the database area focus on certain aspects of data provenance, such se *where-provenance* [35], which is able to reveal "where the data was copied from", but is not able to address questions Q2 – Q5. "Provenance as Annotations" approach [136] extends "Provenance as a Process" approach, because it also deals with certain properties of past processes; such descriptive properties of processes are defined by ontologies to provide structure and semantics to these properties.

In this paper we use the definition of provenance, which is based on Moureau's definition of "Provenance as a Process" but which is combined with "Provenance as Annotations" approach [136] to cover also the descriptive dimension of provenance. The definition is as follows:

Definition 5.1. Provenance of a resource r is a record containing the description of the processes, artifacts, and agents involved in creation of r.

5.3 Provenance as a Research Topic

The main part of the research in provenance was so far concentrated in the areas of scientific data processing and database management systems [136].

[59] `http://www.oed.com`

Scientific Data Processing

In the context of provenance for Scientific Workflow Management Systems (SWfMS), Serra da Cruz et al. [42] and Simmhan et al. [153] present provenance taxonomies and categorize some of the existing workflow systems, in order to motivate today's provenance research and efforts to establish a common provenance model; paper [46] provides an overview of provenance in workflow systems. Paper [27] describes basic components of systems providing provenance retrieval for scientific data products.

Database Management Systems

Provenance was examined in the area of relational databases for years [155, 34, 39]. Buneman et al. [35] focused on the provenance of query results; they distinguish *why-* and *where-provenance*: why-provenance of an output represents the origins that were involved in calculating a single entry of a query result, i.e., it provides a set of witnesses; however, why-provenance does not provide additional information on how the output tuple is actually derived. Where-provenance refers to the exact locations an element of a query result has been extracted from. Green et al. [79] additionally introduce *how-provenance* which, in contrast to why-provenance, describes how the origins were involved in the calculation of the resulting value.

5.4 Provenance on the Web

"Historically, databases and other electronic information sources were trusted, because they were under centralized control: it was assumed that trustworthy and knowledgeable people were responsible for the integrity of data in databases or repositories." [39] This assumption is no longer true – the Web is evolving into a complex information space where the unprecedent volume of documents and data will offer to the information consumer a level of information integration and aggregation that has up to now not been possible. Data is often made available on the Web with no centralized control over its integrity [119], furthermore, data is constantly being copied and combined indiscriminately, which is further boosted in the Linked Data era, leading to inherent problems such as no way how to find out the original source of the statement, provision of poor quality or fraudulent information.

Tracking provenance of data on the Web is essential and represents the cornerstone element when analyzing and assessing the quality of the information by providing the necessary contextualization of the consumed information and enabling the information consumer to justify the usefulness of the used data [136, 60, 83]; paper [155] states "Information about provenance constitutes the proof of correctness [...] and [...] determines the quality and amount of trust [...]". Similarly, as stated in [39], "it is of utmost importance to understand the provenance of data in the resulting database, in order to check the correctness of an ETL specification or assess the quality and trustworthiness of curated data."

However, different Web communities have distinct views of provenance. While some consumers may view the quality of information by focusing on the processes which generated the information, others may focus on the artifacts created by the publication process. Common across these communities is the need to assess the quality and trustworthiness of the consumed information [6]. The generic use of provenance for quality assessment and trust, common across different Web communities, is the fundamental use case for provenance on the Web. In this chapter, provenance is analyzed under this perspective.

Existing works on provenance usually approach provenance under the requirements of scientific workflow systems. The nature of information consumption and publishing on the Web strongly reshapes the requirements of a provenance model suitable for the Web environment. The focus of provenance is shifted in the context of the Web, where provenance should attend a broader audience. Different communities coexist in the Web space, with different perspectives about information, which ultimately drives the way the information is represented, generated or made available via queries. As a result, provenance starts to move in the context of the Web towards a comprehensive and structured description of the history, current state and context of an information resource – the descriptive dimension is combined with the pure provenance as a process approach [136]. We approach provenance under this perspective.

5.5 Requirements for a Provenance Model for the Web

The strategy for building the W3P provenance model for the Web is based on creation of a set of requirements for that provenance model. These requirements are built considering three types of analysis. In the first analysis (Section 5.5.1), considering the centrality of provenance as a way for enabling quality assessment, we investigate a definition for information quality on the Web and outline important information quality dimensions the data provenance should support. Next, in Section 5.5.2, four representative use cases of provenance consumption and publishing on the Web are described. The use cases strongly reflect the focus of the provenance model on quality assessment; quality assessment drives the design of the provenance model. The third analysis covers a literature survey to establish a set of core requirements for the W3P provenance model.

5.5.1 Information Quality on the Web

The definition of a standardized provenance model can strongly impact the effectiveness on which consumers enforce their quality criteria. Therefore, the creation of a comprehensive provenance model is a fundamental step towards enabling information quality assessment on the Web. Considering the centrality of provenance as a tool for enabling quality assessment, we investigated a definition for information quality on the Web and summarized the widely used information quality dimensions, so that we can then discuss to which extent the provenance model proposed enables/supports the quality assessment.

The perception of information quality (term used in the literature interchangeably with data quality) is highly dependent on the fitness for use [112] being relative to the specific task that users have at hand. Information quality is usually described in different works by a series of quality dimensions which represent a set of desirable characteristics for an information resource (see [112] for a survey of the main information quality frameworks). The set of information quality dimensions used in this work were composed by the dimensions described in the works of Wang & Strong [162], Alexander & Tate [1] and the set of most commons information quality dimensions taken from the comprehensive survey of Knight & Burn [112]. Wang & Strong [162] cover a domain independent set of quality dimensions, while [112] and [1] cover quality dimensions for the Web. In this work we revisit quality dimensions introduced in [112, 162, 1] by merging them into a single set of dimensions:

1. **Accuracy/Correctness:** Represents the extent to which the information is correct and accurate enough for its primary intended use (present in [112, 162, 1]).

2. **Compliance:** Covers the extent to which the processes and methodologies behind the data are compliant with the consumers' processes and methodologies (present in [112, 162]).

3. **Completeness:** Covers the sufficiency of information for the information consumer (present in [112, 162]).

4. **Consistency:** Covers the consistency of the data representation and its model (present in [112, 162]).

5. **Interpretability:** Represents the quality of the description/model behind the data. This dimension also covers the suitability of the units or language on which the data is expressed (present in [112, 162]).

6. **Usability:** Represents the extent to which the information is helpful for a specific task. In the context of the Web we complement the definition considering the suitability of use in relation to its primary intended use and potential restrictions on the usage of the data (present in [112, 162]).

7. **Reputation:** Represents the entities (organizations, individuals) which recommend or repudiate the data, and the trustworthiness of the entities behind the production of a data artifact (present in [112, 162, 1]).

8. **Security:** Covers the security mechanisms which enforce the data integrity (present in [112, 162]).

9. **Timeliness:** Represents the extent to which the information is sufficiently up-to-date (present in [112, 162, 1]).

10. **Objectivity:** Represents the extent to which the information is unbiased and impartial (present in [112, 162, 1]).

11. **Accessibility:** Represents the extent to which the information is available and easily retrievable; from the Linked Data perspective this dimension can represent the appropriate choice and reuse of vocabularies (present in [112, 162]).

12. **Navigation:** Covers the extent to which the data is easily found and linked (present in [112, 162, 1]).

13. **Concise:** Represents the extension to which the information is compactly represented (present in [112, 162]).

5.5.2 Provenance Use Cases

This section contains typical use cases of trust decision and quality assessment for applications consuming and publishing provenance information on the Web. These scenarios were developed to maximize the coverage of the use of the provenance model for the Web, both on document and data level. Each use case concentrates on specific provenance problems, with the overlap between some of their features representing the most common provenance uses. The set of use cases summarizes general application areas and is not intended to be an exhaustive investigation of provenance usage in different domains. For more scenarios, please see [71].

Use Case I: Data Integrity and Provenance Tracking in Aggregation of Financial Data

Description: A financial analyst is using an application that consumes Linked Data from a large number of distributed Web datasets. The datasets include open, government, and partner data in the form of stock markets time series, news, blog posts, government data, demographics, previous analysis, third-party qualitative and quantitative analysis, and economic facts. The data is directly referenced in a financial report which provides a summarized overview of the economic context of the previous month.

Provenance use: In the process of building the report the analyst uses provenance to determine the trustworthiness of analysis provided by third party organizations (each organization is an authoritative expert in a specialized market segment). Provenance is also used to determine the analysts (agents) behind the information, since only analysis generated by expert analysts is used. The publisher of the information and its certificate should be available as provenance information in order to be automatically checked. Any news excerpts should have its associated publisher and time information. Each analysis process behind the generation of a report generates a provenance workflow.

Use Case II: Content Aggregation and Social Provenance in a Web Mashup

Description: A startup is creating a mashup to organize information available on the Web about cars. The website will cover a wide range of interests including press releases, technical specifications, reviews, maintenance tips, brand monitoring, sales offers, etc. Free information available from third parties (e.g., Wikipedia) or information provided by partners will be embedded in the website, while copyrighted content will be exposed as links. Tweets and blog posts will be used to monitor the buzz behind a brand or a car. The information of the mashup will be made available as Linked Data.

Provenance use: The provenance of tweets and blog posts (author, creation/modification date, publisher) need to be tracked and will be further used for better filtering of the contents. Readers may be able to support or reject a specific resource and this information should be made available as provenance to other readers and to consumers of the Linked Data made available. Every external content embedded on the website should be explicitly quoted and its source, tracked. Usage terms and licensing of third parties of digital artifacts should be represented together with the provenance information.

Use Case III: Workflow Provenance Tracking, Interoperability, Timeliness and Licensing for Collaboration on the Pharmaceutical Industry

Description: Pharmaceutical organizations are using the Web to cooperate in a common project for Drug Discovery. Each member of the consortium has an access to its internal, partners and public datasets. There are strong cooperation constraints for each partner and trust, security and privacy are key factors to enable an effective collaboration.

Provenance use: Provenance is used to enforce the domain of the partnerships: organization X can cooperate with organization Y in molecular interactions and can cooperate with organization Z in genomic-protein mapping. Each cooperation agreement has an associated time range and terms of usage associated as provenance information with the data. Each group member trusts a different set of public datasets and the provenance of the sources of the data should always be verified. Due to compliance policies and for re-enactment purposes, provenance of the data should also be tracked on the fine grained experimental workflow level. In this scenario provenance is an important tool for experimental investigation and the ability to query and navigate through the model plays an important role for extracting research value from the information collected. In addition, different members of the consortium use different scientific workflow systems which will need to consume the provenance information of different systems. Vocabularies used in a specific dataset and the linkage with other datasets can provide essential information about the understandability of the data and should be appropriately described. The trail of historical changes of a data item should be preserved. Data provided by the consortium for the public use has strong constraints on its usage.

Use Case IV: Geographical and Descriptive Provenance Information for Sensor Networks

Description: A European consortium in climate change is using a set of environmental sensors distributed in different countries. The data collected from the sensors is published on the Web. Since the sensor infrastructure is inherited from different organizations and application domains, there is a strong heterogenity in the conditions and the quality of the data provided. Environment researchers, the end users of the data aggregated from the sensor mashups, need provenance information to determine the quality of the data.

Provenance Use: Provenance is used to track the physical location of the sensor, the sensor type/model, timeliness, owner organization, operating conditions, uncertainties associated with the data, and measurement units.

5.5.3 Literature Review and List of Requirements

Different works in the literature cover distinct perspectives and features of provenance models. Key works in the process of collecting the list of requirements were the extensive survey of the provenance on the Web [136] and the list of requirements for recording and using provenance in eScience experiments [131]. Below, a list of requirements is provided. The requirements are defined by: their incidence in the literature and their existence in available web vocabularies/provenance models, their coverage of the use cases (Section 5.5.2) and their coverage of the quality dimensions in Section 5.5.1 (to which extent the requirement supports the given quality dimension). The requirements detailed here focus on the design of the formal representation (provenance model) and do not address general infrastructure requirements (such as availability, access, or scalability of provenance). In further text, we refer to these requirements by using the expressions Req. 1.

1. **Interoperability:** Maximization of the interoperability with existing provenance models and vocabularies (covered in [136, 137, 59, 62]; use cases I, III; quality dimensions 2, 3, 4, 5, 6, 11).

2. **Extensibility:** Support for addition of domain specific provenance information (requirement based on the multiplicity of provenance models and applications expressing provenance, covered in [4, 33, 36, 44, 46, 53, 54, 55, 61, 68, 153]; use case III; quality dimensions 2, 3, 5, 6, 11, 12).

3. **Well Defined Relational Model/Logical Model/Grounded Semantics:** Suitability for a wider audience, ability to map to the conceptual model of users, appropriate level of abstraction and grounded semantics; have a strong impact on the usability of the model (use cases I, II, III, IV; quality dimensions 4, 5, 6).

4. **Fine-Grained & Coarse-Grained Provenance Information:** Ability to express the description of both fine grained (e.g., statement-level) or coarse grained (dataset/document-level) information resources. The provenance model should be able to describe both types of granularities (covered in [5, 35, 50, 164]; use case III; quality dimensions 2, 3, 5, 6).

5. **Generality:** Coverage of provenance description demands of different communities over the Web (requirement based on the multiplicity of provenance models and applications, covered in [4, 33, 36, 44, 46, 53, 54, 55, 61, 68, 153]; use cases I, II, III, IV; quality dimensions 5, 6, 11).

6. **Data Generation & Transformation (Workflow) Description:** Formalization of the description of the processes behind the generation and transformation of the information. For most use cases it is the core of the provenance description and in some scenarios should be fine grained enough to allow the reproduction of a workflow. Dependencies between artifacts, customizable roles of the agents in the processes, and hierarchies of agents are covered by this category of descriptor (Covered in [136, 137, 131, 46, 53, 50, 47, 169]; use cases I, III; quality dimensions 2, 3, 5, 6).

7. **Spatiality:** Tracking of the geographic location of the information. Spatial information is important in a set of scenarios including tracking of geospatialized artifacts (such as sensor data) and assessment of geospatial trustworthiness and restrictions (covered in [54, 55, 30, 150]; use case IV; quality dimension 3).

8. **Temporality:** Assessment of the timeliness of the information. Provenance consumers will need to track the temporal evolution of the information resource (covered in [83, 84, 49, 63], concept present in most of the scientific workflows [136, 137, 131, 46, 53, 50, 47, 169]; use case I, III, IV; quality dimensions 3, 5, 9).

9. **Contracts, Digital Rights & Licensing:** This requirement covers the formalization of the usage conditions of the published artifact (covered in [54, 55, 30, 150, 41, 65, 132]; use case II, III; quality dimensions 2, 3, 6).

10. **Integrity Mechanisms:** Availability of descriptors for the integrity mechanisms used for both the information resource and its provenance information. Examples are signatures and encryption descriptors (covered in [136, 83, 15, 28, 38]; use cases I, III; quality dimension 8).

11. **Identity Warranties:** Availability of mechanisms which can provide identity warranties for other provenance elements which support an identity (individuals and organizations). Examples of identity warranties are digital certificates (covered in [83, 15, 28, 38]; use cases I, III; quality dimension 8).

12. **Content Description/Annotation:** Availability of content descriptors about information resources – tags, titles, natural language descriptions, justifications (covered in [61, 49, 2]; use cases I, II, III; quality dimension 3, 6).

13. **Change Tracking:** Ability to describe changes and versioning of an information resource (covered in [49, 158]; use case III; quality dimension 3).

14. **Coverage of Social Provenance:** Ability to model support/quote/discourage relationships between entities (individuals and organizations) and information resources (covered in [82, 15, 32]; use cases I, II; quality dimensions 3, 7).

15. **Publishing & Ownership:** Represents the information related to the publisher entities and processes and the ownership over the information resource (covered in [49, 83, 15]; use cases I, II, III, IV; quality dimensions 2, 5, 7).

16. **Query Expressivity and Navigability:** The representation of the provenance model should allow users to launch expressive queries over the model. The provenance model should allow users to navigate through its entities (covered by a large set of different works – Section 4.4 of [136]; use cases III; quality dimensions 3, 6, 11, 12).

17. **Meta-provenance:** Represents descriptors over provenance data, including provenance annotations and permission control over the provenance model entities (covered in [136, 28, 154]; use cases I, III; quality dimension 8).

5.6 W3P Provenance Model

We discussed the requirements for a provenance model for the Web and the rationalities behind each requirement. This section discusses the construction of such model (called W3P) based on the set of introduced requirements and is done in several steps.

Firstly, to maximize the reuse of currently existing vocabularies when constructing the provenance model, in Section 5.6.1, the current vocabularies for describing web resources are surveyed.

Secondly, in Section 5.6.2, we outline the general design decisions yielding from the requirements. Since the provenance model is built over (Semantic) Web standards, the suitability of these standards to the requirements is verified. The vocabularies potentially useful for reuse (identified in Section 5.6.1) must adhere to these general design decisions. Then, in Section 5.6.3, we derive from the identified requirements a set of key provenance concepts, which introduce important categories of terms that should be covered by the W3P provenance model; we also discuss in this section to which extent current vocabularies identified in Section 5.6.1 cover these concepts.

Lastly, in Section 5.6.4, we describe the construction of the W3P provenance model, which adheres to the general design decisions in Section 5.6.2, employs the suitable terms from current vocabularies (Section 5.6.1) to cover the concepts identified in Section 5.6.3, and provides new terms to cover the concepts identified in Section 5.6.3 which are not covered or poorly covered by current vocabularies.

5.6.1 Current Vocabularies

In this section we survey the current vocabularies usable for expressing data provenance on the Web. Further in the text, if we refer to *current vocabularies*, we mean the current vocabularies presented in this section. Table 5.1 depicts abbreviations for the current vocabularies described further, namespace prefix used to reference the terms from these vocabularies, and namespace URI.

Open Provenance Model 1.1. (OPM)

OPM provides a solid foundation for modeling workflow provenance and could be the base for the interoperability of W3P. OPM authors provides an abstract provenance model and also two serialization to OWL – OPMV [168], a lightweight version of the abstract provenance model and OPMO [135], the full version. OPMO reuses terms of OPMV and extends it with new terms.

In OPMV, they define three classes for three core types of *provenance entities* – `opmv:Agent`, `opmv:Process`, and `opmv:Artifact`. Furthermore, they define five predicates expressing *binary relations* between these core types of provenance entities (depicted in Figure 5.1) – `opmv:wasGeneratedBy` expressing that

Vocab.	**Prefix**	**Namespace**
OPM (OPMV)	opmv:	`http://purl.org/net/opmv/ns#`
OPM (OPMO)	opmo:	`http://openprovenance.org/model/opmo#`
DCMI (DC)	dc:	`http://purl.org/dc/terms/`
DCMI (DCTY)	dcty:	`http://purl.org/dc/dcmitype/`
FOAF	foaf:	`http://xmlns.com/foaf/0.1/`
SWP	swp:	`http://www.w3.org/2004/03/trix/swp-2/`
VOID	void:	`http://rdfs.org/ns/void#`
CS	cs:	`http://purl.org/vocab/changeset/schema#`
WGS84	geo:	`http://www.w3.org/2003/01/geo/wgs84_pos#`
CC	cc:	`http://creativecommons.org/ns#`
SIOC	sioc:	`http://rdfs.org/sioc/ns#`
TIME	time:	`http://www.w3.org/2006/time`
TRIX	trix:	`http://www.w3.org/2004/03/trix/rdfg-1/Graph`

Table 5.1: Namespace prefixes for the current provenance vocabularies

an `opmv:Artifact` was generated by a `opmv:Process`, `opmv:used` expressing that an `opmv:Artifact` was used by a `opmv:Process`, `opmv:wasControlledBy` expressing that a `opmv:Process` was controlled by an `opmv:Agent`, `opmv:wasDerivedFrom` expressing that an `opmv:Artifact` was derived from another `opmv:Artifact`, and `opmv:wasTriggeredBy` expressing that a `opmv:Process` was triggered by another `opmv:Process`.

In OPMO, the predicates expressing binary relations in OPMV are further supplemented with the classes `opmo:Used`, `opmo:WasGeneratedBy` (WGB), `opmo:WasDerivedFrom` (WDF), `opmo:WasControlledBy` (WCB) and `opmo:WasTriggeredBy` (WTB); we call such classes as *relation classes*, because they provide an alternative way to express relations between `opmv:Artifact`, `opmv:Agent`, and `opmv:Process` classes; each such relation class corresponds with one predicate in OPMV. The purpose of these relation classes is to provide ways to model n-ary relations in RDF. Each such class has a causal dependency, between the entity denoting the effect of the relation and the entity denoting the cause of the relation. Furthermore, other predicates may be associated with these relation classes to express further details about the relation, e.g., its role (class `opmo:Role`) or timestamp (class `opmo:Time`). In Figure 5.2, an OPM provenance graph is depicted – a directed graph, whose nodes are instances of classes `opmv:Artifact`, `opmv:Process`, `opmv:Agent`, `opmo:Used`, `opmo:WasGeneratedBy`, `opmo:WasDerivedFrom`, `opmo:WasControlledBy`, `opmo:WasTriggeredBy`, `opmo:Role`, and `opmo:Time` [137]. The particular instances of the depicted classes can belong to one or more *accounts* – evidences of provenance (each source may have more evidences as the provenance could have been recorded, e.g., by two different tools).

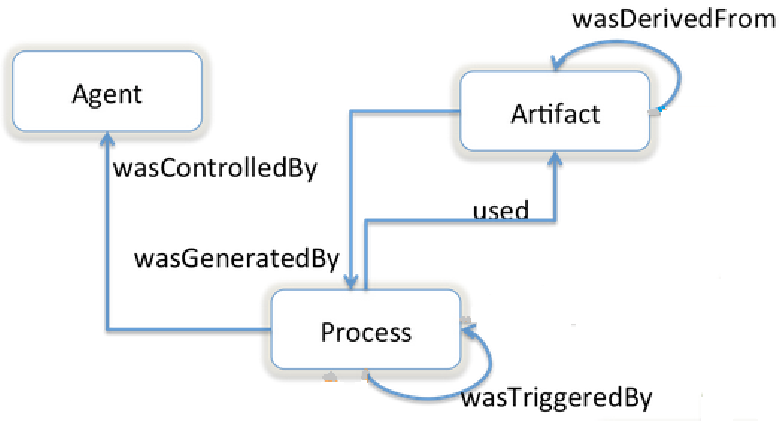


Figure 5.1: OPMV ontology
(Source: `http://open-biomed.sourceforge.net/opmv/ns.html`)

Dublin Core Metadata Initiative (DCMI)

The Dublin Core Metadata Initiative[60] maintains a set of widely accepted metadata to be used with resources [49], such as `dc:creator` or `dc:license` of the resource. The metadata are expressed in RDF data format, so that they can be used to track provenance of Linked Data resources. Most of the relevant terms are from DC namespace, some of them are from DCTY namespace [49].

Friend of a Friend Vocabulary (FOAF)

The Friend of a Friend (FOAF) [32] is an RDF vocabulary capable of describing individual agents, groups of agents, relations between agents (e.g., that a person knows another person), and artifacts the agents created or currently work on (e.g., projects). The core classes are `foaf:Person`, `foaf:Organization`, `foaf:Document`, and `foaf:Project` together with properties characterizing these classes.

Semantic Web Publishing Vocabulary (SWP)

Semantic Web Publishing Vocabulary [15] is an RDF-Schema vocabulary for expressing digital signatures of named graphs and also representing authorities endorsing these signatures.

Vocabulary of Interlinked Datasets (VOID)

VOID [43] is an RDF based schema to describe RDF datasets and linksets; “with VOID, the discovery and usage of Linked Datasets can be performed both effectively and efficiently” [2].

Creative Commons (CC)

Creative Commons vocabulary [41] is an RDF vocabulary for describing licences of resources – what they permit, prohibit, etc.

[60] `http://dublincore.org/metadata-basics/`

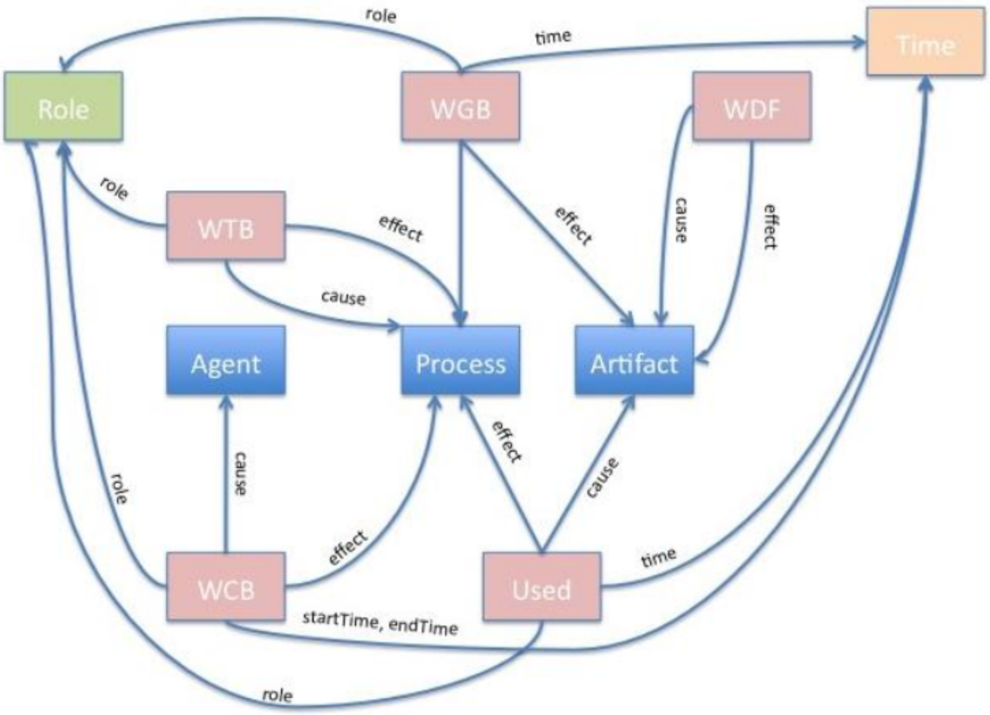


Figure 5.2: Core concepts of OPMO ontology
(Source: `http://openprovenance.org/model/opmo`)

SIOC

SIOC[61] provides an ontology for representing rich data from the social web, such as blog posts, forums [24].

ChangeSet (CS)

ChangeSet[62] is a vocabulary defining terms for describing changes in resources; it introduces the notion of `cs:ChangeSet` encapsulating the deltas between two versions of a resource.

Basic Geo Vocabulary (WGS84)

WGS84 is “a basic RDF vocabulary that provides the Semantic Web community with a namespace for representing lat(itude), long(itude) and other information about spatially-located things” [30].

OWL-Time (TIME)

An OWL ontology representing time instances and intervals [63].

[61] `http://sioc-project.org/`

[62] `http://vocab.org/changeset/schema.rdf`

[63] `http://www.w3.org/2006/time`

TRIX

A single purpose vocabulary to explicitly express that a certain resource is a named graph. Furthermore, TRIX may be used to express hierarchies of named graphs and equivalency of two named graphs [64].

5.6.2 General Design Decisions

W3P is designed to provide a generic model for representing provenance information on the Web. W3P is built over (Semantic) Web standards (HTTP, URIs, RDF/RDFS, OWL, SPARQL); thus, the the suitability of these standards to the outlined provenance requirements has to be verified.

The use of (Semantic) Web standards allows W3P to address Reqs. 1 – 5, 16. Interoperability (Req. 1) is covered by reusing existing provenance vocabularies. Further, by using predicates, such as `owl:equivalentClass`, `owl:equivalentProperty` and `owl:sameAs`, we can map the equivalence of different classes, properties and individuals impacting on interoperability and extensibility (Req. 2). Extensibility (Req. 2) is one of the built-in strengths of the Semantic Web, where schemas can be easily extended and merged. Well Defined Relational Model/-Logical Model/Grounded Semantics (Req. 3) is covered by W3C standards RDF, RDFS and OWL. The use of URIs as identifiers provides the basic infrastructure for unambiguously expressing concepts, impacting also on Req. 3. The provenance model must be independent of data provenance granularity, i.e., allowing users to describe the provenance of different web artifacts including data, documents, or datasets. Fine-Grained & Coarse-Grained Provenance Information (Req. 4) can be partially addressed with the deployment of reification, named graphs or dataset level descriptors. Semantic Web models provide an expressive and generic way to create representations of provenance models both under a graph or a description logic perspective (Generality, Req. 5). SPARQL provides an expressive query language for querying the provenance model covering the Query Expressivity (Req. 16). The use of dereferenceable URIs and RDF data model allows the coverage of the Navigability (Req. 16).

5.6.3 Provenance Concepts and their Coverage by the Current Vocabularies

From the set of requirements (Reqs. 6 – 15, 17), a collection of key *provenance concepts* is identified. The key provenance concepts represent broader categories which are used to verify the provenance coverage of the current vocabularies available on the Web (Section 5.6.1). The concepts not covered sufficiently by current vocabularies motivate the design of new classes and properties in Section 5.6.4.

A summarized overview of the provenance concepts is described in Table 5.2. Every concept is accompanied with its label, description, analysis of the coverage of the provenance concept by current vocabularies, and list of requirements from Section 5.5 motivating the given provenance concept. The coverage of the provenance concept by the given current vocabulary is either complete (the given vocabulary abbreviation is bolded) or partial (the vocabulary abbreviation is not

[64] `http://www.w3.org/2004/03/trix/rdfg-1/Graph`

bolded); in order to get complete coverage, the vocabulary has to provide not only terms to represent the given concept, but also terms to relate the concept to other concepts (e.g., to relate a time of creation to the creation process).

The following discussion clarifies the partial/complete coverage of the concepts by current vocabularies and also introduces for each concept a detailed list of terms (classes and predicates) from the current vocabularies covering that concept [65]. Predicates in the detailed list of terms may be supplemented with their domains and ranges – e.g., a predicate `opmv:used` with the domain `opmv:Process` and range `opmv:Artifact` is abbreviated as `opmv:used` (`opmv:Process` → `opmv:Artifact`). The list of domains and ranges may not be complete (certain rather generic or abstract classes are omitted, e.g., `rdfs:Literal`, `rdfs:Resource`, or `rdf:Property`); if the domain is not introduced, it is written as, e.g., `dc:provenance` (→ `dc:ProvenanceStatement`), if the range is not introduced, it is written as, e.g., `geo:lat` (`geo:SpatialThing`).

Artifact

The concept of an artifact is covered by OPM and DCMI vocabularies; they both provide general relations between two artifacts (e.g., `dc:relation, dc:references`, `opmo:wasDerivedFrom`). Other vocabularies provide only partial coverage: FOAF does not provide enough terms for expressing hierarchies of artifacts, i.e., that one artifact is part of another artifact, VOID provides good dataset descriptors but it is not suitable for describing artifacts with lower granularity, TRIX vocabulary enables to describe named graph artifacts, their equivalency and hierarchy, but terms for describing other types of artifacts are missing. The detailed coverage of the concept artifact by current vocabularies is as follows:

- `opmv:Artifact`, `opmv:wasGeneratedBy` (`opmv:Artifact` → `opmv:Process`), `opmv:wasDerivedFrom` (`opmv:Artifact` → `opmv:Artifact`), `opmv:wasEncodedBy` (`opmv:Artifact` → `opmv:Artifact`)

- `opmo:WasGeneratedBy`, `opmo:WasDerivedFrom`, `opmo:causeWasGeneratedBy`, `opmo:effectWasGeneratedBy`, `opmo:causeWasDerivedFrom`, `opmo:effectWasDerivedFrom`, `opmo:OPMGraph`

- `dcty:Dataset`, `dc:hasPart`, `dc:isPartOf`, `dc:source`, `dc:BibliographicResource`, `dc:PhysicalMedium`, `dc:PhysicalResource`, `dc:Standard`, `dc:references`, `dc:relation`, `dc:isRequiredBy`, `dc:ProvenanceStatement`, `dc:provenance` (→ `dc:ProvenanceStatement`)

- `foaf:Document`, `foaf:Image`

- `void:Dataset`, `void:Linkset`, `void:subset` (`void:Dataset` → `void:Dataset`)

- `trix:Graph`, `trix:equivalentGraph` (`trix:Graph` → `trix:Graph`), `trix:subGraphOf` (`trix:Graph` → `trix:Graph`)

[65] Such list should not be used as an ultimative reference, the particular vocabulary namespace URI should be always confronted to get the full list of terms.

Concept	**Description**	**Coverage**	**Reqs.**
Artifact	Any physical, digital, conceptual, or other kind of artifact that is the input or the product of a process. An artifact can be the origin or a part of a different artifact.	**DCMI** FOAF **OPM** TRIX VOID	6
Process	An operation associated with the generation and transformation of an artifact. A process may be a part of another process.	OPM	6
Agent	Contextual entity acting as a catalyst of a process, enabling, facilitating, controlling, affecting its execution. Agents can form hierarchies and groups.	DCMI **FOAF** OPM	6
Role	A role designates the artifact's or agent's function in a process.	**OPM**	6
Spatial Information	Explicit spatial information associated with a provenance entity.	DCMI FOAF WGS84	7
Temporal Information	Explicit temporal information that could be associated with a provenance entity. Expiration, creation, modification datetime, and valid range are examples of temporal descriptors.	DCMI OPM TIME	8
License	Descriptors specifying the rights associated with the usage of the artifact.	**CC** **DCMI**	9
Integrity and Identity Warranties	Integrity warranties associated with an artifact (e.g., a digital signature). Identity warranties (certification authorities) that issue Integrity Warranties.	FOAF **SWP**	10, 11
Descriptors	Human or machine readable descriptors providing a less constrained detailment over the provenance entity.	DCMI FOAF OPM VOID	12, 17
Change tracking	Represents the tracking of the changes of an artifact.	CS DCMI SIOC	13
Social Descriptors	An individual or organization that advocates an artifact, process, agent or any other entity.	SWP	14
Creator/ Modifier	An individual or organization responsible for creation/modification of an artifact.	**DCMI** FOAF	15
Publisher	An individual or organization responsible for publishing an artifact.	**DCMI**	15
Owner	An organization or an individual which owns the rights over an artifact.	**DCMI** SIOC	15
Host	An organization that provides the infrastructure for the publication of an artifact.		15
User (Data Consumer)	An individual or organization which uses/ consumes/accesses an artifact.		15

Table 5.2: Key provenance concepts, their definition, summarized coverage in vocabularies and requirements (Reqs.)

Process

The concept of a process is covered by OPM, but it is still a partial coverage – OPM does not provide a way to describe hierarchy of processes. The detailed coverage of the concept by current vocabularies is as follows:

- opmv:Process, opmv:used (opmv:Process → opmv:Artifact), opmv:wasControlledBy (opmv:Process → opmv:Agent), opmv:wasPerformedBy (opmv:Process → opmv:Agent), opmv:wasTriggeredBy (opmv:Process → opmv:Process)
- opmo:Used, opmo:WasControlledBy, opmo:WasTriggeredBy, opmo:causeWasTriggeredBy, opmo:effectWasTriggeredBy, opmo:effectWasControlledBy, opmo:causeWasControlledBy

Agent

FOAF covers completely the concept of agent, it provides terms for describing groups of agents. OPM and DCMI cannot express the group membership of agents. The detailed coverage of the concept by current vocabularies is as follows:

- opmv:Agent, opmv:wasControlledBy (opmv:Process → opmv:Agent), opmv:wasPerformedBy (opmv:Process → opmv:Agent)
- dc:Agent, dc:AgentClass
- foaf:Agent, foaf:Person, foaf:Organization, foaf:Group, foaf:membershipClass (foaf:Group → foaf:Agent), foaf:member (foaf:Group → foaf:Agent),

Role

Since a process may have used several artifacts, it is important to identify the roles under which these artifacts were used. Roles are well covered by OPM. The detailed coverage of the concept by current vocabularies is as follows:

- opmo:Role, opmo:role (opmo:Used, opmo:WasControlledBy, opmo:WasGeneratedBy → opmo:Role), opmo:value (opmo:Role)

Spatial Information

WGS84 provides the most precise spatial information by enabling to represent geo-coordinates of locations; however, there are numerous alternative ways (e.g., address, landmark) how to express a location. DCMI provides class dc:Location to represent spatial information and predicate dc:spatial to relate an artifact to dc:Location; however, DCMI does not further specify how the spatial information should be expressed. FOAF provides the predicate foaf:based_near, which is based on the subjective human notion of proximity. The detailed coverage of the concept by current vocabularies is as follows:

- geo:SpatialThing, geo:Point (subclass of geo:SpatialThing), geo:location (→ geo:SpatialThing, subproperty of foaf:based_near), geo:lat (geo:SpatialThing), geo:long (geo:SpatialThing), geo:alt (geo:SpatialThing), geo:lat_long (geo:SpatialThing)

- foaf:based_near (geo:SpatialThing → geo:SpatialThing)
- dc:Location, dc:spatial (→ dc:Location)

Temporal Information

The temporal information concept is covered by OPM (but the concept of validity of resources is missing), DCMI provides a poor set of temporal predicates from the process perspective. TIME provides a low level vocabulary describing the time instant, time interval, proper time interval, etc., but TIME does not provide any terms to relate these time expressions to an artifact. The detailed coverage of the concept by current vocabularies is as follows:

- opmv:wasStartedAt (opmv:Process → time:Instant), opmv:wasEndedAt (opmv:Process → time:Instant), opmv:wasUsedAt (opmv:Process → time:Instant), opmv:wasGeneratedAt (opmv:Process → time:Instant), opmv:wasPerformedAt (opmv:Process → time:Instant)
- opmo:OTime, opmo:exactlyAt (opmo:OTime → xsd:dateTime), opmo:noEarlierThan (opmo:OTime → xsd:dateTime), opmo:noLaterThan (opmo:OTime → xsd:dateTime), opmo:time (opmo:EventEdge → opmo:OTime), opmo:endTime (opmo:WasControlledBy → opmo:OTime), opmo:startTime (opmo:WasControlledBy → opmo:OTime)
- dc:date, dc:dateAccepted, dc:dateCopyrighted, dc:dateSubmitted, dc:temporal (→ dc:PeriodOfTime), dc:PeriodOfTime, dc:created, dc:issued, dc:valid, dc:modified, dc:available
- time:TemporalEntity, time:Instant, time:Interval, time:ProperInterval, time:before (time:TemporalEntity → time:TemporalEntity), time:hasBeginning (time:TemporalEntity → time:Instant), time:hasEnd (time:TemporalEntity → time:Instant), time:inside (time:Interval → time:Instant), time:intervalOverlaps (time:ProperInterval → time:ProperInterval), time:inXSDDateTime (time:Instant → xsd:dateTime)

License

The license concept is covered by both CC and DCMI; CC provides a more detailed description of the license – what it permits, prohibits, or requires. For applications which demand a fine grained model of digital rights/digital contracts, these vocabularies might not be appropriate – the evolution of initiatives such as the Open Digital Rights Language (ODRL) into an RDF vocabulary will cover this missing gap [66]. The detailed coverage of the concept by current vocabularies is as follows:

- cc:license (cc:Work → cc:License), cc:License, cc:Permission, cc:permits (cc:License → cc:Permission), cc:Prohibition, cc:prohibits (cc:License → cc:Prohibition), cc:Requirement, cc:requires (cc:License →

[66] The Open Digital Rights Language (ODRL) Initiative is an international effort aimed at developing and promoting an open standard for policy expressions, see http://www.w3.org/community/odrl/.

cc:Requirement), cc:Jurisdiction, cc:jurisdiction (cc:License → cc:Jurisdiction), cc:legalCode (cc:License, cc:deprecatedOn (cc:License), cc:attributionName (cc:Work), cc:attributionURL (cc:Work), cc:morePermissions (cc:Work), . . . [67]

- dc:license (→ dc:LicenseDocument), dc:LicenseDocument, dc:rights (→ dc:RightsStatement), dc:RightsStatement, dc:dateCopyrighted, dc:Jurisdiction

Integrity and Identity Warranties

SWP provides good coverage of this concept. It is not covered by any other vocabulary. FOAF only addresses SHA1 digests of artifacts. The detailed coverage of the concept by current vocabularies is as follows:

- swp:Warrant, swp:Authority, swp:Graph, swp:SignatureMethod, swp:CanonicalizationAlgorithm, swp:SignatureAlgorithm, swp:DigestMethod, swp:DigestAlgorithm, swp:Key, swp:DSAKey, swp:RSAKey, swp:PGPKey, swp:assertedBy, swp:signature, swp:signatureMethod, swp:signatureAlgorithm, swp:canonicalizationAlgorithm, swp:digest, swp:digestMethod, swp:signatureAlgorithm, swp:hasKey , swp:keyInfo swp:certificate, swp:certificationAuthority, swp:X509Certificate, swp:CertificationAuthority

- foaf:sha1 (foaf:Document)

Descriptors

DCMI provides a comprehensive set of general descriptors which can improve the interpretability of a provenance entity influencing another entity being examined by the data consumer. VOID provides dataset descriptors, such as terms for holding dataset statistics (e.g., number of triples in the dataset); VOID also covers vocabulary descriptors (i.e., terms describing vocabularies used by the datasets) and linkage descriptors (i.e., terms describing links between datasets). FOAF vocabulary provides descriptors for persons, groups, and online accounts. OPM provides generic term for expressing descriptions of entities – class Annotation. Descriptors can be also used at the meta level as descriptors of the provenance records.

Since it is not possible to say that a certain set of descriptors is complete and can express all possible needs in all scenarios, no vocabulary is having a full coverage of the descriptors concept. The detailed coverage of the concept by current vocabularies is as follows:

- foaf:homepage (→ foaf:Document), foaf:workplaceHomepage (→ foaf:Document), foaf:page (→ foaf:Document), foaf:schoolHomepage (→ foaf:Document), foaf:mbox (foaf:Agent), foaf:depiction (→ foaf:Image), foaf:topic (foaf:Document), foaf:primaryTopic (foaf:Document), foaf:weblog (foaf:Agent → foaf:Document), foaf:name, foaf:OnlineAccount, foaf:account
(foaf:Agent → foaf:OnlineAccount), foaf:accountName (foaf:OnlineAccount) . . .

[67] http://creativecommons.org/ns

- dc:title, dc:description, dc:subject, dc:abstract, dc:alternative, dc:language, dc:subject, dc:ProvenanceStatement, dc:provenance (range: dc:ProvenanceStatement), ...

- void:triples (void:Dataset → xsd:integer), void:classes (void:Dataset → xsd:integer), void:properties (void:Dataset → xsd:integer), void:documents (void:Dataset → xsd:integer), void:sparqlEndpoint (void:Dataset), void:uriLookupEndpoint (void:Dataset), void:uriSpace (void:Dataset), void:exampleResource (void:Dataset), void:dataDump (void:Dataset), void:uriRegexPattern (void:Dataset), void:vocabulary (void:Dataset), void:target (void:Linkset → void:Dataset), void:subset (void:Dataset → void:Dataset), void:linkPredicate (void:Linkset), void:subjectsTarget (void:Linkset → void:Dataset)

- opmo:label (opmo:Annotable), opmo:Annotable, opmo:Annotation, opmo:annotation (opmo:Annotable → opmo:Annotation), opmo:pname (opmo:Annotable → xsd:anyURI)

Change Tracking

CS vocabulary is suitable for expressing deltas associated with the original artifact, but it does not provide any terms for expressing an explicit version of an artifact or delta. On the other hand, DCMI provides a way how to express versions of artifacts but cannot express particular relations between different versions of the same artifact, e.g., that a certain version is an update of another version. SIOC defines predicates to track changes of artifacts, but only of sioc:Items not being generic enough for all kinds of artifacts. The detailed coverage of the concept by current vocabularies is as follows:

- cs:ChangeSet, cs:addition (cs:ChangeSet → rdf:Statement), cs:changeReason (cs:ChangeSet), cs:createdDate (cs:ChangeSet), cs:precedingChangeSet (cs:ChangeSet → cs:ChangeSet), cs:removal (cs:ChangeSet → rdf:Statement), cs:statement (cs:ChangeSet → rdf:Statement), cs:subjectOfChange (cs:ChangeSet)

- dc:isVersionOf, dc:hasVersion, dc:replaces

- sioc:earlier_version (sioc:Item → sioc:Item), sioc:later_version (sioc:Item → sioc:Item), sioc:next_version (sioc:Item → sioc:Item), sioc:previous_version (sioc:Item → sioc:Item)

Social Descriptors

The concept of social descriptors is not well covered by the current vocabularies: SWP provides terms to express that an artifact (trix:Graph) was quoted (swp:quotedBy) or asserted (swp:assertedBy) by an agent, but it cannot express that an agent supports another agent, artifact, or process. The detailed coverage of the concept by current vocabularies is as follows:

- swp:Authority, swp:quotedBy, swp:assertedBy

Creator/Modifier

The concept of creator is supported by DCMI and FOAF; DCMI also covers the concept of modifier and contributor. The detailed coverage of the concept by current vocabularies is as follows:

- `dc:creator`, `dc:modifier`, `dc:contributor`
- `foaf:made` (`foaf:Agent`)

Publisher

Publisher is covered only by DCMI, the predicate `dc:publisher`.

Owner

SIOC involves the predicate assigning to an artifact (blog post) a `sioc:UserAccount`, which owns that artifact. However, SIOC cannot be used to express the provenance concept owner, because `sioc:UserAccount` is a subclass of `foaf:OnlineAcount`, which is not an agent. DCMI covers the concept of owner. The detailed coverage of the concept by current vocabularies is as follows:

Owner

- `sioc:has_owner` (`sioc:UserAccount` → `sioc:UserAccount`)
- `dc:rightsHolder` (→ `dc:Agent`)

Host and User

Host and user concepts are not covered by the current vocabularies.

5.6.4 Building the W3P Provenance Model

In this section, we describe the building of the W3P provenance model. W3P provenance model should be intuitive, but still covering all the provenance concepts outlined in Section 5.6.3. The model is independent of granularity, allowing users to describe the provenance of different web artifacts including data, documents, and datasets. The coverage of social provenance is an important feature of the ontology, allowing W3P users to track trust and reputation of entities and artifacts.

The W3P provenance model is formed by (1) a set of terms suggested to be used to cover the provenance concepts identified, some of them are taken from existing vocabularies (reused), some of them are created and (2) a set of mappings (alignments) between these terms. The newly created terms and the created mappings form the content of the new W3PO ontology [100]. W3PO should work as an integration ontology, providing the structure to reuse already consolidated vocabularies under the more structured semantics of a provenance model.

Figure 5.3 depicts an excerpt of the W3P provenance model including terms from the W3PO ontology and terms reused from other vocabularies, so that the introduced provenance concepts are covered. Figure 5.3 depicts the classes from W3PO with solid circles, predicates from W3PO with solid lines. Classes (predicates) from other vocabularies are depicted with dashed circles (lines). Certain popular vocabularies (i.e., DCMI, FOAF, OPM) have their own colors to better visualize the terms defined by these vocabularies, other vocabularies are introduced together with their namespace prefixes (Table 5.1). Data properties of classes `opmv:Artifact`, `opmv:Process`, and `opmo:Entity` and hierarchies of certain properties and classes are depicted separately next to the figure. For clarity, Figure 5.3 does not depict relation classes `opmo:WasDerivedFrom`, `opmo:WasGeneratedBy`, `opmo:Used`, `opmo:WasTriggeredBy`, `w3po:WasInfluencedBy`, and `w3po:WasAssociatedWith`; the only relation class depicted in Figure 5.3, is relation class `opmo:WasControlledBy`.

In the process of building W3P, the reuse of existing vocabularies was maximized. Nevertheless, the analysis of the existing vocabularies identified certain gaps in the representation of various provenance concepts and these gaps provide the scope of W3PO – it covers the missing or poorly covered provenance concepts. Reusing concepts which were partially or poorly covered in other vocabularies could lead to a fragmented, inconsistent or difficult to use vocabulary, corrupting the interpretability of the model (Req. 3); thus, we do not reuse terms from other current vocabularies at all costs. In particular, some of the current vocabularies were designed to be used as metadata annotations, lacking a more structured model behind them. This is an important design issue which directly impacts Reqs. 3, 6, and 16. When designing W3PO, OWL transitive primitives were used for certain property characteristics (`owl:TransitiveProperty`); transitivity can strongly impact Reqs. 3, 6 and 16. In order not to clutter the provenance model, we do not associate W3PO properties with their inverse properties [68]. The complete W3PO ontology can be found at `http://purl.org/provenance/w3p/w3po#`. In further text, namespace prefix `w3po:` is used as an abbreviation for W3PO namespace `<http://purl.org/provenance/w3p/w3po#>`.

All concepts in Table 5.2 are important, but the concepts of artifact, agent, and process (called *core concepts*) have a special role, because the other concepts make sense only together with these core concepts; we can express that a certain process was executed at a certain time (temporal descriptor), artifact is located at a certain place (spatial descriptor), has a certain license (license descriptor), etc.

When thinking about how to represent the core concepts in W3P, either (1) we could have created new terms to represent artifact, agent, and process or (2) we could have reused those terms from OPMV. We decided to go for the second alternative to maximize the reuse of vocabularies (Req. 1). As a result, W3P uses terms `opmv:Agent`, `opmv:Artifact`, and `opmv:Process` for that purpose. W3P also uses all the predicates OPMV defines between an artifact, agent, and process.

In the following paragraphs, we will discuss the ways how W3P covers each provenance concept. We also supplement the description with an advice, the users of the provenance model (e.g., developers of provenance-aware applications, publishers of the datasets) should follow to accomplish their needs.

[68] `http://bit.ly/ZwENvV`

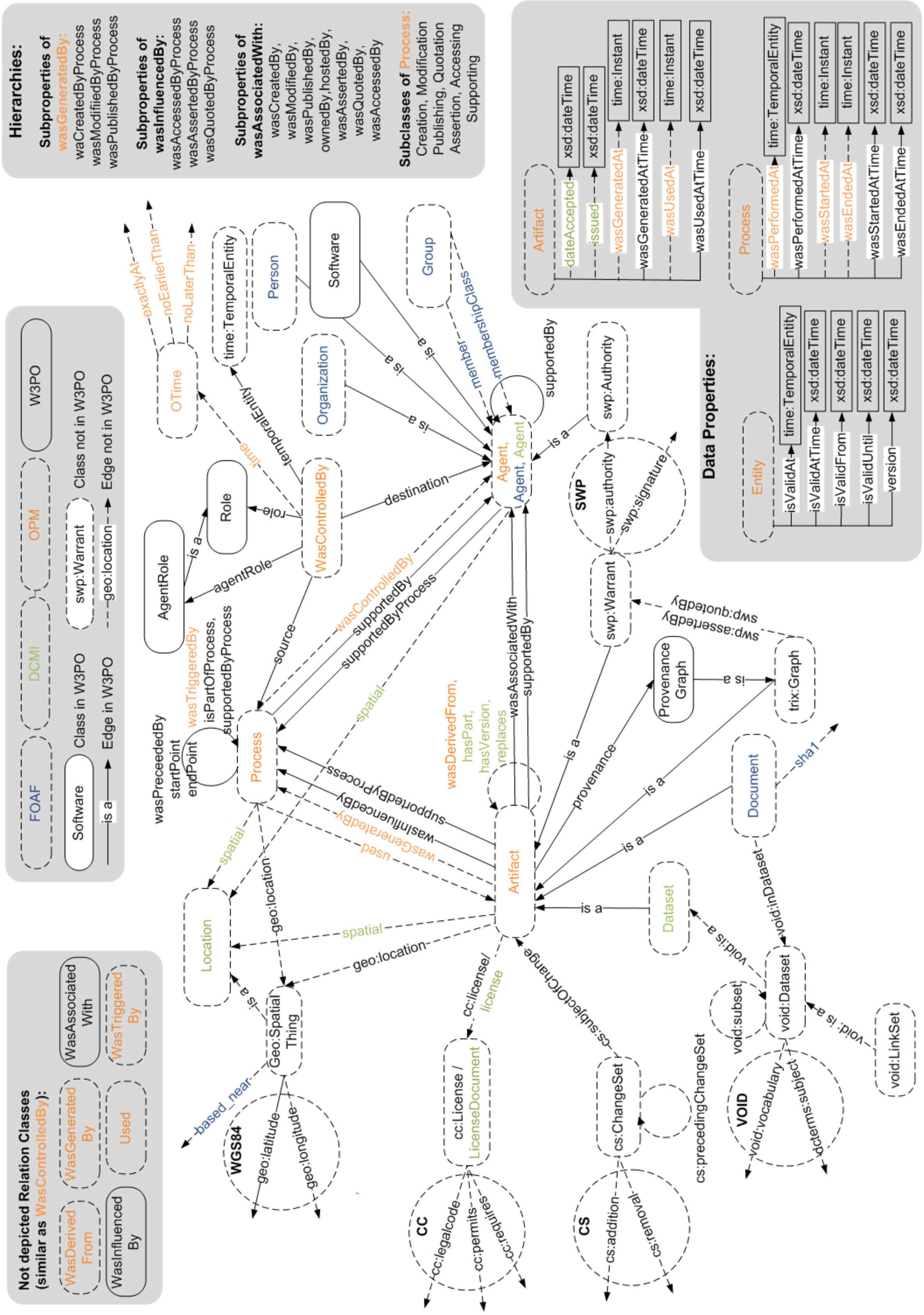


Figure 5.3: W3P provenance model – W3PO and current vocabularies

Artifact

Classes `dcty:Dataset`, `foaf:Document`, `trix:Graph`, and `swp:Warrant` are modeled as subclasses of `opmv:Artifact`, the core concept; `void:Dataset` is already linked to `dcty:Dataset` by the authors of VOID. DCMI provides the predicate (`dc:hasPart`) to express hierarchies of artifacts.

OPMV defines five predicates expressing binary relations between the core classes, which we reuse. Apart from that, we introduce new predicates – `w3po:wasAssociatedWith` and `w3po:wasInfluencedBy`. The former one is useful to relate an artifact (`opmv:Artifact`) with an associated agent (`ompv:Agent`) directly, which is not possible by using the binary relations in OPMV. The latter one is useful to express a more generic relation between an artifact and a process – OPMV provides just the predicate `opmv:wasGeneratedBy`, but, e.g., for expressing that a certain process accessses an artifact, a more generic predicate is needed. For these two predicates (`w3po:wasAssociatedWith` and `w3po:wasInfluencedBy`), we also introduce appropriate relation classes in a similar way OPMO defines relation classes for the five binary relations; i.e., we introduce class `w3po:WasAssociatedWith` for the newly created binary predicate `w3po:wasAssociatedWith` and class `w3po:WasInfluencedBy` for the newly created binary predicate `w3po:wasInfluencedBy`.

Furthermore, W3P introduces alternative predicate `w3po:source` as equivalent predicate for `opmo:effect` and alternative predicate `w3po:destination` as equivalent predicate for `opmo:cause`. The reason for that is that predicates `opmo:cause` and `opmo:effect` are not that intuitive when associated with relation classes, e.g., `opmo:effect` for relation class `opmv:WasControlledBy` is the process being controlled and `opmo:cause` is the agent controlling the process.

Advice 5.1. We encourage to use `trix:Graph` class of artifacts when talking about named graphs, `foaf:Document` class when dealing with documents and `void:Dataset` or `dcty:Dataset` when talking about datasets. Hierarchies of artifacts should be expressed by using DCMI terms. For relating an artifact to another artifact, process, or agent, OPMV binary predicates, OPMO relational classes, or newly introduced W3PO predicates/classes should be used. Together with relation classes, we suggest to use either predicates `opmo:effect` and `opmo:cause` or predicates `w3po:source` and `w3po:destination`.

Agent

Classes `foaf:Agent` and `dc:Agent` are modeled as classes equivalent to `opmv:Agent`. W3P reuses the approach provided by FOAF to express group membership. FOAF provides two ways how to express that `foaf:Agent` belongs to `foaf:Group` – either by using the predicate `foaf:member` to list all the members of the given group or by using predicate `foaf:membershipClass` to associate the given `foaf:Group` instance g with the particular subclass A_g of `foaf:Agents` – instances of A_g are all the `foaf:Agents` being members of the group g. The latter approach enables to define the conditions for being a member of the group; if such condition is satisfied by an agent, such agent is automatically considered as a member (and an instance) of that group. DCMI has also a class `dc:AgentClass` with a similar goal as the class being the value of some `foaf:membershipClass`.

Apart from that, W3PO introduces a class `w3po:Software` to denote non-human agents.

Advice 5.2. We encourage to use OPMV for expressing agents, because OPMV is the core ontology. Nevertheless, FOAF and DCMI can be used as well. To express certain kinds of agents – persons, groups, organizations, FOAF should be used (classes `foaf:Person`, `foaf:Group`, `foaf:Organization`); to express software agents, class `w3po:Software` should be used. Groups of agents should be expressed using FOAF (predicates `foaf:member`, `foaf:membershipClass`).

Process

OPMV provides the core terms `opmv:Process` and binary relations and relation classes to relate process to another process, agent or artifact (see detailed coverage in Section 5.6.3). To support hierarchies of processes, not supported by current vocabularies, W3PO introduces a new predicate `w3po:isPartOfProcess` to express that a certain process is part of another process. W3PO further introduces predicate `w3po:wasPreceededBy` to facilitate the navigation over consequent processes. We also introduce predicates `w3po:startPoint` and `w3po:endPoint` to associate the high-level process with its first and last subprocess.

Advice 5.3. We encourage to use OPM terms for processes and relations between them and to artifacts and agents. W3PO should be used to express hierarchies of processes (`w3po:isPartOfProcess`) and navigation through them (`w3po:wasPreceededBy`, `w3po:startPoint`, `w3po:endPoint`).

Role

OPMO introduces a predicate `opmo:role`, which cannot be used in W3P, because it has too restricted domain – it does not support classes `w3po:WasInfluencedBy` and `w3po:WasAssociatedWith`. As a result, W3PO defines a new predicate `w3po:role` having a generic `opmo:Entity` as a domain and `w3po:Role` as a range. Class `w3po:Role` is a SKOS concept [130], allowing to define hierarchies of roles as desired. We also define a SKOS concept `AgentRole` holding the pre-defined set of agents' roles for the concepts of creator/modifier, publisher, owner, host, user, and social descriptors [100]. Sample role for the creator (`agentRoleTypes:Creator`) is introduced in Listing 10 [69].

Class `opmo:Role` is not recommended to be used, but for its compatibility with `w3po:Role`, we define it also as a SKOS concept and predicate `opmo:value`, containing the label for the `opmo:Role` instance, is set to be equivalent with `skos:prefLabel`.

Advice 5.4. We encourage to use W3PO for roles. We provide certain built-in agents' roles for expressing that the agent is in the role of creator, modifier, publisher, owner, host, user, supporter, or quoter.

Spatial Information

DCMI provides predicate `dc:spatial` to associate spatial information (`dc:Location`) with an artifact, process, or agent. W3P uses these terms to express

[69] Namespace `http://purl.org/provenance/w3p/w3po/agentRoleTypes#` has prefix `agentRoleTypes:`.

general spatial information associated with an artifact, agent, or process. WGS84 provides terms for expressing geographical coordinates of the place. In particular, it introduces predicate `geo:location` expressing geographical coordinates associated with an artifact; such predicate is set as a subproperty of predicate `dc:spatial`. Similarly, class `geo:SpatialThing`, being the core class of WGS84, is a subclass of `dc:Location`. FOAF provides only predicate `foaf:based_near`, with the domain being `geo:SpatialThing`.

Advice 5.5. We suggest to directly use predicate `geo:location` for expressing geographical coordinates or predicate `foaf:based_near` if appropriate for given situation. Otherwise, the generic property `dc:spatial` should be used.

Temporal Information

For expressing temporal information, OPMV uses the vocabulary TIME. TIME vocabulary is in general useful to express time instants and intervals as a part of the W3P predicates. OPMV defines wide range of predicates expressing temporal aspects (see Section 5.6.3). Since OPM is the core ontology of W3P, we use the vocabulary TIME as well, but supplement the model with predicates having `xsd:dateTime` as their ranges, because that simplifies expression of the temporal information. As a result, W3P supplements OPMV predicates `opmv:wasGeneratedAt`, `opmv:wasStartedAt`, `opmv:wasEndedAt`, `opmv:wasUsedAt` and `ompv:wasPerformedAt`, with corresponding alternative predicates `w3po:wasGeneratedAtTime`, `w3po:wasStartedAtTime`, `w3po:wasEndedAtTime`, `w3po:wasUsedAtTime`, and `w3po:wasPerformedAtTime` with ranges being always `xsd:dateTime`.

OPMO provides a way to associate relation classes, such as `opmo:WasControlledBy`, with predicate `opmo:time` having range `opmo:Time`. Then, OPMO defines predicates for the `opmo:Time` class to express that the relation happened exactly at a certain time (the predicate `opmo:exactlyAt`) or not earlier/later than a certain time (predicates `opmo:noEarlierThan`, `opmo:noLaterThan`); all these predicates have already `xsd:dateTime` as their range. We further supplement the terms in OPMO with predicate `w3po:temporalEntity` to associate `time:TemporalEntity` with the relation classes, which is an alternative approach to using predicate `opmo:time`.

Nevertheless, validity of provenance entities cannot be expressed by the terms available in OPM, thus, W3PO introduces new predicates `w3p:isValidAtTime`, `w3p:isValidFrom`, and `w3p:isValidUntil` with ranges being `xsd:dateTime` and `w3p:isValidAt` with range being `time:TemporalEntity`.

Advice 5.6. For expressing the validity range for a provenance entity (artifact, process, agent), W3PO should be used. Otherwise, if using binary relations, OPMV predicates using TIME vocabulary or the alternative W3PO predicates having `xsd:dateTime` as their ranges should be used. If relation classes are used, then OPMO class `opmo:Time` and predicates, such as `opmo:exactlyAt`, should be used; as an alternative to `opmo:Time`, `time:TemporalEntity` may be used, associated with the provenance entity via `w3po:temporalEntity`. DCMI predicates, such as `dc:dateAccepted` or `dc:issued`, may be used, if needed.

License

CC and DCMI both cover the license concept. Thus, W3P reuses classes `dc:LicenseDocument` and `cc:License`, which hold the license document. Both these classes are denoted as equivalent in W3PO. These license documents may be associated with the artifact by using either predicate `cc:license` or `dc:license`, they are denoted as equivalent in W3PO.

Advice 5.7. We suggest to use either CC or DCMI. CC offers richer description of licenses than DCMI, enabling to express what the license permits, prohibits, requires, etc.

Integrity and Identity Warranties

Integrity and identity warranties are covered by SWP vocabulary – `swp:Warrant` is a subclass of `opmv:Artifact` and `swp:Authority` is a subclass of `opmv:Agent`. FOAF defines predicate `foaf:sha`, which can be used for entities of type `foaf:Document` to express SHA1 digest values.

Note that integrity and identity warranties, as introduced in SWP, can be applied only to named graphs (`trix:Graph` instances); it would be a non-trivial work to define them for other artifacts as well.

Advice 5.8. We encourage to use the classes and properties from SWP, but only for named graphs.

Descriptions & Annotations

FOAF, VOID, and DCMI provide descriptors for the proper artifact, i.e., `foaf:Document`, `void:Dataset`, or `dcty:Dataset`.

Set of provenance statements could have been associated with the artifact by using predicate `dc:provenance`; however, the range of predicate `dc:provenance` is `dc:ProvenanceStatement`, which is "a statement of any changes in ownership and custody of a resource since its creation that are significant for its authenticity, integrity, and interpretation."; such definition of provenance statement is too narrow for expressing general provenance information containing the provenance concepts described. Therefore, we introduce and use in W3P a new predicate `w3po:provenance` to associate provenance information with an artifact; the range of such predicate is `w3po:ProvenanceGraph`, being a subclass of `trix:Graph`.

Advice 5.9. Use the descriptors relevant to the type of the described artifact; that determines FOAF, VOID, or DCMI should be used. For associating provenance records with an artifact, use W3PO terms.

Change Tracking

DCMI enables to express that one artifact is replaced by another entity (predicate `dc:replaces`). Furthermore, DCMI allows to express via predicates `dc:hasVersion` or `dc:isVersionOf` that a certain resource is a version of another resource. To specify numerically the version of an artifact in W3P, we introduce a new predicate `w3po:version` not covered by the vocabularies examined.

Alternatively, terms from CS vocabulary can be used to describe the particular changes (deltas) made to the artifact – for that purpose, CS predicate `cs:subjectOfChange` should be used to link `cs:ChangeSet` (the delta) to such artifact.

SIOC enables to express versions of the `sioc:Item`. SIOC properties might be used for expressing versions between `sioc:Items`.

Advice 5.10. For expressing different versions of an artifact, DCMI and W3PO should be used. For tracking changes (deltas) of the original artifact, CS should be used. SIOC properties might be used for expressing versions between `sioc:Items`.

On Modeling Social Descriptors, Creator/Modifier, Publisher, Owner, Host, and User Concepts

Regarding the concepts of social descriptors (supporter, quoter), creator/modifier, publisher, owner, host, and user, there are couple of ways how to model them. Since the W3P provenance model should be generic, we decided to support several ways of expressing the same information, so that each application implementing provenance can choose which one is the most suitable approach for the given case.

The first approach is called *hidden process approach* and is suitable if the particular process behind creation, publication, etc., of an artifact is not interesting in the given situation or does not make sense (e.g., in case of the concepts owner and host).

If we decide for the hidden process approach and if the only important message is that a certain artifact is created, modified, published, asserted, quoted, supported, hosted, or owned by certain agent, it is possible to directly use the corresponding W3PO predicate, e.g., `w3po:wasCreatedBy` (`opmv:Artifact` → `opmv:Agent`). Such hidden process approach is called *simple hidden process approach.* Predicate `w3po:WasAssociatedWith` introduced in Section 5.6.4 is a superproperty for predicates `w3po:wasCreatedBy`, `w3po:wasModifiedBy`, `w3po:wasPublishedBy`, `w3po:ownedBy`, `w3po:hostedBy`, `w3po:wasAssertedBy`, `w3po:wasQuotedBy`, `w3po:wasSupportedBy`, and `w3po:wasAccessedBy`. See Listing 9 for an example of the simple hidden process approach involving predicate `w3po:wasCreatedBy` [70].

```
<http://example.com/dataset/1> a opmv:Artifact ;
  w3po:wasCreatedBy <http://purl.org/knap#me>.
```

Listing 9: Simple hidden process approach

The simple hidden process approach works only for binary relations, so to express n-ary relations using the hidden process approach, one has to use relation classes defined by OPMO or W3PO. We call such approach *qualified hidden process approach.* In Listing 10, terms from OPM and W3PO are used to express that there is an instance of the class `w3po:WasAssociatedWith`, with associated: artifact (expressed by the predicate `w3po:source`, Line 2), agent (expressed by the predicate `w3po:destination`, Line 3), role of that agent (`agentRoleTypes:Creator`, Line 4), and time when the artifact was created (`opmo:exactlyAt`, Lines 5 – 8).

[70] For clarity, prefixes are not included in the listings in this section

```
<http://example.com/relClasses/1> a w3po:WasAssociatedWith ;
  w3po:source <http://example.com/dataset/1>;
  w3po:destination <http://purl.org/knap#me>;
  w3po:agentRole agentRoleTypes:Creator ;
  opmo:time [
      a opmo:OTime ;
      opmo:exactlyAt "2011-11-22T21:32:52"^^xsd:dateTime
  ].
```

Listing 10: Qualified hidden process approach

The second approach supported by W3P is called *revealed process* approach; such approach is suitable if the process (not just the results of the process) is important to be explicitly described in the provenance record, e.g., because certain properties of that process should be emphasized. Listing 11 shows an example of a provenance record expressing that an artifact was created by a process (instance of `w3po:Creation`), which was performed by (`opmo:wasPerformedBy`, Line 4) a certain agent – the creator of the artifact – and it was performed at certain time (`w3po:wasPerformedAtTime`, Line 5). The approach illustrated in Listing 11, when we use two binary relations to express a relation between an artifact and a process and between that process and an agent, is called *simple revealed process approach*; such approach does not involve any relation class.

```
1 <http://example.com/dataset/1> a opmv:Artifact ;
2   w3po:wasCreatedByProcess [
3     a w3po:Creation ;
4     opmo:wasPerformedBy <http://purl.org/knap#me> ;
5     w3po:wasPerformedAtTime "2011-11-22T21:32:52"^^xsd:dateTime
6   ] .
```

Listing 11: Simple revealed process approach

Similarly as in the hidden process approach, if we would like to further qualify either predicate `w3po:wasCreatedByProcess` or predicate `opmo:wasPerformedBy`, we may use instead of each predicate a relation class in a similar way as in the qualified hidden process approach (Listing 10).

Social Descriptors, Creator/Modifier, Publisher, Owner, Host, and User Concepts

Regarding social descriptors, SWP provides terms for quoting entities of type `trix:Graph`; however, such terms cannot be used, because they are only relevant for `trix:Graph` artifact. Thus, we address social descriptors by introducing new terms (similarly as for the creator concept illustrated in Listings 9, 10, and 11) to express quotation of an artifact, assertion (verification) of an artifact, and advocation of an artifact, agent, or process by an agent. Quotation and assertion are indirect means how to support the given provenance entity.

Regarding the concepts of creator, modifier, and publisher, we do not use the DCMI predicates, such as `dc:creator`, `dc:modifier`, and `dc:publisher`, for expressing binary relations in the simple hidden process approach, because they do not define the predicate range properly. FOAF introduces predicate `foaf:maker`, which express the creator of a certain artifact, but W3P rather introduces new W3PO terms (as introduced in Listings 9, 10, and 11) in order to preserve unity of the predicates expressing social descriptors, creator, modifier, publisher, owner, host, and user concepts.

Regarding the concept of owner, SIOC predicate `sioc:has_owner` cannot be used to express the concept owner, because it has a different range and it is used only within the domain of internet discussion forums and blogs. DCMI predicate `rightsHolder` enables to express the concept of owner, but W3P does not use such predicate to preserve unity of the predicates expressing these concepts.

All the newly introduced W3PO terms for concepts creator, modifier, publisher, owner, user, host, and social descriptors can be found at `http://purl.org/provenance/w3p/w3po#`.

Advice 5.11. Use OMP and W3P terms for these concepts. The simple hidden/revealed process approach using binary relations is useful to support Query expressivity and navigability (Req. 16). It is always more difficult to navigate provenance records using qualified approaches. Thus, when deciding whether to use simple revealed process approach or qualified hidden process approach, which both can express the desired information (e.g., that certain artifact was created by certain agent at certain time), the simple revealed process approach (as in Listing 11) should be preferred.

5.6.5 Related Work

There is an extensive list of works in the area of provenance modelling, focused mainly on the domain of scientific workflows. The reader is referred to [136] for a comprehensive survey in that area. This section covers a short discussion on existing works on the definition of provenance models for the Web.

Inference Web project [124] is a Semantic Web based knowledge provenance infrastructure providing interoperable explanations of sources using PML (Proof Markup Language) [44]. The provenance model behind PML, *PML-P*, focuses on tracking provenance in reasoning systems, where the concept of a proof over inference steps determines the attributes of the provenance model. Due to its own purpose, the attributes of PML-P do not provide coverage of the provenance dimensions for a more comprehensive provenance model for the Web; for example, PML-P does not cover the dimension of social provenance.

Groth et al. [80] describe a generic data model for process documentation (the information that describes a process that has occurred), that allows the answer of provenance questions. The model has a precise conceptual definition and it is evaluated with a mash-up use case from the bioinformatics domain. Both Groth's and this work focus on generic (domain independent) provenance models. A major difference is the approach in the definition of the requirements used in the construction of the models: W3P requirements are targeted towards the coverage of provenance representation and use on the Web, while the model described in [80] approaches the problem through a process documentation perspective.

In [83], Hartig proposes a Provenance Model for the Web of Data which generated the *provenance vocabulary.* Hartig proposes two dimensions of provenance for the Web of Data: data access and data creation. However, Hartig's vocabulary lacks enough expressivity for expressing relations between artifacts, processes, and agents. Furthermore, important requirements on the provenance model, such as coverage of social descriptors, licensing, change tracking, and spatiality are not covered by that vocabulary. W3P is also designed to be OPM compatible from

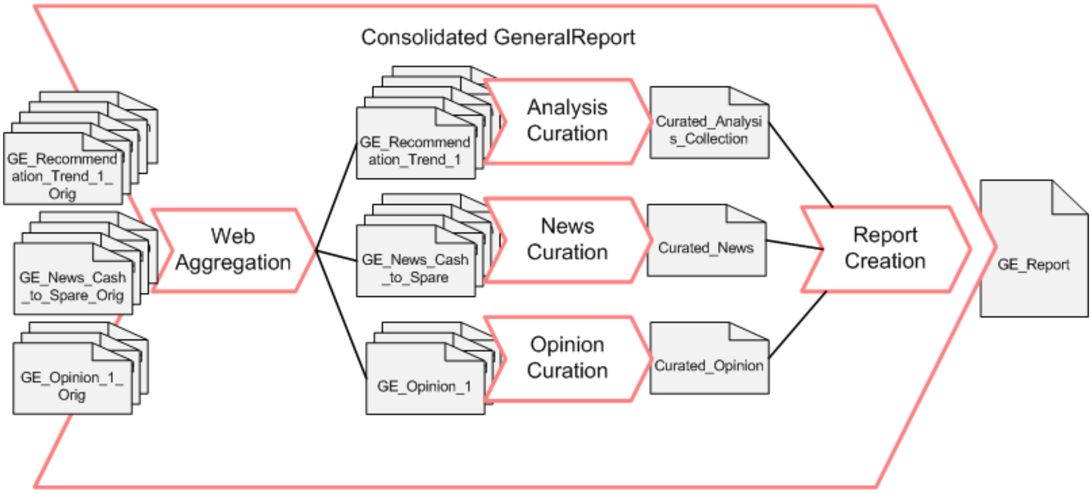


Figure 5.4: Provenance case study workflow

the start, maximizing its interoperability. Another fundamental difference is that W3P uses a requirements based approach for its construction.

Miles et al. [133] describe a detailed mapping between Dublin Core terms and OPM using OPM profiles, deriving relationships between the two vocabularies; such mapping further supplements the mappings defined in W3PO.

Tan [155] distinguishes two granularities of provenance: workflow (or coarse-grained) provenance and data (or fine-grained) provenance. Workflow provenance (mainly addressed by SWfMS systems) represents "the entire history of the derivation of the final output of a workflow" [155]. Data provenance (addressed mainly in databases), in contrast, provides a more detailed view on the derivation of a single value. From the Linked Data perspective, typical methods for describing datasets are coarse grained and descriptive - associated with RDF datasets/documents [157]. Carroll et al. propose named graphs [38] as fine grained methods for tracking data provenance. Bizer et al. [18] use named graphs as a unit of provenance information. Similarly, Ding et al. [52] define the concept of RDF molecules, where each molecule is a group of RDF triples with specific constraints for splitting blank nodes. W3P supports all granularities of provenance.

5.7 Case Study: W3P Aggregation of Financial Data

The W3P provenance model was instantiated using the first use case introduced in Section 5.5.2, where different types of financial data collected from distributed external sources are aggregated, curated and analyzed by a team of analysts in order to generate a daily financial report for a specific company.

The financial report[71] is composed of different types of data (recommendations, fundamental data, news, opinions, analysis) which are consolidated and analyzed in the report creation workflow depicted in Figure 5.4. Each element or collection of elements in the final report (denoted as `GE_Report` in Figure 5.4)

[71] http://purl.org/provenance/scenario/financial.html

has its provenance tracked. External data which was already aggregated by third parties are represented as source artifacts in the report and also have their provenance expressed. In the study, social provenance descriptors play an important role in the process of establishing reputation of external elements among different analysts.

The complete version of the sample provenance descriptor for the daily financial report (GE_Report) can be found at http://purl.org/provenance/scenario/ge_aggregate_provenance_20100524#. Further in this section, we describe only the excerpts of the provenance descriptor. In these excerpts, the namespace prefix rep: represents URI http://purl.org/provenance/scenario/ge_aggregate_provenance_20100524#; declarations of other namespace prefixes are omitted for clarity.

```
1   rep:GE_Report a opmv:Artifact ;
2      rdfs:label "General Electric Report" ;
3      w3po:hostedBy rep:Bluehost ;
4      w3po:ownedBy rep:DERI ;
5      w3po:wasSupportedBy rep:SeanORiain ;
6      opmv:wasGeneratedBy rep:ReportCreation ;
7      opmv:wasGeneratedBy rep:ConsolidatedGeneralReport .
```

Listing 12: Provenance case study – GE_Report artifact

Listing 12 depicts the descriptor of the final artifact (rep:GE_Report). The artifact rep:GE_Report is generated by the rep:ReportCreation process (Line 6), which is the last process of the rep:ConsolidatedGeneralReport process (see Figure 5.4).

```
1   rep:ReportCreation a opmv:Process ;
2      rdfs:label "Report Creation" ;
3      w3po:isPartOfProcess rep:ConsolidatedGeneralReport ;
4      opmv:used rep:Curated_Analysis_Collection ;
5      opmv:used rep:Curated_News ;
6      opmv:used rep:Curated_Opinion ;
7      opmv:wasControlledBy rep:Andre_Freitas ;
8      w3po:wasStartedAtTime "2010-05-24T00:00:00Z"^^xsd:dateTime;
9      w3po:wasEndedAtTime "2010-05-24T00:02:33Z"^^xsd:dateTime ;
10     w3po:wasPrecededBy rep:AnalysisCuration ;
11     w3po:wasPrecededBy rep:NewsCuration ;
12     w3po:wasPrecededBy rep:OpinionCuration .
```

Listing 13: Provenance case study – ReportCreation process

Listing 13 depicts the RDF triples describing the rep:ReportCreation process. It shows how to express that the process consumed resources rep:Curated_Analysis_Collection, rep:Curated_News, and rep:Curated_Opinion (Lines 4 – 6), generated certain resources (see Line 6 of Listing 12), was controlled by agent rep:Andre_Freitas (Line 7), was started and ended at certain times (Lines 8 and 9), was preceded by processes rep:AnalysisCuration, rep:NewsCuration, and rep:OpinionCuration (Lines 10 – 12), and was a part of the high level process rep:ConsolidatedGeneralReport (Line 3).

Listing 14 describes artifacts rep:GE_Opinion_1_Orig (Lines 1 – 8) and artifact rep:GE_Opinion_1 (Lines 10 – 11). These artifacts and process rep:WebAggregation deriving rep:GE_Opinion_1 from rep:GE_Opinion_1_Orig (Line 11) are also depicted in Figure 5.4. Artifact rep:GE_Opinion_1_Orig is generated by some external process performed by agent rep:Jeff_Siegel (Line 5), published by rep:Seeking_Alpha (Line 7), and quoted by rep:YahooFinance (Line 8).

```
rep:GE_Opinion_1_Orig a opmv:Artifact ;
   w3po:wasCreatedByProcess [
       a w3po:Creation ;
       w3po:wasPerformedAtTime "2010-05-23T03:00:00"^^xsd:dateTime ;
       opmv:wasPerformedBy rep:Jeff_Siegel
   ] ;
   w3po:wasPublishedBy rep:Seeking_Alpha ;
   w3po:wasQuotedBy rep:YahooFinance .

rep:GE_Opinion_1 a opmv:Artifact ;
   w3po:wasCreatedByProcess rep:WebAggregation .
```

Listing 14: Provenance case study – `GE_Opinion_1` artifact

5.8 W3P in ODCleanStore

Data feeds inserted to the staging database of ODCleanStore always contain the data graph g and may contain *provenance graph* g^p associated with the data graph g and containing provenance information about that data graph g.

Listing 15 depicts a sample provenance graph holding provenance data about the data graph `<http://source.com/1>`. As we can see, the provenance graph holds (1) the creator of the source, (2) the license of the source, and (3) two sources, expressed in two different ways, from which the data graph was derived from (e.g., extracted).

```
<http://source.com/1> a trix:Graph;
  w3po:wasCreatedBy <http://purl.org/knap#me> ;
  dc:license <http://opendatacommons.org/licenses/pddl/1-0> ;
  opmv:wasDerivedFrom <http://source.com/2> ;
  opmv:wasGeneratedBy [
    a opmv:Process ;
    opmv:used <http://source.com/3> .
  ]
```

Listing 15: Sample provenance graph in ODCleanStore

W3P for Publishers and Consumers

The W3P provenance model is suggested to be used as a way to express the provenance information in the provenance graphs associated with the data graphs submitted to ODCleanStore. Therefore, *data publishers* submitting data to ODCleanStore, should express provenance information using W3P. The query execution module of ODCleanStore, providing the integrated resulting data on the consumer's queries, is then able to supplement the integrated data with provenance information according to W3P provenance model and, thus, provide such provenance information to *data consumers*.

The advantage of using the W3P provenance model to (1) express provenance information of data feeds (the activity of data publishers) or (2) browse and analyze the provenance information (the activity of data consumers) is that W3P satisfies Reqs. 1 – 17 specified in Section 5.5. For data consumers, W3P provenance model also strongly influences the efficiency of the provenance requirements' enforcement as described in Section 5.9.

5.9 Provenance Policies

ODCleanStore framework is able to addressed the objective part of the data quality by enforcing QA policies in quality assessors on the data processing pipelines. Nevertheless, the information quality must be always considered w.r.t. the specific (subjective) requirements of the consumer [141, 112, 18] for his particular task at hand. For example, the consumer may prefer data from the Czech Business Register when looking for data about companies, or may be obliged to use only sources verified by his boss.

These needs are partially supported – the resulting data is accompanied with *data provenance* of the data graphs (sources, source graphs) the data originates from, providing the necessary contextualization for the information consumer to analyze the (subjective) quality of the information [136, 60, 83]. However, there is no automated way how to enforce certain data consumer's requirements, so that the consumed data outputted by ODCleanStore is automatically filtered according to these requirements.

In this section, we describe how consumers can define their own situation-specific policies realizing their requirements; such policies are called *provenance policies* and are capable of filtering certain data sources and preferring others due to certain aspects in the data provenance records associated with these sources. In particular, we describe how these provenance policies can be (1) constructed by data consumers and (2) automatically enforced (applied) as a part of the data consumption process in ODCleanStore. To automatically enforce provenance policies, the *data filtering* component of ODCleanStore is used (Figure 3.1). To that end, data being fetched from the raw data mart of ODCleanStore as a result of the consumer's query are first filtered in the data filtering component and then integrated as described in Chapter 4.

5.9.1 Definition of Provenance Policies

We define a provenance policy $p \in P_{prov}$ as a tuple $(cond, weight)$, where $cond \in C$, C is a set of all valid `GroupGraphPatterns`[72] in the SPARQL query language, and $weight = w(p)$, where $w : P_{prov} \to [-1, 1]$ quantifies the weight of the policy p, s.t. $w(p) \in (0, 1]$ determines a *positive policy* and $w(p) \in [-1, 0)$ determines a *negative policy* p. Set P_{prov} is the infinite set of all provenance policies.

A provenance policy $p = (cond, weight) \in P_{prov}$ can be successfully applied on the provenance named graph g^p if and only if a SPARQL query `ASK FROM NAMED` g^p `WHERE` $\{cond\}$ returns *true*. The successful application is expressed as $a(p, g^p) = true$; otherwise, if the policy was not successfully applied, $a(p, g^p) = false$; $a : P_{prov} \times \mathcal{G} \to \{true, false\}$. If $a(p, g^p) = true$, then p changes the *provenance score* of the graph g according to $w(p)$. Positive policy always increases the provenance score, negative policy decreases.

The condition $cond \in C$ of a policy $p = (cond, weight) \in P_{prov}$ may use the macro `$$graph$$` which is replaced before the query is sent to the underlying SPARQL engine with the name of the data graph to which the policy is trying to be applied (a similar approach was chosen for data normalizer policies). Suppose a condition $c =$ {`$$graph$$ w3po:wasCreatedBy <http://purl.org/knap#me>`},

[72] `http://www.w3.org/TR/rdf-sparql-query/#rGroupGraphPattern`

$c \in C$. Such condition c is matching all the graphs g containing in g^p the triple with the subject being the URI of the graph g, predicate `w3po:wasCreatedBy` and object `<http://purl.org/knap#me>`, i.e., all graphs created by the agent `<http://purl.org/knap#me>`.

5.9.2 Data Filtering Component

The application of provenance policies is a part of the data filtering component as depicted in Figure 3.1. The data filtering component is executed during query execution of ODCleanStore. The data, $Q_x \subseteq \mathcal{Q}$, being fetched from the raw data mart of ODCleanStore as a result of the consumer's URI or keyword query x, are first filtered in the data filtering component and then integrated as described in Chapter 4. The input to the data filtering component (Algorithm 3) is formed by:

- the quads, Q_x, being fetched as a result of the consumers query x
- policies, $P_c \subset P_{prov}$, defined by the consumer c executing the query
- constraints, $F_x \subset F$, customizing the behavior of the algorithm for the given query x
- the desired provenance score threshold $\kappa \in [0, 1]$

The data filtering algorithm can enforce the following constraints F relating to the particular aspects of the provenance policies' application:

- *NoNeg* – Negative policy must not be successfully applied to the provenance graph.
- *ExistsPos* – At least one positive policy must be successfully applied to the provenance graph.
- *PosMajority* – The number of positive policies successfully applied to the provenance graph must prevail over the number of negative policies.
- *PolMandatory* – At least one policy must be successfully applied to the provenance graph.

In Lines 2 – 19 of Algorithm 3, the provenance policies are successively applied to the graphs G_{Q_x} (see Definition 2.8). In Lines 5 – 9, the set $P_a = \{p \in P_c \mid a(p, g^p) = true\}$ of policies successfully applied to the processed graph g is constructed. Based on that, in Lines 10 – 12, the function *eval*, $eval : \mathcal{P}(P_{prov}) \times F \to \{true, false\}$, progressively checks the satisfaction of all constraints F_x w.r.t. to the set of polices P_a. The function $eval(P_a, f)$, executed for policies $P_a \subset P_c \subset P_{prov}$ and the constraint $f \in F$, is defined as follows:

$$eval(P_a, NoNeg) = \begin{cases} true & \nexists p \in P_a : w(p) < 0 \\ false & otherwise \end{cases} \tag{5.1}$$

$$eval(P_a, ExistsPos) = \begin{cases} true & \exists p \in P_a : w(p) > 0 \\ false & otherwise \end{cases} \tag{5.2}$$

Algorithm 3 Provenance Policies Application

```
Input: $Q_x$, $P_c \subseteq P_{prov}$, $F_x \subseteq F$, $\kappa$
Output: $\widetilde{Q_x} = applyProvPolicies(Q_x,\ P_c,\ F_x,\ \kappa)$
 1: $\widetilde{Q_x} \leftarrow \emptyset$
 2: for each graphs $g \in G_{Q_x}$ do
 3:     $P_a \leftarrow \emptyset$
 4:     $flagResult \leftarrow true$
 5:     for each policies $p \in P_c$ do
 6:         if $a(p, g^p)$ then
 7:             $P_a \leftarrow P_a \cup \{p\}$
 8:         end if
 9:     end for
10:     for each flags $f \in F_x$ do
11:         $flagResult \leftarrow flagResult \wedge eval(P_a, f)$
12:     end for
13:     if $flagResult$ then
14:         $s_{prov}(g) \leftarrow min\{\frac{\prod_{p \in P_a}(1+w(p))}{C}, 1\}$
15:         if $s_{prov}(g) \geq \kappa_{pp}$ then
16:             $\widetilde{Q_x} \leftarrow \widetilde{Q_x} \cup (*, *, *, g)$
17:         end if
18:     end if
19: end for
20: return $\widetilde{Q_x}$
```

$$eval(P_a, PosMajority) = \begin{cases} true & |\{p \in P_a \mid w(p) > 0\}| > |\{q \in P_a \mid w(q) < 0\}| \\ false & otherwise \end{cases} \tag{5.3}$$

$$eval(P_a, PolMandatory) = \begin{cases} true & |P_a| > 0 \\ false & otherwise \end{cases} \tag{5.4}$$

If all flags F_x are satisfied for the given set of policies P_a, the algorithm computes in Line 14 the *provenance score* $s_{prov}(g)$ of the graph g based on the weights of the policies P_a successfully applied to the graph g. The constant $C \in \mathbb{N}$ defines the upper boundary for the influence of the positive policies; if $\prod_{p \in P_a}(1 + w(p)) > C$, the provenance score $s_{prov}(g)$ is equivalent to the case when $\prod_{p \in P_a}(1 + w(p)) = C$. The constant C should be set based on the average proportion of positive and negative policies and the average absolute number of positive policies applied to the graphs. Furthermore, the default provenance score of any graph to which no policy was successfully applied should be equal to $1/C$.

Consequently, in Line 15, the algorithm tests whether the provenance score is higher than the required threshold $\kappa \in [0, 1]$; if yes, the quads of the data graph g are added to $\widetilde{Q_x}$ (Line 16). Otherwise, the quads associated with the processed graph g are not included in $\widetilde{Q_x}$.

An output of the data filtering component is the refined collection of quads, $\widetilde{Q_x} \subseteq Q_x$. Such output is used as the input to the data integration component of ODCleanStore.

Function $s : G \to [0, 1]$, defined in Formula 4.1 and being used in the data integration component to compute quality score of the resulting integrated quads, has to be adjusted to take into account provenance score s_{prov} computed by the preceding data filtering component. Instead of using Formula 4.1 for computing $s(g)$, Formula 5.5, computing $s(g)$ as a convex combination of s_{ng}, s_{pu}, and s_{prov}, should be used in the data integration component.

$$s(g) = \gamma_1 \cdot s_{ng}(g) + \gamma_2 \cdot s_{pu}(pub(g)) + \gamma_3 \cdot s_{prov}(g) \tag{5.5}$$

The time complexity of Algorithm 3 is $O(|Q_x|+|G_{Q_x}|\cdot|P_c|\cdot O(a(p, g^p)))$, where $O(|Q_x|)$ yields from loading the quads to the memory and $O(a(p, g^p))$ is the time complexity of applying a single policy $p \in P_c$ to the provenance graph g^p.

Regarding the customization of the provenance data filtering component, so that it can enforce data consumer's provenance requirements, the query format supported by ODCleanStore (introduced in Section 3.2.3) should be extended, so that each query submitted to ODCleanStore can be supplemented with (1) the list of provenance policies the data consumer would like to apply to the data, (2) the constraints F customizing the behavior of the data filtering component, and (3) the desired provenance score threshold $\kappa \in [0, 1]$ (see Section 5.9).

5.9.3 W3P and Efficiency of Provenance Policies' Enforcement

The usability of the provenance policies and efficiency in which the provenance policies may be applied depends on the provenance model used. For example, predicates `dc:license` and `cc:license` both associate an artifact with its license; therefore, there should be a notion of similarity between them in the knowledge base in Figure 1.4. Such notion of similarity is defined in the W3PO ontology being a part of the W3P provenance model. Suppose a policy $p = (cond, weight)$ containing predicate `cc:license` in a certain triple within the condition *cond*. Further, suppose the provenance graph from Listing 15 containing a triple `(<http://source.com/1>`, `dc:license`, `<http://opendatacommons.org/licenses/pddl/1-0>)`. Based on the notion of similarity between the predicates `dc:license` and `cc:license` in the knowledge base, policy p should be applicable to the provenance graph in Listing 15.

Therefore, by using the W3P provenance model, which contains important mappings between reused vocabularies, for expressing provenance policies, the usability and efficiency of provenance policies' application is increased.

5.9.4 Related Work

Researchers have developed and investigated various policy languages to describe trust, quality, and security requirements on the Web [94, 159, 25, 18]; a variety of access control mechanisms generally based on policies and rules have been developed [115, 138, 37].

Bizer [20] defines a context-based trust mechanism as a mechanism which relies on the provenance information associated with a data source (e.g., when the data was created, by who, which process) and is driven by *context policies* customizable by each data consumer. Context policies restrict the resulting data provided to the data consumer to the data with the provenance data satisfying the given context policies. These ideas are implemented in the Linked Data framework WIQA (Web Information Quality Assessment Framework) [18], where users can specify policies in the form of RDF graph patterns using the WIQA-PL policy language. The users can then filter the information in their local storage according to the selected policy, and get justifications "why" the given data satisfies a set of policies. When comparing our provenance policies with policies defined by WIQA-PL, a WIQA-PL policy enables to define which information is filtered positive; a data filtering module based on our policies supports both positive and negative filtering. WIQA supports the provision of justifications by extending the SPARQL language with the construct $EXPL$; in ODCleanStore, justifications are represented by the list of policies being applied to the resulting data.

5.10 Summary

In this chapter, we introduced the definition of provenance, description of the current provenance research focus, and how the data provenance needs for expressing provenance of the data on the Web differ from other domains' needs. Based on that, the necessity for a new provenance model for the Web emerged.

The core part of this chapter, Section 5.5, elaborated the requirements for the provenance model for the Web. Consequently, Section 5.6 built the provenance model for the Web (called W3P), which should be used for expressing and tracking provenance behind the data on the Web. As part of that, we defined a W3PO ontology for holding new terms of the W3P provenance model and also mappings to/between reused vocabularies. In Section 5.7, we applied the W3P provenance model to one of our use cases introduced in Section 5.5.

In Section 5.8, we discussed how ODCleanStore would benefit from the W3P provenance model. Afterwards, in Section 5.9, we discussed the concept of provenance policies realizing the subjective consumers' provenance requirements on the consumed data; the provenance polices are intended to be enforced in the data filtering component of ODCleanStore.

Relevant Author's Publications

The first part of this chapter is covered by the journal paper [60] describing the process of the W3P provenance model creation. Paper [103] describes the concept of provenance policies and how they can be enforced in the data filtering component of ODCleanStore.

Main Contributions

The W3P provenance model for the Web, which is constructed w.r.t. the Requirements 1 – 17 elaborated in Section 5.5.3, is the main contribution of this chapter. The proposed W3P provenance model is built over core Linked Data standards. It is independent of granularity, allowing users to describe the provenance of different web artifacts including data, documents, and datasets. It reuses other vocabularies. The coverage of social provenance is an important feature of the W3P provenance model, allowing W3P users to track trust and reputation of entities and artifacts.

Further contribution is the concept of provenance policies expressing the data consumers' provenance requirements and the intended enforcement of provenance policies in the data filtering component of ODCleanStore.

6. Trust Model for SoSIReČR

In this chapter, we start by introducing the motivational scenario – the SoSIReČR project. The goal of the SoSIReČR project[73] is to leverage the communication and cooperation of the Czech informatics community by creating a social network of its members. Social networks [143, 134] are recognized as a valuable source of information [85]; however, they can be full of malicious entities as well [76]. Therefore, the question of agents' trustworthiness in such social network is of crucial importance. This chapter proposes the trust model for SoSIReČR, which is able to compute trust between members of the Czech Informatics community w.r.t. the particular problematic scenarios introduced in Section 6.1.

In Section 6.2, we describe the concept of trust, important properties of trust, such as its domain specificity and task criticality, and we describe trusting beliefs – factors influencing trust. In Section 6.3, we detail the concept of trusting beliefs as an important building block for trust quantification – we survey the trusting beliefs identified in the literature, select the relevant trusting beliefs for the problematic scenarios in SoSIReČR, evaluate the selection process, and sketch the sources and quantification of these beliefs. In Section 6.4, we survey the current relevant trust metrics for estimating trust in social networks and we discuss their properties and suitability for computing trust in SoSIReČR.

Finally, Section 6.5 defines a trust model for SoSIReČR using the definition of trust introduced in Section 6.2, sources and quantification of beliefs described in Section 6.3, and a trust metric which respects the domain specificity of trust and the fact that quantification of trust should be based on the quantification of relevant beliefs forming trust.

6.1 Motivational Scenario – SoSIReČR Project

In the Czech Republic, the informatics community consists of various entities – persons (students, IT professionals, academics, employers), institutions (companies, universities), and other entities typically enabled/initiated by the institutions and formed by the persons (research groups, projects). Unfortunately, the *communication* and *cooperation* among these entities is not sufficient, which is illustrated in the following problematic scenarios S_1 – S_5:

S_1 Students/IT professionals cannot compare their abilities with (i) other students or IT professionals, (ii) the typical abilities of employees working at certain positions, or (iii) typical level of knowledge of other universities' graduates. As a result, they cannot justify their price in the employment market properly.

S_2 Students/Academics do not know who is working on similar research topics at other universities and, as a result, they cannot unify their efforts to make the research more effective and publish at more prestige conferences.

[73] Social Network of the Computer Scientists in the Regions of the Czech Republic, `http://www.sosirecr.cz/index_en.php`

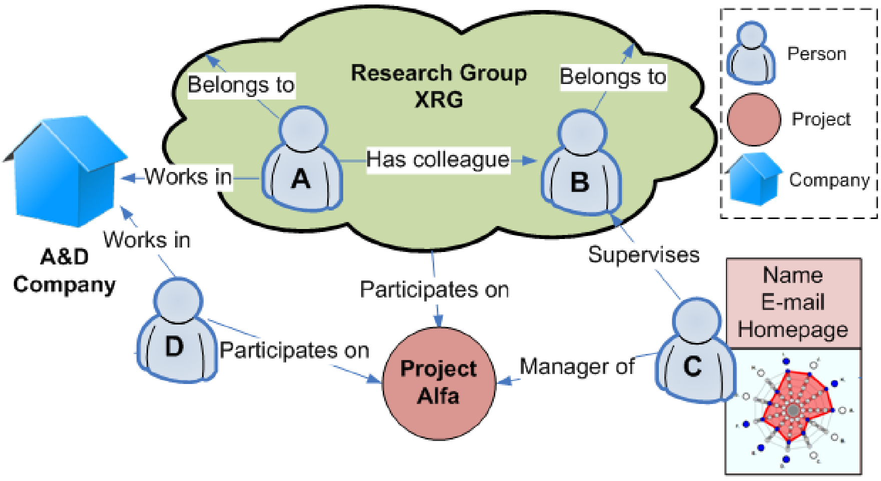


Figure 6.1: Social network behind the SoSIReČR portal

S_3 Companies/universities searching students/IT professionals for their projects cannot quickly and easily find suitable candidates who would like join the project and have the desired expertise.

S_4 Students/IT professionals do not know which companies are looking for new employees and in which domains of expertise.

S_5 Companies do now know the typical aggregated knowledge of students/IT professionals in various regions of the Czech Republic – this information would help them when setting up new branches.

The goal of the SoSIReČR project[74] is to leverage the communication and cooperation of the informatics community by creating a social network of its members (see Figure 6.1) with vertices representing the particular members of the community and edges representing relations between them, e.g., "a student/academic belongs to a research group", "a student/IT professional/academic works on a project/for a company", "a student graduated at the given faculty".

Apart from a general purpose social networking application, such as Facebook[75], the SoSIReČR project focuses on the needs of the Czech informatics community. *ResearchGate*[76], *Epernicus*[77], and *iamResearcher*[78] are examples of foreign projects with goals similar to the SoSIReČR portal's goal – to ensure information sharing and collaboration of members of the informatics community. Nevertheless, they focus merely on the academic domain.

Every member of the informatics community is associated with its *personal* and *professional profile*. A personal profile holds basic information about the entity (e.g., an IT professional is provided with his name, email address, working place, etc.) together with information melted from relations with other entities (e.g., on which projects the IT professional participates). A professional profile

[74] Social Network of the Computer Scientists in the Regions of the Czech Republic, http://www.sosirecr.cz/index_en.php

[75] http://facebook.com

[76] http://www.researchgate.net

[77] http://www.epernicus.com

[78] http://www.iamresearcher.com

of an entity holds information about to which extent the entity (student, IT professional, academic) is an expert in various domains of expertise or to which extent the entity works in and knows the given domain of expertise (university, company). The project uses the ACM Computing Classification System[79] to track expertise in various domains, called *axes* of the professional profile. The expertise is either assigned explicitly by the entities themselves or derived implicitly by taking into account entity's participation in the projects and research groups, entity's contracts in companies, research activities (including published papers), awards obtained, etc.

The social network in the SoSIReČR project is accessible via a Web portal, where users can edit details of their profiles. Apart from a general purpose social networking application, such as Facebook, the project provides a focused (information sharing & collaboration) social networking application intended for the particular target community (Czech informatics community), which does not currently exist; several motivations for its existence are summarized in the problematic scenarios S_1 – S_5. Furthermore, the SoSIReČR project will provide the members of the informatics community with high level of semantization of the stored information, which enables (1) to reuse data already available on the Web (e.g., instances of the Friend of a Friend ontology containing information about friends of a student or DBLP Computer Science Bibliography[80] containing publications of an academic) and (2) to interconnect the social network in the SoSIReČR project with other open social networking application in the future.

To address the problematic scenarios S_1 – S_5, the SoSIReČR portal should support the following types of queries (or simply queries) Q_1 – Q_5 to find for the user the needed information or to help him to start collaboration; each query is supplemented with the particular problematic scenario motivating it:

Q_1 A user would like to obtain the professional profile of another user in the social network to be able to compare his professional profile with other professional profiles (motivated by S_1).

Q_2 A person is searching persons/groups/projects for a future academic collaboration (motivated by S_2).

Q_3 A project (i.e., the project manager of the project) is looking for a student to complete the project team (motivated by S_3).

Q_4 A person is looking for a job/collaboration on the project (motivated by S_4).

Q_5 A company would like to get an aggregated view on the professional profiles of students in the chosen region of the Czech Republic (motivated by S_5).

Query Q_4 is sufficiently solved (at least in the Czech Republic) by various job portals[81]. Nevertheless, our portal will provide this functionality with additional features, such as semantic-rich information or detailed job applicant's expertise using professional profiles, which are not available at most job portals. Other scenarios are not addressed satisfactorily by any other application.

[79] http://www.acm.org/about/class/1998

[80] http://www.informatik.uni-trier.de/~ley/db/

[81] Such as http://www.jobs.cz/en/,http://www.prace.cz/ (in Czech), or http://www.hledampraci.cz/ (in Czech).

6.1.1 Trust in SoSIReČR

Social networks [143, 134], as the one behind the SoSIReČR portal formed by the members of the informatics community, are recognized as a valuable source of information [85]; however, they can be full of malicious entities as well [76]. Therefore, the crucial question is to which extent we can **trust** the entities (members of the informatics community) behind the results on the queries $Q_1 - Q_5$; intuitively, we need to trust that the other entities are honestly providing the professional profiles (Q_1) or have the appropriate competence to join the project (Q_2).

Definition 6.1. A social network behind the SoSIReČR portal is modeled as a directed labeled multigraph $SN = (V, E, a, b, l_E)$, where the vertices, $V \in \mathcal{V}$, represent entities (agents) of the network (members of the informatics community) and edges E are the *relations* between these agents ($\mathcal{V}$ is an infinite set of agents); function $a : E \to V$ assigns to each edge its source vertex, function $b : E \to V$ assigns to each edge its target vertex. function $l_E : E \to R$ is the labeling function; such function associates every edge $e \in E$ with the type of relation $r \in R$.

The set of types R, introduced in Definition 6.1, includes, e.g., relations "has colleague", "has positive experience", "belongs to a group", "participates on the project", "is manager of". Figure 6.1 depicts such relations.

Trust is a crucial concept in human's everyday life and governs the substantial amount of our decisions. When deciding whether to trust or distrust another person in the particular situation, we are influenced (1) by many objective and subjective beliefs, such as the person's honesty, competence, our experience with that person, (2) by our previous experience with that person and his social network (friends, family), (2) by previous experience of our friends with that person, (3) by rumors about that person, (4) by aspects which are not directly connected with the subject of the decision, such as clothing, decency, or loveliness of that person, (5) by the amount of necessity of the trust decision's subject; or (6) by our instantaneous psychological state of mind, unrelated to the trust decision made.

The aspects above (just to mention some of them) influence the human's trust decision and illustrate the complexity of trust as a computational concept and the consequent difficulties to model trust in applications. The application's trust decision process is always a simplification of the human's trust decision process – different simplifications are suitable for P2P networks [95], to provide trustworthy product reviews in an eShop [92], or to ensure trustworthiness of data on the Semantic Web [75].

Let us suppose an instance of Query Q_2: "A young researcher (*seeker, trustor*) is searching another researcher (*target entity*) for future collaboration – writing a paper to a prestigious conference". From the seeker's point of view, the crucial question is how much he can trust that the target entity is the right one for the collaboration. The seeker typically does not know the target entity, trustee, hence, the seeker cannot himself estimate trust in that target entity (there is no *trust relation* between him and the target). Since it was proofed experimentally that

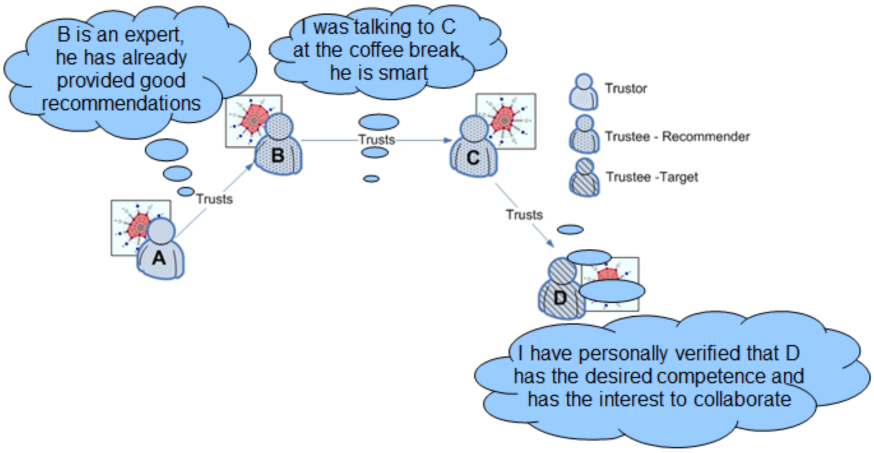

B is an expert, he has already provided good recommendations
I was talking to C at the coffee break, he is smart
Trustor
Trustee - Recommender
Trustee -Target
Trusts
Trusts
Trusts
A
B
C
D
I have personally verified that D has the desired competence and has the interest to collaborate

Figure 6.2: Unclear semantics of the transitive trust

trust in social networks is *transitive*[82], the seeker can (and actually has to) rely on another entity (*a recommender*) having a trust relation to the target [81, 76].

There are lots of trust models, e.g. [76, 171, 148, 123], comprehending trust between two agents as a "black box" and indivisible concept. Since trust is so complex concept [92], semantics of such "black box" trust is ambiguous – a seeker understands the semantics of his *trust relations* in other agents, however, is rather confused regarding trust relations of others. For example, while one seeker from Query Q_2 can trust the target that they can write together a successful paper for the prestigious conference just because he was talking with the target at a coffee break of a conference, another seeker can trust the same target only after personally verifying that the target has the desired competence and has the interest to collaborate (see Figure 6.2). As a result, any trust metric quantifying black box trust between two entities not having a trust relation between them (e.g., B and D in Figure 6.2) has to rely on at least two trust relations ($B \rightarrow C$, $C \rightarrow D$ in Figure 6.2) and, thus, cannot assign clear semantics to the quantified (transitive) trust ($B \rightarrow D$) [85]. Furthermore, to make the things worse, trust models typically lack any domain specificity – the trust models do not distinguish that the seeker may trust the target entity regarding the successfully written paper on *indexing in databases*, but not on *interfaces for component systems*.

To address the issues of *black box* trust models and to allow more fine grained quantification of trust, we comprehend trust (properly defined in Section 6.2) as a concept formed by the set of underlying *trusting beliefs* [57, 126]. Trust is never quantified directly in our approach – neither explicitly by the entities, nor implicitly by the SoSIReČR portal – trust is derived based on the quantifications of the beliefs forming trust. By deriving trust from its beliefs – the simpler and more intuitive concepts, the confusion of social network's members *what trust actually is* is decreased. To address the issues of domain specificity of trust, trusting beliefs should be bound to certain domains/topics for which they are relevant.

Selecting the proper set of beliefs, which (1) would be justified by the literature and (2) suitable for Scenarios S_1 – S_5, is the main goal and contribution of

[82] If an entity A trusts an entity B and the entity B trusts an entity C, then, to some extent, the entity A trusts the entity C.

Section 6.3. Furthermore, we have to also suggest the use of the proper trust metric to derive trust values between two entities not having a trust relation between them. Such trust metric could be based on the current trust metrics, but it has to work with values obtained by quantifying the beliefs and it should respect the domain specificity of trust.

6.2 Concept of Trust

"Manifestations of trust are easy to recognize because we experience and rely on it every day, but, at the same time, trust is quite challenging to define because it manifests itself in many different forms [...], the term is used with wide variety of meanings" [92]. This observation is confirmed in many papers, such as [6, 72, 126].

Definition 6.2, based on the definition proposed by McKnight and Chervany in [126], drives the comprehension of trust in the further text. We selected Definition 6.2 from lots of other definitions [6, 69, 121, 76, 77], because it comprehends trust as the subjective opinion of an entity about another entity and it embodies five essential elements of trust synthesized from the trust literature [126]:

(a) potential negative consequences

(b) dependence

(c) feeling of security

(d) situation specific context

(e) lack of reliance and control

The necessity of these elements is justified in [126], if one of these elements is missing, the term "trusting" conflates with other terms, such as betting (if the feeling of security is not present) or having power over the other (if the trustor has reliance and control over the trustee).

Definition 6.2. *Trust (trusting intention)* is the extent to which one entity *(trustor)* is willing to depend on the other entity *(trustee)* in a given situation with a feeling of relative security, even though negative consequences are possible.

Definition 6.2 is a generic definition, which may be instantiated for Scenarios S_1 – S_5, e.g., for S_3, it would be: "Trust is the extent to which a company is willing to depend on the applicant's work in a given situation (relational database management) with a feeling of relative security (the applicant will do the job properly), even though negative consequences (employee will not work as expected, money will be wasted) are possible".

6.2.1 Domain Specificity of Trust

Lots of papers, such as [74, 148, 123], ignore the extent to which trust is in most cases situation-aware – it is *domain* and *task specific.*

To illustrate the domain specificity, someone who may be trusted for financial advices may not be trusted for film recommendations or to drive safely. There are (semantic web) applications, such as FilmTrust[83], where the notion of trust is restricted to a particular domain by the intended use of the application (movies recommendation in FilmTrust); in these cases, the lack of domain specificity may be justified [76]. However, in case of the majority of applications, including the SoSIReČR portal, it is necessary to acknowledge and comprehend trust as a domain specific concept; in SoSIReČR, the domains are determined by the axes of the professional profile.

Definition 6.3. Let us define a *domain hierarchy* as a directed acyclic continuous graph $F = (D, R_D)$, i.e., a directed tree, where the set D represents particular domains and R_D represents relations of "being a subdomain" between domains D.

We suppose there is always one domain $d \in D$ called *root domain* covering all domains; it is the only domain not being a subdomain of any other domain. If this precondition is not met, we may model the domain hierarchies as directed, acyclic, but not necessarily continuous graphs, i.e., as directed forests[84].

In the SoSIReČR portal, the domain hierarchy is the ACM classification hierarchy[85], which is a favorite hierarchy to classify research papers. ACM classification is supplemented with the artificial root domain to satisfy Definition 6.3.

6.2.2 Trust Value

The *extent* in Definition 6.2 is called *trust value*. It can be quantified either on a discrete [117, 76] or continuous scale [121, 81, 171, 160, 76, 72]. In general, discrete trust levels are easily seizable by humans; on the other hand, continuous *trust values* provide more accurate expressions of trust.

In [76], Golbeck internally uses continuous trust values $tv \in [1, 10]$ to express trust between agents in the social network; however, externally, the information consumer is provided with ten discrete *trust levels* ranging from "absolute trust" ($tv = 10$) to "absolute distrust" ($tv = 1$), with $tv = 5$ expressing the "neutral trust" (neither positive, nor negative) or the absence of trust. We agree that discrete trust levels are easily seizable by information consumers expressing the trust in other entities in the social network. Regrettably, it is hard to imagine that entities will consistently subjectively map their trust to others on a ten point scale (as proposes Golbeck in [76]) – there is no guarantee for the information consumer that someone "really trusted" is always expressed as 9 or 8.

Definition 6.4. Let us define $T = \{\tau_{a_1}, \ldots, \tau_{a_{|V|}}\}$, a set of partial trusting functions $\tau_{a_i} : V \times D \to [-1, 1] \cup \perp$, where $1 \leq i \leq |V|$ and $SN = (V, E, a, b, l_E)$ is the social network. Let us suppose the domain hierarchy $F = (D, R_D)$. Every partial function $\tau_{a_i}(a_j, d)$ is assessing trust value of the trustor a_i in a trustee a_j w.r.t. the domain $d \in D$. Let us define partial function $\tau : V \times V \times D \to [-1, 1] \cup \perp$ which is a union of all partial functions τ_{a_i} for all $a_i \in V$. We call such function as *trust metric*.

[83] http://trust.mindswap.org/FilmTrust/

[84] http://mathworld.wolfram.com/Forest.html

[85] http://www.acm.org/about/class/1998

Table 6.1: Task criticality levels

Task criticality	**Sample usage**	**Query**	**Financial loss**
Very High (4)	A company is looking for a Java expert, who will undergo a 2 months training costing 80 000 $	Q3 – Q5	Significant
High (3)	A company is hiring a programmer for part-time routine job	Q2 – Q5	Medium
Normal (2)	A researcher is looking for someone to cooperate with on the writing of the next paper	Q1 – Q5	Small/No
Low (1)	A researcher is comparing his profile with a profile of another researcher at a different university	Q1 – Q2	No

Since trust is never going to be assessed manually by the agents in the SoSIReČR project, but is quantified based on the quantification of its beliefs (see Section 6.3), we use continuous trust value $\tau(a_i, a_j, d)$ ranging from absolute distrust ($\tau(a_i, a_j, d) = -1$) to maximum trust ($\tau(a_i, a_j, d) = 1$); trust value may be also undefined ($\tau(a_i, a_j, d) = \perp$); $a_i, a_j \in V$, $d \in D$. Furthermore, using negative values for distrust and positive for trust is a more natural way how to represent trust and distrust values; it was already used by [121], one of the first works formalizing trust. We assume that $\tau(a_i, a_i, d) = 1$, $\forall d \in D$, i.e., an agent a_i trusts himself.

Paper [121] argues that if $i \neq j$, $1 \leq i, j \leq V$, $a_i, a_j \in V$, $d \in D$, then $\tau(a_i, a_j, d) < 1$, because no agent can be 100% sure that another agent will behave like expected. On the other hand, [121] argues that $\tau(a_i, a_j, d) = -1$ can be ascribed through a thoughtful judgement typically based on the negative experience with the agent a_j in the past [121]. We agree with such arguments and also incorporate this assumption to Definition 6.2; however, rather than restricting the interval for the trust value to $[-1, 1)$, we assume that even if $\tau(a_i, a_j, d) = 1$, still, some negative consequences are possible (although not probable).

6.2.3 Task Criticality

Apart from the domain specificity of trust, also the *task criticality* influences the trust value. It really differs whether an agent, a student, decides to trust another agent's advices when preparing school homework for the university course "Financial accounting" or whether an agent, a director of a company, decides to trust somebody to be hired as a financial expert. Obviously, in the latter case, the trust value between the director and the financial expert must be higher before the trusting intention from Definition 6.2 can yield in hiring that financial expert.

To reflect the task criticality, a different *threshold*, $\kappa \in [0, 1]$, should be used for the trust values computed by trust metric τ in different situations. For example, Query Q_3 (see Section 6.1) typically requires higher threshold κ than Q_1. If the trust value is above the given threshold κ, the trustor is willing to depend on the trustee according to Definition 6.2.

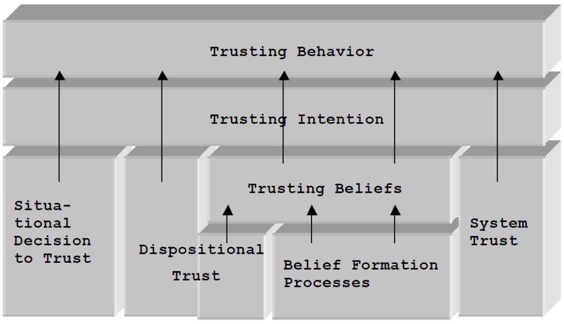


Figure 6.3: Trusting intentions and its sources (Source: [126])

Table 6.1 lists the suggested task criticality levels (influencing the value κ) together with their sample use – the types of queries typically associated with these task criticality levels. Each level is associated with the expected financial loss (if any) in case the trusting intention is misplaced; expected financial loss should be use as the primary indicator when determining to which criticality level the task belongs [26].

6.2.4 Trust Ingredients

We agree that trusting intention in Definition 6.2 is made up of underlying interpersonal and situation-specific *trusting beliefs* [57, 126]; for example, a trustor may believe that the trustee is competent and honest (in the given domain). Relevant beliefs are surveyed in Section 6.3.

Apart from trusting beliefs, paper [126] identified other *trust ingredients* supporting the formation of the trusting intention – *system trust*, *dispositional trust*, and *situational decision trust*. System trust enables a trustor to feel more secure in taking risks with others because of (1) structural assurance safeguards (e.g., a person feels rather safe to depend on the other, because there is a contract between them enforceable by law) or (2) situational normality's reduction of uncertainty (normally, the surgery operation of this type is successful). Dispositional trust is the extent to which a person has a consistent tendency to trust others, without the respect of the particular situation; therefore, it is a stance that others are generally trustworthy people or it is a stance that "irrespective of whether people are good or bad [...], one will obtain better outcomes by trusting them" [126]. Finally, situational decision to trust is the extent to which the trustor has formed an intention to trust every time the given situation arises, irrespective of one's beliefs about the attributes of the trustee [149]. Figure 6.3 depicts the trust ingredients mentioned.

Although all the trust ingredients mentioned influence the trusting intention in Definition 6.2, to lower the amount of variables in the trust model, we focus in our work on trusting beliefs and suppose the existence of the same system trust for all situations, fixed and optimistic dispositional trust for all entities, and no situational decision to trust – trusting intention is driven just by the underlying interpersonal and situation-specific *trusting beliefs*.

Determining the particular set of trusting beliefs supporting the trusting intention in Scenarios $S_1 - S_5$ is the main goal of Section 6.3.

6.2.5 Trust Intention versus Trust Behavior

Figure 6.3 also relates the concept of trusting intention described in Definition 6.2 with the *trusting behavior*. Whereas trusting intention is a cognitive-based construct (willing to depend), trusting behavior is a behavior-based construct (depends) [126]. In other words, the trusting behavior is a consequence of trusting intention; it is actually the action of trusting – in case of trusting behavior, we have already built up a trusting intention and, if above the given threshold κ, we are about to start depending on the other entity [57]. We have decided to define trust as a cognitive-based construct (trusting intention), because it is closer to the formation of trustor's beliefs.

6.2.6 Trust and Reputation

Concept of trust is often intermixed with the concept of reputation. Reputation is what is generally said or believed about a person‘s or thing‘s character or standing [85]. Thus, trust is a subjective view of an entity (trustor) about trustworthiness of another entity (trustee); reputation is a collective measure of trustworthiness, i.e., what all the entities in the social network think about trustworthiness of the target entity. An agent may trust another agent because of his good reputation or despite his bad reputation.

6.3 Trusting Beliefs

In this section, firstly, we survey the trusting beliefs relevant for the trusting intention according to Definition 6.2. Secondly, we select the relevant trusting beliefs for the SoSIReČR portal (Section 6.3.3) and evaluate that selection process (Section 6.3.4). Lastly, the sources for the selected beliefs and their quantifications are sketched (Section 6.3.5).

6.3.1 Survey of Trusting Beliefs

In this section we survey the set of trusting beliefs (or simply beliefs), L, influencing interpersonal trusting intentions according to Definition 6.2, with a special focus on trust in informatics literature. We do not consider beliefs forming trust of an entity in a resource – therefore, data provenance and all data quality dimensions, such as accuracy, timeliness, or relevance are omitted [60].

Since many *labels* for beliefs obtained from the literature are synonyms representing the same beliefs, we clustered the obtained labels into the set of beliefs presented in Table 6.2 [86]. The first column in Table 6.2 contains the main labels chosen to represent the beliefs, supplemented, in the brackets, with other labels for the beliefs in the same cluster. Labels for the beliefs are provided as they appear in the literature [87]. Further columns in Table 6.2 represent for every belief its description, and introduce references to papers supporting the given belief. If

[86] The reasons for considering two labels as two different representations of the same belief are discussed further.

[87] With one exception – the label for the belief practice was originally called “experience” in [85], however, this label collides with the belief experience in Table 6.2.

Table 6.2: Identified trusting beliefs

Belief's labels	**Description**	**Papers**
Affinity (Similarity)	A trustee has characteristics in common with a trustor, such as shared tastes, standards, values, viewpoints, interests, or expectations.	[85, 170, 73, 160, 123, 171]
Competence ((Cap)ability)	A trustee has an ability to do for a trustor what the trustor needs.	[96, 56, 57, 13, 77]
Experience (Track record, History of encounters)	A trustor has an experience with a trustee; the trustor has evidence about the trustee's previous interaction with that trustor.	[70, 85, 13, 121, 140]
Expertise	A trustee is considered as an expert in the particular domain.	[85, 96, 70]
Honesty (Bias, Impartiality)	A trustee is honest, has no malicious intentions towards trustor, the trustee tells the truth.	[85, 70, 13, 77]
Practice	A trustee has an experience of solving similar problems in the given domain, but without extensive expertise.	[85, 29]
Reputation	Reputation of a trustee is what is generally said or believed about the trustee's character or standing.	[56, 95, 140, 92]
Willingness (Likely to help, Motivation)	A trustee will do what a trustor needs, he is motivated to do that.	[57, 96, 70]

not specified otherwise, the beliefs are domain specific. The description of the beliefs follows:

Affinity (Similarity): Trust-based recommender systems [160, 123, 171] assume that trust reflects similarity between users. Papers [170, 73] show a strong and significant correlation between trust and similarity; they state that "recommendations [of entities] only make sense when obtained from like-minded people exhibiting a similar taste". Paper [85] defines affinity as an extent to which a trustor has characteristics – shared tastes, standards, values, viewpoints, interests, or expectations – in common with the trustee and confirms that affinity is an important belief in subjective (taste-like) domains.

Competence (Capability, Ability): In [56], trust is presented as a function of capability. Paper [13] states that "to trust an entity [...] means to believe in its capabilities". In [57], they argue that in order to trust an agent, we need to know his competence – ability to do what the trustor needs. Paper [96] indirectly states that trustee's competence influences trust in that entity. In [77], they introduce trust as a complex concept formed by many beliefs, including competence of the trusted person.

Experience (Track record, History of encounters): The label experience is rather ambiguous, because it is used to denote an experience (practice) of a trustee in the particular domain and also to denote an experience (track record) of a trustor with a trustee regarding the particular domain. To distinguish these two beliefs, the former one is denoted as *practice* and discussed later. Here, we discuss experience according to the latter meaning. Papers [70, 121] state that the previous experience of a trustor with a trustee influences the amount of trustor's trust in the trustee. Paper [85] defines track record of a recommender as an experience of the trustor with recommendations from that recommender. In [13], when deriving trust, they compare the number of positive and negative experiences of a trustor with the trustee. Finally, paper [140] defines trust as a "subjective expectation an agent has about another's future behaviour based on the history of their encounter"; "history of their encounter" is their mutual experience.

Expertise (Authority): Expertise of a trustee is an extent to which the trustee "has relevant expertise in the domain of the recommendation-seeking", which "may be formally validated through qualifications or acquired over time" [85]. Paper [96] explains the importance of experts' recommendations when seeking for trustworthy information/ recommendation. In [70] they introduce label *authority* with the meaning "being an expert". Although the idea is understandable, it is rather confusing. The authority of an entity typically implies the existence of expertise of that entity in the given domain; however, it is not always true – someone (e.g., a general or officer) can be an authority just because he has a power over the others.

Honesty (Bias, Impartiality): Paper [13] emphasizes the role of trustee's honesty when deciding how much to trust recommendations from that trustee. An impartial trustee is defined in [85] as someone who "does not have vested interests in a particular resolution to the scenario"; for example, a vendor of LCD monitors might be dishonest regarding the properties of his products. In [77], honesty of a trustee is emphasized as one of the beliefs forming trust. In [70] they state that "a biased source may convey certain information that is misleading or untrue". The label *bias* has an opposite meaning than honesty or impartiality; however, it still points to the same belief, only observed from the opposite side.

Practice: Paper [85] specifies that the trustworthy entity needs to have "practice of solving similar scenarios in the domain", but not necessarily with extensive expertise. Lots of papers describing an algorithm for locating experts, e.g., [29], are actually locating entities with high practice, who may be experts, but it is hard to verify that.

Reputation: Reputation is "what is generally said or believed about a person's or thing's character or standing"[88]. In [56], company's trust to an employee is presented (among other beliefs) as a function of the individual's reputation. Reputation is very often, e.g., in [95, 140, 92], comprehended as an indivisible concept, similarly to trust; whereas trust is considered as a subjective opinion of a trustor, reputation is in this case considered as a collective measure of trustworthiness [92].

Willingness (Likely to help, Motivation): Paper [57] states that "willingness to do what the trustor needs is a crucial belief". In [96], they specify that trustworthy trustee is the one who is likely to help the trustor. According to [70], a trustee may be more believable, if there is a motivation for the trustee to provide accurate information/to participate on the project.

[88] Definition taken from the Concise Oxford Dictionary.

6.3.2 Reliability and Social Proximity

Reliability (Dependability)

Paper [13] emphasizes the role of reliable recommendations in the process of determining trust. In [96], a trustworthy trustee is characterized as a reliable entity. Paper [57] distinguishes two meanings of trust – *core* trust and *reliance* trust; the latter one emphasizes the importance of the trustee's reliance. According to [77], trust is a composition of many different beliefs, including reliability and dependability. In [97] trust is defined as "assumed reliance on some person or a confident dependence on the character, ability, strength or truth of someone". Apart from that, many definitions of trust include reliance of the trusting intentions, not of the trustees; e.g., paper [69] presents a definition of trust as "what an observer knows about an entity and can rely upon to a qualified extent".

Most of the papers [13, 96, 57] consider reliability as an alternative label for trust, at least in some situations. Only the paper [77] comprehends reliance as one of the beliefs forming trust. We agree with the majority, and consider reliability as an alternative label for trust.

Social Proximity

Many papers also argue that social proximity (a trustor and a trustee are friends, colleagues, or acquaintances) matters when seeking recommendations in social networks [76, 85]. Nevertheless, the question is whether the social proximity matters (1) because of higher trust of the recommenders closer to the trustor or (2) because these recommenders are more easily accessible, are more willing to provide any recommendation, and the trustor can better assess their suitability to give recommendations in the given situation (the trustor is more aware of what knowledge they may posses) [85]. We agree with the latter reason, therefore, we assume that social proximity relations between entities (e.g., being a colleague) help when quantifying beliefs, such as honesty or competence; however, as long as the beliefs are quantified, these relations do not play significant role when determining trust.

6.3.3 Trusting Beliefs in SoSIReČR

Table 6.3 identifies for Scenarios $S_1 - S_5$ in Section 6.1 the selected trusting beliefs $L_{sel} \subseteq L$ we would like to quantify and use for the quantification of trust in the SoSIReČR portal; L is the set of beliefs identified in Table 6.2. In Table 6.3, the abbreviation "T", respectively "R", denotes that a certain combination of the belief (row) and the scenario (column) is relevant when forming trusting intentions where the trustee is a target entity, respectively a recommender.

For all Scenarios $S_1 - S_5$, honesty (truthfulness) of a trustee is important – if a trustor does not know whether the trustee (and especially the target entity) is honest about the particular axis of his professional profile, it is hard to believe the quantification of practice, expertise, or willingness [89].

[89] Honesty is considered as an interpersonal belief (as the other beliefs), because the content of the profiles is actually what the entities say about themselves.

Table 6.3: Trusting beliefs in Scenarios $S_1 - S_5$

Belief	**Description**	$\mathbf{S_1}$	$\mathbf{S_2}$	$\mathbf{S_3}$	$\mathbf{S_4}$	$\mathbf{S_5}$
Experience	Does the trustor have a previous experience with the trustee in the given axe(s) of the professional profile?		T/R	T/R	T/R	
Expertise	What is the trustee's expertise in the relevant axe(s) of the professional profile?	T/R	T/R	T/R	T/R	T/R
Honesty	Is the trustee/recommender honest when specifying the given axe(s) of his professional profile?	T/R	T/R	T/R	T/R	T/R
Practice	What is the trustee's practice in the relevant axe(s) of the professional profile?	T/R	T/R	T/R	T/R	T/R
Willingness (to cooperate)	Would the target entity be willing to cooperate with the trustor for the duration of the project/common work regarding the given axe(s) of the professional profile?		T	T	T	

An experience of a trustor with a trustee is of crucial importance when any cooperation/collaboration is needed – it is used in Scenarios $S_2 - S_4$. In S_5, professional profiles of lots of trustees are collected during the aggregation of profiles; therefore, it is hardly assumable that the trustor will evaluate his experience with all these trustees. For S_1, experience with the trustee is not that important, the trustor is not intending to cooperate/collaborate with the trustee; what really matters is the trustee's professional profile.

Willingness is selected for Scenarios $S_2 - S_4$, where a trustor is looking for a collaboration with the target entity; in S_1, S_5, and when trusting a recommender in all Scenarios $S_1 - S_5$, the willingness of the trustee is not necessary, the information/profiles' details of the trustee (who is a recommender) are provided automatically, so there is no need for "willingness to provide information in the profile".

Practice and expertise are important in all Scenarios $S_1 - S_5$, where a trustor needs to know the competence of a trustee in the selected axe(s) of the professional profile.

Reputation, a collective measure of trustworthiness, may supplement trust, but, on the other hand, requires subjective trust relations for its quantification. Thus, reputation can be computed only based on the previous trust computation. And if the reputation is computed and counted among the selected beliefs, trust must be again recomputed, which leads to further recomputation of the reputation, etc. Because of the recursive computation of trust which is influenced by the reputation belief, reputation is not part of L_{sel} (originally described in paper [107]). We plan to incorporate reputation in the future as another belief which may influence trust.

Competence is considered as a combination of beliefs expertise and practice. The belief affinity is not considered, because the influence of (character) affinity when deriving trust between entities in the social network is marginal in the objective domains (such as informatics), where the trustee's competence is more important than his character similarity [85].

6.3.4 Beliefs' Selection Process Evaluation in SoSIReČR

In this section we evaluate the trusting beliefs' selection process presented in Section 6.3.3 by consulting it with members and non-members of the informatics community. To do that, we created a questionnaire consisting of four model situations $S = \{S_{2T}, S_{2R}, S_{3T}, S_{5T}\}$ successively corresponding with Scenarios: S_2, where the trustee is a target entity (hence abbreviated as S_{2T}); S_2, where the trustee is a recommender; S_3 and S_5 in which the trustees are target entities. We omitted Scenario S_1, because it is rather simple, and S_4, which is an analogy of S_3, just seen from the opposite perspective. The description of the model situations S is as follows:

- **Situation** S_{2T}**:** Imagine you are a young fellow with an interesting idea for a journal article and you plan to contact a researcher who knows the domain "searching in object databases", whether he would help you with the preparation of the paper for a prestigious conference VLDB. You have at hand professional profiles of other researchers and their preliminary expression of interest. Which beliefs (factors) from Table 6.4 influence your choice of the most suitable researchers for the academic collaboration?

- **Situation** S_{2R}**:** If you have not found any suitable researcher you could contact and collaborate with, which beliefs from Table 6.5 influence your choice of the persons (*recommenders*) you ask for a recommendation of the suitable researcher for a collaboration?

- **Situation** S_{3T}**:** The European project, which continues in the next year, is looking for a programmer of mobile applications to complete the existing team of programmers. You are responsible for the selection of that programmer. Suppose you are presented with tens of professional profiles of programmers in the social network. Which beliefs from Table 6.6 influence your choice of the three programmers – appropriate (trustworthy) candidates for the given position?

- **Situation** S_{5T}**:** Imagine you are an employee of a fast growing IT company that is programming applications for mobile devices. Your task is to create a report to the Executive Director of the company, who wants to establish new branch in some region of the Czech Republic and, thus, wants to know the potential abilities of students and recent graduates in the various regions of the Czech Republic. The SoSIReCR portal will provide you with the aggregated professional profile of trustworthy students and graduates for each region of the Czech Republic. Which beliefs from Table 6.7 are important to denote the given student or graduate as trustworthy – that is, as a person whose professional profile is included in the aggregated professional profile for the given region?

Table 6.4: Description of beliefs for Situation S_{2T}

Belief	Description
Experience	You or your colleagues at the university have good previous experience with the given researcher
Expertise	The researcher has already published several papers at the most prestigious conferences
Honesty	The researcher is telling the truth, his professional profile corresponds with the reality
Practice	The researcher has already published lots of paper at average conferences
Willingness (to cooperate)	The researcher is willing to cooperate with you in the following 3 months (the idea is interesting for him, he has no deadlines for other projects, the priority of the cooperation with you is high)

Table 6.5: Description of beliefs for Situation S_{2R}

Belief	Description
Experience	You or your colleagues at the university have good previous experience with the given recommender (he has already given you good advices in the past)
Expertise	The recommender is an expert in the given domain, he works in the important research center.
Honesty	The recommender is telling the truth, his professional profile corresponds with the reality
Practice	The recommender has practise in the given domain (he worked 5 years in a company X, however, he was doing rather routine tasks

Table 6.6: Description of beliefs for Situation S_{3T}

Belief	Description
Experience	You or your colleagues at the university have good previous experience with that programmer (You have already worked with the programmer on the project of the similar scope)
Expertise	The programmer has lots of certificates regarding programming of mobile applications or programming in general
Honesty	The programmer is telling the truth, his professional profile corresponds with the reality
Practice	The programmer worked for 5 years in the company X developing applications for mobile devices
Willingness (to cooperate)	The programmer is willing to participate on the project (the job description is interesting for him, the salary conditions are acceptable for him)

Table 6.7: Description of beliefs for Situation S_{5T}

Belief	**Description**
Expertise	The student or graduate has some certificates regarding programming of mobile applications or programming in general, he is an expert in the given domain of applications for mobile devices
Honesty	The student or graduate is telling truth, his professional profile corresponds with the reality
Practice	The student or graduate has practise in programming applications for mobile devices

In each situation S, the respondent is presented with a set of trusting beliefs L_{sel} introduced in Table 6.3 and the respondent's goal is to mark for each such belief $b \in L_{sel}$ one choice ($C_0^{b,s}$, $C_1^{b,s}$, $C_2^{b,s}$, or $C_3^{b,s}$) expressing to which extent the belief b influences trust of the trustor (respondent) in the trustee in the given situation $s \in S$. The choices (four levels of influence) are the same for all situations with the meanings: the given belief b has *no influence* ($C_0^{b,s}$), *minimal influence* ($C_1^{b,s}$), *influence* ($C_2^{b,s}$), or *substantial influence* ($C_3^{b,s}$) in the given situation s. The choice $C_3^{b,s}$ means that if the quantification of the belief b is not satisfactory, it will penalize the considered entity heavily, possibly obstructing any potential trusting intention with that entity in the situation s. The choice $C_2^{b,s}$ ($C_1^{b,s}$) means that if the quantification of the belief b is not satisfactory, it is a major (minor) issue, which will penalize (slightly penalize) the trustee.

The questionnaire was completed by 104 respondents (81% of men) with ages between 20 and 69. Most of the respondents were informatics (81%), the main target group of the SoSIReČR portal. We dispatched the questionnaire to the region coordinators cooperating on the SoSIReČR project and managing different regions of the Czech Republic, therefore, the selection of respondents should have been sufficiently random, at least in the sense that the respondents were selected mainly by the regions' coordinators, not by the authors themselves. Full questionnaire (translated to English) is available at `http://www.ksi.mff.cuni.cz/~knap/files/Questionnaire.pdf`.

Metrics for Evaluating the Results

Table 6.8 summarizes for the belief $b \in L_{sel}$ and the situation $s \in S$ the number of choices $C_i^{b,s}$ (abbreviated as $\#C_i^{b,s}$, or simply as $\#i$ if the belief and the situation are obvious) selected by the respondents; $i \in \{0, 1, 2, 3\}$. For some combinations of the belief and the situation, the results are not defined, which corresponds with the empty spaces in Table 6.3.

Suppose that for a belief $b \in L_{sel}$, a situation $s \in S$, and the numbers $i, j \in \{0, 1, 2, 3\}$, $i \neq j$, we have a null hypothesis $H_0^{b,s,i,j}$: "*$\#C_i^{b,s}$ is equal to $\#C_j^{b,s}$*". Then, using the binomial test, suppose that we reject the null hypothesis $H_0^{b,s,i,j}$ with p-value $p_j < 0.05$. If $\forall k \in \{0, 1, 2, 3\}$, $k \neq i$, the hypothesis $H_0^{b,s,i,k}$ can be rejected in the way described and $\#C_i^{b,s} > \#C_k^{b,s}$, we accept the hypothesis $H^{b,s,i}$: "*$\#C_i^{b,s}$ is the prevailing number of choices for the belief b in the situation s, i.e., the belief b has in the situation s the level of influence C_i*" and this result

Table 6.8: Number of choices $C_i^{b,s}$ selected by the respondents for S_{2T}, S_{2R}, S_{3T}, and S_{5T}

	$\mathbf{S_{2T}}$				$\mathbf{S_{2R}}$			
Belief	**#0**	**#1**	**#2**	**#3**	**#0**	**#1**	**#2**	**#3**
Experience	0	15	22	67	0	25	32	47
Expertise	0	24	52	28	0	23	60	21
Honesty	0	16	27	61	3	25	51	25
Practice	15	21	49	19	20	55	25	4
Willingness	3	4	37	60	-	-	-	-
	$\mathbf{S_{3T}}$				$\mathbf{S_{5T}}$			
Belief	**#0**	**#1**	**#2**	**#3**	**#0**	**#1**	**#2**	**#3**
Experience	0	15	31	58	-	-	-	-
Expertise	0	16	60	28	7	20	49	28
Honesty	0	3	**49**	**52**	0	6	41	57
Practice	0	18	**39**	**47**	8	23	53	20
Willingness	0	14	**43**	**47**	-	-	-	-

is statistically significant, with $p = max_{k|k \neq i}\{p_k\}$; for $p < 0.01$, the appropriate $\#C_i^{b,s}$ is highlighted in Table 6.8 with a dark grey color, for $0.01 \leqq p < 0.05$, the appropriate $\#C_i^{b,s}$ is highlighted in Table 6.8 with a light grey color.

In Situation S_{3T}, for beliefs $b \in \{$*honesty*, *practice*, *willingness*$\}$, we cannot accept the hypothesis $H^{b,S_{3T},i}$ for any $i \in \{0, 1, 2, 3\}$; however, when comparing the sum $\#C_2^{b,S_{3T}} + \#C_3^{b,S_{3T}}$ with the sum $\#C_0^{b,S_{3T}} + \#C_1^{b,S_{3T}}$, the first sum prevails and this result is statistically significant with $p < 0.01$. Therefore, we can accept the hypothesis that the beliefs *honesty*, *practice*, and *willingness* have *influence* or *substantial influence* in Situation S_{3T} (denoted by bolded font in Table 6.8).

Discussion

Table 6.8 shows that an experience has a *substantial influence* in all situations; simply, if a trustor has a positive experience with a trustee, he is much more willing to depend on the trustee. The belief honesty does not have a *substantial influence* in Situation S_{2R}, probably because of lower influence of expertise and practice of a trustee (recommender), and, thus lower needs for honesty of the recommender in S_{2R}. Whereas honesty and willingness have *substantial influence* in S_{2T}, we cannot say that in S_{3T}; the reason for that may be that there is a lack of *system trust* [126] in S_{2T} – the trustee is not bounded by any contract – thus, there is a higher need for honesty and willingness in S_{2T}.

The belief expertise does not have a *substantial influence* in any of the situations, which is rather surprising, especially in Situation S_{3T}. The reason for that may be that most of the respondents comprehend the position of a mobile application programmer as rather standard position not requiring any extensive expertise above the generic programming skills. Practice is more important in S_{3T} (hiring a programmer) than in S_{2T} (writing a paper); this corresponds with the previous hypothesis that programming is comprehended as a rather routine job; however writing a good paper needs expertise more than practice. Finally, practice has *minimal influence* in S_{2R} – a trustee with a vast expertise is more useful when searching for recommendations.

Summary

The evaluation confirmed (with the exception of practice in Situation S_{2R}) that all the selected beliefs in Table 6.3 have an *influence* or a *substantial influence* on trust; these results are statistically significant, with the significance level $\alpha = 0.01$ or $\alpha = 0.05$, respectively. The evaluation also provided the first *estimation of weights* these beliefs should have when quantifying trust.

6.3.5 Towards Sources and Quantification of Selected Trusting Beliefs in SoSIReČR

In this section, we describe the relevant sources, which serve as an evidence for quantifying the beliefs L_{selq}. The set of beliefs L_{selq} is based on the set L_{sel} selected for SoSIReČR in Section 6.3.3. The difference is that L_{selq} contains competence, a more abstract belief representing beliefs expertise and practice as one belief, and it does not contain belief willingness, which needs further investigation. Thus, $L_{selq} = \{honesty, competence, experience\}$. The evaluation presented in Section 6.3.4, which was done for the set L_{sel}, remains relevant also for L_{selq}, because the only difference is that one belief from the evaluation is not considered and two beliefs evaluated as two distinct beliefs are now considered as one more abstract belief.

The list of sources for the beliefs is not complete, it rather represents the sources being the "low-hanging fruits" to start with. We also discuss in this section the quantification of these sources and quantification of the beliefs based on the quantification of these sources; however, preparation of the particular quantification formulas for the beliefs and the configuration of the proper weights of different sources contributing to the quantification of the beliefs is an on-going and future work.

There are three types of beliefs' sources – *implicit*, *explicit*, and *belief*. Explicit sources are based on the explicit user's input. Implicit source are deduced based on the social network analysis methods or data already available in the social network; we further distinguish internal and external implicit sources – internal implicit sources are those available in the social network behind the SoSIReČR portal, external implicit sources are those available in the external social networks or data silos. If the quantification of a belief l influences quantification of another belief l', we say that l is a belief source for l'.

Suppose that $S_l \subseteq S$ is the set of sources being an evidence for the belief l, S is a set of all sources for all selected beliefs L_{selq}. For each source $s \in S_l$, it is important to distinguish (1) the weight of the source, $w(s) \in [0, 1]$, it has when quantifying the belief l, $\sum_{s \in S_l} w(s) = 1$, and (2) the value, $val(s, u, v, d)$, of each source $s \in S_l$ w.r.t. the particular $u, v \in V$, and $d \in D$. The value of each source may be comprehended as instantiation of that source for the particular combination of u, v, and d. We suppose that all the values for the sources are normalized, i.e., $val(s, u, v, d) \in [-1, 1] \cup \perp$, $\forall s \in S$. The higher the non-negative value of the source $s \in S_l$, the more evidence has the source s about positive quantification of the belief l, the lower negative value of the source s, the more evidence has the source s about negative quantification of the belief l. Not all the sources has to provide the whole spectrum of values, e.g., they may provide only binary values 1 and -1 or a positive evidence $[0-1]$.

Definition 6.5. Suppose we want to quantify the belief $l \in L_{selq}$. Then, let us define $B^l = \{\beta^l_{a_1}, \ldots, \beta^l_{a_{|V|}}\}$, a set of partial belief functions $\beta^l_{a_i} : V \times D \rightarrow [-1, 1] \cup \perp$, $1 \leq i \leq |V|$, $F = (D, R_D)$ is the domain hierarchy. Thus, there is one partial belief function for every belief l and agent a_i in the social network SN. Let us define a partial *belief function* $\beta^l : V \times V \times D \rightarrow [-1, 1] \cup \perp$ which is a union of all partial functions $\beta^l_{a_i}$ for all $a_i \in V$.

The partial beliefs function $\beta^l(u, v, d)$ quantifying belief l gives certain weights $w(s_i)$ to all values $val(s_i, u, v, d)$ of the sources $s_i \in S_l$ contributing to the quantification of that belief l for the given agents u, v, and domain d; $1 \leq i \leq |S_l|$. Certain beliefs may rely on the quantification of other beliefs, as in the case of competence relying on the quantification of honesty.

Suppose we quantify the value of the source s for the given domain d and agents u, v, i.e., $val(s, u, v, d)$. However, we do not have any value for the domain d', i.e., $val(s, u, v, d')$. Since the domains D form the tree (as described in Definition 6.3), we may employ the algorithm *TopicTrustLocal* described in Paper [110], so that the value $val(s, u, v, d)$ may be used to estimate the value $val(s, u, v, d')$, where d' is a more specific or generic domain than d. Obviously, when estimating $val(s, u, v, d')$, the value $val(s, u, v, d)$ should have an impact on $val(s, u, v, d')$ which indirectly correlates with the distance $\delta(d, d')$ (such distance $\delta(d, d')$ is equal to the number of relations of "being a subdomain" between d and d' in the domain hierarchy, see Definition 6.3). In paper [110], we use the classical distance decay model [142] for determining the proper impact of $val(s, u, v, d)$ on $val(s, u, v, d')$. Such mechanism described in Paper [110] may be used in general to influence the computed value $val(s, u, v, d)$ with the computed values for more generic or more specific domains.

Honesty

Honesty, quantified by $\beta^{hon}(u, v, d)$, is a belief of an agent u that the value on the particular axis $d \in D$ of the professional profile of agent v corresponds with the reality.

- **Explicit Source.** Every agent u can denote another agent v as being honest/dishonest regarding the provided information in the particular axis d of agent v's professional profile. Such explicit source may be expressed as a positive or negative honesty relation (u, v, d) – $(u, v, d) \in E_{PH}$, if the honesty relation is positive, E_{PH} is the set of all positive honesty relations, or $(u, v, d) \in E_{NH}$, if the honesty relation is negative, E_{NH} is the set of all negative honesty relations.

- **Explicit Source.** Agent u can denote agent v as being dishonest regarding agent v's professional profile as a whole; such relation (u, v) is called *global negative honesty relation*; $(u, v) \in E_{GNH}$, where E_{GNH} is the set of all global negative honesty relations.

The honesty belief, $\beta^{hon}(u, v, d)$, is then quantified based on the introduced sources as follows:

$$\beta^{hon}(u, v, d) = \begin{cases} -1 & if\ (u, v, d) \in E_{NH}\ \vee\ (u, v) \in E_{GNH} \\ 1 & if\ (u, v, d) \in E_{PH} \\ \bot & otherwise \end{cases} \tag{6.1}$$

Competence

Competence, quantified by $\beta^{comp}(u, v, d)$, is a belief of an agent u that the agent v is competent regarding the domain $d \in D$.

- **Explicit Source.** Value on the particular axis d of the professional profile of agent v, expressing agent v's competence w.r.t. the domain d. An agent might be a person, such as a student, IT professional, or academic.

- **(Internal) Implicit Source.** Competence of agent v w.r.t. d is influenced by the axis d of the professional profile of (1) research groups agent v belongs/belonged to (should take into account the the size of the group, how long does the agent v participate in that group) and (2) projects v is/was working on (should take into account the length of the project).

- **Belief Source.** Honesty of the axis d of the agent v's professional profile. Honesty of the professional profiles of research group/projects the agent v belongs to/participates on. Thus, $hon \in S_{comp}$, $val(hon, u, v, d) = \beta^{hon}(u, v, d)$.

- **(External) Implicit Source.** Research papers obtained from various portals, such as DBLP, ACM[90], CiteSeer[91], or IEEE[92]. Projects obtained from CORDIS portal[93].

- **(Internal) Implicit Source.** As was discussed in Section 6.3.2, we do not comprehend social proximity between two persons as a belief, but rather as a factor, which really helps when quantifying beliefs. In the social network behind the SoSIReČR project, social proximity relations are represented by the professional social relations, such as that two agents collaborate on the same project, work in the same research group etc. To further help quantifying competence $\beta^{comp}(u, v, d)$, we decided to use social proximity between the agents u and v. If an agent u is working in the same research group as v, even if there is no special positive honesty relation $(u, v, d) \in E_{PH}$, the competence belief should be higher than in case of absence of such social proximity. The reason for that is that an agent u should be able to evaluate, thanks to his social proximity with agent v, the values of the axis d of the agent v's professional profile. If the agent u wants to express explicit honesty relation, e.g., a negative honesty relation, he can do that, and such explicit statement will overwrite any implicit social proximity value. Social

[90] http://dl.acm.org/

[91] http://citeseerx.ist.psu.edu/index

[92] http://ieeexplore.ieee.org

[93] http://cordis.europa.eu/home_en.html

proximity source values are computed with the variation of the Appleseed metric described in Section 6.4.3.

The quantification, $\beta^{comp}(u, v, d)$, is driven by a convex combination of the source values S_{comp} associated with agents u,v, and domain d as follows:

$$\beta^{comp}(u, v, d) = \sum_{s \in S_{comp}} w(s) \cdot val(s, u, v, d) \tag{6.2}$$

Social proximity source values are taken into account only if no honesty relations (u, v, d) is present (this adjustment is not incorporated in Formula 6.2). More experiments are needed to conduct the suggestions for the weights $w(s)$ of sources from S_{comp}.

Due to the nature of the sources we currently have, we decided to start just with the belief competence; later on, we will differentiate between practice and expertise; e.g., we take into account the quality of the conferences (according to a certain list of conference rankings).

Experience

Experience, quantified by $\beta^{exp}(u, v, d)$, is a belief of an agent u expressing a certain experience with agent v regarding domain d.

Explicit Sources. Every agent u can express an experience with another agent v regarding situation $s \in S$ and domain d. Each such situation represents a different explicit source – evidence of the experience belief and may contribute differently to the quantification of the experience belief. The set of situations S should involve: collaboration in a research group v focusing on the domain d, cooperation on certain project v covering the domain d, writing common paper with agent v focusing on the domain d, etc. The portal should support the agents in expressing such experiences by automatically generating, e.g., feedback questionnaires when certain project finishes, paper is published, etc.

The quantification, $\beta^{exp}(u, v, d)$, is driven by a convex combination of the source values and weights associated with these sources for agents u, v, domain d, and situation s. More experiments are needed to conduct the suggestions on the weights of the particular sources.

6.3.6 Related Work

McKnight and Chervany [126] conducted an extensive survey of various beliefs the trusted entities should have (the survey is based on the interdisciplinary papers published between years 1960 and 1995) and group these beliefs to four categories:

(1) *Benevolence*: A trustee cares about the welfare of a trustor and is therefore motivated to act in the trustor's interest. A benevolent person does not act opportunistically.

(2) *Competence*: A trustee has the ability to do for a trustor what the trustor needs to have done.

(3) *Honesty*: A trustee makes good faith agreements, tells the truth, and fulfils any promises made.

(4) *Predictability*: Trustee's actions are consistent enough that a trustor can forecast what the trustee will do in a given situation.

Although focused only on informatics literature, our survey has lots of similarities. Competence and honesty is comprehended similarly. The category predictability is a function of the belief experience – if a trustor knows what were the actions of a trustee in the past, he can predict the future behavior of the trustee. The category benevolence has a substantial overlap with the belief willingness and is related to our belief honesty ("A benevolent person does not act opportunistically").

6.4 Trust Metrics

Quantification and propagation of trust and distrust in social networks (trust metrics) have been studied in lots of papers, e.g. [171, 76, 81]. Although we have already explained why propagation of trust as a black box concept is complicated and sentenced to failure, we can consider leveraging of the techniques proposed in these papers so that (1) the trust values are quantified based on the quantification of the beliefs and (2) the domain specificity of trust is considered.

To that end, this section discusses the basic requirements on the trust metrics for SoSIReČR and, then, describes the particular relevant trust metrics from the literature which satisfy these requirements – *TidalTrust*, *Advogato*, and *Appleseed*. Afterwards, numerous properties of these metrics are discussed and the suitability of these trust metrics for the SoSIReČR portal is considered. Appleseed is declared as the winner. Section 6.5 describes the trust model for SoSIReČR and leverages the trust metric Appleseed, so that it supports the quantification of beliefs as described in Section 6.3 and the domain specificity of trust. As a result, such leveraged trust metric may be used as the metric τ from Definition 6.4.

Categorization of Trust Metrics

Ziegler and Laursen [171] present categorization of trust metrics – they distinguish *global* and *local* trust metrics.

Global trust metrics, such as [145, 95, 148], compute the *reputation* of the particular agent in the social network based on the average of trust estimations the others have in that entity. Global metrics assign trust ranks based upon complete social trust network information. These global metrics violate our assumption about subjectiveness of the trust one agent has in another agent (see Definition 6.2); thus, we focus further on the local trust metrics.

Local trust metrics, such as those introduced in [76, 117], comprehend trust as a subjective opinion of one entity in another entity. Local trust metrics compute trust of certain agent (trustor). Ziegler and Laursen [171] further distinguish two types of local metrics – *scalar* and *group* local metrics. Group metrics compute trust between the trustor and a group of other entities at once (in parallel); scalar metrics compute only trust between a trustor and one other agent at once.

Social Trust Network

A local trust metric operates on top of a *social trust network*, a social network with edges representing the trust relations between agents (see Definition 6.6). We suppose such social trust network for every trust metric discussed further – TidalTrust, Advogator, and Appleseed. Set X in Definition 6.6 is defined differently for every trust metric and is discussed further. If there is no trust relation between two agents in a social trust network, then the trust value between them is not defined, but it may be computed by the appropriate trust metric.

Definition 6.6. Let us define a *social trust network*[94], which is a directed weighted graph $STN(V, E_T, \omega)$, where V represents the set of agents from the social network $SN = (V, E, a, b, l_E)$, $E_T \subseteq V \times V$ represents the set of *trust relations* between them, and $\omega : V \times V \rightarrow X$ is the function labeling the trust relations with the trust values from X.

Furthermore, let us define a trust relation path (see Definition 6.7) and distinct trust relation path (see Definition 6.8). We will use these definitions when discussing the trust metrics.

Definition 6.7. A *trust relation path*, $\pi_{u,v}$, between two entities u and v in $STN(V, E_T, \omega)$, $u, v \in V$, is a progression of trust relations $(u = v_1, v_2), (v_2, v_3), \ldots,$ $(v_i, v_{i+1}), \ldots, (v_{n-1}, v_n = v)$, $v_i \in V$, pairwise distinct, so that each $e_i = (v_i, v_{i+1}) \in E_T$; where $1 \leqq i \leqq n$. The expression $e \in \pi_{u,v}$ denotes that $e \in E_T$ is on path $\pi_{u,v}$.

Definition 6.8. Two paths $\pi_{u,v}$ and $\pi'_{u,v}$ are distinct if $\exists e \in \pi_{u,v}$ and $\exists e' \in \pi'_{u,v}$, s.t. $e \neq e'$. The number of distinct trust relations involved in $\pi_{u,v}$ is denoted as $|\pi_{u,v}|$, whereas $\#\pi_{u,v}$ denotes the number of distinct trust relation paths between u and v.

Trust Transitivity

Trust value is transitive, if the existence of trust relations $(a, b) \in E_T$ and $(b, c) \in E_T$ in $STN(V, E_T, \omega)$ leads to a (computed) trust value between a and c being derived based on $\omega(a, b)$ and $\omega(b, c)$. As stated in [160] "there have been fierce discussions in the literature whether or not trust is transitive; from the perspective of network security (where transitivity would, for example, imply accepting a key with no further verification based on trust) or formal logics (where transitivity would, for example, imply updating a belief store with incorrect, impossible, or inconsistent statements) it may make sense to assume that trust is not transitive [93, 40, 88]". On the other hand, Golbeck [76] and Guha et al. [81] show experimentally that trust in social networks similar to the social network in Definition 6.6 is transitive and may propagate along the trust relations. Based on that experiments, trust transitivity is one of the requirements for the trust metric.

[94] A concept of social trust networks is similar to the web of trust introduced in PGP system.

6.4.1 TidalTrust

The *TidalTrust* metric is in detail described in [76], we depict here only the core formula (Formula 6.3) for computing the trust value $\widetilde{\tau_{tt}}(u, v)$ (see Definition 6.9).

Definition 6.9. Suppose the social trust network $STN(V, E_T, \omega)$, where ω is defined as $\omega : V \times V \to [1, 9]$. Let us define a partial trust function $\widetilde{\tau_{tt}} : V \times V \to [1, 9] \cup \perp$; $\widetilde{\tau_{tt}}(u, v)$ is assessing the trust value between the trustor $u \in V$ and the trustee $v \in V$ *without respect* of any domain.

The trust value $\widetilde{\tau_{tt}}(u, v)$ is computed as a *restricted* weighted average over distinct trust relation paths between u and v (see Formula 6.3). In Formula 6.3, $n(u) = \{x \in V : (u, x) \in E_T\}$, E_T is the set of edges from $STN(V, E_T, \omega)$. In [76], the entity u is called *source* and v *sink*.

$$\widetilde{\tau_{tt}}(u, v) = \begin{cases} \omega(u, v) & (u, v) \in E_T \\ \frac{\sum_{x \in n(u) | \omega(u,x) \geq max} \omega(u,x) \widetilde{\tau_{tt}}(x,v)}{\sum_{x \in n(u) | \omega(u,x) \geqq max} \omega(u,x)} & \text{otherwise} \end{cases} \tag{6.3}$$

Since entities are more likely to connect with entities they trust highly [76], Formula 6.3 restrict the weighted average by defining the threshold $max \in [1, 9]$ for $\omega(u, x)$.

If there are more trust relation paths from the source u to the sink v, only the shortest paths required to connect u and v are considered in the computation of the restricted weighted average in Formula 6.3. This approach preserves the benefits of shorter path lengths [76].

6.4.2 Advogato

Advogato[95] trust metric, presented in [117, 147], was used to "determine which users are trusted by members of an online community" [171].

Definition 6.10. Suppose the social trust network $STN(V, E_T, \omega)$, where ω is defined as $\omega : V \times V \to \{0, 1\}$. Let us define a partial trust function $\widetilde{\tau_{ad}} : V \times V \to \{0, 1\} \cup \perp$; $\widetilde{\tau_{ad}}(u, v)$ is assessing the trust value between the trustor $u \in V$ and the trustee $v \in V$ *without respect* of any domain.

The further description of the Advogato trust metric is based on the description in [171]. Input to the Advogato trust metric is $m \in N$ and a trust seed $s \in V$. Let us suppose capacities $C_V : V \to \mathbb{N}$, which are assigned to every vertex $x \in V$ based on the shortest path distance from the seed s to x; $C_V(s) = m$, the capacity of each successive distance level $lv + 1$ is equal to the capacity of the previous level lv divided by the average outdegree of trust relations $e \in E_T$ extending from lv.

The basic idea of the Advogato metric is to employ Ford-Fulkerson integer maximum network flow algorithm [58] to compute $\widetilde{\tau_{ad}}$. However, the integer maximum network flow algorithm requires only one sink and capacities on the edges. As a result, the social trust network behind Advogato trust metric has to be adjusted before the integer maximum network flow algorithm can be launched: every vertex x with $C_V(x) \geq 1$ is represented by two new vertices x^+, and x^- and

[95] `http://www.advogato.org/`

edge (x^-, x^+), the original capacity $C_V(x)$ is enforced as capacity $C_E(x^-, x^+) = C_V(x)$, extra super-sink z is created, and edge (x^-, z) is added for each x^- with the capacity $C_E(x^-, z) = 1$. The detailed steps of the transformation are described in [171].

After the transformation and execution of the Ford-Fulkerson integer maximum network flow algorithm, the agents $X \subseteq V$ trusted by s (i.e., those agents $x \in X$, for which $\widetilde{\tau_{ad}}(s, x) = 1$) are exactly those agents $x \in X$ for which there is a flow from nodes x^- to the super-sink z.

6.4.3 Appleseed

Ziegler and Laursen [171] proposed the Appleseed trust metric calculating trust for a collection of entities at once by energizing the selected entity (the seed node, the trustor) and spreading the energy to other entities connected by trust relations in the social trust network. The idea of the Appleseed metric is based on the spreading activation models, first proposed in [146]. The description of the Appleseed metric is based on paper [171].

Definition 6.11. Suppose the social trust network $STN(V, E_T, \omega)$, where ω is defined as $\omega : V \times V \to [0, 1]$. Let us define a partial trust function $\widetilde{\tau_{ap}} : V \times V \to (0, in^0) \cup \perp$; $\widetilde{\tau_{ap}}(u, v)$ is assessing the trust value between the trustor $u \in V$ and the trustee $v \in V$ *without respect* of any domain.

Algorithm 4 summarizes the idea of the Appleseed trust metric. The input to the algorithm is formed by the vertex s (source, to which energy is injected), the amount of energy in^0 injected to the source s, the spreading factor f, accuracy threshold T_c serving as the convergence criterium, and the social trust network $STN(V, E_T, \omega)$ the metrics operate on. Algorithm 4 runs in iterations; an iteration is denoted by $i \in \mathbb{N}^0$, V_i holds the nodes reached in iteration i or in the previous iterations $j < i$, $j \in \mathbb{N}^0$; $trust_i(x)$ denotes the trust value $\widetilde{\tau_{ap}}(s, x)$ valid in iteration i. Let $in_i(x)$ denote the energy influx into node $x \in V$ in iteration i.

In every iteration i, every node $x \in V_{i-1}$, which received certain energy $in_{i-1}(x)$ in the previous iteration $i - 1$, distributes its energy along its outgoing trust relations; this is a core part of the algorithm depicted in Lines 13 – 23. Parameter f (spreading factor) influences the portion of energy $f \cdot in_{i-1}(x)$ that the node x distributes in iteration i among successors (Line 22), while retaining for itself $(1 - f) \cdot in_{i-1}(x)$ of energy; the retained energy is stored as $trust_i(x)$ (Line 12).

Algorithm 4 uses edge weight normalization – the energy distributed along the edge $(x, u) \in E_T$ (Line 22) depends on its relative weight w, i.e., $\omega(x, u)$ compared to the sum of weights of all outgoing edges of x (Line 21). Such normalization of trust values is used in many global trust metrics, such as [145]; however, the edges are not weighted there. Serious problems with data normalization occur when the edges are weighted as depicted in Figure 6 of paper [171]. To illustrate date, suppose that an energy influx to nodes $b \in V$ and $d \in V$ is the same; consequently, if b has only one trust relation to $c \in V$ with $\omega(b, c) = 0.25$ and d has three trust relations to $e, f, g \in V$, $\omega(d, e) = 1$, $\omega(d, f) = 1$, and $\omega(d, g) = 1$, then, c would be trusted three times as much as e, f, g. Appleseed alleviates such problem of data normalization by backward propagation of part of the energy from every node back to the source; this is realized in Lines 18 and 19 by adding the edge (u, s), i.e., $E_T = E_T \cup \{(u, s)\}$, with weight $\omega(u, s) = 1$. Such backward propagation also ensures that there are no dead ends – nodes, which would accumulate energy.

To sum up the algorithm, Lines 11 – 24 demonstrates one iteration of Appleseed; it processes progressively nodes V_{i-1} discovered up to iteration $i - 1$. In Line 12, trust rank of the node $x \in V_{i-1}$ is updated; note that only $(1 - f)$ of the energy $in_{i-1}(x)$ is added to the current trust rank of x. In Lines 13 – 23, every edge (trust relation) of the node x is examined. If we discover new node $u \notin V_i$, we initialize the node's trust value $trust_i(u)$, energy influx $in_i(u)$, and realize backward trust propagation as explained in the previous paragraph. In Line 21, we calculate the relative weight w of the processed trust relation and, based on that, influx $in_i(u)$ to the node u, the target of the trust relation, is computed in Line 22.

The distribution of the energy in the social trust network ends when the convergence criterium given by the accuracy threshold T_c is met, i.e., when the energy of any node in the current iteration i does not change significantly (as depicted in Line 25 of Algorithm 4). The convergence criterium must be met after a certain amount of iterations, because the spreading factor f ensures that less and less energy is flowing through the social trust network each iteration.

The output of the trust metric Appleseed is an assignment function holding for each $x \in V_i$ the trust values $\widetilde{\tau_{ap}}(s, x)$; $\widetilde{\tau_{ap}}(s, x) = trust_i(x)$, where i is the last iteration. The nodes $y \in V_T \setminus V_i$ not reached in iteration i have the trust value $\widetilde{\tau_{tt}}(s, y) = 0$.

Distrust in Appleseed

Appleseed trust metric can be extended to support distrust relations, i.e., trust relations (u, v) when $\omega(u, v) \in [-1, 0]$. Function $\widetilde{\tau_{apd}}$ realizes the Appleseed trust metric with distrust.

Definition 6.12. Suppose the social trust network $STN(V, E_T, \omega)$, where ω is defined as $\omega : V \times V \to [-1, 1]$. Let us define a partial trust function $\widetilde{\tau_{apd}} : V \times V \to (-in^0, in^0) \cup \perp$; $\widetilde{\tau_{apd}}(u, v)$ is assessing the trust value between the trustor $u \in V$ and the trustee $v \in V$ *without respect* of any domain.

The only difference between Definition 6.11 and Definition 6.12 is in the range of the trust function and the function ω. In Appleseed with distrust, the computation with distrust relations may be directly incorporated to the iterative process of Algorithm 4. To realized that, Line 22 in Algorithm 4 has to be changed as follows (function $sign(x)$ returns the sign of value x):

$$in_i(u) \leftarrow in_i(u) + out(x, i) \cdot sign(\omega(x, u)) \cdot w \tag{6.4}$$

$$out(x, i) = \begin{cases} f \cdot in_{i-1}(x) & if \ (in_{i-1}(x) \geq 0) \\ 0 & otherwise \end{cases} \tag{6.5}$$

As a result, energy $in_i(u)$ distributed along the edge (x, u) in Line 22 may be negative, but it is distributed only if $in_{i-1}(x)$ is positive (if $in_{i-1}(x) < 0$ then $out(x, i) = 0$ and no energy $in_i(u)$ is distributed) [171]. Such approach ensures

Algorithm 4 Appleseed trust metric

```
Output: $Trust_A(s \in V, in^0 \in \mathbb{R}^{\geq 0}, f \in [0,1], T_c \in \mathbb{R}^{\geq 0}, STN(V, E_T, \omega))$
 1: $in_0(s) \leftarrow in^0$
 2: $trust_0(s) \leftarrow 0$
 3: $i \leftarrow 0$
 4: $V_0 \leftarrow \{s\}$
 5: repeat
 6:   $i \leftarrow i + 1$
 7:   $V_i \leftarrow V_{i-1}$
 8:   For each $x \in V_{i-1}$ do
 9:     $in_i(x) \leftarrow 0$
10:   end for
11:   For each $x \in V_{i-1}$ do
12:     $trust_i(x) \leftarrow trust_{i-1}(x) + (1 - f) \cdot in_{i-1}(x)$
13:     For each $(x, u) \in E_T$ do
14:       if $u \notin V_i$ then
15:         $V_i \leftarrow V_i \cup \{u\}$
16:         $trust_i(u) \leftarrow 0$
17:         $in_i(u) \leftarrow 0$
18:         $E_T \leftarrow E_T \cup (u, s)$
19:         $\omega(u, s) \leftarrow 1$
20:       end if
21:       $w \leftarrow \omega(x, u) / \sum_{(x,u') \in E_T} \omega(x, u')$
22:       $in_i(u) \leftarrow in_i(u) + f \cdot in_{i-1}(x) \cdot w$
23:     end for
24:   end for
25:   $m \leftarrow max_{y \in V_i}\{trust_i(y) - trust_{i-1}(y)\}$
26: until $(m \leq T_c)$
27: return $\{(x, \widetilde{\tau_{ap}}(s, x) = trust_i(x)) \mid x \in V_i\}$
```

that the Appleseed metric with distrust does not distribute positive energy as a result of the multiplication of the negative influx to node u, i.e., $in_i(u) < 0$, and the negative weight w of the outgoing trust relation in Line 22 of Algorithm 4.

The Appleseed metric with distrust still converges, but the amount of energy the metric is working with is not invariant in every iteration of the algorithm. Lastly, the Appleseed metric with distrust ensures that in case of no distrust relation in the social trust network, the metric works as depicted in Algorithm 4. In further text, if talking about Appleseed trust metric, we mean Appleseed with distrust realized by trust function $\widetilde{\tau_{apd}}$.

Further extensions which may customize the Appleseed trust metric involve – the limitation of the number of nodes the energy may reach, the upper bound on trust relation paths' lengths, or the adjustment of the spreading factor for the source node, so that source node is not accumulating any energy.

Table 6.9: Trust properties of the selected trust metrics

Property	**TidalTrust**	**Advogato**	**Appleseed**
Type of local trust metric	scalar	group	group
Trust transitivity	YES	YES	YES
Composability	YES	YES	YES
Computation locus	central	central	central
Trust decay	NO	NO	YES
Bottleneck property	YES	YES	YES
Deterministic computation	YES	NO	YES
Weighted trust relations	YES	NO	YES
Trust values	continuous [1-10]	binary 0,1	continuous $(-in^0, in^0)$
Distrust support	NO	NO	YES
Normalization of the trust value	NO	YES	YES
Trust value is a rank	NO	NO	YES

6.4.4 Summary of the Trust Metrics

Table 6.9 summarizes various properties of the trust metrics discussed in the previous sections. For the last two properties, *NO* is better, for the rest of the properties, *YES* is better, if applicable. The non-obvious properties and their values are discussed further:

- *Type of local trust metric* was discussed in Section 6.4, page 149. Trust transitivity was discussed in Section 6.4, page 150.

- *Composability.* Golbeck [76] particularizes a *composability* property of the trust metric, which we illustrate on the TidalTrust metric: if there are more recommenders $V' \subseteq V$ having trust relation with the trustee v, i.e., $(x, v) \in E_T$, $x \in V'$, and there is a trust value between trustor u and these recommenders above certain threshold, i.e., $\widetilde{\tau_{tt}}(u, x) > \kappa$, the trust value $\widetilde{\tau_{tt}}(u, v)$ should be composed from opinions (trust relations) of all recommenders $x \in V'$. This property is satisfied by all metrics.

- *Computation locus* denotes whether the trust relations between individuals are evaluated and quantified on a single machine (central computation locus) or their evaluation and quantification is distributed in the network (distributed computation locus). All metrics have a central computation locus.

- *Trust decay.* In contrary to Golbeck's approach [76], Guha et al. [81] suggest appropriate trust discounting – *trust decay* – with the increasing lengths of trust relation paths. We agree with this demand, since trust decay realistically models the concept of trust – a user trusts more his direct friends than the friends of his friends. TidalTrust considerers only the shortest paths when composing the trust value, but it does not satisfy the trust decay property. Advogato does not satisfy trust decay property, because the trusting value is expressed only as a binary value. Appleseed addresses the trust decay by including the spreading factor f.

- *Bottleneck property.* Thesis [147] proposes the bottleneck property of the attack-resistant trust metrics. The bottleneck property, informally stated, is that the "trust quantity accorded to an edge (s, t) is not significantly affected by changes to the successors of t" [147]. Advogato satisfies that property, because if there is any change to successors of t, the edge (s, t) must have already been processed. Spreading factor f of Appleseed trust metric is crucial for maintaining the bottleneck property. TidalTrust supports the bottleneck property by using only the shortest trust relation paths when computing trust value between two agents.

- *Deterministic computation.* The computation is deterministic for Appleseed and TidalTrust. Advogato trust metric is non-deterministic, because Ford-Fulkerson algorithm computing integer maximum network flow can non-deterministically select the edge the flow will go through.

- *Distrust support.* Lots of approaches, including TidalTrust and Advogato, consider only full trust and degrees of trust. But distrust is semantically different from low trust [81, 171]. Only Appleseed has direct support for distrust.

- *Normalization of the trust value.* Trust value is normalized in Advogato and Appleseed – the more trust relations the entity defines, the less energy (trust) each target entity of these trust relations receives. However, in Appleseed, certain mechanisms, such as non-linear normalization (considering, e.g., squares of $\omega(x, u')$ instead of just $\omega(x, u')$ in Line 21 of Algorithm 4) and backward propagation (Line 18), mitigate the negative effects of the normalization.

- *Trust value is a rank.* The range of the computed trust value may be either equal to the range of ω in the social trust network $STN(V, E_T, \omega)$ (as in case of TidalTrust and Advogato), or it may differ (in case of Appleseed). The trust value computed by the Appleseed trust metric must be considered only as a rank of the agent – the higher the rank, the more the agent is trustworthy.

The time complexity of TidalTrust is $O(E_T)$, if $E_T \gg V$ in the social trust network $STN(V, E_T, \omega)$ [76]. The time complexity of Advogato is $O(E_T \cdot f)$, where f is the maximum flow, if the standard Ford-Fulkerson algorithm is used [58]. The time complexity of Appleseed is $O(E_T \cdot i)$, where i is the number of iterations executed by the algorithm. As described in [171], for the social trust network with 572 nodes, average outdegree of five trust relations per node, $T_c = 0.01$, and $in^0 = 200$, the Appleseed algorithm terminates in 38 iterations; if we change the input energy to $in^0 = 800$, the algorithm terminates in 45 steps. As debated in [171], the convergence of the Appleseed algorithm takes place rapidly even in larger networks.

Discussion

For the purpose of the SoSIReČR portal, Advogato is not a suitable trust metric, because it does not allow the weighted trust relations. TidalTrust does not introduce trust decay, a logical property of the trust metric. Furthermore, both Advogato and TidalTrust do not support distrust, having a major drawback. Another disadvantage of TidalTrust is that it is a scalar metric.

The advantage of TidalTrust over Appleseed is that Appleseed suffers from the data normalization; however, the Appleseed trust metric defines methods how to mitigate that problem – non-linear normalization and backward propagation of energy.

The disadvantage of Appleseed is that the trust values computed by the metric $\widetilde{\tau_{apd}}$ do not correspond with the trust values prescribed by the function ω in the social trust network $STN(V, E_T, \omega)$. However, as described in [171], the trust values $\widetilde{\tau_{apd}} \in (-in^0, in^0) \cup \perp$ may be adjusted to $\widetilde{\tau_{apd}} \in [-1 \pm \epsilon, 1 \pm \epsilon] \cup \perp$ by tuning the input energy in^0. In such case, the time complexity of Appleseed is $O(E_T \cdot i \cdot m)$, where i is the number of iterations executed by the algorithm and m is the number of the preliminary runs of the Applessed algorithm needed to tune the amount of input energy.

We chose the Appleseed trust metric as the trust metric τ used in the SoSIReČR trust model, because it covers all the relevant properties well and it does not have any major drawback.

6.4.5 Related Work

Kuter and Golbeck propose in [113] an algorithm SUNNY for trust inference in social networks; the computed trust value is supported by a measure of confidence in the computed value; the confidence measure is derived probabilistically, based on similarities of entities' ratings. They evaluated the algorithm on the FilmTrust network [76], where the entities rated various films, and compared the algorithm with TidalTrust algorithm – SUNNY's average error was 6.5% lower, performing much more better for $p < 0.05$ in the standard two-tailed t-test. Nevertheless, the computation of the confidence values seems to be non-trivial (unfortunately, no time complexity of the algorithm is given) and the resulting improvement over TidalTrust is rather minor.

Guha et al. [81] developed a framework of trust propagation schemes. They introduced several ways of propagating trust in social networks. Except for *direct propagation* (use of trust transitivity), they propose other atomic propagations – *co-citation* (if $\tau(i_1, j_1, d) \geqq 0$, $\tau(i_1, j_2, d) \geqq 0$, and $\tau(i_2, j_2, d) \geqq 0$, we can infer $\tau(i_2, j_1, d) \geqq 0$), *transpose trust* (if $\tau(u, v, d) \geqq 0$, then $\tau(v, u, d) \geqq 0$), and *trust coupling* (if $\tau(u_1, v, d) \geqq 0$, $\tau(u_2, v, d) \geqq 0$, then $\tau(u, u_1, d) \geqq 0$ implies $\tau(u, u_2, d) \geqq 0$). In the trust metrics in Section 6.4, transitivity of trust is used. Transpose trust propagation is in contrast with our assumption of trust subjectivity and asymmetry. The co-citation and trust coupling atomic propagations are not used, because they are vulnerable to attacks – a malicious entity can easily simulate the prerequisites of co-citation or trust coupling propagations and obtain an extra trust.

PageRank [145] and HITS [98] algorithms are trust metrics which are used to rank nodes in the graphs – social networks. PageRank and HITS algorithms are in their original versions global trust metrics, unsuitable for the definition of trust introduced in Definition 6.2. There are lots of variations of PageRank, some of them introduce a personalized version of PageRank, such as [161]; unfortunately, they compute a matrix as a result of the trust function (including all vertices V in the social network) instead of a vector of these vertices, bringing performance issues. Furthermore, PageRank works only with non-negative values. We can remove negative edges, however, in that case, as was pointed in [81, 171], we cannot distinguish between negative values (distrust) and no values at all; alternatively, we may shift the interval $[-1, 1]$ for the function τ to the positive numbers, but in that case, the semantics of distrust is blurred with "low trust".

6.5 Trust Model for SoSIReČR

In this section we define a trust model for SoSIReČR. We suppose that the concept of trust is comprehended as in Definition 6.2. Let us suppose that $L_{selq} \subseteq L$ is a set of beliefs relevant for the SoSIReČR trust model as defined in Section 6.3.3 and refined in Section 6.3.5 including the beliefs: honesty, competence, and experience. Based on the quantification of these beliefs, the trust value τ (see Definition 6.4) will be computed. Before formalizing the notion of the trust model for SoSIReČR, let us define *beliefs directly contributing to trust* (Definition 6.13) and a *social trust beliefs network* (Definition 6.14), a basic data structure, which will be used to hold the quantified beliefs directly contributing to trust.

Definition 6.13. Let us denote a belief $l \in L_{selq}$ as a belief *directly contributing to trust*, if that belief is directly used by the trust function τ to compute trust value. Such set of such beliefs is denoted as L'_{selq}.

From the quantification of beliefs in Section 6.3.5, the beliefs competence and experience are the beliefs directly contributing to trust. The belief honesty is a belief source for the quantification of the competence belief, thus, it is not considered as the belief directly contributing to trust.

Definition 6.14. Let us define a *social trust beliefs network*, which is a directed labeled multigraph $BSTN = (V, A, s, t, l_A)$, where the set V represents the set of agents from the social network $SN = (V, E, a, b, l_E)$, A is the set of *beliefs relations* between agents, function $s : A \to V$ assigns to each edge its source vertex (agent), function $t : A \to V$ assigns to each edge its target vertex (agent), $l_A : A \to L'_{selq} \times D \times [-1, 1]$ is the labeling function; such function associates every edge $e \in A$, $s(e) = u$, $t(e) = v$, with (1) the belief $l \in L'_{selq}$ represented by the edge e, (2) the domain $d \in D$ the belief is relevant for, and (3) the belief's value obtained as a result of belief function $\beta^l(u, v, d)$ (see Definition 6.5); $F = (D, R_D)$ is the domain hierarchy (see Definition 6.3).

All the beliefs $l \in L'_{selq}$ are quantified for every domain $d \in D$ (see Section 6.3.5) and the quantifications are stored as edges (beliefs relations) in the social trust beliefs network. The trust model used in the SoSIReČR portal is formalized as follows:

Definition 6.15. Trust model is a tuple $((S, val, w), (L_{selq}, B), F, BSTN, \tau)$, where S is the set of sources for the beliefs L_{selq}, val and w are functions quantifying the value and weight of every source $s \in S$ w.r.t. domain hierarchy $F = (D, R_D)$ (see Section 6.3.5), L_{selq} is the set of selected beliefs relevant for the SoSIReČR project, B is the set of beliefs functions β^l, $l \in L_{selq}$ (see Definition 6.5), $F = (D, R_D)$ is the domain hierarchy (see Definition 6.3), $BSTN = (V, A, s, t, l_A)$ is the social trust beliefs network, τ is the trust metric (see Definition 6.4).

Function val is used together with function w to quantify the sources from S using the domain hierarchy $F = (D, R_D)$; function B quantifies beliefs in L_{selq} using the domain hierarchy F and quantified set of sources S. The quantification of beliefs may run in several iterations, because certain beliefs may be belief sources for other beliefs – in such case, the beliefs being sources for other beliefs have to be quantified first. Based on the quantification of the beliefs directly contributing to trust, the social trust beliefs network, $BSTN = (V, A, s, t, l_A)$, is constructed. Finally, the trust metric τ operates on top of the social trust beliefs network $BSTN$ and computes the trust value $\tau(u, v, d)$ between the trustor $u \in V$ and trustee $v \in V$ w.r.t. domain $d \in D$.

The trust metric τ has to be either a completely new trust metric for the SoSIReČR trust model or it can be a certain trust metric from Section 6.4, which is leveraged to support beliefs forming the trust value and domain specificity of trust.

Based on the relevant trust metrics in the literature and their properties (see Section 6.4), we decided to use Appleseed as the implementation of τ in the SoSIReČR trust model. To use the Appleseed trust metrics, a mapping of the social trust beliefs network (Definition 6.14) to the social trust network (Definition 6.6) has to be defined, so that Appleseed can internally operate on top of the social trust network as depicted in Section 4. As a part of the mapping, beliefs relations have to be combined to the trust relations and also the domain for the trust relation has to be fixed, because Appleseed does not take into account domain specificity of trust. The social trust network, Appleseed will operate with, can be constructed when needed during the execution of the Appleseed algorithm (and only to the extent needed by the execution) and may be cached for further runs of the Appleseed algorithm. Therefore, the whole social trust network, one for every domain $d \in D$, will not be built upfront.

In the next section, we describe how the algorithm Appleseed my be adjusted to support beliefs and domain specificity of trust. We suppose that the Appleseed trust metric discussed further supports distrust as indicated in Section 6.4.3.

Leveraged Appleseed Algorithm Supporting Beliefs

To leverage the Appleseed trust metric, the list of the input parameters in Algorithm 4 has to be extended with the domain $d \in D$, for which the trust metric τ should be computed, and with the social trust beliefs network $BSTN$. On the other hand, the social trust network $STN(V, E_T, \omega)$ is removed from the input parameters and is initialized in Line 1 of Algorithm 4 with the empty set of edges E_T. Furthermore, a new line – $computeTrustRels(x, d, BSTN, STN)$ – should be added between Lines 12 and 13 of Algorithm 4, so that all outgoing trust relations for the currently processed vertex x are constructed before the energy is spread along these trust relations; the trust relations are constructed based on

Algorithm 5 Compute trust relations

```
Output: $computeTrustRels(x \in V, d \in D, BSTN(V, A, s, t, l_A), STN(V, E_T, \omega))$
1: $V_x \leftarrow \{u \in V \mid \exists e \in A, s(e) = x, t(e) = u\}$
2: for each $u \in V_x$ do
3:     if $(x, u) \notin E_T$ then
4:         $E_T \leftarrow E_T \cup (x, u)$
5:         $\omega(x, u) \leftarrow \alpha_1 \beta^{comp}(x, u, d) \cdot \alpha_2 \beta^{exp}(x, u, d)$
6:     end if
7: end for
```

the convex combination of the quantified beliefs represented as beliefs relations in the social trust beliefs network $BSTN = (V, A, s, t, l_A)$.

The function *computeTrustRels*, added to the original Appleseed algorithm, is executed as depicted in Algorithm 5. The core idea of Algorithm 5 is to map the beliefs relations to trust relations and add such trust relations to the social trust network Algorithm 4 can operate on. Line 1 of Algorithm 5 prepares the set V_x containing all the neighbors of the vertex x in the social trust beliefs network $BSTN$. Lines 2 – 7 then process these neighbors and for each such neighbor u, the trust relation is added to the social trust network in Line 4 (if it does not exist yet). Function $\omega(x, u)$ in Line 5 labels the created trust relation between agents x and u with the trust value being a convex combination of the quantified beliefs β^l directly contributing to trust, $l \in L'_{selq}$; $\alpha_1, \alpha_2 \in [0, 1]$, $\alpha_1 + \alpha_2 = 1$.

For completeness, Algorithm 6 depicts the leveraged Appleseed trust metric with the distrust support. It can be further optimized, so that it remembers for which vertices in V the trust relations were already created; for those, the function $computeTrustRels(x, d, BSTN, STN)$ may not need to be executed. Furthermore, Algorithm 6 has the social trust network $STN(V, E_T, \omega)$ always initialized with an empty set of edges E_T; this may be improved and the social trust network may be initialized from the cache if it was already computed for the same domain $d \in D$ in the past.

6.6 Summary

In this chapter, we started by introducing the SoSIReČR project and the particular problematic scenarios $S_1 - S_5$ emphasizing the importance of trust in the social network behind the SoSIReČR portal.

In Section 6.2, we described the concept of trust, the important properties of trust, such as its domain specificity, and the needs for quantifying trust based on the quantification of a set of beliefs forming trust. In Section 6.3, we detailed the concept of trusting beliefs – the important building blocks for the quantification of trust. We surveyed the trusting beliefs identified in the literature, selected the relevant trusting beliefs for the SoSIReČR project, evaluated the selection process, and sketched the sources and quantification of the subset of these beliefs. In Section 6.4, we surveyed the relevant trust metrics for estimating trust in social networks, discussed their properties and suitability for computing trust in SoSIReČR.

Section 6.5 defines a trust model for the SoSIReČR project using the def-

Algorithm 6 Leveraged Appleseed trust metric with distrust support

```
Output: $Trust_A(s \in V, in^0 \in \mathbb{R}^{\geq 0}, f \in [0,1], T_c \in \mathbb{R}^{\geq 0}, d \in D, BSTN(V, A, s, t, l_A))$
initialize$(STN(V, E_T, \omega))$
$in_0(s) \leftarrow in^0$
$trust_0(s) \leftarrow 0$
$i \leftarrow 0$
$V_0 \leftarrow \{s\}$
repeat
    $i \leftarrow i + 1$
    $V_i \leftarrow V_{i-1}$
    for each $x \in V_{i-1}$ do
        $in_i(x) \leftarrow 0$
    end for
    for each $x \in V_{i-1}$ do
        $trust_i(x) \leftarrow trust_{i-1}(x) + (1 - f) \cdot in_{i-1}(x)$
        $computeTrustRels(x, d, BSTN, STN)$
        for each $(x, u) \in E_T$ do
            if $u \notin V_i$ then
                $V_i \leftarrow V_i \cup \{u\}$
                $trust_i(u) \leftarrow 0$
                $in_i(u) \leftarrow 0$
                $E_T \leftarrow E_T \cup (u, s)$
                $\omega(u, s) \leftarrow 1$
            end if
            $w \leftarrow \omega(x, u) / \sum_{(x,u') \in E_T} \omega(x, u')$
            $in_i(u) \leftarrow in_i(u) + out(x, i) \cdot sign(\omega(x, u)) \cdot w$
        end for
    end for
    $m \leftarrow max_{y \in V_i}\{trust_i(y) - trust_{i-1}(y)\}$
until $(m \leq T_c)$
return $\{(x, \tau(s, x) = trust_i(x)) \mid x \in V_i\}$
```

inition of trust introduced in Section 6.2, sources and quantification of beliefs described in Section 6.3, and a leveraged Appleseed trust metric respecting the domain specificity of trust and supporting beliefs forming the trust value. The trust metric being leveraged was selected based on the survey of trust metrics in Section 6.4.

Relevant Author's Publications

Paper [64] describes the SoSIReČR project. The survey of the trusting beliefs in the literature, the selection of the relevant trusting beliefs for the SoSIReČR portal, and the evaluation of such selection is covered by papers [107, 101, 102]. Paper [108] covers the discussions behind the concept of trust, properties of trust, and the survey of trust metrics. Paper [110] describes the particular domain hierarchies available (apart from the ACM Computing Classification System) and discusses, how the quantification of a belief for the given domain may be derived from the quantification of the same belief for a more specific or more generic domain in the domain hierarchy.

Main Contributions

The main contributions involve:

- the survey of the trusting beliefs in the literature
- the selection of the relevant trusting beliefs for the SoSIReČR portal
- the formalization of the trust model for the SoSIReČR portal
- the comparison of the relevant trust metrics and the leveraging of the trust metric for the SoSIReČR trust model

7. Summary, Lessons Learned, and Future Work

In Section 1.5, we outlined the main contributions $C1 - C4$ of the thesis, which we recall, further describe, and justify in this chapter – every such contribution is described in its own section. For each contribution, we (1) describe the state of the art before the contribution was realized and then the current state of the art after our contribution, (2) particularize the impact of the contribution, and (3) discuss latest related activities and future work.

Finally, we also describe in Section 7.5 to which extent we managed to address Problem P7 (Trustworthy linked data consumption), outlined in Section 1.1 as the main goal to which we were aiming our thesis.

7.1 ODCleanStore (Contribution C1)

In this section, we discuss the contribution of, the general impact of, and the future work associated with the ODCleanStore tool as a whole and w.r.t. data cleansing, linking, and quality assessment not associated with a more specific contributions in Sections 7.2 and 7.3.

7.1.1 State of the Art

Before

To the best of our knowledge, when we started working on the ODCleanStore tool, there existed only two other projects with the overlapping functionality – LDIF and LDM (described in more detail in Section 3.3.1). LDM covers the data processing pipeline; however, it does not provide support for crucial transformers, such as linker, generic data quality assessor, or data fusion component. LDM does not provide any query execution module.

At the time we started working on ODCleanStore, LDIF, did not provide any way how to address data fusion and computation of the integrated quality. Furthermore, LDIF did not provide any administration interface which may be used for setting up a pipeline, monitoring the pipeline execution, debugging the pipeline, or managing transformers available on the pipeline. Lastly, LDIF did not provide any query execution module.

Now

We implemented ODCleanStore, a Linked Data management tool, which allows data cleansing, linking, quality assessment, query execution, and which is able to provide data consumers with integrated and customized views on the data, supplemented with data provenance and quality scores. The full documentation and the latest version of the tool is available at `http://sourceforge.net/p/odcleanstore`. ODCleanStore is released under an open license and is suggested to be used in any environment where the goal is to increase the efficiency of

Linked Data management or the efficiency of Linked Data consumption. The main contributions of ODCleanStore are as follows:

- a data processing pipeline for automated cleansing, linking, and quality assessment (directly addressing Problems P1, P2, and P5)
- a support for user specific pipelines, custom transformers (not just cleaners, linkers, and quality assessors), which may be easily added (supporting P1, P2, P3, and P5)
- a query execution module including data integration and data filtering modules, which are discussed in more detail in Sections 7.2 and 7.3
- query execution module provides the resulting data in various RDF and non-RDF serializations (HTML, TriG, or RDF/XML), thus, resulting data is easily used by the web applications consuming Linked data, such as the Linked Data browser Alice is using
- a prototype Linked Data browser, which is able to call output web service of ODCleanStore and provide data consumers with the possibility to browse the integrated data and associated (provenance) metadata; such browser is a part of the standard distribution of ODCleanStore and is also discussed in Section 3.2.2 (supporting P7)
- an administration interface for setting up a pipeline, monitoring the pipeline's execution, debugging the pipeline, managing the transformers available, managing policies for transformers, and managing the query execution module (supporting P1, P2, P3, P5, and P7)
- ODCleanStore as a platform for direct application of the research activities and contributions conducted in Chapters 4, 5, and 6 (supporting P1 – P7)

Currently, to the best of our knowledge, apart from LDM and LDIF, there is no other tool having significant overlap with ODCleanStore. LDM did not progress in any way, it was deployed internally in a single company and did not maintained since that time. On the other hand, LDIF starts providing basic monitoring capabilities and also provides data fusion component. Still, ODCleanStore (1) has its unique features not implemented in LDIF, (2) is applied in various domains (see Section 7.1.2), and (3) serves as a solid base for further research conducted in the direction towards trustworthy Linked Data integration and consumption.

7.1.2 Impact

ODCleanStore is used to prepare data marts (see Figure 1.4) for the `http://opendata.cz` portal; data marts are accessible via the list of available data marts at `http://linked.opendata.cz`. We also plan to provide the users of the portal with the user interface wrapping the output web service of ODCleanStore, so that users can directly query the raw data mart in ODCleanStore and browse the results in a similar way as illustrated in Section 3.2.2.

Furthermore, ODCleanStore is used to cleanse, link, and integrate public procurement data coming from various data sources, such as European portal of public contracts, TED[96], or Czech national portal of public contracts, ISVZUS[97]. Such data will appear on the list of the available data marts, i.e., at `http://linked.opendata.cz`, after clarifying the licensing issues.

We are also working on the INTLIB project[98] – the goal of that project is to create a certified methodology for mining the semantics from the specific categories of documents (such as legislation documents) and consequent processing, cleansing, searching and presenting of the obtained data as Linked Data. In case of legislation documents, the extracted semantics might represent the entities (e.g., a president, a citizen) mentioned in the collection of legislation documents and their rights and obligations described by these legislation documents. ODCleanStore is currently being tested to be used for the processing, cleansing, and searching on top of the extracted semantic information.

7.1.3 Latest Related Activities and Future Work

Together with the Semantic Web Company[99], i.e., the authors of the Linked Data Manager tool (see Section 3.3.1), we collaborate on the common ETL tool for RDF data processing, which will be based on ODCleanStore. When compared with the data processing module in ODCleanStore, such ETL tool will provide (1) its own scheduling capabilities, (2) a possibility to define not only transformers, but also extractors and loaders on the pipelines, (3) better user interface, and (4) better environment for debugging the pipelines. In that activity, we will reuse at maximum the data processing pipeline in ODCleanStore and employ the experience we gained while developing ODCleanStore. ODCleanStore will not be superseded completely, the query execution module, a core part of this thesis, will not be implemented in such ETL tool; however, the ETL tool may submit (load) the processed resulting data to the input web service of ODCleanStore, so that it can become available in the raw data mart for further data querying.

We also cooperate with the Agile Knowledge Engineering and Semantic Web (AKSW) research group at the University of Lepzig[100] and the Department of Computer Science, Systems and Communication at the University of Milan-Bicocca on the further data cleansing and quality assessment techniques. As a part of that activity, we will address the relevant quality assessment dimensions (such as accuracy, completeness, consistency, or timeliness) for the public procurement data and create new transformers – specialized quality assessors – for the data processing pipeline of ODCleanStore; such transformers will become available for users of ODCleanStore. As part of this activity, ODCleanStore will be also extended to support a vector of quality assessment scores, each for one particular quality assessment dimension. Later on, we will also try to generalize the implemented cleaners and quality assessors to other domains (apart from the public procurement one).

[96] `http://ted.europa.eu/`

[97] `http://www.isvzus.cz/`

[98] A project of the Technology Agency of the Czech Republic, project number TA02010182.

[99] `http://www.semantic-web.at/`

[100] `http://aksw.org/About.html`

The complete list of future intended features of ODCleanStore tool is provided at `http://sourceforge.net/p/odcleanstore/wiki/Future%20extensions/`.

7.2 Data Fusion in ODCleanStore (Contribution C2)

In this section, we discuss the data fusion and integrated quality computation which is a part of the data integration component in ODCleanStore. Data fusion and integrated quality computation is described in detail in Chapter 4.

7.2.1 State of the Art

Before

To the best of our knowledge, there was no data fusion tool for RDF data. Section 4.9.2 presents some of the non-RDF data fusion tools, which were available, and their comparison with the data fusion module in ODCleanStore.

Now

We described and implemented in ODCleanStore a data fusion algorithm, being a crucial part of the data integration component in ODCleanStore. The data fusion algorithm helps to create integrated views for data consumers by solving the data conflicts. Furthermore, it also computes the quality of the integrated data and supplements the resulting integrated data with justifications of the computed quality and information about the source the integrated data originates from. Data fusion may be also customized by data consumers.

The novel customizable data fusion algorithm implemented in ODCleanStore presents one of the main contributions of the thesis. The detailed contributions of the data fusion algorithm are as follows:

- The data fusion algorithm supports the typical conflict handling strategies [101].

- Every resulting integrated quad is supplemented with (1) the integrated quality score and (2) source graphs contributing to the computation of the integrated (object) value of the quad.

- The data fusion algorithm is customizable – conflict handling policies may be customized on the global and per predicate level, a multivalue flag (see Section 4.5.2) may be set on the global and per predicate level, a data fusion error strategy may be selected.

Apart from the data fusion component in ODCleanStore, another Linked Data fusion software was developed in parallel – Sieve [127]; Sieve adds quality assessment and data fusion capabilities to the LDIF framework. Sieve offers functionality similar to our data fusion component; however, the purpose of Sieve

[101] Except of the conflict handling strategy AVOID, which was not implemented in the current version of ODCleanStore.

in LDIF is different - it fuses data while being stored to the database and not during execution of queries, thus, in the consequent query executions, it does not provide any data fusion customization. On the other hand, the data fusion algorithm in ODCleanStore must be fast enough to return the result in a reasonable time during the consumer's query; as we show in Section 4.7.2, our data fusion algorithm can accomplish that. Furthermore, whereas Sieve computes data quality only for whole named graphs, ODCleanStore provides quality estimation of each statement resulting from the data fusion.

7.2.2 Impact

The data fusion algorithm was implemented in ODCleanStore and provided as a part of ODCleanStore. The data fusion algorithm allows data consumers to customize how the data is integrated. The integrated data is supplemented with justified quality scores and provenance metadata, thus, the data fusion algorithm contributes significantly to the trustworthy Linked Data consumption.

7.2.3 Latest Related Activities and Future Work

Currently, our team is responsible for adjusting the data fusion component, so that it can be used not only during the query execution, but also when ODCleanStore is preparing the specialized data marts. Furthermore, such adjusted data fusion component should be also pluggable as a new type of transformer on the data processing pipeline of ODCleanStore, thus, covering the same functionality as Sieve in LDIF as well. The details about these efforts are available at `http://github.com/mifeet/cr-batch`.

The data fusion algorithm limits the expressivity of queries data consumers can submit to the query execution module of ODCleanStore. The future work will investigate the ways how the data fusion algorithm may work in case of complex SPARQL queries and how the descriptive and provenance metadata of the resulting integrated quads should be computed and provided together with the integrated data.

7.3 Data Provenance (Contribution C3)

In Chapter 5, we described the W3P provenance model for the Web, being a major contribution of the thesis. In Section 5.9, we described the provenance requirements, consumers may have on the consumed Linked Data, and how these requirements can be enforced by the data filtering module of ODCleanStore. We also emphasized in Section 5.8 the role of the W3P provenance model in ODCleanStore.

7.3.1 State of the Art

Before

The focus of provenance research papers was mainly on the domain of databases and scientific workflow management systems. There was no provenance model for the Web, except for the one proposed in [83]. However, such model lacked enough expressivity for expressing relations between artifacts, processes, and agents. Furthermore, important requirements on the provenance model, such as coverage of social descriptors, licensing, change tracking, and spatiality, were not covered by that model.

Now

We delivered the W3P provenance model for the Web, which is constructed w.r.t. the Requirements 1 – 17 elaborated in Section 5.5.3. The proposed W3P provenance model is built over core Linked Data standards. It is independent of granularity, allowing users to describe the provenance of different web artifacts including data, documents, and datasets. The coverage of social provenance is an important feature of the W3P provenance model, allowing W3P users to track trust and reputation of entities and artifacts. W3P is reusing vocabularies being available by that time and defines a W3PO ontology for holding (1) new terms not covered sufficiently by other vocabularies and (2) mappings between the reused vocabularies. The W3P provenance model should be used for expressing and tracking provenance of the data on the Web.

Since the time we published the W3P provenance model in [60], certain other efforts have appeared. The most important efforts are the efforts of the W3C Provenance Incubator Group and the consequent W3C Provenance Group; we describe these efforts and compare them with our results in Section 7.3.2.

7.3.2 W3C Provenance Activities

W3C Provenance Incubator Group

The goal of the W3C Provenance Incubator Group[102] was to provide a roadmap for covering provenance on the Web. Our paper [60], being a major source for Chapter 5 describing the W3P provenance model, was submitted to the journal in December 2009. The W3C Provenance Incubator Group was established shortly before that but the first outputs were conducted in 2010. We joined the W3C Provenance Incubator Group, so that we were able to promote ideas of the W3P provenance model there.

The W3C Provenance Incubator Group initiated broader discussion on the use cases of tracking provenance data on the Web and defined requirements (called dimensions in their case and in the rest of this section) for the data provenance from the users' perspective [103].

Table 7.1 summarizes the dimensions they proposed and also describes how these dimensions are addressed by our requirements for the W3P provenance model discussed in Section 5.5.3. In general, the W3P provenance requirements on generality (Req. 5), integrity mechanisms (Req. 10), identity warranties (Req. 11), and query expressivity and navigability (Req. 16) address many W3C Provenance Incubator Group's dimensions summarized in Table 7.1. In case of

[102] `http://www.w3.org/2005/Incubator/prov/`

[103] `http://www.w3.org/2005/Incubator/prov/wiki/Provenance_Dimensions`, see `http://www.w3.org/2005/Incubator/prov/wiki/User_Requirements` for a detailed discussion

the dimension *attribution*, a user should know (1) which agent contributed to the artifact in question, (2) which agent executed/endorsed a particular artifact or process, (3) the roles of that agents and (4) the integrity and identity warranties supporting the credibility of that endorsement; furthermore, the user should be able to effectively query the provenance information. Similar requirements correspond with the dimensions *accountability* and *trust*; trust is typically based on attribution – trustor has to know who did what with the particular artifact. In case of the dimensions *justification* and *entailment*, descriptions about processes, artifacts, and agents are crucial; furthermore, the dimension *justification* requires fine-grained provenance and temporal information available, the dimension *entailment* requires well defined logical model grounded in semantics. The dimension *understandability* is supported by the requirement for the fine-grained and coarse-grained provenance information, well defined logical model grounded in semantics, query expressivity and navigation, and extensibility of the model (to enable creation of, e.g., new roles to further increase the understandability). The dimension *interoperability* is supported by the requirement for the interoperability and generality of the proposed model and also further supported by the extensibility requirement (the provenance model should be extendable with new terms to incorporate new domain specific ontologies).

The W3C Provenance Incubator Group also mentioned provenance concepts we outlined in Table 5.2. Apart from the provenance concepts motivating the W3P provenance model in Chapter 5, they argue for the concept of *recipe*, which further describes how the artifact was created, e.g., a recipe may hold the XSL template which generated the artifact. From the W3P model's perspective, a recipe is yet another artifact. Similarly, the concepts *collections of entities*, and *views/accounts* containing other provenance entities are comprehended in W3P as artifacts. W3C Provenance Incubator Group's provenance concepts *derivation*, *generation*, *use*, *ordering of processes*, *participation*, and *control* are all represented by proper relations between the concepts of agent, artifact, and process in W3P. W3C Provenance Incubator Group's provenance concepts *location*, *version*, *provenance container*, *role*, and *time* are directly supported by W3P provenance concepts.

The goal of the W3C Provenance Incubator Group was to provide a roadmap for covering provenance on the Web. The W3C Provenance Incubator Group covered different communities with interests in the provenance space and its final output is an important guideline for future work on the provenance area. On the other hand, the main objective of Chapter 5 was to propose a provenance model for the Web (W3P provenance model), defined over a set of requirements and maximizing the reuse and coverage of existing vocabularies.

W3C Provenance Group

The W3P Provenance Group (started in 2011) follows the results of the W3C Provenance Incubator Group. The W3C Provenance Group defines the generic provenance model for the Web, which is then expressed as an ontology PROV-O [156]; no terms are reused from other ontologies. As depicted further, lots of the provenance terms defined in PROV-O are semantically similar to the terms used by the W3P provenance model. To some extent, the ideas behind creation of the W3P provenance model are also reflected in the PROV-O.

Table 7.1: Provenance dimensions, their description and how they are supported by our requirements for the W3P provenance model (relevant W3P provenance concepts are introduced in the brackets)

Dimension	**Description**	**W3P Requirement**
Object	The artifact that a provenance statement is about.	6 (Artifact), 5
Attribution	The sources or agents that contributed to creation of an artifact in question.	6 (Agent, Artifact, Role) 15, 10, 11, 5, 16
Process	The activities (or steps) that were carried out to generate or access the artifact at hand.	6 (Process), 5
Evolution and versioning	Records of changes to an artifact over time and what entities and processes were associated with those changes.	13, 8, 6 (Agent, Artifact, Process)
Justification for decisions	Documentation recording why and how a particular decision is made.	6, 4, 8, 5, 10, 11, 16, 15
Entailment	Explanations showing how facts were derived from other facts.	6, 3, 5, 10, 11, 16
Understandability	How to enable the end user consumption of provenance.	3, 4, 2, 16
Interoperability	Combining provenance produced by multiple different systems.	1, 2, 3, 5
Comparison	Comparing artifacts through their provenance.	5, 6 (Artifact), 16
Accountability	Using provenance to assign credit or blame. Accountability requires that the users can rely on the provenance record and authenticate its sources.	6 (Artifact, Agent, Activity, Role), 10, 11, 5, 16, 9, 15, 8, 7
Trust	Using provenance to make trust judgments.	6 (Agent, Artifact, Role) 10, 11, 5, 16, 9, 15, 8, 7, 14
Imperfections	Dealing with imperfections in provenance records.	6, 3, 5, 16
Debugging	Using provenance to detect bugs or failures of processes.	6, 3, 5, 16

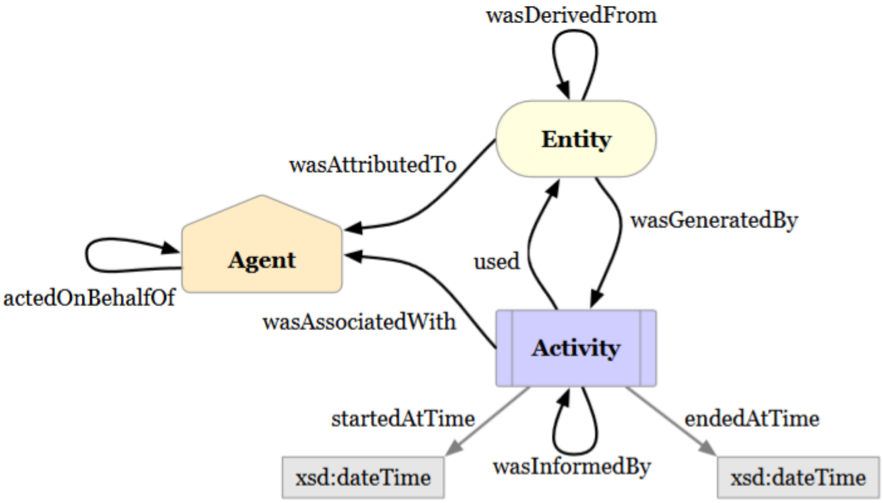


Figure 7.1: Core concepts of PROV-O ontology (Source: [156])

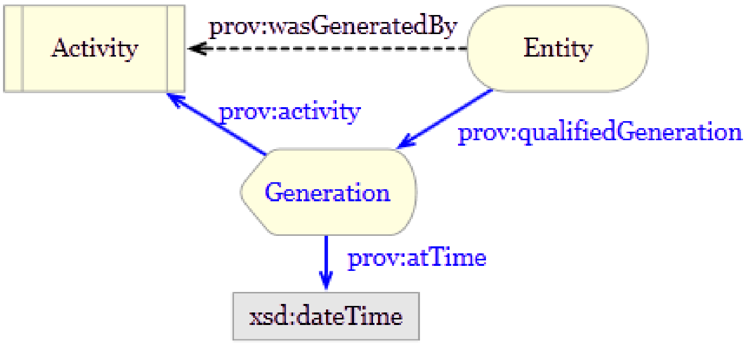


Figure 7.2: Qualified usage of the property `wasGeneratedBy` in PROV-O (Source: [156])

PROV-O defines basic classes and properties to hold entities (artifacts in W3P), activities (processes in W3P), agents, relations between them, and basic time expressions, as depicted in Figure 7.1. Furthermore, PROV-O defines ways how to deal with hierarchies of entities and agents, it defines a couple of types of agents, and also further predicates covering the relations depicted in Figure 7.1, e.g., PROV-O defines the predicate `hadPrimarySource` as a special case of `opmv:wasDerivedFrom`. Our provenance model, W3P, proposed in Chapter 5, covers all these concepts.

PROV-O also defines qualified versions of the predicates in Figure 7.1 to express additional attributes of binary relations, e.g., Figure 7.2 shows how the basic property `wasGeneratedBy`, expressing that a certain entity was generated by a certain activity, may be expanded, “qualified”, by introducing the class `Generation` about which further facts can be expressed, such as the time of the entity’s generation. Creating an auxiliary class (e.g., the class `Generation`) is a common approach to deal with N-ary relations in RDF data model[104]. Our W3P provenance model supports binary predicates and relation classes in a similar way.

[104] `http://www.w3.org/TR/swbp-n-aryRelations/`

7.3.3 Impact

Our provenance model W3P was one of the first provenance models for the Web. It motivated other provenance efforts, such as the papers [114, 152, 167], which directly cite our paper [60] describing the W3P provenance model. Furthermore, the ideas behind the creation of the W3P provenance model are confirmed by the currently ongoing standardization of the provenance model proposed by the W3C Provenance Group (see Section 7.3.2).

W3P provenance model is suggested to be used for expressing provenance of the data submitted to ODCleanStore. As a result, the query execution module of ODCleanStore, providing the integrated resulting data on the consumer's queries, can supplement the resulting data with provenance information according to W3P provenance model. Since the W3P provenance model satisfies the requirements outlined in Section 5.5.3, it brings better experience to the data consumers, e.g., in terms of the provenance information navigability. Furthermore, W3P provenance model significantly influences the efficiency of the provenance policies' enforcement in the data filtering component of ODCleanStore (see Section 5.9); as a result, W3P provenance model contributes significantly to the goal of providing trustworthy Linked Data to the data consumers.

7.3.4 Latest Related Activities and Future Work

Since the provenance model PROV-O will be a W3C standard and, thus, will become a de-facto standard for expressing and tracking provenance on the Web, we should align W3P provenance model with PROV-O. Other provenance models, such as the one defined in [83], follow the same direction. Regarding the core concepts in W3P, instead of using OPM, we should start using the core PROV-O classes `Agent`, `Entity`, and `Activity`, thus, replacing OPMV classes `opmv:Agent`, `opmv:Artifact`, and `opmv:Process`. PROV-O also defines its own terms for expressing the concepts of role, spatial information, and temporal information; it partially covers the social descriptors concept. A detailed alignment of W3P provenance model is a future work; in general, we should prefer the use of PROV-O terms if possible, because it will be the W3C standard.

W3P in ODCleanStore

Currently, provenance information behind the data graphs inserted to ODCleanStore has to be prepared manually in an (RDF) editor for every data feed submitted to ODCleanStore. Future work involves creation of tools supporting the automated creation of provenance information as the data is produced by the external applications, so that the provenance information can be automatically provided to ODCleanStore. Moreover, if the incoming data graphs do not contain provenance information expressed according to the W3P provenance model, it should be mapped (if possible) to the W3P provenance model as the data is processed by the data processing pipelines in ODCleanStore.

Lastly, every transformer on the data processing pipeline of ODCleanStore should describe its transforming activities on the processed data and store such information in the provenance graphs associated with the processed data feeds.

Provenance Policies

Future work includes finishing the implementation of the provenance data filtering module in ODCleanStore according to Section 5.9 and its release with the next versions of ODCleanStore. Furthermore, the parameters of the URI and keyword queries should be extended to accommodate (1) the list of provenance policies the data consumer would like to apply to the data, (2) the constraints F customizing the behavior of the data filtering algorithm and (3) the desired provenance score threshold $\kappa \in [0, 1]$ (see Section 5.9).

Moreover, before the provenance policies are applied as a part of the data filtering component in ODCleanStore, reasoning should be performed on top of these policies to increase the efficiency of the provenance policies' enforcement. To illustrate that, suppose that a policy $p = (\mathit{cond}, \mathit{weight})$ contains property X in a certain triple pattern within the condition *cond*; further, a provenance graph g contains a triple with property Y being a more specific property than X. As a result, reasoning should realize that Y is a more specific version of X and, thus, the data filtering module should consider the application of policy p to the provenance graph g. The future work is also to measure the performance of the provenance policies' enforcement with such reasoning capabilities enabled.

Linked Data Browser Supporting Provenance Policies

Future work also involves evaluation of the provenance policies' usability. To that end, we will extend the prototype Linked Data browser we implemented (see Section 3.2.2), so that the data consumer can not only examine the data and provenance metadata returned by the query execution module of ODCleanStore, but he can also define new provenance requirements.

The general idea of such extension is as follows. Every data consumer is working in his private data space containing his provenance requirements in the form of provenance policies. The new provenance policy may be created in a user friendly way by browsing the provenance data and selecting one or more provenance concepts (e.g., the predicate `w3po:wasPublishedBy` and the corresponding object value) which will then form the condition of the new provenance policy.

When the browser receives integrated view on the data returned as a result of a certain query, the consumer can not only browse the returned data and metadata, but also select one or more *relevant*[105] *provenance policies* defined previously in his private workspace, and, as a result, further filter the resulting integrated view; the list of relevant and suggested policies will be automatically preselected from the consumer's private space based on the query result.

After selecting one or more suggested relevant provenance policies, the data filtering and data integration components are executed once more for the same query and the data is further refined according to the consumer's latest provenance requirements. In that way, the consumer can iteratively refine the resulting data for the given query by iteratively adjusting the provenance policies.

[105] A provenance policy is relevant for the data graph contributing to the integrated view, if such policy can be successfully applied to that graph.

7.4 Trust Model (Contribution C4)

In Chapter 6, we described the trust model for the SoSIReČR portal.

7.4.1 State of the Art

Before

There are lots of trust models, e.g. [76, 171, 148, 123], comprehending trust between two agents as a "black box" and indivisible concept. Since trust is so complex concept [92], and the trust metrics of these trust models typically assume transitivity of trust, semantics of such transitive and domain independent trust computed by these trust metrics is ambiguous. Paper [85] illustrates the problem of "black box" trust and serves as the main motivation for the trust model for the SoSIReČR portal.

Now

The trust model for the SoSIReČR portal addresses the problem of "black box" trust by surveying the trusting beliefs in the literature relevant for the SoSIReČR portal and employing a trust metric, which computes the trust value based on the quantification of these beliefs; trust model for SoSIReČR also respects the domain specificity of trust.

7.4.2 Impact

Trust model for the SoSIReČR portal guides the implementation of the trust module in SoSIReČR. The trust metric of the SoSIReČR trust model computing trust values among members of the informatics community is used as a part of the portal's query service executing queries $Q_1 - Q_5$ introduced in Section 6.1; the relevancy of the results on these queries may be resorted w.r.t. the trust values provided by that trust metric. The SoSIReČR portal is expected to attract thousands of members of the Czech informatics community in the next years. Those members will rely on the trust module implemented according to the trust model in Chapter 6.

7.4.3 Latest Related Activities and Future Work

Our future work is divided into two areas discussed in two separate subsections: (1) work related to improvements and further implementation of the SoSIReČR trust model in the SoSIReČR trust module and (2) application of the proposed trust model to social networks of data consumers and publishers using the OD-CleanStore tool.

SoSIReČR Trust Model

In this case the future work mainly focuses on finishing the implementation of the proposed sources' and beliefs' quantification (outlined in Section 6.3.5) for the SoSIReČR trust module. Currently, honesty belief's sources, an explicit source for competence – the value on the axis of a professional profile, and a source for the generic experience (w.r.t. the root domain) are implemented. The future work also includes description of the quantification of the belief willingness and how the belief reputation should be incorporated to the trust model to shape the computed trust value.

Furthermore, weights of the particular sources contributing to the quantification of the identified beliefs should be experimentally set. Moreover, the weights of the beliefs directly contributing to the trust value according to Definition 6.13 should be evaluated and experimentally set. During these experiments, we should take into account that weights of the beliefs forming trust may differ for different types of queries $Q_1 - Q_5$.

The future work also includes adjustments to the trust metric. Since the number of outgoing edges in the social trust network in Definition 6.6 may vary substantially (e.g., a person can have just one colleague, or a person can belong to tens of groups), we already did preliminary experiments with normalizing the energy spread along the trust relations in Appleseed without respect of the sum of weights of the outgoing edges. To realize that, if $\sum_{(x,u')\in E_T} \omega(x,u') < M$ (see Line 21 of Algorithm 4), then the suggested approach is to replace Line 21 with a new line $w \leftarrow \omega(x,u)/M$, where $M \in \mathbb{R}^+$, e.g., $M = 20$. As a result of such adjustment, lots of energy may disappear from the system, which does not affect the correctness of the trust metric, but it may influence the ability to provide reasonable resulting values in the range $[-1 \pm \epsilon, 1 \pm \epsilon]$. To verify this idea and set the appropriate value for M more experiments are needed.

Social Networks Behind ODCleanStore

In this case the future work is to (1) create a social network including the data publishers submitting data to ODCleanStore and data consumers consuming data from ODCleanStore and (2) define a proper trust model, inspired by the trust model for SoSIReČR, which will lead to the construction of a social trust beliefs network containing these data publishers and consumers.

Such social trust beliefs network will be used for establishing trustworthiness of data publishers as observed by data consumers; such trustworthiness of the publishers influences the trustworthiness of the data originating from these publishers [70]. As a result, ODCleanStore can provide further justifications about the trustworthiness of the data presented to data consumers, filter the processed and provided data based on its trustworthiness, or adjust the data integration process w.r.t. data trustworthiness.

Furthermore, such social trust beliefs network will be used to share provenance policies (described in Section 5.9) among trustworthy agents in that network. As depicted in Section 7.5, such social trust beliefs network may be also used to share other kinds of requirements (policies), which may enforce data quality or trustworthy agents requirements on the consumed data.

7.5 Trustworthy Linked Data Consumption

Let us recall Alice, an investigative journalist from Scenario 1.1, and Problem P7 we outlined in Section 1.1. The goal of the trustworthy Linked Data consumption was that Alice may express various requirements on the consumed data and these requirements will be automatically enforced as the data is prepared for her. In particular, she may require only trustworthy data – data with certain quality, data with certain provenance records behind (e.g., coming only from certain sources), or data published only by trustworthy agents.

State of the Art

Current state of the art, after considering all the contributions of the thesis, is as follows. ODCleanStore helps to ensure that Alice will be provided with data already being cleansed (Problem P2), linked (P2), transformed (P1), quality assessed (P5), trustworthy (P4, P5) and integrated (P3). The evidence of trustworthiness of the resulting data provided to Alice is supported by the justified quality scores (Chapter 4) and data provenance (Chapter 5) of the integrated data. Alice may customize the data integration process by specifying various requirements on the data fusion algorithm and these are enforced in ODCleanStore (see Section 4.8). Alice may also browse the integrated data in the raw data mart by using prototype Linked Data browser, which is available with the current distribution of ODCleanStore.

After finishing the implementation of provenance policies' enforcement in ODCleanStore, ODCleanstore will be able to automatically enforce Alice's requirements on the provenance data, expressed as provenance policies, by applying these provenance policies as part of the data filtering module (Section 5.9). Alice will be also provided with an user interface, where she can create and manage her provenance policies as sketched in Section 7.3.4.

Policies as a General Approach to Address Consumers' Requirements

The concept of policies and their application as part of the data filtering module in ODCleanStore (see Figure 3.1) is a promising approach to realize the vision of the trustworthy Linked Data consumption. The data filtering component should be extended in the future to support enforcement of other kinds of requirements shaping the consumed data. For example, w.r.t. the quality score, certain policies should allow data consumers to address the following requirements: (1) different consumers may have different thresholds for considering the given data as having enough quality score for the task at their hands and (2) different consumers may prioritize different dimensions of the information quality; e.g., accuracy of the data graph may be more important than its completeness.

To reach the vision of trustworthy Linked Data consumption described in Problem P7 in Section 1.1, apart from the definition of policies expressing the desired consumers' requirements, it is equally important to provide the graphical user interface – a browser of the consumed data, associated provenance information, quality scores, etc. – where the data consumer can browse the evidences for the data trustworthiness and specify his requirements on the data. The idea of a browser for provenance policies mentioned in Section 7.3.4 should be taken as a starting point.

Bibliography

[1] J. E. Alexander and M. A. Tate. *Web Wisdom; How to Evaluate and Create Information Quality on the Web.* L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1999.

[2] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain, 2009. CEUR-WS.org.

[3] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[4] E. W. Anderson, J. P. Ahrens, K. Heitmann, S. Habib, and C. T. Silva. Provenance in Comparative Analysis: A Study in Cosmology. *Computing in Science and Engg.*, 10(3):30–37, May 2008.

[5] D. W. Archer, L. M. L. Delcambre, and D. Maier. A Framework for Fine-grained Data Integration and Curation, with Provenance, in a Dataspace. In J. Cheney, editor, *TAPP'09: First workshop on on Theory and practice of provenance*, San Francisco, CA, Feb. 2009. USENIX Association.

[6] D. Artz and Y. Gil. A Survey of Trust in Computer Science and the Semantic Web. *Web Semant.*, 5(2):58–71, 2007.

[7] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino. Methodologies for Data Quality Assessment and Improvement. *ACM Comput. Surv.*, 41(3):1–52, 2009.

[8] C. Batini and M. Scannapieco. *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[9] D. Beckett. RDF/XML Syntax Specification (Revised), February 2004.

[10] D. Beckett and T. Berners-Lee. Turtle – Terse RDF Triple Language. W3C team submission, World Wide Web Consortium (W3C), Jan. 2008.

[11] T. Berners-Lee. Notation 3, 1998. `http://www.w3.org/DesignIssues/Notation3.html`, Retrieved 07/03/2013.

[12] T. Berners-Lee. RFC 2396: Uniform Resource Identifiers (URI). Technical report, MIT, 1998.

[13] T. Beth, M. Borcherding, and B. Klein. Valuation of Trust in Open Networks. In *Proc. 3rd European Symposium on Research in Computer Security – ESORICS '94*, pages 3–18. Springer-Verlag, 1994.

[14] P. V. Biron and A. Malhotra. XML Schema Part 2: Datatypes Second Edition, W3C Recommendation. October 2004.

[15] C. Bizer. Semantic Web Publishing Vocabulary (SWP) User Manual. Technical report, Freie Universitat Berlin, 2006. http://wifo5-03.informatik.uni-mannheim.de/bizer/wiqa/swp/SWP-UserManual.pdf, Retrieved 07/03/2013.

[16] C. Bizer. Quality-Driven Information Filtering in the Context of Web-Based Information Systems. Dissertation, 2007. http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/DisertationChrisBizer.pdf, Retrieved 07/03/2013.

[17] C. Bizer and R. Cyganiak. The TriG Syntax. Technical report, FU Berlin, 2007. http://www.wiwiss.fu-berlin.de/suhl/bizer/TriG/Spec/TriG-20070730/, Retrieved 07/03/2013.

[18] C. Bizer and R. Cyganiak. Quality-driven Information Filtering Using the WIQA Policy Framework. *Web Semantics*, 7(1):1–10, 2009.

[19] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1 – 22, 2009.

[20] C. Bizer and R. Oldakowski. Using Context- and Content-based Trust Policies on the Semantic Web. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt. '04, pages 228–229, New York, NY, USA, 2004. ACM.

[21] C. Bizer and A. Schultz. The R2R Framework: Publishing and Discovering Mappings on the Web. In *Proceedings of the First International Workshop on Consuming Linked Data*, Shanghai, China, 2010. CEUR-WS.org.

[22] C. Bizer, J. Volz, G. Kobilarov, and M. Gaedke. Silk - A Link Discovery Framework for the Web of Data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOV)*, Madrid, Spain, April 2009. CEUR-WS.org.

[23] J. Bleiholder and F. Naumann. Data fusion. *ACM Comput. Surv.*, 41(1):1:1–1:41, Jan. 2009.

[24] U. Bojars and J. G. Breslin. SIOC Core Ontology Specification. Technical report, 2010. http://sioc-project.org/ontology, Retrieved 07/03/2013.

[25] P. Bonatti and D. Olmedilla. Driving and Monitoring Provisional Trust Negotiation with Metapolicies. In *POLICY '05: Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 14–23, Washington, DC, USA, 2005. IEEE Computer Society.

[26] P. Bonhard and A. M. Sasse. "I thought it was terrible and everyone else loved it" — A New Perspective for Effective Recommender System Design. In T. McEwan, J. Gulliksen, and D. Benyon, editors, *People and Computers XIX — The Bigger Picture*, pages 251–265. Springer London, 2006.

[27] R. Bose and J. Frew. Lineage Retrieval for Scientific Data Processing: a Survey. *ACM Comput. Surv.*, 37(1):1–28, 2005.

[28] U. Braun, A. Shinnar, and M. Seltzer. Securing Provenance. In *HOTSEC'08: Proceedings of the 3rd conference on Hot topics in security*, pages 1–5, Berkeley, CA, USA, 2008. USENIX Association.

[29] J. G. Breslin, U. Bojars, B. Aleman-meza, H. Boley, L. J. Nixon, A. Polleres, and A. V. Zhdanova. Finding Experts using Internet-based Discussions in Online Communities and Associated Social Networks. In *First International ExpertFinder Workshop*, Berlin, Germany, 2007.

[30] D. Brickley. Basic Geo (WGS84 lat/long) Vocabulary. W3C Semantic Web Interest Group – Informal collaboration, 2003. `http://www.w3.org/2003/01/geo`, Retrieved 07/03/2013.

[31] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Technical report, 2 2004. `http://www.w3.org/TR/2004/REC-rdf-schema-20040210/`, Retrieved 07/03/2013.

[32] D. Brickley and L. Miller. FOAF Vocabulary Specification 0.98. Namespace Document, 2010. `http://xmlns.com/foaf/spec/`, Retrieved 07/03/2013.

[33] P. Buneman, A. Chapman, J. Cheney, and S. Vansummeren. A Provenance Model for Manually Curated Data. In L. Moreau and I. Foster, editors, *Proceedings of the International Provenance and Annotation Workshop 2006 (IPAW'2006)*, volume 4145, pages 162–170. Springer, 2006.

[34] P. Buneman, S. Khanna, and W. C. Tan. Data Provenance: Some Basic Issues. In *Foundations of Software Technology and Theoretical Computer Science, 20th Conference, FST TCS*, volume 1974 of *Lecture Notes in Computer Science*, pages 87–93. Springer, 2000.

[35] P. Buneman, S. Khanna, and W.-C. Tan. Why and Where: A Characterization of Data Provenance. In *Proceedings of the 8th International Conference on Database Theorie (ICDT)*, London, United Kingdom, 2001. Springer.

[36] P. Buneman and W.-C. Tan. Provenance in Databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 1171–1173, New York, NY, USA, 2007. ACM.

[37] S. Cantor, J. Kemp, R. Philpott, and E. Maler. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2. 0. OASIS, 2005. `http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf`, Retrieved 07/03/2013.

[38] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, Provenance and Trust. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2005. ACM.

[39] J. Cheney, L. Chiticariu, and W.-C. Tan. Provenance in Databases: Why, How, and Where. *Found. Trends databases*, 1(4):379–474, 2009.

[40] B. Christianson and W. S. Harbison. Why Isn't Trust Transitive? In *Proceedings of the International Workshop on Security Protocols*, pages 171–176, London, UK, 1997. Springer-Verlag.

[41] C. Commons. Describing Copyright in RDF. `http://creativecommons.org/ns#`, Retrieved 07/03/2013.

[42] S. M. S. d. Cruz, M. L. M. Campos, and M. Mattoso. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. In *Proceedings of the 2009 Congress on Services*, SERVICES '09, pages 259–266, Washington, DC, USA, 2009. IEEE Computer Society.

[43] R. Cyganiak, J. Zhao, K. Alexander, and M. Hausenblas. Vocabulary of Interlinked Datasets (VoID). DERI Vocabularies, 2011. `http://vocab.deri.ie/void/`, Retrieved 07/03/2013.

[44] P. P. da Silva, D. L. Mcguinness, and R. Fikes. A Proof Markup Language for Semantic Web Services. *Inf. Syst.*, 31(4):381–395, June 2006.

[45] P. P. da Silva, D. L. McGuinness, N. D. Rio, and L. Ding. Inference Web in Action: Lightweight Use of the Proof Markup Language. In *Proceedings of the 7th International Semantic Web Conference (ISWC)*, pages 847–860, Karlsruhe, Germany, 2008. Springer.

[46] S. B. Davidson, S. C. Boulakia, A. Eyal, B. Ludäscher, T. M. McPhillips, S. Bowers, M. K. Anand, and J. Freire. Provenance in Scientific Workflow Systems. *IEEE Data Eng. Bull.*, 30(4):44–50, 2007.

[47] S. B. Davidson and J. Freire. Provenance and Scientific Workflows: Challenges and Opportunities. In *SIGMOD Conference*, pages 1345–1350, 2008.

[48] U. Dayal. Query Processing in a Multidatabase System. In *Query Processing in Database Systems*, pages 81–108. Springer, 1985.

[49] DCMI Usage Board. DCMI Metadata Terms. DCMI Recommendation, 2012. `http://dublincore.org/documents/dcmi-terms/`, Retrieved 07/03/2013.

[50] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-Science: An Overview of Workflow System Features and Capabilities. *Future Gener. Comput. Syst.*, 25(5):528–540, May 2009.

[51] L. G. Demichiel. *Performing Database Operations over Mismatched Domains*. PhD thesis, Stanford, CA, USA, 1989. UMI Order No: GAX89-25855.

[52] L. Ding, T. Finin, Y. Peng, P. P. da Silva, and D. L. McGuinness. Tracking RDF Graph Provenance using RDF Molecules. In *Proceedings of the 4th International Semantic Web Conference*, Galway, Ireland, November 2005. Springer.

[53] A. Dolgert, L. Gibbons, C. D. Jones, V. Kuznetsov, M. Riedewald, D. Riley, G. J. Sharp, and P. Wittich. Provenance in High-Energy Physics Workflows. *Computing in Science and Engineering*, 10(3):22–29, 2008.

[54] J. Dozier and J. Frew. Computational provenance in hydrologic science: a snow mapping example. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1890):1021–1033, 2009.

[55] P. D. Eagan and Ventura. Enhancing Value of Environmental Data: Data Lineage Reporting. *Journal of Environmental Engineering*, 119(1):5–16, 2007.

[56] D. J. Essin. Patterns of Trust and Policy. In *Proceedings of the 1997 workshop on New security paradigms*, NSPW '97, pages 38–47, New York, USA, 1997. ACM.

[57] R. Falcone and C. Castelfranchi. *Trust and Deception in Virtual Societies*, chapter Social Trust: A Cognitive Approach, pages 55–90. Kluwer Academic Publishers, 2001.

[58] L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404.

[59] S. M. Freire, J. and L. Moreau. Second provenance challenge, 2007. `http://twiki.ipaw.info/bin/view/Challenge/`, Retrieved 07/03/2013.

[60] A. Freitas, T. Knap, S. O'Riain, and E. Curry. W3P: Building an OPM based provenance model for the Web. *Future Generation Comp. Syst.*, 27(6):766–774, 2011.

[61] J. Futrelle. Harvesting rdf triples. In L. Moreau and I. Foster, editors, *Proceedings of the International Provenance and Annotation Workshop 2006 (IPAW'2006)*, volume 4145, pages 64–72. Springer, 2006.

[62] J. Futrelle and J. D. Myers. Tracking Provenance Semantics in Heterogeneous Execution Systems. *Concurrency and Computation: Practice and Experience*, 20(5):555–564, 2008.

[63] L. M. R. Gadelha and M. Mattoso. Kairos: An Architecture for Securing Authorship and Temporal Information of Provenance Data in Grid Enabled Workflow Management Systems. In *Proceedings of International Conference on e-Science and Grid Computing*, pages 597–602, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[64] J. Galgonek, T. Knap, M. Kruliš, and M. Nečaský. SMILE - A Framework for Semantic Applications. In *Proceedings of OTM 2010 Workshops*, pages 53–54, Crete, Greece, 2010. Springer.

[65] R. García, R. Gil, I. Gallego, and J. Delgado. Formalising ODRL Semantics using Web Ontologies. In R. Iannella, S. Guth, and C. Serrao, editors, *Open Digital Rights Language Workshop, ODRL'2005*, pages 33–42, Lisbon, Portugal, 2005. ADETTI.

[66] S. H. Garlik, A. Seaborne, and E. Prud'hommeaux. SPARQL 1.1 Query Language. W3C Proposed Recommendation, 2012. `http://www.w3.org/TR/sparql11-query/`, Retrieved 07/03/2013.

[67] P. Gearon, A. Passant, and A. Polleres. SPARQL 1.1 Update. Technical report, W3C, 2012. Published online on November 8th, 2012 at `http://www.w3.org/TR/2012/PR-sparql11-update-20121108/`, Retrieved 07/03/2013.

[68] A. Gehani and U. Lindqvist. VEIL: A System for Certifying Video Provenance. In *Proceedings of the Ninth IEEE International Symposium on Multimedia*, ISM '07, pages 263–272, Washington, DC, USA, 2007. IEEE Computer Society.

[69] E. Gerck. Toward Real-World Models of Trust: Reliance on Received Information. Basil Blackwell, Oxford, 1990. `http://www.safevote.com/papers/trustdef.htm`, Retrieved 07/03/2013.

[70] Y. Gil and D. Artz. Towards Content Trust of Web Resources. *Web Semant.*, 5(4):227–239, 2007.

[71] Y. Gil, J. Cheney, P. Groth, O. Hartig, S. Miles, L. Moreau, and P. P. da Silva. Provenance XG Final Report. W3C Incubator Report, 08 December 2010. `http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/`, Retrieved 07/03/2013.

[72] J. Golbeck. Trust on the World Wide Web: A Survey. *Found. Trends Web Sci.*, 1(2):131–197, 2006.

[73] J. Golbeck. Trust and Nuanced Profile Similarity in Online Social Networks. *ACM Trans. Web*, 3(4):1–33, September 2009.

[74] J. Golbeck and J. Hendler. Reputation Network Analysis for Email Filtering. In *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, 2004.

[75] J. Golbeck, B. Parsia, and J. Hendler. Trust Networks on the Semantic Web. In *Proceedings of 7th International Workshop on Cooperative Intelligent Agents*, pages 238–249, Helsinki, Finland, 2003. Springer.

[76] J. A. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, College Park, MD, USA, 2005.

[77] T. Grandison and M. Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3(4), 2000.

[78] J. Grant and D. Becket. RDF Test Cases - N-Triples. Technical report, W3C Recommendation, 2004. `http://www.w3.org/TR/rdf-testcases/#ntriples`, Retrieved 07/03/2013.

[79] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance Semirings. In *Proceedings of the 26th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 31–40. ACM, 2007.

[80] P. T. Groth, S. Miles, and L. Moreau. A Model of Process Documentation to Determine Provenance in Mash-ups. *ACM Trans. Internet Techn.*, 9(1), 2009.

[81] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of Trust and Distrust. In *Proceedings of the 13th international conference on World Wide Web (WWW 2004)*, pages 403–412, New York, NY, USA, 2004.

[82] A. Harth, A. Polleres, and S. Decker. Towards A Social Provenance Model for the Web. Positional Paper, 2007. `http://vmserver14.nuigalway.ie/xmlui/bitstream/handle/10379/527/harth-etal-2007.pdf`, Retrieved 07/03/2013.

[83] O. Hartig. Provenance Information in the Web of Data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOW 2009)*, Madrid, Spain, April 2009. CEUR-WS.org.

[84] O. Hartig and J. Zhao. Using Web Data Provenance for Quality Assessment. In J. Freire, P. Missier, and S. S. Sahoo, editors, *SWPM*, volume 526 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[85] T. Heath. *Information-seeking on the Web with Trusted Social Networks – from Theory to Systems.* PhD thesis, Milton Keynes, UK, 2008.

[86] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space.* Morgan & Claypool, 1st edition, 2011.

[87] I. Herman, R. Swick, and D. Brickley. Resource Description Framework (RDF) / W3C Semantic Web Activity, January 2007. `http://www.w3.org/RDF/`, Retrieved 07/03/2013.

[88] J. Huang and M. S. Fox. An Ontology of Trust: Formal Semantics and Transitivity. In *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*, ICEC '06, pages 259–270, New York, NY, USA, 2006. ACM.

[89] R. Isele and C. Bizer. Learning Linkage Rules Using Genetic Programming. In *Proceedings of the 6th International Workshop on Ontology Matching*, Bonn, Germany, 2011. CEUR-WS.org.

[90] R. Isele, A. Jentzsch, and C. Bizer. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In *Proceedings of the 14th International Workshop on the Web and Databases 2011 (WebDB)*, Athens, Greece, 2011.

[91] I. Jacobs and N. Walsh. Architecture of the World Wide Web, Volume One. Technical report, W3C, Dec. 2004. `http://www.w3.org/TR/2004/REC-webarch-20041215/`, Retrieved 07/03/2013.

[92] A. Josang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2):618–644, March 2007.

[93] A. Jøsang and S. Pope. Semantic Constraints for Trust Transitivity. In *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling - Volume 43*, APCCM '05, pages 59–68, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[94] L. Kagal, T. Finin, and A. Joshi. A Policy Based Approach to Security for the Semantic Web. In *Proceedings of the 2nd International Semantic Web Conference (ISWC)*, pages 402–418, Sanibel Island, FL, USA, 2003. Springer.

[95] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM.

[96] H. Kautz, B. Selman, and M. Shah. The Hidden Web. *AI Magazine*, 18:27–36, 1997.

[97] A. Kini and J. Choobineh. Trust in Electronic Commerce: Definition and Theoretical Considerations. In *Proceedings of the 31st Annual Hawaii International Conference on System Sciences*, pages 51–61, Washington, DC, USA, 1998. IEEE Computer Society.

[98] J. M. Kleinberg. Hubs, Authorities, and Communities. *ACM Comput. Surv.*, 31(4es), Dec. 1999.

[99] G. Klyne and J. J. Carroll, editors. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation. World Wide Web Consortium, Feb. 2004.

[100] T. Knap. The W3PO Ontology. Working Draft, 2010. `http://purl.org/provenance/w3p/w3po#`, Retrieved 07/03/2013.

[101] T. Knap. Trusting Beliefs: A Different Way to Comprehend Trust in Social Networks. 8th Extended Semantic Web Conference (ESWC), Positional paper, Greece, 2011. `http://www.ksi.mff.cuni.cz/~knap/publications/2011/eswc-pp.pdf`, Retrieved 07/03/2013.

[102] T. Knap. Trusting Beliefs: A Way to Comprehend Trust between Members of the Czech Informatics Community. Extended Semantic Web Conference (ESWC) Summer School, Poster, Greece, 2011. `http://www.ksi.mff.cuni.cz/~knap/publications/2011/eswc-school-poster.pdf`, Retrieved 07/03/2013.

[103] T. Knap. Provenance Policies for Subjective Filtering of the Aggregated Linked Data. In *Proceedings of the 5th International Conference on Advances in Databases, Knowledge, and Data Applications*, DBKDA'13, pages 95–99, Seville, Spain, 2013. IARIA.

[104] T. Knap, J. Klímek, J. Mynarz, M. Nečaský, and J. Stárka. OpenGov - Towards More Transparent Public Contracts. Indian-summer school on Linked Data (ISSLOD), Poster, Germany, 2011. `http://www.ksi.mff.cuni.cz/~knap/publications/2011/isslod-poster.pdf`, Retrieved 07/03/2013.

[105] T. Knap, J. Michelfeit, J. Daniel, P. Jerman, D. Rychnovský, T. Soukup, and M. Nečaský. ODCleanStore: A Framework for Managing and Providing Integrated Linked Data on the Web. In *Proceedings of 13th International Conference on Web Information Systems Engineering (WISE)*, pages 815–816, Paphos, Cyprus, 2012. Springer.

[106] T. Knap, J. Michelfeit, and M. Nečaský. Linked Open Data Aggregation: Conflict Resolution and Aggregate Quality. In *COMPSAC Workshops*, pages 106–111, Izmir, Turkey, 2012. IEEE Computer Society.

[107] T. Knap and I. Mlýnková. Revealing Beliefs Influencing Trust between Members of the Czech Informatics Community. In *Proceedings of the 3rd International Conference on Social Informatics (SocInfo)*, pages 226–239, Singapore, 2011. Springer.

[108] T. Knap and I. Mlýnková. Web Quality Assessment Model: Trust in QA Social Networks. In *Proceedings of 8th International Conference on Ubiquitous Intelligence and Computing (UIC)*, pages 252–266, Banff, Canada, 2011. Springer.

[109] T. Knap and I. Mlýnková. Quality Assessment Social Networks: A Novel Approach for Assessing the Quality of Information on the Web. Proceedings of the 8th International Workshop on Quality in Databases of VLDB '10: 36th International Conference on Very Large Data Bases, 2010. `http://www.vldb2010.org/proceedings/files/vldb_2010_workshop/QDB_2010/Paper1_Knap_Mlynkova.pdf`, Retrieved 07/03/2013.

[110] T. Knap and I. Mlýnková. Towards Topic-based Trust in Social Networks. In *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing (UIC)*, pages 635–649, Xi'an, China, 2010. Springer.

[111] T. Knap, M. Nečaský, and M. Svoboda. A Framework for Storing and Providing Aggregated Governmental Linked Open Data. In *Proceedings of Joint International Conference on Electronic Government and the Information Systems Perspective, and Electronic Democracy (EGOVIS/EDEM)*, pages 264–270, Vienna, Austria, 2012. Springer.

[112] S. A. Knight and J. Burn. Developing a Framework for Assessing Information Quality on the World Wide Web. *Informing Science Journal*, 8:159–172, 2005.

[113] U. Kuter and J. Golbeck. SUNNY: A New Algorithm for Trust Inference in Social Networks Using Probabilistic Confidence Models. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pages 1377–1382, Vancouver, Canada, 2007. AAAI Press.

[114] N. Kwasnikowska, L. Moreau, and J. Van den Bussche. A Formal Account of the Open Provenance Model. December 2010. Submitted for publication.

[115] K. Lawrence and C. Kaler. WS-Trust Specification. Technical report, OASIS, 2007. `http://docs.oasis-open.org/ws-sx/ws-trust/200512`, Retrieved 07/03/2013.

[116] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

[117] R. Levien. Attack-Resistant Trust Metrics. *Computing with Social Trust (journal)*, pages 121–132, 2009.

[118] E.-P. Lim and R. H. L. Chiang. A Global Object Model for Accommodating Instance Heterogeneities. In *Proceedings of the 17th International Conference on Conceptual Modeling (ER)*, pages 435–448, Singapore, 1998. Springer.

[119] C. A. Lynch. When Documents Deceive: Trust and Provenance as New Factors for Information Retrieval in a Tangled Web. *J. Am. Soc. Inf. Sci. Technol.*, 52(1):12–17, 2001.

[120] F. Manola and E. Miller. RDF primer. *W3C Recommendation*, 10:1–107, 2004. `http://www.w3.org/TR/rdf-primer/`, Retrieved 07/03/2013.

[121] S. P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, April 1994.

[122] Martin Hepp. GoodRelations Vocabulary, 2011. `http://purl.org/goodrelations/v1#`, Retrieved 07/03/2013.

[123] P. Massa and B. Bhattacharjee. Using Trust in Recommender Systems: An Experimental Analysis. In *Proceedings of the 2nd International Conference on Trust Management (iTrust)*, pages 221–235, Oxford, UK, 2004. Springer.

[124] D. L. McGuinness and P. P. da Silva. Infrastructure for Web Explanations. In *Proceedings of the 2n International Semantic Web Conference (ISWC)*, pages 113–129, Sanibel Island, FL, USA, 2003. Springer.

[125] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. Technical report, W3C - World Wide Web Consortium, January 2004. `http://www.w3.org/TR/owl-features/`, Retrieved 07/03/2013.

[126] D. H. Mcknight and N. L. Chervany. The Meanings of Trust. Technical report, University of Minnesota, Carlson School of Management, 1996.

[127] P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: Linked Data Quality Assessment and Fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123, Berlin, Germany, March 2012. ACM.

[128] J. Michelfeit and T. Knap. Linked Data Fusion in ODCleanStore. In *Proceedings of the 11th International Semantic Web Conference (Posters & Demos)*, Boston, USA, 2012. CEUR-WS.org.

[129] J. Michelfeit, D. Rychnovský, J. Daniel, P. Jerman, T. Soukup, and T. Knap. ODCleanStore, Linked Data Management Tool, Programmer's Guide. Technical Report, 2011. `http://sourceforge.net/p/odcleanstore/home/`, Retrieved 07/03/2013.

[130] A. Miles, B. Matthews, M. Wilson, and D. Brickley. SKOS Core: Simple Knowledge Organisation for the Web. In *Proceedings of the 2005 International Conference on Dublin Core and metadata applications: vocabularies in practice*, pages 1:1–1:9, Madrid, Spain, 2005. Dublin Core Metadata Initiative.

[131] S. Miles, P. Groth, M. Branco, and L. Moreau. The Requirements of Using Provenance in e-Science Experiments. *Journal of Grid Computing*, 5(1):1–25, 2007.

[132] S. Miles, P. Groth, and M. Luck. Handling Mitigating Circumstances for Electronic Contracts. In *Proceedings of the AISB 2008 Symposium on Behaviour Regulation in Multi-agent Systems*, pages 37–42, Aberdeen, UK, Apr. 2008. The Society for the Study of Artificial Intelligence and Simulation of Behaviour.

[133] S. Miles, L. Moreau, and J. Futrelle. OPM Profile for Dublin Core Terms. Draft, 2009.

[134] S. Milgram. The Small World Problem. *Psychology Today*, 1:61, 1967.

[135] L. Moreau. Open Provenance Model (OPM) OWL Specification. Working Draft, 2010. `http://openprovenance.org/model/opmo`, Retrieved 07/03/2013.

[136] L. Moreau. The Foundations for Provenance on the Web. *Found. Trends Web Sci.*, 2(2-3):99–241, Feb. 2010.

[137] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche. The Open Provenance Model Core Specification (v1.1). *Future Generation Computer Systems*, 27(6):743 – 756, 2011.

[138] T. Moses. eXtensible Access Control Markup Language (XACML) Version 2.0. Technical report, OASIS Access Control TC, Feb. 2005. `http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf`, Retrieved 07/03/2013.

[139] A. Motro and P. Anokhin. Fusionplex: Resolution of data inconsistencies in the integration of heterogeneous information sources. *Information Fusion*, 7(2):176–196, 2006.

[140] L. Mui, M. Mohtashemi, and A. Halberstadt. A Computational Model of Trust and Reputation for E-businesses. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS)*, pages 188–, Washington, DC, USA, 2002. IEEE Computer Society.

[141] F. Naumann and C. Rolker. Assessment Methods for Information Quality Criteria. In *Proceedings of the 5th International Conference on Information Quality (IQ)*, pages 148–162. MIT, 2000.

[142] J. C. Nekola and P. S. White. The Distance Decay of Similarity in Biogeography and Ecology. *Journal of Biogeography*, 26(4):867–878, 1999.

[143] M. E. J. Newman. Models of the Small World. *J. Stat. Phys*, pages 819–841, 2000.

[144] A.-C. Ngonga Ngomo and S. Auer. LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Spain, 2011. IJCAI/AAAI.

[145] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[146] M. R. Quillian. Semantic Memory. In M. Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, MA, 1968.

[147] L. R. Attack Resistant Trust Metrics. PhD thesis, UC Berkeley, Berkeley, CA, USA, 2003.

[148] M. Richardson, R. Agrawal, and P. Domingos. Trust Management for the Semantic Web. In *Proceedings of the 2nd International Semantic Web Conference (ISWC)*, pages 351–368, Sanibel Island, FL, USA, 2003. Springer.

[149] W. H. Riker. The Nature of Trust. In Tedeschi, J. T. (Ed.), Perspectives on Social Power, pages 63 – 81, Chicago, Aldine Publishing Company, 1971.

[150] N. D. Rio, P. P. da Silva, A. Q. Gates, and L. Salayandia. Semantic Annotation of Maps Through Knowledge Provenance. In *Proceedings of the 2nd International Conference on GeoSpatial Semantics (GeoS)*, pages 20–35, Mexico City, Mexico, 2007. Springer.

[151] A. Schultz, A. Matteini, R. Isele, C. Bizer, and C. Becker. LDIF : Linked Data Integration Framework. In *Proceedings of the Second International Workshop on Consuming Linked Data (COLD)*, Bonn, Germany, 2011. CEUR-WS.org.

[152] Y. Simmhan and R. Barga. Analysis of Approaches for Supporting the Open Provenance Model: A Case Study of the Trident Workflow Workbench. *Future Gener. Comput. Syst.*, 27(6):790–796, June 2011.

[153] Y. L. Simmhan, B. Plale, and D. Gannon. A Survey of Data Provenance in e-Science. *SIGMOD Rec.*, 34(3):31–36, September 2005.

[154] A. Syalim, Y. Hori, and K. Sakurai. Grouping Provenance Information to Improve Efficiency of Access Control. In *Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance (ISA)*, pages 51–59, Seoul, Korea, 2009. Springer.

[155] W. C. Tan. Provenance in Databases: Past, Current, and Future. *IEEE Data Eng. Bull.*, 30(4):3–12, 2007.

[156] S. S. Timothy Lebo and D. McGuinness. PROV-O: The PROV Ontology. W3C Editor's Draft, 2012. `http://dvcs.w3.org/hg/prov/raw-file/default/ontology/Overview.html`, Retrieved 07/03/2013.

[157] N. Toupikov, J. Umbrich, R. Delbru, M. Hausenblas, and G. Tummarello. DING! Dataset Ranking using Formal Descriptions. In *WWW 2009 Workshop: Linked Data on the Web (LDOW2009)*, Madrid, Spain, 2009. CEUR-WS.org.

[158] S. Tunnicliffe and I. Davis. Changeset vocabulary, 2005.

[159] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS Policy and Domain Services: Toward a Description-logic Approach to Policy Representation, Deconfliction, and Enforcement. In *Proceedings of 4th IEEE International Workshop on Policies for Distributed Systems and Networks (Policy)*, pages 93–96, Lake Como, Italy, 2003. IEEE Computer Society.

[160] F. Walter, S. Battiston, and F. Schweitzer. A Model of a Trust-based Recommendation System on a Social Network. *Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, February 2008.

[161] F. E. Walter, S. Battiston, and F. Schweitzer. Personalised and Dynamic Trust in Social Networks. In *Proceedings of the 3rd ACM conference on Recommender systems (RecSys)*, pages 197–204, New York, NY, USA, 2009. ACM.

[162] R. Y. Wang and D. M. Strong. Beyond Accuracy: What Data Quality Means to Data Consumers. *J. Manage. Inf. Syst.*, 12(4):5–33, 1996.

[163] W. E. Winkler. The State of Record Linkage and Current Research Problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[164] A. Woodruff and M. Stonebraker. Supporting Fine-grained Data Lineage in a Database Visualization Environment. In *Proceedings of the 13th International Conference on Data Engineering*, pages 91–102, Birmingham, England, Apr. 1997. IEEE Computer Society.

[165] L. L. Yan. Towards Efficient and Scalable Mediation: The AURORA Approach. In *Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research (CASCON)*, page 23, Toronto, Ontario, Canada, 1997. IBM.

[166] L. L. Yan and M. Tamer. Conflict Tolerant Queries in AURORA. In *Proceedings of the 4th IFCIS International Conference on Cooperative Information Systems (CoopIS)*, pages 279 – 290, Edinburgh, Scotland, 1999. IEEE Computer Society.

[167] P. H. Yanfeng Shu, Kerry Taylor and C. Peters. Modelling Provenance in Hydrologic Science: a Case Study on Streamflow Forecasting. *Journal of Hydroinformatics*, 14(4):944–959, 2012.

[168] J. Zhao. Open Provenance Model Vocabulary Specification. Working Draft, 2010. `http://open-biomed.sourceforge.net/opmv/ns.html`, Retrieved 07/03/2013.

[169] J. Zhao, C. Wroe, C. Goble, R. Stevens, D. Quan, and M. Greenwood. Using Semantic Web Technologies for Representing e-Science Provenance. In *Proceedings of Third International Semantic Web Conference (ISWC2004)*, pages 92–106, Hiroshima, Japan, Nov. 2004. Springer-Verlag.

[170] C.-N. Ziegler and J. Golbeck. Investigating Interactions of Trust and Interest Similarity. *Decision Support Systems*, In Press, Corrected Proof, 2007.

[171] C.-N. Ziegler and G. Lausen. Propagation Models for Trust and Distrust in Social Networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.