Web Semantics: Science, Services and Agents on the World Wide Web 7 (2009) 1–10

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Quality-driven information filtering using the WIQA policy framework

Christian Bizer[a,*], Richard Cyganiak[b]

[a] Freie Universität Berlin, Germany
[b] Digital Enterprise Research Institute, NUI Galway, Ireland

ARTICLE INFO

Article history:
Received 2 August 2007
Received in revised form
30 November 2007
Accepted 17 February 2008
Available online 21 April 2008

Keywords:
Information quality
Quality-driven information filtering
Semantic Web

ABSTRACT

Web-based information systems, such as search engines, news portals, and community sites, provide access to information originating from numerous information providers. The quality of provided information varies as information providers have different levels of knowledge and different intentions. Users of web-based systems are therefore confronted with the increasingly difficult task of selecting high-quality information from the vast amount of web-accessible information. How can information systems support users to distinguish high-quality from low-quality information? Which filtering mechanisms can be used to suppress low-quality information? How can filtering decisions be explained to the user? This article identifies information quality problems that arise in the context of web-based systems, and gives an overview of quality indicators as well as information quality assessment metrics for web-based systems. Afterwards, we introduce the WIQA—Information Quality Assessment Framework. The framework enables information consumers to apply a wide range of policies to filter information. The framework employs the Named Graphs data model for the representation of information together with quality-related meta-information. The framework uses the WIQA-PL policy language for expressing information filtering policies against this data model. WIQA-PL policies are expressed in the form of graph patterns and filter conditions. This allows the compact representation of policies that rely on complex meta-information such as provenance chains or combinations of provenance information and background information about information providers. In order to facilitate the information consumers' understanding of filtering decisions, the framework generates explanations of why information satisfies a specific policy.

1. Introduction

The World Wide Web is a global information space consisting of information from a multitude of autonomous information providers [17]. Web-based information systems provide access to this information space. They integrate information from multiple providers and present integrated information to their users.

The key success factor of the web is the vast amount of web-accessible information. On the other hand, its openness and the autonomy of information providers make the web vulnerable to inaccurate, misleading, or outdated information. Information quality problems arise in various application domains of web-based information systems:

Search engines provide access to billions of web documents and an increasing number of structured information sources. The quality of provided information varies widely and the huge amount of accessible information obscures relevant information.

News portals aggregate news articles from a wide range of newspapers and news agencies and assemble them according to user's interests. As news providers have different views of the world and different levels of knowledge, news may be biased or inaccurate.

Financial information portals integrate stock quotes, financial news, company profiles, and analyst reports from multiple information sources. The expertise of information providers on specific markets and companies varies widely and investors are confronted with conflicting advice.

Online communities like MySpace, Facebook, Flickr, or YouTube are used by large numbers of information providers to share information. The quality of provided information varies widely, and again the amount of accessible information blurs relevant information.

Semantic Web applications. The Semantic Web [15] is a global information space consisting of Linked Data [4]. Semantic Web applications enable users to navigate and query this information space. Assuring information quality is problematic within Semantic Web applications as they operate on an unbound, dynamic set of autonomous data sources.

* Corresponding author. Tel.: +49 30 838 54057.
E-mail addresses: chris@bizer.de (C. Bizer), richard@cyganiak.de (R. Cyganiak).

1570-8268/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.websem.2008.02.005

2. Problem statement

Information providers have different levels of knowledge, different views of the world, and different intentions. Therefore, provided information may be wrong, biased, outdated, incomplete, and inconsistent.

Before information from the web is used to accomplish a specific task, its quality should be assessed according to task-specific criteria. Based on the assessment result, information may be accepted or rejected for a specific task.

In everyday life, we use a wide range of different policies to assess the quality of information: we might accept information from a friend on restaurants, but distrust him on computers; regard scientific papers only as relevant, if they have been published within the last 2 years; or believe foreign news only when they are reported by several independent sources. Which policy is chosen depends on the specific task at hand, our subjective preferences, and the availability of information quality-related meta-information, such as ratings or background information about information providers.

This article introduces an innovative solution to quality-driven information filtering in the context of web-based information systems. Instead of having the designer of an information system decide for the user on a single, fixed method to distinguish high-quality from low-quality information, the user is empowered to employ a similar wide range of filtering policies as she is using in the off-line world.

The article makes the following contributions to the research on policy frameworks for web-based information systems:

- While there is a lot of work on policy frameworks for access control and privacy protection, the article highlights the need of web-based systems for information filtering policies. The article gives an overview of different types of quality indicators that are relevant in the context of web-based systems and discusses information quality assessment metrics for these systems.
- Information filtering may rely on various quality indicators, such as provenance meta-information or ratings. Therefore, information filtering frameworks need a flexible means to represent information together with quality-related meta-information. The article proposes to employ the Named Graphs [8] data model to fulfill this requirement and to represent information together with quality-related meta-information as an integrated model.
- The article proposes a new approach to expressing information filtering policies. Instead of relying on a specific policy ontology or a generic rules language, WIQA-PL policies are expressed as graph patterns and filter conditions. This approach closely couples information representation and policy formulation and allows the compact representation of policies that rely on complex meta-information such as provenance chains or combinations of provenance information and background information about information providers.

The article is structured as follows. Sections 3 and 4 give an overview of the concept of information quality and discuss information quality assessment metrics. Chapter 6 describes the WIQA information filtering framework and the WIQA-PL policy language. Chaper 7 shows how the WIQA framework is used to extend a web browser with information filtering capabilities. Chapter 8 describes the evaluation of the WIQA framework, while Chapter 9 compares the framework with related work.

3. Information quality

Compared to concepts like data integrity and security which have been studied in detail since the introduction of relational database technology, the notion of information quality is relatively young and its general conceptualization as well as the methods developed to assess information quality are still highly diverse.

The concept of information quality is a domain-specific subconcept of the general concept of quality. A popular definition for quality is given by Joseph Juran. He defines quality as "fitness for use" [20]. Juran's definition has been adopted by most authors working on information quality. Information quality is commonly defined as the fitness for use of information [38,37,29,13,23]. This definition implies two important aspects:

- Information quality is task-dependent. A user might consider the quality of a piece of information appropriate for one task but not sufficient for another task.
- Information quality is subjective, as a second less quality-concerned user might consider the quality of the same piece of information appropriate for both tasks.

Information quality is commonly conceived as a multidimensional construct [38,29,36,32,9], as the "fitness for use" may depend on various factors such as accuracy, timeliness, completeness, relevancy, objectivity, believability, understandability, consistency, conciseness, availability, and verifiability [38,29,10].

These information quality dimensions are not independent of each other and typically only a subset of the dimensions is relevant in a specific situation. Which quality dimensions are relevant and which levels of quality are required for each dimension is determined by the specific task at hand and the subjective preferences of the information consumer [29,38].

4. Information quality assessment

Information quality assessment is the process of evaluating if a piece of information meets the information consumer's needs in a specific situation [29,10]. Information quality assessment involves measuring the quality dimensions that are relevant to the information consumer and comparing the assessment results with the information consumer's quality requirements.

An *information quality assessment metric* is a procedure for measuring an information quality dimension. Assessment metrics rely on *quality indicators* and calculate an assessment score from these indicators using a *scoring function*. Assessment metrics are heuristics that are designed to fit a specific assessment situation [34,39].

The types of information which may be used as quality indicators are highly diverse. Besides the information to be assessed itself, assessment metrics may rely on meta-information about the circumstances in which information was created, on background information about the information provider, or on ratings provided by the information consumer herself, other information consumers, or domain experts. Fig. 1 shows an abstract view on an information exchange situation. All types of information that may be used as quality indicators are shaded gray.

Information quality assessment metrics can be classified into three categories according to the type of information that is used as quality indicator: (1) information content itself; (2) information about the context in which information was claimed; (3) ratings about information itself or the information provider.

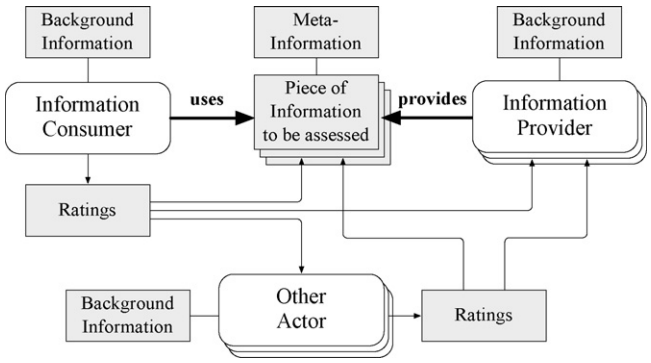

Fig. 1. Abstract view on an information exchange situation.

4.1. Content-based metrics

Content-based metrics use information to be assessed itself as quality indicator. The metrics analyze the information content or compare information with related information. Which metrics are applicable depends on the type of information to be analyzed.

In the case of natural language texts or loosely structured documents, such as HTML pages, it is possible to apply various text analysis methods for information quality assessment. In general, these methods derive assessment scores by matching terms or phrases against a document and/or by analyzing the structure of the document. Within deployed web-based information systems, text analysis methods are widely used to assess the relevancy of documents, to detect spam, and to scan websites for offensive content.

In the case of formalized information, one possibility is to apply simple rule-based metrics. For instance, a metric to assess the believability of a sales offer could be to check if the price lies above a specific boundary. If the price is too low, the offer might be considered bogus. Frequently, formalized information contains values that are grossly different or inconsistent with the remaining set. There are various statistical methods to identify such outliers [24]. Within the context of information quality assessment, outlier detection methods can be used as heuristics to assess quality dimensions such as accuracy or believability.

4.2. Context-based metrics

Context-based metrics employ meta-information about the information content and the circumstances in which information was created, e.g., who said what and when, as quality indicator. For instance, an important quality indicator for assessing the believability of information is meta-information about the identity of the information provider. That is, assumptions about the believability of information providers are extended to information they provide. An example for a simple heuristic to assess the believability of information is to check whether an information provider is contained in a list of trusted providers. Other meta-information that might influence believability are the identities of the contributors and the publisher of information as well as the source from which information is retrieved.

Instead of relying solely on meta-information, information quality assessment metrics can also combine meta-information with background information about the application domain. Another metric for assessing the believability dimension could, for instance, be based on the role of an information provider in the application domain ("Prefer product descriptions published by the manufacturer over descriptions published by a vendor" or "Disbelieve everything a vendor says about its competitor."), his membership in a specific group ("Believe only information from authors working for certain companies."), or his former activities ("Believe only information from authors who have already published several times on a topic." or "Believe only reports from stock analysts whose former predictions proved correct to a certain percentage.").

4.3. Rating-based metrics

Rating-based metrics rely on explicit ratings about information itself, information sources, or information providers. Ratings may originate from the information consumer herself, other information consumers, or domain experts.

The design of rating systems has been widely studied in computer science [19]. Seen from an abstract perspective, rating-based quality assessment involves two processes: The acquisition of ratings and the calculation of assessment scores from these ratings [10]. Within the assessment process, a scoring function calculates assessment scores from the collected ratings. The scoring function decides which ratings are taken into account and might assign different weights to ratings. Designing scoring functions is a popular research topic and various authors have proposed different algorithms. Approaches to classifying the proposed algorithms are presented by Zhang et al. [40] and Ziegler [41].

4.4. Accuracy of assessment results

A general problem of information quality assessment is that assessment results are often imprecise [29,33,10]. This is especially true for the the context of web-based systems where the availability and the quality of quality indicators is often not optimal. For instance, because of the decentralized nature of the web and the autonomy of information providers, meta-information about web content is often incomplete. The quality of ratings is often uncertain, as the expertise of raters is unknown in many cases. Because of these problems, information quality assessment often has to trade accuracy for practicability [29].

The imprecision of assessment results is relativized by the fact that users of web-based information systems are accustomed to tolerating a certain amount of low-quality information. For them, the benefit of having access to a huge information-base is often higher than the costs of having some noise in the answers [28]. Therefore, the goal of practical information quality assessment is to find heuristics which can be applied in a given situation and that are sufficiently precise to be useful from the perspective of the information consumer.

5. Information filtering policies

Quality-based information filtering policies are heuristics for deciding whether to accept or reject information to accomplish a specific task [10].

A information filtering policy consists of a set of assessment metrics, for assessing the quality dimensions that are relevant for the task at hand and a decision function which aggregates the resulting assessment scores into an overall decision on whether information satisfies the information consumer's quality requirements. Each assessment metric relies on a set of quality indicators and specifies a scoring function to calculate an assessment score from these indicators. The decision function weights assessment scores depending on the relevance of the different quality dimensions for the task at hand.

Information consumers may choose a wide range of different policies to decide whether to accept or reject information. When choosing a policy that fits a specific situation, an information consumer has to answer the following questions: Which information quality dimensions are relevant in the context of the task at hand? Which information quality assessment metric should be used to

assess each dimension? How should the assessment results be compiled into an overall decision on whether to accept or reject information?

The relevance of the different quality dimensions is determined by the task at hand. The choice of suitable assessment metrics for specific quality dimensions is restricted by several factors:

Availability of quality indicators. Whether an assessment metric can be used in a specific situation depends on the availability of the quality indicators that are required by the metric. Assessing quality dimensions like timeliness is possible in many cases, as the required quality indicators are often available. Accessing other dimensions like accuracy or objectivity often proves difficult, as it might involve the information consumer or experts verifying or rating information.
Quality of quality indicators. The choice of assessment metrics is also influenced by the quality of the available quality indicators. If an information consumer is in doubt about the quality of certain indicators, she might prefer to choose a different assessment metric which relies on other indicators.
Understandability. The key factor for an information consumer to trust assessment results is his understanding of the assessment process. Therefore, relatively simple, easily understandable, and traceable assessment metrics are often preferred [19].
Subjective preferences. The information consumer might have subjective preferences for specific assessment metrics. She might, for example, consider specific quality indicators and scoring functions more reliable than others. Thus, there is never a single best policy for a specific task, as the subjectively best policy differs from user to user.

6. The WIQA framework

The *WIQA—Information Quality Assessment Framework* is a set of software components that can be employed by applications which process information of uncertain quality and want to enable their users to filter information using a wide range of different quality-based information filtering policies.

The framework has been designed to fulfill the following requirements: (1) flexible representation of information together with quality-related meta-information; (2) enable users to employ different information filtering policies; (3) ability to generate explanations about the filtering process.

The WIQA framework consists of two components: a Named Graph store for representing information together with quality-related meta-information, and a filtering and explanation engine which enables applications to filter information and to retrieve explanations about filtering decisions.

Fig. 2 gives an overview of the components of the WIQA framework. An application deposits unfiltered information in the framework's Named Graph store. The application also provides a set

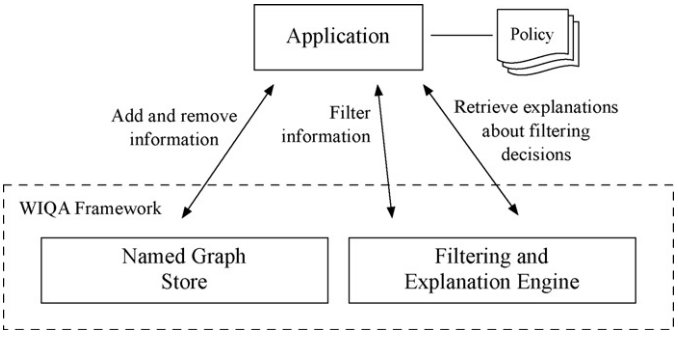

Fig. 2. Overview of the WIQA framework.

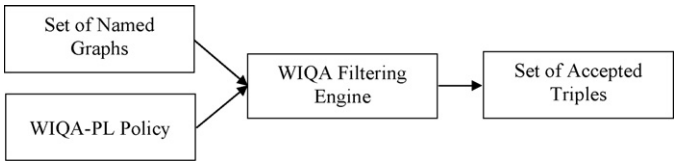

Fig. 3. The WIQA filtering process.

of filtering policies which fit the specific requirements of the application domain. When the application requests the filtering engine to apply one of these policies, the filtering engine provides a view on the Named Graph store which contains only information that fulfills the policy. Fig. 3 illustrates this filtering process.

6.1. Representing quality-related information

The WIQA framework uses Named Graphs [8] as a flexible data model for representing both quality-related meta-information and information to be filtered itself. Named Graphs are an extension of the RDF data model [22]. Like in RDF, the atomic units of information are subject–predicate–object triples. A set of triples forms an RDF graph. But unlike in RDF, these graphs are named with URI references. Graph names can be used as the subject and object of triples, which allows the expression of meta-information about graphs.

Within the WIQA framework, each Named Graph is assumed to contain information published by a specific information provider at a certain point in time. Provenance information about each graph is represented within other graphs using the Semantic Web Publishing (SWP) Vocabulary [10]. In SWP, graph provenance is recorded in the form of a *warrant* attached to the graph. A warrant is authorized by an actor, called its *authority*. The warrant expresses a propositional attitude towards one or more Named Graphs, e.g., assertion or quotation [10]. Warrants can be enriched with additional meta-information and may be digitally signed.

This approach to representing primary information together with quality-related meta-information as an integrated model provides for fine-grained provenance tracking, the representation of provenance chains, and the representation of provenance information together with background information about information providers.

The example graph set shown in Fig. 4 demonstrates how the Named Graphs data model is used to represent partial descriptions of a resource while keeping track of the provenance of different pieces of information. The example also demonstrates how the data model is used to represent provenance chains. The example uses the TriG syntax [5]. Namespace declarations are omitted. The graph set consists of four Named Graphs. The first graph originates from Intel and contains general information about the company (lines 1–5). The second graph originates from an individual and contains a newsgroup posting and several trust ratings for companies and other individuals (lines 7–16). The third graph originates from Yahoo Finance (lines 18–26). It contains additional information about Intel as well as provenance information about the second graph. The fourth graph originates from an information aggregation service and contains provenance information about the first and third graph (lines 28–38). The third and the forth graph taken together represent the provenance chain that according to Yahoo Finance, the newsgroup posting described in the second graph was authored by Mark Scott.

6.2. Expressing policies

Information consumers use a wide range of different policies for determining whether to accept or reject information. The WIQA framework allows the expression of such policies using the declar-

```
1.  fd:GraphFromIntel {
2.   <http://www.intel.com/c>
3.      rdf:type fin:Corporation ;
4.      fin:country iso:US ;
5.      foaf:homepage <http://www.intel.com> . }
6.
7.  fd:GraphFromMarkScott {
8.   <urn:x-ISIN:US4581401001> fin:posting
9.      "As we have seen in ..."@EN .
10.  <mailto:mark@scott.com>
11.     fin:positiveRating
12.        <http://www.analyst-house.com/c> ;
13.     fin:positiveRating
14.        <mailto:pSmith@deutsche-bank.de> ;
15.     fin:negativeRating
16.        <mailto:reynolds@ft.com> . }
17.
18. fd:GraphFromYahooFinance {
19.   <urn:x-ISIN:US4581401001>
20.      rdf:type fin:Share ;
21.      fin:emitter <http://www.intel.com/c> .
22.  fd:GraphFromMarkScott
23.     swp:assertedBy fd:MsWarrant .
24.  fd:MsWarrant
25.     swp:authority <mailto:mark@scott.com> ;
26.     dc:date "2005-11-15"^^xsd:date . }
27.
28. fd:GraphFromAggregator {
29.  fd:GraphFromIntel
30.     swp:assertedBy fd:IntelWarrant .
31.  fd:IntelWarrant
32.     swp:authority <http://www.intel.com/c> ;
33.     dc:date "2005-10-21"^^xsd:date .
34.  fd:GraphFromYahooFinance
35.     swp:assertedBy fd:YFWarrant .
36.  fd:YFWarrant
37.     swp:authority <http://www.yahoo.com/c> ;
38.     dc:date "2005-11-20"^^xsd:date . }
```

Fig. 4. Example set of Named Graphs.

ative WIQA-PL policy language. The policy language is able to: (1) use whatever quality-related information that is available in the application context as quality indicators; (2) express a wide range of context-, content-, and rating-based assessment metrics over these indicators; (3) combine those assessments into an overall filtering decision.

The following sections define the semantics of WIQA policies. A WIQA policy specifies conditions that a triple from a set of Named Graphs has to fulfill in order to be *accepted* by the policy (Fig. 3). We rely on the SPARQL semantics given by Pérez et al. [31] for basic definitions.

6.2.1. Named Graphs and named graph patterns

Let U be the set of all URIs, B the set of blank nodes, L the set of RDF literals, and V the set of variables. U, B, L and V are pairwise disjoint. We denote by T the set of RDF terms, the union $U \cup B \cup L$.

A triple $(s, p, o) \in T \times U \times T$ is called an *RDF triple*. An *RDF graph* is a set of RDF triples. A *named graph* is a tuple (g, G), where g, the *graph name*, is an URI, and G is an RDF graph.

A *named graph pattern* is a pair (n, P), where $n \in (U \cup V)$ and P is a *graph pattern*. A graph pattern, for the purposes of this paper, is either a tuple from $(T \cup V) \times (U \cup V) \times (T \cup V)$ (a *triple pattern*); or an expression $(P_1 \text{ AND} P_2)$, where P_1 and P_2 are graph patterns; or an expression (P FILTERR), where P is a graph pattern, and R is a boolean *filter condition* composed of constant RDF terms, variables, comparison and boolean operators, and function calls.

6.2.2. Semantics of WIQA policies

A *WIQA policy* consists of one or more named graph patterns $P_1 \ldots P_i$, and zero or more additional filter conditions $R_1 \ldots R_j$. The *policy pattern* is defined as the expression

$$P_1 \text{ AND} \ldots \text{AND } P_i \text{ FILTER } R_1 \ldots \text{FILTER } R_j$$

The connectives AND and FILTER over named graph patterns are defined in analogy to those given in Ref. [31] over graph patterns. For the policy to be meaningful, at least one of the named graph patterns must involve at least one of the *referring variables* ?GRAPH, ?SUBJ, ?PRED, and ?OBJ, which are used to express conditions that triples have to fulfill to be accepted by the policy.

In Ref. [31], the *evaluation* of a graph pattern P over an RDF graph G, denoted by $[[P]]_G$, is defined as a set of *mappings* from variables to RDF terms that, informally stated, fulfill the conditions imposed by the graph pattern. We will now extend the notion of evaluation to cover Named Graphs.

Let (n, P) be a named graph pattern, and N a set of Named Graphs. The evaluation of (n, P) over N, denoted by $[[(n, P)]]_N$ is defined as follows:

$$\begin{array}{ll} \{\mu \mid (n, G) \in N \text{ and } \mu \in [[P]]_G\}, & \text{if } n \in U \\ \{\mu \bowtie \mu' \mid (g, G) \in N \text{ and } \mu \in [[P]]_G\}, & \text{if } n \in V \end{array}$$

where μ' is the singleton mapping $\{n \mapsto g\}$.

Let Q be a WIQA policy, P_Q its policy pattern, and N a set of Named Graphs. Then a triple (s, p, o) in Named Graph $(g, G) \in N$ is said to be *accepted by* Q if there exists a mapping m such that $m \in [[P_Q]]_N$ and m compatible to the mapping

$$\{?\text{GRAPH} \mapsto g, ?\text{SUBJ} \mapsto s, ?\text{PRED} \mapsto p, ?\text{OBJ} \mapsto o\}$$

Less formally speaking, a triple is accepted if its origin graph, subject, predicate and object, respectively, fulfill the conditions that the patterns and filter conditions of the policy impose on the variables ?GRAPH, ?SUBJ, ?PRED and ?OBJ.

6.2.3. The WIQA-PL syntax

WIQA policies are expressed using the WIQA-PL syntax. The syntax is based on the syntax of the SPARQL query language [35] in order to make it easier for people who already know SPARQL to learn WIQA-PL. The complete grammar of the WIQAP-PL syntax given in Ref. [10].

Fig. 5 shows an example of a WIQA-PL policy. Lines 1–4 specify the policy name and description. The PATTERN clause restricts information to originate from analysts who achieved a StarMine score above 80. It consists of two named graph patterns.

The first named graph pattern requires provenance information about graphs to be contained in the graph `fd:GraphFromAggregator`. It contains two triple patterns which require provenance information to be expressed using the SWP properties `swp:assertedBy` and `swp:authority`. The first pattern binds the names of asserted graphs to the referring variable `?GRAPH`. The second pattern binds URIs that identify authorities to the variable `?authority`. The second triple pattern is connected to the first by sharing the variable `?warrant`.

The second graph pattern requires authorities to be an instance of the class `fin:Analyst` and to have a `fin:benchmark` prop-

```
1. NAME "Information from highly rated analysts"
2. DESCRIPTION "Accept only information that
3.   has been asserted by analysts who
4.   achieved a StarMine score above 80."
5. PATTERN {
6.   GRAPH fd:GraphFromAggregator {
7.     ?GRAPH swp:assertedBy ?warrant .
8.     ?warrant swp:authority ?authority . }
9.   GRAPH fd:BackgroundInformation {
10.    ?authority rdf:type fin:Analyst .
11.    ?authority fin:benchmark ?benchmark .
12.    FILTER (?benchmark > 80) . }
13. }
```

Fig. 5. WIQA-PL policy: accept only information that has been asserted by analysts who achieved a StarMine score above 80.

erty whose value is bound to `?benchmark`. The FILTER clause in line 12 restricts the benchmark score to values above 80. The triples that describe authorities have to occur in the graph `fd:BackgroundInformation`.

6.2.4. Extension functions

Quality-based information filtering policies rely on a wide range of different, application domain specific assessment metrics. Therefore, WIQA-PL provides an extension mechanism for invoking arbitrary, application domain specific functions. For instance, rating-based filtering policies may use various scoring algorithms to calculate a score for an entity from a network of ratings. Content-based filtering policies may rely on natural language processing methods to analyze text or may use various statistical methods to compare a piece of information with related information. By including domain specific functions, the WIQA framework can be extended to fit the requirements of different application domains.

There are two types of extension functions. Conditional extension functions are used as part of FILTER conditions and can be combined with the built-in logical operators. They compute boolean values from mappings, and are used in determining if the condition holds for a particular mapping.

The second type are result set filters. They are applied to the entire set of mappings that satisfy a policy's patterns and conditions. Result set filters can perform arbitrary modifications to the set, though typical applications are ranking and counting within the result mappings. Both types of extensions have access to the original unfiltered dataset.

Three example extension functions have been implemented: The `wiqa:morePositiveRatings` and the `wiqa:tidalTrust`[18] conditional functions implement different rating-based scoring algorithms; the `wiqa:count` result set filter allows the formulation of quantity constraints. A WIQA-PL policy that uses the `wiqa:count` function is given in Fig. 6.

6.3. Explaining assessment results

The accuracy of information quality assessment results is often uncertain due to the limited availability of quality indicators and due to the uncertain quality of the quality indicators themselves. Therefore, the user's final decision whether to trust or distrust assessment results depends on her understanding of the assessment metrics and quality indicators that were used in the assessment process. Information systems can support users in this trust decision by providing explanations of *why* information satisfies a given filtering policy.

```
1. NAME "Asserted by analysts with at least
2.       3 positive ratings."
3. DESCRIPTION "Accept only information that
4.              has been asserted by analysts
5.              who have received at least 3
6.              positive ratings."
7. PATTERNS {
8.
9.   GRAPH fd:GraphFromAggregator
10.       { ?GRAPH swp:assertedBy ?warrant .
11.         ?warrant swp:authority ?auth .
12.         EXPL "it was asserted by " ?auth
13.              " and " . }
14.
15.  GRAPH ?graph2
16.        { ?auth rdf:type fin:Analyst . }
17.
18.  GRAPH fd:GraphFromAggregator
19.        { ?graph2 swp:assertedBy ?warrant2 .
20.          ?warrant2 swp:authority ?auth2 .
21.          EXPL ?auth2 " claims that "
22.               ?auth " is an analyst." . }
23.
24.  GRAPH ANY
25.        { ?rater fin:positiveRating ?auth .
26.          FILTER (wiqa:count(?rater) > 2) .
27.          EXPL ?auth "has received positive
28.                      ratings from" . }
29.
30.  GRAPH fd:BackgroundInformation
31.        { ?rater fin:affiliation ?company .
32.          EXPL ?rater "who works for"
33.               ?company . }
28.  }
```

Fig. 6. WIQA-PL policy using the `wiqa:count` extension function and including explanation templates.

Making information filtering decisions comprehensible and traceable requires diverse forms of explanations. The content of suitable explanations depends on the assessment metrics that are used within a policy and on the current task of the user. For less important tasks, the user will be contented with short, simple to comprehend explanations. For other, more important tasks the user will require explanations to contain detailed information about the assessment process and the quality indicators that were used in the process.

Explanations for rating-based metrics. Ratings might be subjective and raters may try to influence rating systems by providing unfair ratings. Therefore, explanations for rating-based assessment metrics should contain the ratings that were used in the evaluation and explain the calculation steps of the scoring algorithm. More detailed explanations might provide provenance information about ratings and background information about raters.

Explanations for context-based metrics. An explanation for a policy that relies on provenance information should list the information providers. The explanation might also contain additional background information about information providers in order to

The triple: Siemens AG has positive analyst report: "As Siemens agrees partnership with Novell unit SUSE ..."

fulfills the policy: Accept only information that has been asserted by analysts who have received at least 3 positive ratings.

because: it was asserted by Peter Smith and

- Deutsche Bank claims that Peter Smith is an analyst.
- Financial Times claims that Peter Smith is an analyst.

Peter Smith has received positive ratings from

- Mark Scott who works for Siemens.
- David Brown who works for Intel.
- John Maynard who works for Financial Times.

Fig. 7. Example explanation.

support information consumers in judging their trustworthiness. *Explanations for content-based metrics* detail why information content itself satisfies the requirements of an assessment metric. For instance, an explanation for a statistical metric should contain the data that was used in the calculation and describe the calculation process. An explanation for a text analysis method might list relevant keywords and explain how the overall score for a text was calculated.

The WIQA framework combines two explanation generation mechanisms. First, a template mechanism is used to generate the parts of an explanation which explain why constraints that are expressed as graph patterns are satisfied. When a user requests an explanation why an accepted triple fulfills the policy, *explanation templates* that are part of the WIQA-PL policy syntax are instantiated with variable bindings from the matching solutions that led to the acceptance of the triple. As a convention, instantiated templates should form a phrase that completes the sentence: "The piece of information fulfills the policy because ..." Second, the explanation is supplemented with explanation parts that explain why constraints that are expressed using WIQA extension functions are satisfied. Extension functions may conduct complex calculations such as rating-based scoring or statistical evaluations. In order to make their calculations comprehensible, extension functions can generate custom, function-specific explanations. The complete algorithm that is used by the WIQA framework to combine template and extension function generated explanation parts is given in Ref. [10].

Fig. 6 shows a WIQA policy containing explanation templates. Explanation templates are marked by the `EXPL` keyword. The policy requires that information is stated by an analyst, and the analyst must have received at least three positive ratings. The explanation will display: (1) the source of the information; (2) who claims that the source is an analyst; (3) the raters and their affiliation. An example explanation generated by this policy is given in Fig. 7.

6.4. Implementation

A Java implementation of the WIQA framework is available. It builds on the *NG4J—Named Graphs API*,[1] the *Jena—Semantic Web framework*[2] and the *ARQ SPARQL processor*.[3] As the WIQA framework relies on ARQ for graph pattern matching, it can take advantage of performance gains of ARQ's graph pattern matching algorithm.

The implementation of the WIQA framework is available under the terms of the GNU General Public License and can be downloaded from the WIQA website.[4]

7. The WIQA browser

The WIQA browser is an example application that uses the WIQA framework. The browser demonstrates how information quality filtering capabilities can be integrated into a standard web browser. The browser enables users to extract structured information from web pages. Extracted information from different web pages is stored in a local repository and can be browsed, sorted, and searched. The content of the local repository can be filtered using quality-based information filtering policies. In order to help users to understand the filtering decisions, the browser can display explanations of why a piece of information satisfies a selected policy.

The WIQA browser is based on the Piggy Bank extension for the Firefox web browser developed by the SIMILE project at the Massachusetts Institute of Technology [16]. The WIQA browser uses Piggy Bank functionality to extract structured information from web pages and to display and navigate extracted information. The WIQA browser stores information together with provenance meta-information as a set of Named Graphs. It uses the WIQA—Filtering and Explanation Engine to filter stored information and to generate explanations about filtering decisions.

While a user browses the web, the WIQA browser runs in the background and analyzes the visited web pages. Whenever the browser can extract structured information items from a page, it displays a data coin icon in the status bar, indicating that the user can switch to an *information item view* of the web page. This view lists all information items that have been extracted from the page. Next to each item is a save button that stores the item in the local repository.

Whenever the user saves information from a webpage into the local repository, the browser creates a new Named Graph for this visit of the page and stores the current timestamp, the URL of the page, and the authority (website URL) together with the actual information. The new graph is named with an UUID [25]. Provenance meta-information is represented using the Semantic Web Publishing vocabulary and the Dublin Core [30] vocabulary.

In a 1997 note on web browser user interfaces, World Wide Web inventor Tim Berners-Lee envisioned that browsers could offer what he called the "Oh, yeah?"-button [3]. Whenever a surfer questions the quality and trustworthiness of displayed information, she should press this button and the browser would display an explanation of why information should be considered trustworthy. The WIQA browser realizes the "Oh, yeah?"-button.

The user can load a WIQA policy suite into the browser. After selecting a policy from a selection panel, the content of the local repository is filtered and the view is updated to show only information matching the policy. As shown in Fig. 8, a small button is

[1] http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/.
[2] http://jena.sourceforge.net/.
[3] http://jena.sourceforge.net/ARQ/.
[4] http://www.wiwiss.fu-berlin.de/suhl/bizer/wiqa/.

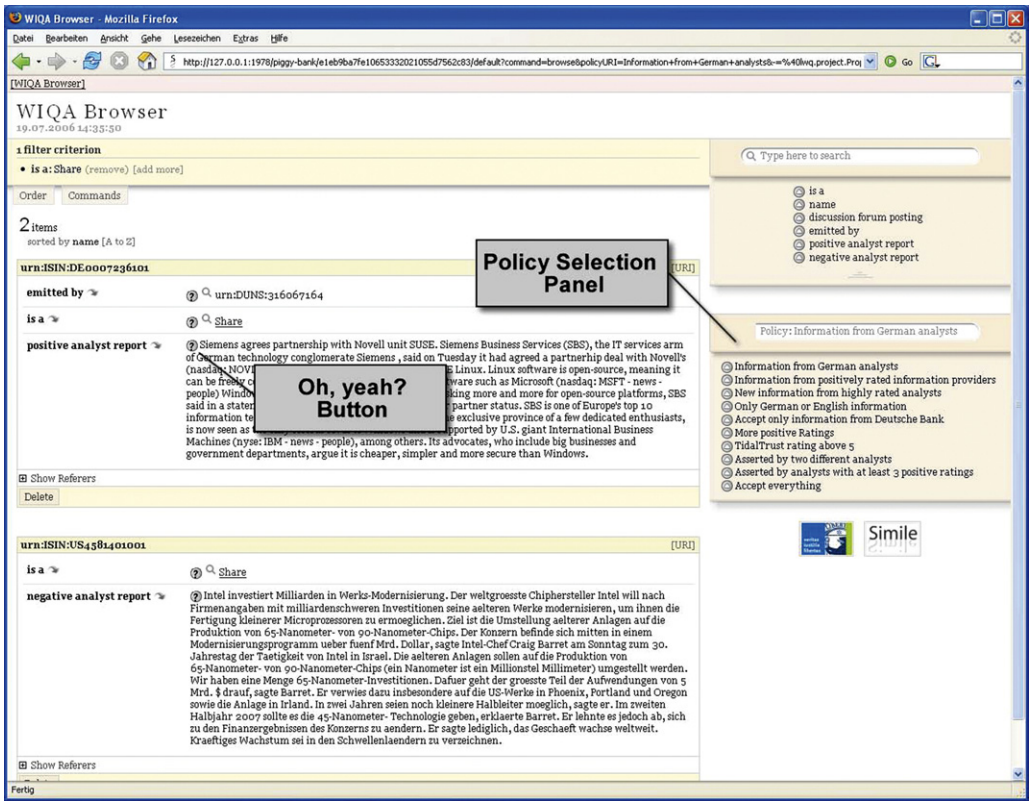

Fig. 8. When the user selects a policy from the policy selection panel on the right-hand side, the left-hand view updates to show only matching information. The "Oh, yeah?"-buttons open new windows with explanations of why a piece of information satisfies the selected policy.

located next to each piece of information. Pressing any of these buttons opens a new window that displays an explanation generated using the active policy.

8. Evaluation

The WIQA framework and the WIQA browser are working prototypes. In order to verify the utility of the Named Graphs data model to represent information together with quality-related meta-information and to demonstrate the utility of the WIQA-PL policy language, we did a preliminary evaluation of the WIQA browser against a financial information integration use case.

We gathered a test dataset consisting of financial news, discussion forum post and information about analyst reports from Yahoo Finance[5] and Google Finance[6] using screen scraping techniques. The resulting dataset represents provenance information using the same techniques as the example graph set shown in Fig. 4. The dataset was enhanced with artificial ratings for users and analysts. The dataset is available on the WIQA browser website.[7]

We developed a set of information filtering policies that take advantage of the different types of quality-related meta-information that are available in the test dataset. This policy suite is also found on the WIQA browser website.

The evaluation of the Named Graphs data model and the WIQA-PL policy language against the use case provided a first indicator for the utility of both technologies. Using the Named Graphs data model, it was possible to represent primary information, quality-related meta-information as well as background information about information providers as a compact, integrated model. WIQA-PL's graph pattern approach proved suitable for expressing complex information provenance oriented policies against the model. The extension function mechanism proved suitable for extending the WIQA framework with different rating-based scoring functions that were required by the use case.

As a next step, we plan to evaluate the WIQA framework against further use cases. Current candidates are news portals, search engines within knowledge management systems, and online communities that are used by large numbers of information providers to share information.

9. Related work

This chapter compares the WIQA framework with related work.

9.1. Database views

Accepted graphs within the WIQA framework can be compared to views in the context of relational databases. WIQA's explanation capabilities relate to work within the database community on explaining data lineage and view generation. An approach to explaining view generation has been developed by Cui and Wisdom in the context of the Stanford University WHIPS data warehousing system [12]. For a given data item in a materialized view, the authors propose a lineage tracing algorithm to identify the set of source data items that produced the view item. The algorithm is applicable to aggregate-select-project-join views and can be employed by data warehouse analysis tools to provide drill-down functionality from view items to source data items. What distinguishes the WIQA framework from the work within the relational data base community is the underlying data model. By employing a variation of the

[5] http://finance.yahoo.com/.
[6] http://finance.google.com/.
[7] http://sites.wiwiss.fu-berlin.de/suhl/bizer/wiqa/browser/.

RDF data model, the WIQA framework is tailored towards the integration of heterogeneous information from the web. For instance, integrating two partial descriptions of the same object while keeping track of the provenance of different pieces of information can simply be achieved within the Named Graphs data model, but is tricky using the relational data model.

9.2. Inference Web

A related approach to explaining information quality in the context of web-based information systems has been developed by the Inference Web project at the Stanford University Knowledge Systems Laboratory [27]. The project aims at making query answers more transparent by providing explanations about information sources as well as inference processes that are used to derive query results. The Inference Web infrastructure includes a registry containing details on information sources, reasoners, languages, and rewrite rules; a portable proof specification language; and a proof and explanation browser. Inference Web and the WIQA framework assume different application scenarios. While the WIQA framework is tailored towards a simple web-based information integration scenario, Inference Web assumes an agent community consisting of actively reasoning agents that cooperatively derive query answers from shared knowledge. Therefore, Inference Web focuses on explaining distributed reasoning paths [26], while the WIQA framework generates explanations of why subjective information filtering policies are satisfied.

9.3. TRELLIS

A further system that employs the RDF data model and provides information quality assessment functionality is the TRELLIS information analysis tool [14] developed at the University of Southern California. TRELLIS aims at supporting intelligence analysts in selecting quality information within a military setting. As an analyst makes a decision on which sources to dismiss and which to believe, TRELLIS captures the derivation of the decision in a semantic markup. The system then uses these annotations to derive an information quality assessment of the source based on the annotations of many individuals. Compared with the WIQA framework, TRELLIS supports only one fixed ontology for capturing quality-related meta-information and a single hard-coded assessment policy, while the WIQA framework is ontology independent and may employ a wide range of different filtering policies.

9.4. REI, KAoS and PROTUNE

There are several well established frameworks for enforcing access control policies within Semantic Web settings: REI [21] provides for positive and negative authorization as well as obligation policies and supports remote policy management. The KAoS framework [2] is aimed at agent environments and provides a service architecture for constraining the execution of actions in relation to various aspects of the situation. Within both frameworks, policies are expressed using a specific policy ontology. The PROTUNE framework [7] uses a rule-based policy language for specifying access control policies, privacy policies and certain classes of business rules.

All three frameworks assume a different application scenario from the WIQA framework. The frameworks are designed for situations where agents actively exchange messages in order to determine whether certain actions can be performed or certain information should be accessible. In contrast, the WIQA framework is designed for classic Web scenarios where information has been published by different information providers and where information consumers want to determine the subset of the available information that fulfills their information quality requirements.

Information quality assessment relies on different types of meta-information such as fine-grained provenance information or provenance chains. The WIQA framework provides for the representation of complex meta-information by employing the Named Graphs data model. REI, KAoS and PROTUNE use pure triple-based data models for the internal representation of information. This makes it difficult to represent meta-information about specific sets of triples. For instance, it is difficult to represent partial descriptions of the same object while keeping track of the provenance of different pieces of information or to model provenance chains which require the representation of meta-information about meta-information.

9.5. Almendra and Schwabe

An approach for translating abstract information quality requirements into concrete filtering policies is presented by Almendra and Schwabe from the Pontificia Universidade Catolica do Rio de Janeiro in Ref. [1]. Their work is based on the work presented in this article. It also employs the Named Graphs data model, the Semantic Web Publishing Vocabulary and the TriQL.P policy language [6], an earlier version of the WIQA-PL policy language. In addition to our work, where each policy must explicitly specify all the conditions that triples must fulfill to be accepted, they propose an ontology for expressing information quality requirements (TrustPoints) and an algorithm that automatically derives information filtering policies from these requirements by combining policy fragments. Given adequate tool support, their translation mechanism provides a valuable extension to our work as it reduces the technical knowledge required from a policy author.

10. Conclusion

This article highlighted the need of web-based information systems for task-specific information filtering policies and gave an overview of different metrics that can be used to assess information quality in the context of web-based information systems. We demonstrated how the Named Graphs data model can be employed to represent information together with quality related-meta information as an integrated model. We developed the WIQA-PL policy language which closely couples information representation and policy formulation and provides for expressing content-, context- and rating-based assessment policies.

As future work, we plan to integrate the WIQA framework into further applications besides the WIQA browser. Interesting candidates for being upgraded with information quality filtering capabilities are news portals and newsfeed aggregators, search engines within knowledge management systems, and online communities that are used by large numbers of information providers to share information.

A further, more visionary application domain for the WIQA framework is the Semantic Web as a whole. A growing number of data sources have started to expose their content as Linked Data [11]. As this number further increases, technologies for selecting high-quality information from this web of data are needed and it would be exciting to integrate the WIQA framework into a Semantic Web search engine, which crawls data from different sources and provides query capabilities over crawled data.

References

[1] V. Almendra, D. Schwabe, Real-world Trust Policies, in: Proceedings of the Semantic Web Policy Workshop, 2005.

[2] A. Uszok, et al., 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 2003), 2003.
[3] T. Berners-Lee, Cleaning up the User Interface, Section—The "Oh, yeah?"-Button, 1997, http://www.w3.org/DesignIssues/UI.html.
[4] T. Berners-Lee, Linked Data, 2006, http://www.w3.org/DesignIssues/LinkedData.html.
[5] C. Bizer, The TriG Syntax, 2005, http://www.wiwiss.fu-berlin.de/suhl/bizer/TriG/.
[6] C. Bizer, R. Cyganiak, T. Gauss, O. Maresch, The TriQL.P Browser: filtering information using context-, content- and rating-based trust policies, in: Semantic Web and Policy Workshop at the 4th International Semantic Web Conference, 2005.
[7] P.A. Bonatti, D. Olmedilla, Policy language specification. project deliverable d2, working group i2, eu noe rewerse, Tech. rep., 2005.
[8] J. Carroll, C. Bizer, P. Hayes, P. Stickler, Named Graphs, Journal of Web Semantics 3 (4) (2005) 247–267.
[9] Y. Chen, Q. Zhu, N. Wang, Query processing with quality control in the World Wide Web, World Wide Web Journal 1 (4) (1998) 241–255.
[10] C. Bizer, Quality-driven information filtering in the context of web-based information systems, Ph.D. Thesis, Freie Universität Berlin, 2007.
[11] C. Bizer, et al., Interlinking open data on the web, in: Poster at 4th European Semantic Web Conference, 2007.
[12] Y. Cui, J. Widom, Practical lineage tracing in data warehouses, citeseer.ist.psu.edu/article/cui99practical.html, in: Proceedings of the 16th International Conference on Data Engineering, 2000.
[13] M. Eppler, P. Muenzenmayer, Measuring information quality in the web context: a survey of state-of-the-art instruments and an application methodology, in: International Conference on Information Quality, 2002.
[14] Y. Gil, V. Ratnakar, Trusting information sources one citizen at a time, in: Proceedings of the 1st International Semantic Web Conference, 2002.
[15] I. Herman, W3C Semantic Web Activity, 2006, http://www.w3.org/2001/sw/.
[16] D. Huynh, S. Mazzocchi, D. Karger, Piggy Bank: experience the semantic web inside your web browser, in: Proceedings of the 4th International Semantic Web Conference, 2005.
[17] I. Jacobs, N. Walsh, Architecture of the World Wide Web, Volume One, W3C Recommendation, 2004, http://www.w3.org/TR/webarch/.
[18] J. Golbeck, Computing and applying trust in Web-based Social Networks, Ph.D. Thesis, University of Maryland, 2005.
[19] A. Jøsang, R. Ismail, C. Boyd, A Survey of Trust and Reputation Systems for Online Service Provision, 2006, URL citeseer.ist.psu.edu/738255.html.
[20] J. Juran, The Quality Control Handbook, 3rd ed., McGraw-Hill, New York, 1974.
[21] L. Kagal, T. Finin, A. Joshi, A policy-based approach to security for the semantic web, in: Proceedings of 2nd International Semantic Web Conference (ISWC'03), LNCS 2870, Springer, 2003.
[22] G. Klyne, J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax—W3C Recommendation, 2004, http://www.w3.org/TR/rdf-concepts/.
[23] S.-A. Knight, J. Burn, Developing a framework for assessing information quality on the World Wide Web, Informing Science Journal 8 (2005) 160–172.
[24] E. Knorr, R. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, International Journal on Very Large Data Bases 8 (3–4) (2000) 237–253, citeseer.ist.psu.edu/knorr00distancebased.html.
[25] P. Leach, M. Mealling, R. Salz, RFC 4122: A Universally Unique IDentifier (UUID) URN Namespace, 2005, http://tools.ietf.org/html/4122.
[26] D. McGuinness, Explaining reasoning in description logics, Ph.D. Thesis, Rutgers—The State University of New Jersey, 1996, URL citeseer.ist.psu.edu/mcguinness96explaining.html.
[27] D. McGuinness, P. da Silva, Infrastructure for web explanations, citeseer.ist.psu.edu/mcguinness03infrastructure.html, in: Proceedings of the 2nd International Semantic Web Conference, 2003.
[28] F. Naumann, From databases to information systems—information quality makes the difference, citeseer.ist.psu.edu/naumann01from.html, in: Proceedings of the 6th International Conference on Information Quality, 2001.
[29] F. Naumann, Quality-Driven Query Answering for Integrated Information Systems, Springer, Berlin/Heidelberg/New York, 2002.
[30] M. Nilsson, A. Powell, P. Johnston, A. Naeve, Expressing Dublin Core Metadata Using the Resource Description Framework (RDF)—Dublin Core Working Draft, 2006, http://dublincore.org/documents/dc-rdf/.
[31] J. Pérez, M. Arenas, C. Gutierrez, Semantics and complexity of sparql, in: International Semantic Web Conference, 2006.
[32] B. Pernici, M. Scannapieco, Data quality in web information systems, Journal on Data Semantics 1 (2003) 48–68.
[33] L. Pipino, Y. Lee, R. Wang, Data quality assessment, Communications of the ACM 45 (2002) 211–218.
[34] L. Pipino, R. Wang, D. Kopcso, W. Rybold, Developing Measurement Scales for Data-Quality Dimensions, M.E. Sharpe, New York, 2005.
[35] E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF, 2005, http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/.
[36] T. Redman, Data Quality for the Information Age, Artech House, London, 1996.
[37] D. Strong, Y. Lee, R. Wang, Data quality in context, Communications of the ACM 40 (5) (1997) 103–110.
[38] R. Wang, D. Strong, Beyond accuracy: what data quality means to data consumers, Journal of Management Information Systems 12 (4) (1996) 5–33.
[39] R. Wang, M. Ziad, Y. Lee, Data Quality, Kluwer Academic Publishers, Norwell, 2000.
[40] Q. Zhang, T. Yu, K. Irvin, A Classification scheme for trust functions in reputation-based trust management, in: Proceedings of the Workshop on Trust, Security, and Reputation on the Semantic Web, 2004.
[41] C.-N. Ziegler, Towards decentralized recommender systems, Ph.D. Thesis, Universität Karlsruhe, 2005.