# Quality Assessment Social Networks: A Novel Approach for Assessing the Quality of Information on the Web

Tomáš Knap
Department of Software Engineering
Faculty of Mathematics and Physics
Charles University
Malostranské nám. 25
Prague, Czech Republic
tomas.knap@mff.cuni.cz

Irena Mlýnková
Department of Software Engineering
Faculty of Mathematics and Physics
Charles University
Malostranské nám. 25
Prague, Czech Republic
irena.mlynkova@mff.cuni.cz

## ABSTRACT

The unprecedented volume of information on the Web brings the practical difficulties for the information consumers to assess the quality of information. The wide variety of web users and distinct situations at their hands pose the difficulties to the quality assessment (QA) process, which must be customizable according to the information consumer's needs. To that end, we (1) introduce the Web Quality Assessment model formalizing the QA process driven by the customizable sets of QA policies and (2) propose a novel concept of the QA social networks improving the QA process by reinforcing the number of relevant QA policies successively applied to the consumed resources by sharing the QA policies between users who are trusting each other. Since the ability to assess the information quality will play a fundamental role in the continued evolution of the Web, we are convinced that the concept of QA social networks can contribute in this area.

## 1. INTRODUCTION

The advent of Linked Data [8] in the recent years accelerates the evolution of the Web into a giant information space where the unprecedented volume of resources will offer to the information consumer a level of information integration and aggregation 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 an associate cost which will ultimately affect decision making, system usage and uptake.

Therefore, 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 [23].


*VLDB '10,* September 13-17, 2010, Singapore


IQ is usually described in different works by a series of *IQ dimensions* which represent a set of desirable characteristics for an information resource [3, 27, 5, 23]. Wang & Strong [27] present an extensive survey of IQ dimensions, based on the results of the questionnaire given to the panel of human subjects; papers [3, 2] cover IQ dimensions for the Web. In this paper, we neither intend to present our set of IQ dimensions for the Web, nor discuss the current sets of IQ dimensions; we merely exploit the universal way how the quality of unprecedented volume of resources on the Web can be assessed and mediated to the information consumer.

According to traditional practices on the Web, the information consumers "make judgments about which sources to rely on based on prior knowledge about a source's perceived reputation, or past personal experience about its quality relative to other alternative sources" [14]. This approach is impractical, (1) having scalability issues with the increasing information available on the Web and (2) eliminating a priori any kind of non-human agents, where only the automated judgments are feasable.

Naumann and Rolker [23] argue that assessing IQ is rightly considered difficult, because IQ dimensions are "often of subjective nature and can therefore not be accessed automatically". Knight and Burn [3] argue similarly that the perception of IQ is highly dependent on the fitness for use being relative to the specific task that users have at their hands. Furthermore, the Web is emerging as a global information space where the documents and data can be reused, aggregated and interconnected in new and unexpected ways, thus, we have to cope with the data with the beforehand unknown structure.

To that end, our approach is based on the idea of customizable set of *QA policies*, which drive the *QA process* assessing the quality of the resources based on the set of IQ dimensions preferred by the information consumer. The result of the QA process is the *QA score*, a metric directly determining the rank of the resource among other consumed resources, and the *QA color* of the resource, which serves as a first signal to the information consumer whether the resource is trustworthy (green color), trustworthy with warnings (yellow), unknown, in the sense that no QA policies were successfully applied to this resource (grey), or distrusted (red). QA policies belong to the *QA analyses* – implementations of one or more IQ dimensions. Every QA analysis is associated with the customizable weight – a coefficient determining how much the result of the analysis

influences the final QA score. The set of QA analyses and policies, unambiguously determining the QA process, is hold in the *QA profile* of an information consumer.

From the information consumer's perspective, the basic workflow of the QA process is as follows. Firstly, the consumer defines his QA profile determining which QA analyses and policies should participate in the QA process and queries the Web. Consequently, the set of resources relevant to the query is returned, based on the result of the conventional search engine, such as Google[1], or semantic search engine, such as Sindice[2]. In this moment, the QA process is entered – the QA analyses and policies specified in the consumer's QA profile are fetched, the QA analyses are launched, and the QA policies belonging to these analyses are tried to be applied to the resources; the successfully applied QA policies are reflected in the QA score and QA color computed for each processed resource. Afterwards, the resources are ranked according to their QA score and returned to the information consumer together with the information about their QA color and the set of successfully applied policies.

The idea of policies and rules is not novel and was successfully used by variety of access control mechanisms [19, 22, 11]. What is novel in our QA process is the concept of *QA social network* representing a social network [24, 20] an information consumer is part of. Every user of the particular social network can (1) specify his QA profiles and (2) express how much he trusts his acquaintances in various areas of human expertise [18]. Consequently, such a network can be used to inherit the QA profiles of the consumer's acquaintances; the QA policies defined inside these QA profiles can be used in the QA process together with the consumer's QA policies. The hypothesis is that if every user defines a few QA policies and pinpoints a few users in the QA social network he trusts in, the QA process successfully applying more QA policies yields to better ranking results.

Therefore, the goal of this paper is the definition of the generic *Web Quality Assessment model* (*WQA model*) – a model that can be used for ranking resources based on their QA score. The definition of the WQA model enables us to formally introduce the concepts of QA analyses, policies, and profiles, explain the application of QA policies and the function of the QA social networks. Our main contribution is therefore twofold:

- A generic Web Quality Assessment model for ranking resources according to their QA score
- A novel concept of QA social networks, the persistence of such networks, and the incorporation of such networks to WQA model.

The paper is organized as follows. The rest of Section 1 illustrates the WQA model concepts on the financial motivational scenario. Section 2 presents the *Restricted WQA model*, the WQA model not utilizing the concept of QA social networks. Further, Section 3 defines the concept of QA social networks, their persistence and utilization as part of the WQA model. Section 4 experimentally evaluates the contribution of the QA social networks to the ranking of the resources. Section 5 reviews related work and the paper is rounded off with a conclusion and discussion of future work in Section 6.

[1]http://www.google.com
[2]http://sindice.com/

## 1.1 Financial Scenario

We have chosen the *financial scenario* (*FS*) to illustrate the basic concepts of WQA model, especially the QA social networks. We have chosen the scenario in which the QA is crucial – the poorer quality of a resource, the bigger finance losses of a company.

Alice, a financial analyst and the information consumer, is preparing the financial report focusing on the summarized overview of the economic context of the executives of Fortune500 companies in the first quarter 2010. To achieve her goal, she is using a Linked Data [8] *mashup application* consuming and aggregating information included in stock markets time series, financial reports, government data, demographics, previous analysis, and third-party qualitative and quantitative analysis, physically distributed around the globe in various resources, such as RDF/XML [6] documents, named graphs [12], database tables, and XHTML pages. The mashup application accommodates an implementation of the WQA model helping Alice to decide the right resource for her task at hand by ranking the consumed resources according to their QA scores.

Since provenance or lineage of resources provides Alice the necessary contextualization to analyze the quality of the provided information [21, 16, 13] and allows the transfer of trust from entities behind the resources to the information in the resources [14], she decides to accommodate the Data Provenance QA analysis to her QA profile. Consequently, she defines the *QA criteria* (*QAC*) of the Data Provenance QA analysis as follows:

(1) Raw stock market information published by the official operators NYSE, LSE, NIKKEI, BOVESPA[3] is considered trustworthy. (2) Any news excerpts which do not introduce the source they quote are considered distrusted. Up-to-date news (w.r.t the first quarter 2010) are preferred. (3) Open resources using metadata prescribed by DBpedia[4] are prioritized. (4) The GovTrack.us, Geonames, and US Census Bureau[5] open resources are considered trustworthy. (5) Resources containing qualitative and quantitative analyses must be signed by a secure algorithm for digital signatures. All resources' hosts must provide a certificate issued by a trustworthy Certification Authority. (6) Signed resources are preferred when several contradicting statements occurs. (7) Due to the strategic importance of the report, the qualitative analysis generated by junior analysts are considered untrustworthy and are not used in the report. (8) Trends published by NYSE and supported by at least three subjects are considered stronger and prioritized.

QAC (1 – 8) directly corresponds with one or more QA policies. The range of QA policies available is strongly impacted by the used provenance model for tracking provenance on the Web [16, 13]. Although it is not mandatory for Alice, we suppose that she uses X-Prov Provenance Model[6], introduced in our previous work [13].

Further, let us assume that Alice ($A$) is part of the social network $SN$ formed by Alice's colleagues working in

[3]www.nyse.com/, http://www.londonstockexchange.com/, http://e.nikkei.com/, http://www.bmfbovespa.com.br/
[4]http://dbpedia.org/
[5]www.govtrack.us/, www.geonames.org/, www.census.gov/
[6]http://www.prov4j.org/xprov/

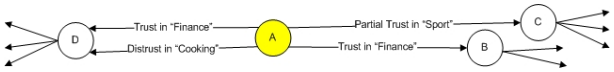


**Figure 1: Sample social network Alice is part of.**

the financial sector and her other friends. Alice expresses in the social network $SN$ her trust in colleagues Bob ($B$) and David ($D$) regarding "finance", partial trust in Cyril ($C$) regarding "sport", and distrust in David regarding "cooking" (these relations are called *trust relations*). Similarly, Bob, Cyril, and David can express trust relations to other people. Since it has been widely documented that social networks have the properties of small world networks, where the average distances between nodes in the network are small and connectance of nodes is high [24, 20], there is a chance of close (indirect) relationship of Alice and other financial experts. If we assume that all users in the social network $SN$ define their own QA profiles, such a social network $SN$ is the QA social network.

## 2. RESTRICTED WQA MODEL

In this section, we formally describe (1) the Restricted Web Quality Assessment model (RWQA model), which is equal to the WQA model not including the concept of QA social networks, and (2) the simple ranking algorithm *SRANK* using the RWQA model to rank resources on the Web.

A resource, $r \in R$, is a basic unit of consumed information to which QA policies can be applied. Every resource is identified by the unique URI in a format which is compatible with Linked Data principles for creating URIs[7] [8]. Every resource $r$ is supplemented with *score* and *color* for holding the QA score and QA color of the resource (these variables are accessible as $r.score$, respectively $r.color$). Furthermore, we suppose that every resource is annotated by appropriate metadata holding the *topic* of the resource[8], for example by utilizing Dublin Core Metadata Initiative[9]; the topic of the resource $r$ is accessible as $r.topic$. The sample topics (introduced in FS) are *finance*, *sport*, and *cooking*.

Let $A$ denote the set of QA analyses and $P$ the set of QA policies. An information consumer, $c \in C$, who queries the Web and consumes the resources $R_q \subseteq R$ retrieved from the Web as the answer on the query $q \in Q$, can specify the QA profile $f \in F_c$ which drive the QA process ($F_c$ is the set of all QA profiles specified by the consumer $c$). Every QA profile $f$ contains a set of analyses $A_f \subseteq A$ and a set of QA policies $P_f \subseteq P$ associated with $f$. Every QA profile $f$ must be associated with one of the five *quality requirement* ($QR$) *levels* (accessible as $f.qrl$). Various QR levels cover the demand on different requirements on the quality of the resources in different situations. Table 1 lists the admissible QR levels together with their sample usage and the financial loss associated with the chosen level. The financial loss should be the primary factor for determining to which QR level the QA profile belongs.

The designer of the analysis $a \in A$ must define a set of templates, $T_a \subseteq T$ ($T$ represents the set of all templates), in order to prescribe the form in which the QA policies belonging to the analysis $a$ may be constructed. The QA policy $p^t \in P$ denotes the policy $p \in P$ together with an additional information that the policy $p$ was created as an instance of the template $t \in T$. The algorithm behind the execution of the analysis $a$ is constructed with templates $T_a$ in mind, and, thus, any $q^x \in P$, can be applied to a resource $r \in R_q$ during the execution of the analysis $a$ as long as $x \in T_a$. Every QA policy must be created as an instance of the particular template $t \in T$.

**Table 1: Quality Requirement Levels**

| QR level | Sample usage | Financial loss |
|---|---|---|
| Crucial (4) | Creating financial report driving the future business | Significant |
| Important (3) | Completing product dimension in the data warehouse | Small |
| Normal (2) | Preparing homework for the university course | Insignificant / No |
| Below Normal (1) | Arguing with a friend about the ice hockey match result | No |

A QA policy $p^t \in P$ is a 4-tuple ($qaLev$, $qaWhat$, $qaCond$, $qaVal$), where $qaLev \in QAL^t$, $qaWhat \in QAW^t$, $qaCond \in QAC^t$, and $qaVal \in QAV^t$. The elements of the 4-tuple are referenced as $p^t.qaLev$, $p^t.qaWhat$, $p^t.qaCond$, $p^t.qaVal$. The template $t \in T$ particularizes the sets $QAL^t$, $QAW^t$, $QAC^t$, and $QACV^t$ with the admissible values for the instances of the template $t$. The $l \in QAL^t \subseteq \{TRUST(T), DISTRUST(D), PRIORITIZE(P), DEPRIORITIZE(DP)\}$ expresses whether the user wants to trust, prioritize, deprioritize, or distrust the resource satisfying the condition $c \in QAC^t$ with the left operand $w \in QAW^t$ and the right operand $v \in QAV^t$. The conditions $c \in QAC^t \subseteq \{Eq, Less, LessEq, More, MoreEq\}$ successively correspond to the basic arithmetic operators $=, <, \leqq, >, \geqq$. If $w \in QAW^t$ is wrapped in quotation marks, it is an arbitrary string, otherwise it represents RDF class or property. If $v \in QAV^t$ is enclosed by quotation marks, it represents the particular string admissible as the right operand of $c \in QAC^t$; otherwise, it represents the name of the RDF class or property which can occur as the right operand of $c \in QAC^t$. The particular semantics of $v$ is defined by the content of $w \in QAW^t$ and $c \in QAC^t$; $v$ can involve special strings "$*$", "$[N]$", "$[Z]$", and "$[R]$" representing, successively, arbitrary string, natural number $\mathbb{N}$, whole number $\mathbb{Z}$, and real number $\mathbb{R}$. The characters ", $*$, [, ], and \ are metacharacters in $v$, similarly, " and \ are metacharacters in $w$; the metacharacters can be escaped by \.

To conduct an example of QA policies and templates, the QA policy $p^t = (T, publishedBy, Eq, "NYSE")$ is generated by QAC (1) in FS, where *publishedBy* is the RDF property (defined in XProv vocabulary [13]) for holding publishers of the resource. The template $t$ can be then ($\{T, D\}$, $\{publishedBy\}$, $\{Eq\}$, $\{"*"\}$). Obviously, various QA policies can be written in more concise way. For example, QAC (6) generates QA policy $q^s = (P, hasSignature$[10]$, More, "0")$, where the parameters $More \in QAC^s$ and "0" $\in$

[7] http://www.w3.org/DesignIssues/LinkedData.html
[8] The current activities in the area of data provenance and annotations show that these assumptions are being fulfilled by more and more resources, see [21] and http://twiki.ipaw.info/bin/view/Challenge.
[9] http://dublincore.org/
[10] RDF property defined in XProv vocabulary [13], http://www.prov4j.org/xprov/

$QAV^s$ are redundant and can be omitted. The advantage of having only single 4-tuple representation of QA policies is the easier formalism.

We say that the QA policy $p^t$ can be *successfully applied* to the resource $r$, if $w \in QAW^t$ satisfies (as the left operand) the condition $c \in QAC^t$ with the right operand $v \in QAV^t$. The application of the QA policy $p^t$ is represented by a function $a(p^t) : P \to \{false, true\}$. Every policy $p^t$ is successfully applied to the resource $r$ (i.e. the resource $r$ *satisfies* the QA policy $p^t$) if and only if $a(p^t) = true$; otherwise the QA policy $p^t$ cannot be successfully applied to the resource $r$.

We distinguish policies presented so far (called hereafter *simple policies*, denoted with a flag $p.simple = true$) and *compound policies* ($p.simple = false$). A compound policy $cp$ consists of a set of simple policies; the expression $|cp|$ denotes the number of simple policies *forming* (involved in) the compound policy $cp$. If $p_1, \ldots, p_{|cp|} \in P$ are simple policies forming the compound policy $cp$ (denoted as $p_i \propto cp$, $1 \leqq i \leqq |cp|$), then $a(cp) = a(p_1) \wedge \ldots \wedge a(p_{|cp|})$. Furthermore, $\forall p_i, p_j \propto cp$, $1 \leqq i, j \leqq |cp|$, $\neg cp.simple : p_i.qaLev = p_j.qaLev$. The compound policy $cp$ is written as a set of 4-tuples representing the simple policies forming $cp$. An example of the compound policy is the QA policy $cp^t = \{(P,$ *publishedBy*, $Eq$, "$NYSE$"), ($P$, *supportedBy*, $MoreEq$, "3")$\}$ generated by QAC (8) in FS. A simple policy $p^t$ always *belongs* to an analysis $a \in A$ based on the template $t \in T_a$ it instantiates (denoted as $p^t.belongsTo = a$). A compound policy $cp^t$ neither belongs to an analysis ($cp.belongsTo = \perp$), nor instantiates a template ($t = \perp$).

Every QA policy $p$ has an associated *importance* (accessible as $p.imp \in \mathbb{R}^+$); the bigger number denotes the higher importance. The importance of the simple policy $p$ is either defined explicitly or implicitly by inheriting the importance of the analysis $p.belongsTo$. The shortcut $a.imp$ refers to the importance of the analysis $a \in A$, $a.imp \in \mathbb{R}^+$. If the importance of the compound policy $cp$ is explicitly stated, it overrules any importance defined by the simple policies forming the compound policy. Otherwise, the importance of the compound policy $cp$ is computed as $cp.imp = \max_{i=1}^{|cp|}\{p_i.imp \mid p_i \propto cp\}$.

The effect of the successful policy application is quantified by a function $lScore : P \to \mathbb{R}$ according to Formula (1), where the function $bv : P \to (0, 1]$ represents the *basic value* associated with the successful application of the QA policy. The basic value is further meliorated by the importance of the QA policy. For a simple policy $p^t$, $bv(p^t) = t.bv$ ($t.bv$ is specified within the template $t \in T$). For a compound policy $cp$, $bv = \max_{i=1}^{|cp|}\{bv(p_i) \mid p_i \propto cp\}$. For example, $bv(p^t)$ can be equal to 1 for QA policy $p^t = (T,$ *publishedBy*, $Eq$, "$NYCE$") generated by QAC (1) in FS. On the other hand, $bv(q^s)$ can be equal to $1/m$ for QA policy $q^s = (P,$ *supportedBy*, $MoreEq$, "3") generated by the part of QAC (8) in FS prioritizing resources supported by at least three persons ($m$ is the total number of the consumed resources satisfying the policy $q^s$ as well).

Together with $lScore(p)$, we define a function $lColor : P \to \{GREEN,$ *YELLOW*, *RED*, *GREY*$\}$, which qualifies every successful policy application $a(p)$ to the resource $r$ with the $lColor(p)$ assigned for simple policy $p$ according to Formula (2) and for compound policy $p$ according to Formula (3). In Formula (3), $nG$ ($nR$) represents the number of simple QA policies $p_i \propto p$, $1 \leqq i \leqq |p|$, for which $lColor(p_i) = GREEN(RED)$.

$$lScore(p) = \begin{cases} \text{bv(p)*p.imp}, & \text{if } p.qaLev \in \{T, P\} \\ \text{-bv(p)*p.imp}, & \text{otherwise} \end{cases} \quad (1)$$

$$lColor(p) = \begin{cases} GREEN, & \text{if } p.qaLev = T \\ RED, & \text{if } p.qaLev = D \\ GREY, & p.qaLev \in P, DP \end{cases} \quad (2)$$

$$lColor(p) = \begin{cases} GREY, & \text{if } (nR = 0) \wedge (nG = 0) \\ GREEN, & \text{if } (nR = 0) \wedge (nG > 0) \\ RED, & \text{if } (nR > 0) \wedge (nR \geqq nG) \\ YELLOW, & \text{if } (nR > 0) \wedge (nR < nG) \end{cases} \quad (3)$$

## 2.1 QA Process and Algorithm SRANK

Algorithm 1 depicts the ranking algorithm SRANK utilizing the RWQA model. The input to Algorithm 1 is the QA profile $f \in F_c$ ($c \in C$) and $R_q$, the ordered set of resources returned as an answer on the consumer's query $q \in Q$. In Line 1, the QA policies *qaps* are taken from the QA profile $P_f$ of the consumer $c$; in Line 2, these policies are applied to the resources $R_q$ during the QA process executed according to Algorithm 2. The result of Algorithm 2 is the modified set $R_q$, where each resource $r_i \in R_q$, $1 \leqq i \leqq |R_q|$, is supplemented with *QA annotations* $r_i.score$ and $r_i.color$. In Line 3, the permutation $\phi : R_q \to \widehat{R_q}$, mapping the ordered set of resources $R_q$ to $\widehat{R_q}$, is applied to the set $R_q$ to sort the set $R_q$ according to decreasing QA scores of the resources. The ordered set $\widehat{R_q}$ is the output of Algorithm 1.

**Algorithm 1** Ranking algorithm SRANK

**Input:** $f \in F_c$, $R_q \subseteq R$,
**Output:** $\widehat{R_q} = rankedResources(f, R_q)$
1: $qaps \leftarrow P_f$
2: $R_q \leftarrow qualityAssessment(R_q, qaps)$
3: $\widehat{R_q} \leftarrow \phi(R_q)$
4: **return** $\widehat{R_q}$

Algorithm 2 depicts the QA process for the ordered set of resources $R_q$ and QA policies $\widetilde{P} \subseteq P$ which are going to be applied to the resources. The output (and the result of the QA process) is the set of resources $\widetilde{R_q}$ holding the QA annotations for each resource $r \in \widetilde{R_q}$.

Every resource $r$ holds in variables $r_G$, $r_Y$, $r_R$ (initiated in Line 3 in Algorithm 2) the number of GREEN, YELLOW, and RED colors as the results of the function $lColor$. In Lines 4 – 11, the policies $\widetilde{P}$ are tried to be applied to the resource $r$. If they are successfully applied (Line 5), the function $lScore$ is computed and added to $r.score$ (Line 6). Consequently, in Lines 7 – 9, the function $lColor$ is computed and the appropriate color counter is increased. In Line 12, a function $c : R \to \{GREEN,$ *YELLOW*, *GREY*, *RED*$\}$ computes the QA color for the resource $r$ according to Formula (4) by comparing the variables $r_G$, $r_Y$, and $r_R$.

$$c(r) = \begin{cases} \text{GREY}, & \text{if } (r_R = r_Y = r_G = 0) \\ \text{GREEN}, & \text{if } (r_R = r_Y = 0) \wedge (r_G > 0) \\ \text{RED}, & \text{if } (r_R > 0) \wedge (r_R \geqq (r_G + r_Y)) \\ \text{YELLOW}, & \text{otherwise} \end{cases} \quad (4)$$

**Algorithm 2** Quality Assessment Process

**Input:** $R_q \subseteq R$, $\widetilde{P} \subseteq P$
**Output:** $\widetilde{R_q} = qualityAssessment(R_q, \widetilde{P})$

```
1: for all resources r ∈ R_q do
2:    r.score ← 0
3:    r_G, r_Y, r_R ← 0
4:    for all QA policies p ∈ P̃ do
5:       if a(p) then
6:          r.score ← r.score + lScore(p)
7:          if lColor(p) = GREEN then
8:             r_G ← r_G + 1
9:          end if
            Similarly for YELLOW, and RED; r_Y, and r_R
10:      end if
11:   end for
12:   r.color ← c(r)
13: end for
14: R̃_q ← R_q
15: return R̃_q
```

# 3. WQA MODEL

In this Section, we define the WQA model by extending the RWQA model with the concept of QA social networks. A *social network* [24, 20] is typically modeled as a directed graph, where the vertices of the graph represent entities of the network (persons, groups, organizations) and the edges relations between these entities [15, 28]. The social network provides an important mechanism for sharing QA policies (grouped in QA profiles) between the entities participating in the social network. For example, a person in a company may reuse the QA profile of his workgroup, inheriting the QA policies of that workgroup.

Such a social network, which enables an entity to specify the *trust* [18, 4] in another entity w.r.t the *r.topic* of the consumed resource $r$, is called *social trust network* ($STN$). The trust specified in another entity is quantified as *topic trust value* (*ttValue*) and an edge between two entities in STN is called *trust relation*. The $ttValue$, $ttValue \in [1, 9]$, is ranging from complete distrust ($ttValue = 1$) to complete trust ($ttValue = 9$), with $ttValue = 5$ expressing the neutral state – either neutral trust (neither positive, nor negative) or the absence of trust [18, 4]. The *QA social network* consists of the social trust network and QA profiles associated with the entities in STN.

Obviously, the information consumer is more willing to inherit QA profiles of entities he trusts more regarding the particular topic. For example, in the social network $SN$ visualized in Figure 1, Alice specifies that she trusts David regarding "finance". Thus, in case of the consumed resources $r$ with $r.topic =$ "$finance$", Alice prefers the QA profile of David before the QA profile of Cyril.

For the sake of simplicity, we assume in Subsections 3.1 – 3.3 that the entities in QA social network are only persons (= users) $U$, $C \subseteq U$. Groups of persons as entities are discussed in Subsection 3.4.

## 3.1 Persisting QA Social Networks

To enable the concept of QA social networks, we have to be able to persist the social trust networks and QA profiles associated with entities in STN.

Social networks coexisting in the Web information space[11] are typically tightly connected with the applications utilizing them (such as MySpace[12]). Contrary, the Friend of a Friend (FOAF) project[13] is an ontology to describe persons (instances of RDF class `foaf:Person`) and relations between them in an open and application independent way using RDF model. Moreover, in our previous work [18], we have developed an extension of FOAF ontology, called Topic-based Trust Module[14], having the capability to persist STN (the sample persistence of STN is in [18, p. 4 – 6]). Since there are no other open, application independent, machine readable, and widely used implementations of social networks with the possibility to persist trust relations, we decided to choose FOAF-based social network together with the Topic-based Trust Module extension.

In order to store QA profiles of users in the STN, we developed a FOAF extension module called Quality Assessment Profile Module[15], with the default namespace *foafq*. The main purpose of the Quality Assessment Profile Module is the capability to persist the content of QA profiles – QA policies, QA analyses, and QR levels. Moreover, Quality Assessment Profile Module defines a RDF property `foafq:hasQAProfile` to associate a QA profile with an instance of the RDF class `foaf:Person`.

## 3.2 Inheriting Profiles in QA Social Networks

So far, we were discussing how to persist QA social networks. This subsection formulates Algorithm 3 collecting (inheriting) QA policies from the QA profiles involved in the QA social network the consumer is part of. Algorithm 3 is then used in Algorithm 4 to improve the QA process and the ranking algorithm SRANK depicted in Algorithm 1.

Algorithm 3 collects QA policies $\widehat{P} \subseteq P$ for two users $c$, $u \in U$ and QA profile $f \in F_c$ ($c \in C$). The QA policies $\widehat{P}$ are formed by the policies $P_f$ of the information consumer $c$ with profile $f$ and *not conflicting* and *relevant* policies from QA profiles of his acquaintances (users $\widehat{U} \subseteq U$) *trusted* above the certain *threshold*. Which policies are relevant and not conflicting and which users in the QA social network are the users trustworthy enough to inherit QA profiles from is explained further.

In Line 1, Algorithm 3 executes the TTA algorithm explained in detail in our previous work [18]. For its execution, the algorithm TTA needs an object *ttap* - an input to Algorithm 3 - which has fields *ttap.topic*, *ttap.stn*, and *ttap.th*. The algorithm TTA is using the underlying social trust network *ttap.stn* to compute the topic trust value $ttValue \in [1, 9]$ of the information consumer $c$ in other user $u$ w.r.t the particular topic *ttap.topic*. The variable *ttap.th* refers to the hierarchy of topics which is used by the algorithm TTA. The topic hierarchies are further explained and discussed in [18, p. 6 – 7].

Since we want to inherit QA policies only from users we trust above the certain *threshold*, the computed $ttValue$ must be above this *threshold* (Line 2); the *threshold* $\geqq 5$ means that the QA profiles of untrustworthy users are not

[11] See [15, p. 10 – 28] for the list of web-based social networks
[12] http://www.myspace.com
[13] http://xmlns.com/foaf/spec/
[14] The full ontology can be downloaded at http://www.ksi.mff.cuni.cz/~knap/qa/ttm.owl
[15] The full ontology can be downloaded at http://www.ksi.mff.cuni.cz/~knap/qa/qapm.owl

inherited. In Line 4, QA policies from the QA profile $g$ are examined only if the QA profile $g \in F_u$ ($u \neq c$) has the same or higher (more strict) QR level (see Table 1) than the profile $f \in F_c$. If we find the corresponding profile $g \in F_u$, the policies $P_g$ are potential candidates for QA policies to be inherited and are further processed in Lines 5 – 17.

**Algorithm 3** Collection of QA policies from QA social network

**Input:** $Q \subseteq P$, $c \in U$, $f \in F_c$, $u \in U$, $ttap \in TTAP$
**Output:** $\widehat{P} = collectPolicies(Q, c, f, u, ttap)$

```
1: ttValue ← TTA(ttap.topic, c, u, ttap.stn, ttap.th)
2: if ttValue ≧ threshold then
3:    qaps ← Q
4:    if (c ≠ u) ∧ (∃g ∈ F_u) ∧ (g.rql ≧ f.rql) then
5:       for all QA policies p ∈ P_g do
6:          if (p.simple ∧ p.belongsTo ∈ A_f) ∨
            (¬p.simple ∧ (∀p_i ∝ p : p_i.belongsTo ∈ A_f))
            then
7:             if ∀q ∈ P_f : ¬(p ⊗ q) then
8:                p.ttValue ← ttValue
9:                if (p.imp = ⊥) ∧ p.simple then
10:                  p.imp ← p.belongsTo.imp
11:               else if (p.imp = ⊥) ∧ ¬p.simple then
12:                  p.imp ← max_{i=1}^{|p|} {p_i.belongsTo.imp |
                                                 p_i ∝ p}
13:               end if
14:               qaps ← qaps ∪ {p}
15:            end if
16:         end if
17:      end for
18:   end if
19:   for all users un ∈ u.neighbours do
20:      qaps ← qaps ∪ collectPolicies(qaps, c, f, un, ttap)
21:   end for
22:   return qaps
23: end if
```

The QA analyses executed during the QA process are fully determined by the profile $f$ of the consumer $c$. Therefore, the QA policy $p \in P_g$ is inherited only if it is *relevant* – the condition in Line 6 holds. Since the policies $P_f$ of the information consumer $c$ are considered as *authoritative*, the policy $p \in P_g$ is inherited only if it is *not conflicting* with any other policy $q \in P_f$ (denoted as $\neg(p \otimes q)$, see Line 7). Two simple policies $p^s$ and $q^t$ are *conflicting* ($p^s \otimes q^t$) if $(s = t) \wedge (p^s.qaLev \neq q^t.qaLev) \wedge (p^s.qaWhat = q^t.qaWhat) \wedge (p^s.qaCond = q^t.qaCond) \wedge (p^s.qaVal = q^t.qaVal)$. A simple policy $p$ is conflicting with a compound policy $cp$ ($p \otimes cp$) if $\exists p_i \propto cp$, $1 \leqq i \leqq |cp| : (p_i \otimes p)$. Two compound policies $cp$ and $cq$ are conflicting ($cp \otimes cq$) if $\exists p_i \propto cp$, $1 \leqq i \leqq |cp|$, $\exists q_j \propto cq$, $1 \leqq j \leqq |cq| : (p_i \otimes q_j)$.

If both conditions in Lines 6 and 7 are satisfied, the QA policy $p$ is added to the list of policies already inherited (Line 14). Before that, $ttValue$ computed in Line 1 is stored within the policy $p$ for further use (Line 8); if $p.imp$ is not defined, Lines 9 – 13 derive the value of $p.imp$.

In Lines 19 – 21, if the $ttValue$ computed by the TTA algorithm is above the given $threshold$, QA policies of the users, with whom the user $u$ is directly connected by a trust relation in STN ($u.neighbours$), are examined in the next iteration of Algorithm 3 and possibly added to the list of QA policies $qaps$.

## 3.3 QA Process and Algorithm RANK

Algorithm 4 depicts the ranking algorithm $RANK$ utilizing the WQA model. The input to Algorithm 4 is a consumer $c \in C$, a profile $f \in F_c$, a set of resources $R_q \subseteq R$ returned as an answer on the consumer's query $q$, and $ttap \in TTAP$, which represents an object holding information necessary for Algorithm 3 executed in Line 1[16]. The output of the Algorithm 4 is the ordered set of resources, $\widehat{R_q}$, firstly defined in Subsection 2.1.

**Algorithm 4** Ranking algorithm RANK

**Input:** $c \in C$, $f \in F_c$, $R_q \subseteq R$, $ttap \in TTAP$
**Output:** $\widehat{R_q} = rankedResources(c, f, R_q, ttap)$

```
1: qaps ← collectPolicies(P_f, c, f, c, ttap)
2: rqaps ← refinePolicies(qaps)
3: R_q ← qualityAssessment(R_q, rqaps)
4: R̂_q ← φ(R_q)
5: return R̂_q
```

In Line 1, the QA policies $qaps$, which are going to be applied to the resources $R_q$ during the QA process, are collected from the QA social network according to Algorithm 3 (compare this step with the equivalent step in Algorithm 1).

Algorithm 3 intentionally does not check whether two or more inherited policies are conflicting between each other and simply inherits them all. Algorithm *refinePolicies* in Line 2 ensures that $\nexists p_i \neq p_j$, $p_i, p_j \in qaps$, $1 \leqq i, j \leqq |qaps| : (p_i \otimes q_j)$ by choosing the prevailing QA policies from the sets of conflicting policies. The algorithm *refinePolicies* is rather technical, thus only the main idea is explained. In the first phase, all QA policies $qaps$ are separated to the classes of equivalence $\equiv$. In case of simple policies, the two policies $p^t$ and $q^s$ are equivalent ($p^t \equiv q^t$) as long as $(s = t) \wedge (p^s.qaWhat = q^t.qaWhat) \wedge (p^s.qaCond = q^t.qaCond) \wedge (p^s.qaVal = q^t.qaVal)$; the two compound policies $cp$ and $cq$ are equivalent if $\forall p_i \propto cp$, $\exists q_j \propto cq : p_i \equiv q_j$ and $\forall p_i \equiv q_j$, $p_k \equiv q_l$, $p_i \neq p_k : q_j \neq q_l$, $1 \leqq i, k \leqq |cp|$, $1 \leqq j, l \leqq |cq|$. Consequently, for each equivalence class, one arbitrary policy $q \in [q]_{\equiv}$ is selected. Further, every policy $p_i \in [q]_{\equiv}$, $1 \leqq i \leqq |[q]_{\equiv}|$, contributes with the value ($p_i.ttValue * p_i.imp$) to one of four $qaLev \in \{T, D, P, DP\}$ for which $p_i.qaLev = qaLev$. As we can see, the influence of the QA policy $p_i$ on the score of the particular $qaLev$ is influenced by the computed $p_i.ttValue$ and $p_i.imp$. Based on the $qaLev$ with the highest score, the resulting $q.qaLev$ is computed. The field $q.ttValue$, and $q.imp$ is no longer necessary and can be set to $\perp$. Such a policy $q$ is inserted to the refined set of policies $rqaps$ as a representant of $[q]_{\equiv}$.

In Line 3, the QA process itself is executed (Algorithm 2). Finally, in Line 4, the permutation $\phi$ is applied to the set of resources $R_q$ with the QA annotations present and the ordered set of resources $\widehat{R_q}$ is returned. Lines 3 and 4 are the same as Lines 2 and 3 in the algorithm SRANK depicted in Algorithm 1.

## 3.4 Group Entities in QA Social Networks

We have mentioned that entities in QA social networks can be persons or groups of persons. The previous chapters

[16]We suppose that all resources $r \in R_q$ have the same $r.topic$. If not, Algorithm 4 can be executed several times – each time the resources with the same topics are processed.

assumed that all entities are persons; in this Subsection, we present an extension working with *groups* of persons.

Every information consumer $c \in C$ can be part of several groups $g \in G$. Each $g$ can define its QA profile in a similar way as users can do. If the consumer $c$ is a member of the particular group $g$, during the process of collecting policies from QA social networks, the QA policies of the group $g$ are merged to the set of all policies used during the QA process of the consumer $c$. There is one exception from this behavior – if the particular group $g_a$ is an *authoritative group*, the policies of the information consumer, the other groups the information consumer is part of, and other users in QA social network are inherited only as long as they are not conflicting with any policy defined by group $g_a$. Obviously, the consumer is associated with at most one authoritative group. In other aspects, the groups behave like the users. An example of an authoritative group can be the organization where the analyst Alice is working – the organization as an entity can prescribe in its QA profile some QA policies mandatory to all employees.

Regarding the persistence of the groups, FOAF ontology contains RDF class `foaf:Agent` to represent a generic entity (a person or group) in STN and the property `foaf:member` to express the membership of the `foaf:Agent` to the given `foaf:Group`. Quality Assessment Profile Module can be easily extended with the RDF properties `foafq:memberOf` (represents the inverse property to the property `foaf:member`, which is suitable for tracking trust relations between persons and groups) and `foafq:authoritativeGroup` (holds the authoritative group of the person).

The problem is that the current version of the TTA algorithm [18] internally uses RDF property `foaf:knows`, which is defined only for instances of `foaf:Person`. Moreover, we would have to update (1) the algorithm TTA to accept trust relations defined by `foafq:memberOf` and (2) the Topic Trust Module to work with instances of `foaf:Agent`. Therefore, we decided to leave the support of the entity group as future work adherent to the new versions of the Topic Trust Module and the TTA algorithm.

# 4. EVALUATION OF THE WQA MODEL

In this part we would like to present the preliminary simulation of our WQA model to show that the inheritance of QA profiles in QA social networks brings improvements to the ranking algorithm. In our simulation, a consumer $c \in C$ constructs a query $q \in Q$ which yields to the ordered set $R_q \subseteq R$. Consequently, $R_q$ is an input to the QA process resulting in the ordered set $\widehat{R_q}$, defined by the permutation $\phi$ introduced in Subsection 2.1.

## 4.1 Initialization of the Model

For the simulation of the WQA model, we examined 10000 queries $q_i \in Q_I \subseteq Q$, $0 < i \leqq 10000$, the resources ($|R_q| = 10$)[17] in the answer for each $q_i$ are chosen randomly from a pool of 10000 resources (i.e. $|R| = 10000$), the information consumer $c \in C$ is chosen randomly from a pool of 1000 users ($|U| = 1000$), and the rest of the users form QA social network[18]. All the assignments above are performed using random selection, with replacement (i.e. the covariance between two consumers and two sets of resources selected for two different queries is zero), from uniform distributions over the sets $R$ and $U$.

Although the QA profile can accommodate a wide range of QA analyses, we utilize only the provenance analysis $a_p$[19]. The provenance analysis defines single template $t = (\{TRUST, DISTRUST\}, \{MC_i\}^{20}, \{Eq\}, \{MV_j\}^{21})$, $1 \leqq i \leqq |QAW^t|$, $1 \leqq j \leqq |QAV^t|$. Every resource $r$ is associated with $|QAW^t|$ RDF triples having each $w \in QAW^t$ as the RDF property and $v \in QAV^t$ as the property value. The property values of the triples are accessible as $r.triple[x].v$, $1 \leqq x \leqq |QAW^t|$; $v$ is chosen for each triple at random with replacement from uniform distribution over $QAV^t$.

The number of acquaintances of an user $u \in U$ in a QA social network is a random number chosen from a normal distribution with the chosen mean $\mu_A = 7$ and standard deviation $\sigma_A = 1$.

The topic trust value ($ttValue$) is not computed (since it is not an evaluation of the TTA algorithm), rather $ttValue$ is randomly chosen from a normal distribution with the mean $\mu_T$ and standard deviation $\sigma_T$. The propagation of trust and distrust in social networks was studied in [15, p. 76] in the Trust Project where "the average trust rating was 7.25, with a standard deviation of 2.30". Since [15] uses the same quantification of trust (the range [1..9]) and the social network in Trust Project is similar to our social trust network, we decided to put $\mu_T = 6.0$ (a more pessimistic mean value than in the Trust Project) and $\sigma_T = 2.3$.

The number of QA policies a user $u$ in a QA social network defines is a random number chosen from a normal distribution with the chosen mean $\mu_P = 7$ and standard deviation $\sigma_P = 1$. We ensure that each user defines at least one QA policy. All the generated policies are simple policies $p^t$, $p^t.imp = 1$, and $p^t.belongsTo = a_p$. For each such generated policy $p^t$, $p^t.qaWhat$, respectively $p^t.qaValue$, are selected at random with replacement from uniform distributions over $QAW^t$, respectively $QAV^t$. To deduce the $p^t.qaLev$, we introduce an ideal ordering of $QAV^t$ values according to a magical oracular that knows general and objective trustworthiness of the values $v \in QAV^t$. The ideal ordering is governed by $\gamma : QAV^t \to 1, 2, \ldots, |QAV^t|$, ranking the $QAV^t$ values from the most trusted to the most distrusted. For simplicity, we choose that $\gamma(MV_i) = i$, $1 \leqq i \leqq |QAV^t|$, i.e. $MV_i$ is ranked to the $i$-th position in the ideal ordering. We suppose that $MV_i$ values for $1 \leqq i \leqq (|MV_i|/2)$ are considered trustworthy (to some extent) and $MV_i$ for $(|MV_i|/2) < i \leqq |MV_i|$ are distrusted.

Table 2 describes how probable is that the user $u$ will choose the right $p^t.qaLev$ for the particular $v \in QAV^t$, i.e. $TRUST$ for the first half of the values and $DISTRUST$ for the rest of the values $QAV^t$ according to $\gamma$. The trust and distrust probabilities in Table 2 are based on the ideal ordering (the better the rank according to $\gamma$, the bigger the probability that the user will choose the right $p^t.qaLev$) and meliorated by $ttNorm \in [0, 1]$, a normalized $ttValue$ between the consumer and the user $u$, $ttNorm = (ttValue - 1)/8)$.

To conduct an example, suppose that $ttValue$ between

[17] We have tried various numbers of resources with very similar resulting graphs.

[18] The values for $R$ and $U$ were arbitrarily chosen to allow fast simulation.

[19] We assume that $a_p$ uses X-Prov provenance model [13], however, this fact is not important for the simulation.

[20] The particular names of admissible RDF properties are not important, we denote them as $MC_1 \ldots MC_n$, $|QAW^t| = n$.

[21] The particular names of objects in RDF triples are not

**Table 2: Probabilities for Generated Trust Levels**

| $\mathbf{v} \in \mathbf{QAV^t}$ | Trust prob | Distrust prob |
|---|---|---|
| $MV_1 - MV_5$ | $60 * ttNorm$ | $40 * (1 - ttNorm)$ |
| $MV_6 - MV_{10}$ | $58 * ttNorm$ | $42 * (1 - ttNorm)$ |
| $MV_{11} - MV_{15}$ | $56 * ttNorm$ | $44 * (1 - ttNorm)$ |
| $MV_{16} - MV_{20}$ | $54 * ttNorm$ | $46 * (1 - ttNorm)$ |
| $MV_{21} - MV_{25}$ | $52 * ttNorm$ | $48 * (1 - ttNorm)$ |
| $MV_{26} - MV_{30}$ | $48 * (1 - ttNorm)$ | $52 * ttNorm$ |
| $MV_{31} - MV_{35}$ | $46 * (1 - ttNorm)$ | $54 * ttNorm$ |
| $MV_{36} - MV_{40}$ | $44 * (1 - ttNorm)$ | $56 * ttNorm$ |
| $MV_{41} - MV_{45}$ | $42 * (1 - ttNorm)$ | $58 * ttNorm$ |
| $MV_{46} - MV_{50}$ | $40 * (1 - ttNorm)$ | $60 * ttNorm$ |

**Table 3: Determination of the Values for the Number of Policies (#Pol), Number of Acquaintances (#Acq), and the Topic Trust Value (ttValue)**

| Range | #Pol | #Acq | ttValue |
|---|---|---|---|
| $\|gv\| < \sigma$ | 7 | 7 | 6 |
| $(-2) * \sigma < gv \leqq (-1) * \sigma$ | 5 | 5 | 5 |
| $\sigma \leqq gv < 2 * \sigma$ | 9 | 9 | 7 |
| $(-3) * \sigma < gv \leqq (-2) * \sigma$ | 3 | 3 | 4 |
| $2 * \sigma \leqq gv < 3 * \sigma$ | 11 | 11 | 8 |
| $gv \leqq (-3) * \sigma$ | 1 | 1 | 3 |
| $gv \geqq 3 * \sigma$ | 13 | 13 | 9 |

consumer $c$ and user $u$ computed by TTA is 6 (i.e. $ttNorm = 0.625$), the user $u$ defines the policy $p^t$, where $p^t.qaValue = MV_{32}$. According to Table 2, the probability of choosing the trust level $TRUST$, $prob_t$, is equal to $t/t + d$, where $t = 46 * (1 - 0.625)$ and $d = 54 * (0.625)$ (Line 7 in Table 2). Therefore $prob_t \approx 0.34$ and $prob_d \approx 0.66$, where $prob_d$ is the probability of choosing trust level $DISTRUST$. The resulting probabilities correspond with the fact that $ttValue > 5$ – the user is more trustworthy then untrustworthy, and $\gamma(MV_{32}) = 32$, thus, the selected $p^t.qaValue$ is in the second half according to function $\gamma$, where the distrusted values occur.

Table 3 depicts how the values for the number of policies, number of acquaintances, and topic trust value are chosen according to the randomly generated value $gv$ of the normal distribution and it's distance from the mean measured in the multiples of the standard deviation $\sigma$.

## 4.2 Metrics, Simulations & Results

In order to quantify the contribution of the inheritance of QA policies in QA social networks to the ranking algorithm RANK, we compare for every query $q \in Q_I$ the ordered set $\widehat{R_q}$ with the ordered set $\widehat{IR_q}$. In $\widehat{IR_q}$, the resources $r_i \in R_q$, $1 \leqq i \leqq |R_q|$, are ordered according to the decreasing values of the function $iScore : R_q \to \mathbb{Z}$ computed according to Formula (5), where $\omega : QAV^t \to \{-1, 1\}$ is defined in Formula (6). Further, we define a permutation $\chi : R_q \to \widehat{IR_q}$ mapping the ordered set of resources $R_q$ to $\widehat{IR_q}$. Obviously, $\widehat{IR_q}$ differs for different $q \in Q_I$, therefore, we call the ranking according to $\widehat{IR_q}$ as an *ideal ranking* and the ranking according to $\widehat{R_q}$ as *computed ranking* for some $q$.

important, we denote them as $MV_1 \dots MV_m$, $|QAV^t| = m$.

$$iScore(r_i) = \sum_{j=1}^{|QAW^t|} \omega(r_i.triple[j].v) \tag{5}$$

$$\omega(qaValue) = \begin{cases} 1, & \text{if } \gamma(qaValue) \leqq (|QAV^t|/2) \\ -1, & \text{otherwise} \end{cases} \tag{6}$$

Suppose that for $q \in Q_I$, we denote $r_q$ the resource for which $\chi(r_q) = a_q$ and $\phi(r_q) = b_q$; then, to measure the success of QA social networks, we define the set of *average distance metrics* $\delta_x$, $1 \leqq x \leqq |R_q|$, according to Formula (7). The expression $a_q^x$ in Formula (7) denotes the rank $x$ of the resource $a_q \in \widehat{IR_q}$ according to the decreasing results of the function $iScore$; similarly, $b_q^y$ denotes the rank $y$ of the resource $b_q \in \widehat{R_q}$ according to decreasing $b_q.score$. The metrics $\delta_x$ express the average relative distance of $\phi(r_q)^x$ and $\chi(r_q)^y$ measured $\forall q \in Q_I$. To express $\delta_x$ added to or subtracted from the ranks in ideal ranking sequence, we define *average rank metrics* $\xi_x$ as $\xi_x = \delta_x + x$, $1 \leqq x \leqq |R_q|$.

$$\delta_x = (\sum_{q=1}^{|Q_I|} a_q^x - b_q^y)/|Q_I| \tag{7}$$

The set of metrics $\delta_x$, $1 \leqq x \leqq |R_q|$, is further used to compute the *average global distance metric* $\Delta$ according to Formula (8). The absolute value in Formula (8) occurs because we do not want to distinguish positive and negative $\delta_x$ values – they would falsify the metric $\Delta$.

$$\Delta = \sum_{x=1}^{|R_q|} |\delta_x| \tag{8}$$

Finally, we define the *applied policies metric* $\#Pol$ for computing the average number of successfully applied QA policies per resource.

Figure 2 depicts the set of metrics $\xi_1$ – $\xi_{10}$. The y-axis represents the values for the average rank metrics (denoted as $\xi$). The x-axis represents $\forall q \in Q_I$ the ranked positions 1 – 10 of the resources $R_q$. The solid blue line $IR$ represents the ideal ranking, whereas the lines with markers represent the computed ranking. The line labeled as $SRANK$ denotes the results of the ranking algorithm SRANK, $RANK$ denotes the results of the ranking algorithm RANK, and $RANK_d$ denotes the results of the algorithm RANK, where the QA profiles were inherited only from the users having the distance $\leqq d$ from the information consumer (in the particular QA social network) – the distance is computed as the lowest number of edges (trust relations) in STN between the information consumer and the particular user. We say that $RANK_d$ uses *inheritance level* $d$.

Consequently, Figure 3A shows the metric $\Delta$ (y-axis) for the different ranking algorithms $SRANK$, $RANK_d$, and $RANK$ (x-axis). Figure 3B depicts the metric $\#Pol$ (y-axis) for the same ranking algorithms (x-axis).

As we can see in Figure 2, the higher the inheritance level used in the ranking algorithm, the more the computed ranking advances towards the results of the ideal ranking.

The ranking algorithm $SRANK$ seems to work fine for the first several ranked positions, however, this is caused by a very small amount of successfully applied QA policies per resource (see Figure 3B), which means that for the majority of resources no QA policy was successfully applied. This

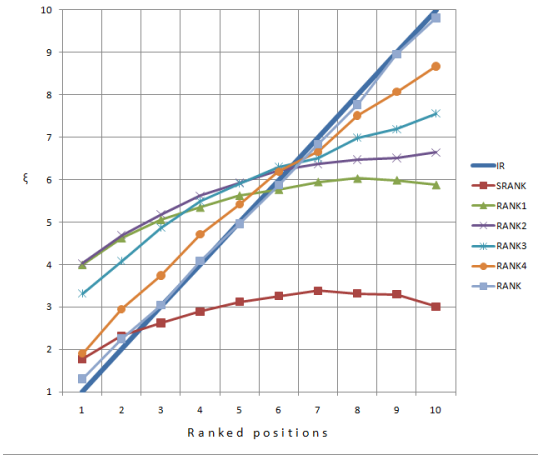


**Figure 2: Average ranks metrics ($\xi_1 - \xi_{10}$)**

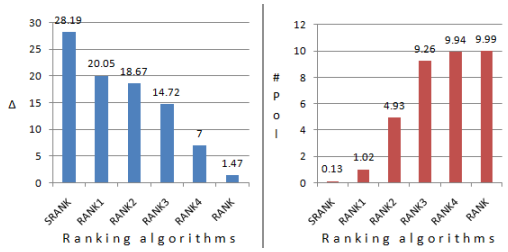


**Figure 3: (A) Average global distance metric $\Delta$ (B) Applied policies metric $\#Pol$**

leads to many resources $r$ with $r.score = 0$ and $r.color = GREY$ occupying the same rank.

The almost ideal ranking reached by ranking algorithm $RANK$ is caused by the relatively small set of $QAW^t$ and $QAV^t$. This corresponds with Figure 3B, where we can see that in case of the algorithms $RANK_3$, $RANK_4$, and $RANK$ the metric $\#Pol$ almost reaches $|QAW^t|$, which is the maximum amount of not conflicting successfully applied policies per resource. On the other hand, it shows that the algorithms collectPolicies (Line 1) and refinePolicies (Line 2) in Algorithm 4 are designed reasonably – the metric $\Delta$ is decreasing (see Figure 3A), despite the stagnating $\#Pol$ in Figure 3B.

We experimented with the sizes of the sets $QAW^t$ and $QAV^t$ and the observed results are summarized in Table 4 showing that the algorithm RANK outperforms SRANK in all cases having at least three times lower $\Delta$.

## 5. RELATED WORK

Researchers have developed and investigated various policy languages to describe trust and security requirements on the Web (such as [9, 17, 26, 7]); a variety of access control mechanisms generally based on policies and rules have been developed (such as [19, 22, 11]). Although these approaches support customization of the policies, to the best of our knowledge, they do not utilize the power of the social network to share policies.

Paper [10] coins the term Data Quality Social Network. Nevertheless, their social network is not intended to be used for sharing policies – it is used for adjusting weights of the quality dimensions participating in the QA process. Moreover, each Data Quality Social Network is restricted to users collaborating on the similar tasks.

**Table 4: The Metric $\Delta$ for Algorithms SRANK and RANK and Different $|QAW^t|$ and $|QAV^t|$**

| $\|QAW^t\|$ | $\|QAV^t\|$ | SRANK | RANK |
|---|---|---|---|
| 10 | 50 | 28.19 | 1.47 |
| 20 | 100 | 34.63 | 7.1 |
| 50 | 50 | 32.95 | 9.82 |
| 50 | 100 | 36.81 | 10.01 |

Ziegler and Laursen [28] extensively examined the propagation of trust and distrust in social networks and present Appleseed trust metric. Our trust metric (TTA algorithm) always computes trust between two persons - information consumer and another user in a QA social network. On the other hand, Appleseed calculates trust for a collection of nodes at once by energizing the selected node (an information consumer) and spreading the energy to other users connected by trust relations. The problem of this algorithm is the normalization of the trust values - the more trust relations the user is part of, the less energy (trust) each of these trust relations obtain.

The QA process can comprise many analyses [1], such as the analysis of data provenance [21, 16, 13], data timeliness and popularity [25], reputation of resources [4] etc. Our focus was not to enumerate the approaches to particular analyses, but to present the generic approach for QA process on the Web. For the purpose of FS in Subsection 1.1 working with the provenance analysis, we extensively analyzed the issues of provenance on the Web and proposed XProv provenance model for the Web[22] in [13].

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a generic Web Quality Assessment model for assessing the quality of information on the Web. Since the unprecedented volume of information is being published as open data in the Web information space and millions of information consumers are assessing the quality of information, the QA process in the WQA model is driven by a set of customizable QA policies grouped to QA profiles suitable for different tasks at the consumers' hands. As part of the WQA model, we propose and detail a novel concept of QA social networks enabling the sharing of QA profiles and QA policies between users in the social network.

We empirically evaluated the contribution of the QA social networks to the QA process by introducing two ranking algorithms - SRANK and RANK; the algorithm SRANK utilizes the Restricted WQA model, whereas the algorithm RANK uses the full WQA model including the concept of QA social networks. We compared the average global distance metric $\Delta$ of the sequence of resources ranked by both algorithms w.r.t the ideal ranking sequence (see Table 4 for details) and the results acknowledge the significant decrease in $\Delta$ for the ranking algorithm RANK. In other words, our experiments show the positive effect of QA policies sharing within the QA social networks – the resources ranked according to the algorithm RANK are much more closer to the ideal ranking sequence.

[22] http://prov4j.org/xprov

Future work involves the finishing of the particular WQA model reference implementation, which is in parallel developed. Moreover, we will (1) conduct experiments with real world data using FOAF social networks[23] supplemented with trust relations and QA profiles using our FOAF extension modules, and (2) study the behavioral patterns of the information consumers and users in QA social networks to better justify the settings of our experiments, such as the number of QA policies provided by the average user on the Web. We are convinced that the QA social networks have the indisputable role in the future QA process on the Web.

## 7. ACKNOWLEDGMENTS

We would like to thank to Andre Freitas, Sean O'Riain, and Edward Curry[24] for discussion of the initial ideas and implementation of the simple prototype of the WQA model. The work presented in this article has been funded in part by the Czech Science Foundation (GACR, grant number 201/09/H057), grant SVV-2010-261312, and GAUK 3110.

## 8. REFERENCES

[1] B. Aleman-Meza, C. Halaschek-Wiener, I. B. Arpinar, C. Ramakrishnan, and A. P. Sheth. Ranking Complex Relationships on the Semantic Web. *IEEE Internet Computing*, 9:37–44, 2005.

[2] J. E. Alexander and M. A. Tate. *How to Evaluate and Create Information Quality on the Web*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1999.

[3] S. ann Knight and J. Burn. Developing a Framework for Assessing Information Quality on the World Wide Web. *Informing Science Journal*, 8:159–172, 2005.

[4] D. Artz and Y. Gil. A Survey of Trust in Computer Science and the Semantic Web. *Web Semant.*, 5(2):58–71, 2007.

[5] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino. Methodologies for Data Quality Assessment and Improvement. *ACM Comput. Surv.*, 41(3):1–52, 2009.

[6] D. Beckett. RDF/XML Syntax Specification (Revised). W3C Recommendation, 10 February 2004. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.

[7] C. Bizer and R. Cyganiak. Quality-driven Information Filtering Using the WIQA Policy Framework. *Web Semant.*, 7(1):1–10, 2009.

[8] 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.

[9] 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.

[10] I. Caballero, E. Verbo, M. A. Serrano, C. Calero, and M. Piattini. Tailoring Data Quality Models Using Social Network Preferences. In *DASFAA Workshops*, pages 152–166, 2009.

[23] Consumed according to Linked Data principles [8]

[24] All from Digital Enterprise Research Institute (DERI), National University of Ireland, Galway

[11] S. Cantor, J. Kemp, R. Philpott, and E. Maler. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. Technical report, 2005.

[12] 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.

[13] A. Freitas, T. Knap, S. ORiain, and E. Curry. W3P: Building an OPM based Provenance Model for the Web. Submitted to Future Generation Computer Systems, The International Journal of Grid Computing and eScience. (http://www.ksi.mff.cuni.cz/~knap/qa/prov10.pdf).

[14] Y. Gil and D. Artz. Towards Content Trust of Web Resources. *Web Semant.*, 5(4):227–239, 2007.

[15] J. A. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, College Park, MD, USA, 2005.

[16] O. Hartig. Provenance Information in the Web of Data. April 2009. LDOW2009, 2009, Madrid, Spain.

[17] L. Kagal, T. Finin, and A. Joshi. A Policy Based Approach to Security for the Semantic Web. In *2nd International Semantic Web Conference (ISWC2003)*, September 2003.

[18] T. Knap and I. Mlýnková. Towards Topic-Based Trust in Social Networks. 7th International Conference on Ubiquitous Intelligence and Computing, 2010. (http://www.ksi.mff.cuni.cz/~knap/qa/tt10.pdf).

[19] K. Lawrence and C. Kaler. WS-Trust Specification. Technical report, March 2007.

[20] S. Milgram. The Small World Problem. *Psychology Today*, 1:61, 1967.

[21] L. Moreau. The Foundations for Provenance on the Web. *Foundations and Trends in Web Science*, November 2009.

[22] T. Moses. Extensible Access Control Markup Language Version 2.0. Technical report, 2005.

[23] F. Naumann and C. Rolker. Assessment Methods for Information Quality Criteria. In *In Proceedings of the International Conference on Information Quality (IQ*, pages 148–162, 2000.

[24] M. E. J. Newman. Models of the Small World. *J. Stat. Phys*, pages 819–841, 2000.

[25] 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.

[26] 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 Policy 2003*, pages 93–96, 2003.

[27] 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.

[28] C.-N. Ziegler and G. Lausen. Propagation Models for Trust and Distrust in Social Networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.