On Creating a Trusted and Distributed Data Source Environment*

Roman Špánek, Martin Řimnáč, Zdeňka Linková

Institute of Computer Science,
Pod Vodárenskou věží 2, Prague 8, Czech Republic
Technical University of Liberec,
Hálkova 6, Czech Republic
{spanek,rimnacm,linkova}@cs.cas.cz
http://www.cs.cas.cz/~{spanek,~rimnacm,~linkova}

Abstract. *Despite the tremendous research activity in the field of searching engines for the Internet, current searching engines still face some severe limitations. The paper presents an idea of a distributed data source environment to be build on the current state of the art technologies available on the Internet. The paper combines recent advances in the fields of a data inconsistency, a data integration and reputations of sources for further refinements of data searching and sharing processes. The paper generalizes the data binary formalism narrowly connected with the ideas of the semantic web into the* $\langle 0, 1 \rangle$ *interval to enable the consideration of uncertainty at various levels.*

Key words: semantic web, search engines, data integration, reputation system

1 Introduction

Recent advances achieved in communication technologies have let to the rapid emergence of many new commercial applications (e.g. e-commerce, banking or travel services). On the other hand, such a progress in technologies cannot concentrate only on communication protocols, new data exchange formats or advance end user interfaces, but it has also to give users a way how to exchange their private data or to manage their preferences.

The paper brings a vision of a distributed and trusted data framework based on the semantic web ideas. The framework aims at accessing distributed data -

* The work was supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) "Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realization", partly by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications" and by Ministry of Education, Youth and Sports of the Czech Republic, project No. 1M0554 Advanced Remedial Technology and Processes.

from the web resources or services - respecting users' preferences, which can be based at least on their experiences with particular data providers.

Our vision can be clarified by a simple scenario: *Let be resource A, resource B and resource C available providing general information for traveling. Furthermore, a user issues a question of which country Prague is the capital city. Assume that resources provide the following as the output for the query: A provides the Czech Republic; B provides France; C provides Prague to be an ordinary city in the Czech Republic. The user has to make a complicated decision which resource he/she will trust. Such a decision, on the other hand, is much simplified if a user has had some previous experiences with resources or if a queried framework can provide some additional information about these resources.*

The semantic web ideas bear trust in mind with the very top level of trust in the well known semantic web layer architecture. On the other hand, trust has been used in many research areas for different purposes [1],[2],[3],[4],[5]. All these applications for managing trust justify our assertion that trust is very important, and that its importance for the next generation of Internet services is indisputable.

The paper presents a refined framework that utilizes the following three research fields to overcame the problem of sharing and accessing the data over various services on the (future) Internet:

- data processing
- data integration
- dynamic trust management

The paper is organized as follows. Section 2 presents the related approaches. Section 3 starts with a proposal of an internal formalism, and shows its connections into the semantic web and relational databases. Then the issue of inconsistency checking and solving is presented. This issue leads to the advanced source quality analysis through a trust definition given in section 4. Finally, the main advantages of this work are given as well as future aims.

2 Related Work

2.1 Trust Management

Trust management systems can be categorized according to the way adopted for establishing and evaluating trust as follows:

- *credential* and *policy based trust management*;
- *reputation based trust management*, and;
- *social network based trust management.*

Policy based approach has been proposed in the context of open and distributed service architectures [6], [7] as well as in the context of Grids [8] as the solution to the problem of authorization and an access control in open systems. Its focus is on the trust management mechanisms employing different policy

languages and engines for specifying and reasoning on rules for the trust establishment. Since the primary aim of such systems is to enable access control, the trust management is limited to verification of credentials and restricting an access to resources according to policies defined by a required resource owner [9].

On the contrary, *Reputation based* trust management systems provide a way in which entities may evaluate and build a trust relationship between resource provider and requester. The reputation approach has emerged in the context of electronic commerce systems, e.g. eBay. In distributed settings, reputation-based approaches have been proposed for managing trust in public key certificates, P2P systems XREP, mobile ad-hoc networks, and recently, also in the Semantic Web [3], NICE [4], DCRC/CORC [10], EigenTrust [5], EigenRep[11].

Social network based trust management systems utilize, in addition, social relationships between entities to infer trust. In particular, the social network based system views the whole structure as a social network with relationships defined amongst entities. Examples of such trust management systems include Regret [12], NodeRanking [13].

2.2 Partitioning problem for parallel sparse-matrix vector multiplication

A linear system of equations are widely solved by iterative solvers on parallel computers [14],[15]. The parallelization is necessary as the size of the matrix might be huge. The goal of the partitioning is to enable parallelization of a sparse-matrix vector product. In order to avoid the communication of vector components during the linear vector operation a partition scheme is adopted. It means that all vectors used in the solver are decomposed conformally.

Graph Model for Decomposition An undirected graph $G = (V, E)$ is defined as a set of vertices V and a set of edges E.

$\Pi = P_1, P_2, \cdots, P_K$ is a *K-way partition* of graph $G = (V, E)$ if the following conditions hold:

- each P_k, $1 \leq k < l \leq K$; $P_k \neq \oslash$
- $P_k \cup P_l = 0$ for all $1 \leq k < l \leq K$
- $\bigcup_{k=1}^{K} P_k = V$

The graph partitioning problem can be defined as the task of dividing a graph into two or more parts such that the cutsize (amount/weight of edges connection the parts) is minimized, while the balance criterion (well-proportioned) on part weights is maintained.

The graph model has nevertheless face some limitations[1] therefore a hypergraph model was proposed.

[1] e.g. the graph model does not express the real communication volume implied by the partitions

Hypergraph Models for Decomposition K-way partitioning of hypergraphs is defined similarly to the graph model[16], but instead of a graph is partitioned a hypergraph $H = (U, N)$.

The most important advantage of the hypergraph model is ability to fit the real communication needed better resulting in low cost partitions. We relax details here and readers are referred to [16] for details.

3 Formalism and Global Data Source Matrix Representation

Let be an environment consisted of several sources $S_l \in \mathscr{S}$. The *source* S_i covers the set of *attributes* $\mathscr{A}_{S_l}$ together with their *active domains* $\mathscr{D}_\alpha^{S_l}(A), \forall A \in \mathscr{A}_{S_l}$, subsets of values from the *attribute domains* $\mathscr{D}(A)$, which are covered by the source S_l. All values covered by the source S_l are denoted $\mathscr{D}_\alpha^{S_l} = \bigcup_{\forall A \in \mathscr{A}_{S_l}} \mathscr{D}_\alpha^{S_l}(A)$. Further *data* in the source are represented by a set of *implications* $\mathscr{I}_{S_l}$ between *elements* $\mathscr{E}_{S_l} \subseteq \mathscr{A}_{S_l} \times \mathscr{D}_\alpha^{S_l}$, the attribute-value pairs. These implications can be seen as the instances of *functional dependencies* $A_i \rightarrow A_j$ between (unary) attributes $A_i, A_j \in \mathscr{A}^{S_l}$. Such instances can be expressed by the binary *repository matrix* $\boldsymbol{\Phi}^{S_l}$, which is defined as [17]

$$\boldsymbol{\Phi}^{S_l} = [\phi_{ij}^l];\ \phi_{ij}^l = \begin{cases} 1 & \text{if } e_i \rightarrow e_j \in \mathscr{I}^{S_l} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Similarly, the *active attribute domain matrix* is defined as

$$\boldsymbol{\Delta}_{S_l} = [\delta_{ij}^l];\ \delta_{ij}^l = \begin{cases} 1 & \text{if } \exists v \in \mathscr{D}_\alpha^{S_l} : e_i = (A_j, v) \in \mathscr{E}_{S_l} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Within the repository matrix $\boldsymbol{\Phi}_{S_l}$ covers instances of functional dependencies, the *functional depedency matrix* $\boldsymbol{\Omega}_{S_l}$ can defined as

$$\boldsymbol{\Omega}_{S_l} = \boldsymbol{\Delta}_{S_l} \boldsymbol{\Phi}_{S_l} \boldsymbol{\Delta}_{S_l}^T \tag{3}$$

3.1 Accessing the Repository Matrix

Let be a query represented by an element activation vector $\boldsymbol{x}$. The repository matrix can be accessed by two basic operators:

- *Generalization*, which returns the elements, which are implied by any element in the query
$$\boldsymbol{y} = \boldsymbol{\Phi}_{S_l} \boldsymbol{x} \tag{4}$$
- *Specialization* (Restriction), which returns elements, which implies any element in the query
$$\boldsymbol{y} = \boldsymbol{\Phi}_{S_l}^T \boldsymbol{x} \tag{5}$$

3.2 Integrating Sources

This mechanism can be used for one source. In many real situations, the complete result can be reached only if several sources are used. This is the reason, why the sources are connected together via mediators representing mapping rules.

The mediator structure can be in general very complex. One of possible solutions is to consider a mapping, which transforms the instances of the local sources into terms of virtual global repository elements. These mediators connecting local sources S_l with the global repository can be expressed by the binary matrix Γ_{S_l}:

$$\Gamma_{S_l} = [\gamma_{ij}^l];\ \gamma_{ij}^l = \begin{cases} 1 & \text{if } \exists e_i \sim e_j,\ \ e_i \in \bigcup_{\forall S \in \mathscr{S}} \mathscr{E}_S, e_j \in \mathscr{E}_{S_l} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The virtual repository can be consequently represented by

$$\Phi_{\mathscr{S}} = \sum_{\forall S_l \in \mathscr{S}} \Phi_{\mathscr{S}}^{S_l} \quad \text{where } \Phi_{\mathscr{S}}^{S_l} = \Gamma_{S_l} \Phi_{S_l} \Gamma_{S_l}^T \tag{7}$$

4 Reputation System Proposal

The previous section shows a possibility of presentation and basic operations on a global data repository created by integration of various distributed data sources. It was also shown, that criteria used for data integration are sometimes insufficient leaving the global repository inconsistent. Hence, this section presents a new criterion that might be used for further refinements of the integration rules – *trust* of the distributed sources.

Trust between entities may be built by so called reputation systems. The main aim of the reputation systems is to provide entities in the system with a set of rules, restrictions, behavior observations, transaction histories, etc. allowing deduction of trust.

4.1 Design of the Reputation System

Our motivation scenario given before also assumed a set of distributed data sources to be integrated into one virtual repository. The integration might have been admittedly done wrong; some integration rules might have been invalid or the virtual repository might contain some inconsistencies. To overcame this problem, we propose to use trust of resources as an additional measure positively influencing the integration process.

The following list gives the design principles of the reputation system:

- to prefer resources with accurate (globally acceptable) data
- to prefer resources providing new and accurate data
- distributed as well centralized implementation

Firstly we would to introduce a model for representation of set of resources some of them providing the same data. At first glance, one might think of a graph model describing resources as vertices and references between sources and their functional dependencies as edges. Even thought the graph model might be considered sufficient, it has some severe drawbacks influencing especially efficient distributed implementation (see section 4.3). Therefore we propose a hypergraph model.

In a hypergraph $H = (U, N)$, a hyperdge $n_j \in N$ can connect arbitrary many vertices[2] and one vertex can be a pin of more hyperedges [18]. In our hypergraph model the following relations hold:

- vertices $u \in U = \{[i,j] : \phi_{ij}^{\mathscr{S}} > 0\}$ represent instances of functional dependencies in the virtual integrated repository matrix $\Phi_{\mathscr{S}}$.
- a hyperedge $S_I \in N = \mathscr{S}$ represents source
- pins of a hyperedge (set of vertices connected by the hyperedge) $pins(S_I) = \{[i,j] : \phi_{ij}^{I} > 0\}$ represent instances of functional dependencies presented at source S_I
- $hyperedges(\phi_{ij})$ represents a set of sources which present instance ϕ_{ij}.

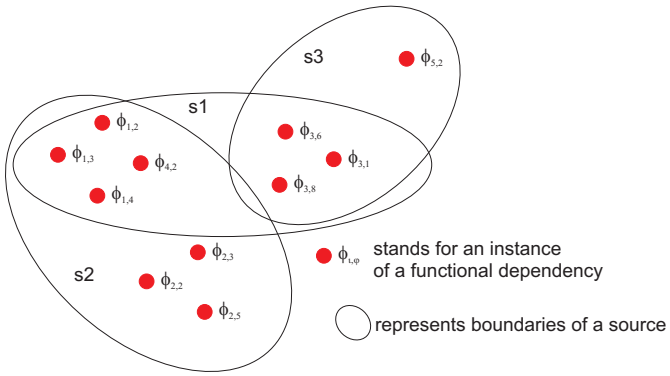

Fig. 1. An example situation

It follows, that one source can assert one or more instances of functional dependencies and, in addition, one instance can be common for several sources. The hypergraph model fits this perfectly. In Figure 1 is depicted an example of a hypergraph describing instances some of which are common for some sources. The figure shows three distinct sources S_1, S_2, S_3 (represented by ovals) and their 11 instances (spots). For example, source S_3 provides 4 instances $(\phi_{5,2}, \phi_{3,6}, \phi_{3,1}, \phi_{3,8})$, from which $\phi_{3,6}, \phi_{3,1}, \phi_{3,8}$ are common also for the source S_1.

[2] The upper bound is naturally given by the amount of vertices $|U|$ of a hypergraph and the lower bound is 1 vertex

4.2 On Calculation of Quality of Sources

The hypergraph model gives us a natural way to describe a set of sources asserting some instances which might be shared amongst them. In the following is presented the calculation of trust of sources corresponding to a source quality.

In our reputation system the overall trust value for a source is assembled from the following components:

$$a(s_i^k) = \sum_{u \in pins(s_i^k)} \frac{(|hyperedges(u)| - 1)}{|U|} = \frac{||\Phi_{\mathscr{S}}^{S_i^k} \odot \sum_{\forall S_j \in \mathscr{S} - \{S_i\}} \Phi_{\mathscr{S}}^{S_j^k})||}{||\Phi_{\mathscr{S}}^k||} \quad (8)$$

Equation (8) represents the fact that the instances confirmed by more sources is more likely to be true. The overall factor value is naturally weighted by the level of trust of the other sources claiming the same dependency.

$$b(s_i^k) = \epsilon b^{k-1} + (1 - \epsilon) \sum_{u \in pins(s_i^k) \wedge u \notin pins(s_i^{k-r})} \frac{|hyperedges(u)| - 1}{|U|} \quad (9)$$

Equation (9), on the other hand, reflects the fact, that a new information is very worthy, but only in the case that some other sources confirmed it. The attribute is also weighted by the level of trust of sources confirming the dependency.

$$c(s_i^k) = \sum_{u \in pins(s_i^k)} hits(s_i^k) \quad (10)$$

The important information about resource trustworthiness is amount of queries comming from the global repository to the local resource. For this purpose we utilize a *hits* factor. The $hits(s_i^k)$ factor is increased when ever the query was transformed onto a source s_i.

$$d(s_i^k) = \frac{1 - |noOfInconsistences(s_i^k)|}{||\Phi_{S_i}||} \quad (11)$$

The last factor influencing the overall trust of a source is amount of inconsistences found during the integration process. This is reflected by factor $d(s_i^k)$ given in (11).
Finally, the reputation of source S_i is given linear combination of components [3]

$$\rho_{S_i}^k = a(s_i^k) \cdot (1 + b(s_i^k)) \cdot c(s_i^k) \cdot d(s_i^k) \quad (12)$$

This trust can effect all relevant mediators using weighted virtual matrix definition

$$\Phi_{\mathscr{S}} = \sum_{\forall S_l \in \mathscr{S}} \rho_{S_l}^k \cdot \Gamma_{S_l} \Phi_{S_l^k} \Gamma_{S_l}^T \quad (13)$$

[3] See section 4.4 for further details on the reputation of resource.

4.3 Efficient Distributed Implementation

One of the design principles mentioned in section 4.1 was the efficient distributed implementation. From our point of view, an implementation of our proposal is efficient if the overall load can be uniformly distributed onto resources. Numerial algebra uses the hypergraph partition model for paralelization of computer rograms so that comunication between paralel computers is minimized.

It follows that our proposal can straightforwardly use the hypergraph partitioning to enable load distribution onto resources. In addition to this we canalso use many efficient representation and implementation of hypergraphs [15], [16] well known from numerical mathematics.

4.4 Analysis of Attacks

Even thought the world with only honest sources would be a wonderful place to live, it is not the case of the current and very probably future Internet. Currently running computer systems are often under attack of hackers, viruses, etc. In our proposal we introduced, besides the other attributes, a level of trust of a source as an additional parameter influencing integration and accessing of decentralized data. In this scenario, the level of trust has direct impact on result of a given query. Therefore the reputation system managing the levels of trust must be able to cope with certain threats. In the following paragraphs we put the reputation system under investigation against well known attacks on reputation systems.

Fake transaction Malicious colluding peers always cooperate with others in order to receive strong reputation. They then provide misinformation to promote actively malicious peers [19].

In our system, the responsibility of creating transaction and evaluation a feed-back (whether transaction was done correctly with demanded results or otherwise) lays upon our framework. In other words, the framework presented does not require any feed-back from sources, since it can generate feed-backs automatically upon each integration step.

Collusion Multiple malicious peers cooperating together to cause more damage [20],[21].

Figure 2 shows an example situation of 3 sources sharing the same fake functional dependencies to boost their trust. If the overall reputation given in (12) had included only parameter $a(s_i^k)$ (8), the reputation system would have been affected very easily by collusion. In our proposal reputation of resource includes also parameters $b(s_i^k)$ (9) and $c(s_i^k)$ (10). Parameter $b(s_i^k)$ prices newly introduced functional dependencies, thereby giving more trust to sources that introduce new valuable data. Parameter $c(s_i^k)$, on the other hand, gives more trust to sources that provide data often used in queries. In collision, the sources providing fake or meaningless data are punished by parameter $c(s_i^k)$. Moreover sources in collision would unlikely provide a new information that is accepted by some trusted sources (parameter $b(s_i^k)$).

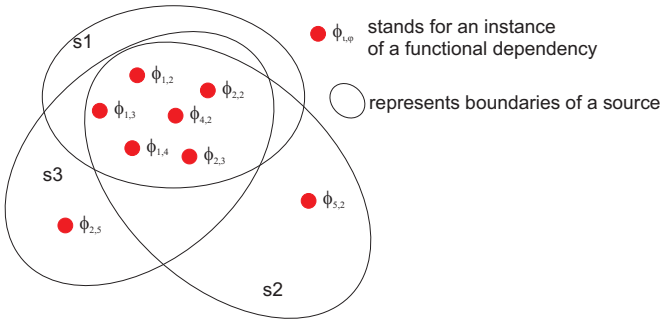

Fig. 2. An example situation of collision attack on reputation system. Sources co-operate in generation possible false dependencies that are shared to boost parameter a^k of the overall trust given in 12

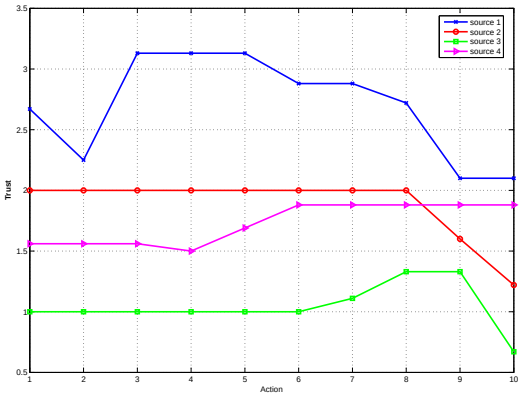

Fig. 3. Trust of sources

Whitewashers Peers purposefully leave and rejoin the system with a new identity in an attempt to shed any bad reputation they have accumulated previously [22].

In our framework, entities are not anonymous peers but rather real servers with at least IP addresses. Under such circumstances, it is unlikely to assume such type of attack as making a new identity might be expensive and starting level of trust is given only by parameter $a(s_i^k)$.

	S1								S2							
Action	*Num-inst-L*	*Sum-coincide-LL*	*Num-inconsit-LG*	*Num-confirm*	*Ratio-coincide*	*Ratio-inconsistency*	*Ratio-confirm*	***Level of trust***	*Num-inst-L*	*Sum-coincide-LL*	*Num-inconsit-LG*	*Num-confirm*	*Ratio-coincide*	*Ratio-inconsistency*	*Ratio-confirm*	***Level of trust***
Initial configuration	3	8	0	0	*2,67*	*0,00*	*0,00*	***2,67***	3	6	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Add consit. into S1	4	9	0	0	*2,25*	*0,00*	*0,00*	***2,25***	3	6	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Confirm it by S2	4	10	0	1	*2,50*	*0,00*	*0,25*	***3,13***	4	8	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Add consit. into S4	4	10	0	1	*2,50*	*0,00*	*0,25*	***3,13***	4	8	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Confirm it by S2	4	10	0	1	*2,50*	*0,00*	*0,25*	***3,13***	5	10	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Confirm it by S1	5	12	0	1	*2,40*	*0,00*	*0,20*	***2,88***	5	10	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Add consist. into S3	5	12	0	1	*2,40*	*0,00*	*0,20*	***2,88***	5	10	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Confirm it by S1	6	14	0	1	*2,33*	*0,00*	*0,17*	***2,72***	5	10	0	0	*2,00*	*0,00*	*0,00*	***2,00***
Add inconsit.(against S2) into S1	7	15	1	1	*2,14*	*0,14*	*0,14*	***2,10***	5	10	1	0	*2,00*	*0,20*	*0,00*	***1,60***
Add inconsit.(against S3) into S2	7	15	1	1	*2,14*	*0,14*	*0,14*	***2,10***	6	11	2	0	*1,83*	*0,33*	*0,00*	***1,22***

	S3								S4							
Action	*Num-inst-L*	*Sum-coincide-LL*	*Num-inconsit-LG*	*Num-confirm*	*Ratio-coincide*	*Ratio-inconsistency*	*Ratio-confirm*	***Level of trust***	*Num-inst-L*	*Sum-coincide-LL*	*Num-inconsit-LG*	*Num-confirm*	*Ratio-coincide*	*Ratio-inconsistency*	*Ratio-confirm*	***Level of trust***
Initial configuration	2	4	1	0	*2,00*	*0,50*	*0,00*	***1,00***	3	7	1	0	*2,33*	*0,33*	*0,00*	***1,56***
Add consit. Into S1	2	4	1	0	*2,00*	*0,50*	*0,00*	***1,00***	3	7	1	0	*2,33*	*0,33*	*0,00*	***1,56***
Confirm it by S2	2	4	1	0	*2,00*	*0,50*	*0,00*	***1,00***	3	7	1	0	*2,33*	*0,33*	*0,00*	***1,56***
Add consit. into S4	2	4	1	0	*2,00*	*0,50*	*0,00*	***1,00***	4	8	1	0	*2,00*	*0,25*	*0,00*	***1,50***
Confirm it by S2	2	4	1	0	*2,00*	*0,50*	*0,00*	***1,00***	4	9	1	0	*2,25*	*0,25*	*0,00*	***1,69***
Confirm it by S1	2	4	1	0	*2,00*	*0,50*	*0,00*	***1,00***	4	10	1	0	*2,50*	*0,25*	*0,00*	***1,88***
Add consist. into S3	3	5	1	0	*1,67*	*0,33*	*0,00*	***1,11***	4	10	1	0	*2,50*	*0,25*	*0,00*	***1,88***
Confirm it by S1	3	6	1	0	*2,00*	*0,33*	*0,00*	***1,33***	4	10	1	0	*2,50*	*0,25*	*0,00*	***1,88***
Add inconsit.(against S2) into S1	3	6	1	0	*2,00*	*0,33*	*0,00*	***1,33***	4	10	1	0	*2,50*	*0,25*	*0,00*	***1,88***
Add inconsit.(against S3) into S2	3	6	2	0	*2,00*	*0,67*	*0,00*	***0,67***	4	10	1	0	*2,50*	*0,25*	*0,00*	***1,88***

Fig. 4. Trust of sources

5 Experimental Results

In Figure 4 are shown tabular form of results for our experiments where the following holds: for each source (*S1,S2,S3,S4*) are in columns shown total number of instances in the source (*Num-inst-L*), sum of coinciding instances in local source (*Sum-coincide-LL*), number of inconsistent instances in the global repository (*Num-inconsist-LG*), number of confirmed instances (*num-confirm*), ratio of coincide instances to total number of instances in the source (*Ratio-coincide*), the same ratio for inconsistent instances (*Ratio-inconsistency*) and for confirmed instances (*Ratio-confirm*). The last column shows the overall level of trust.

In the rows are given actions taken during the simulation; to the initial configuration in the first row was added a new consistent instance into source $S1$. This instance was confirmed in the next row by source $S2$. After that, source $S4$ introduced consistent instance, consequently confirmed by sources $S2, S1$. In the next step, source $S3$ introduced a new instance and this instance was confirmed by source $S1$ consequently. Until this moment all sources are gaining trust, since all instances are consistent in the global repository. One can also see that source $S1$ gained more trust as had been actively introducing or confirming new instances. On the other hand, source $S2$ gained no more trust, as it had simply been copying data from other sources. The next action, add of inconsistency by source $S1$ against source $S2$ caused decline of trust of both involved sources. The next addition of other inconsistency caused even more decline (see Figure 3 for graph representation).

The experiments show, that the reputation system can bring additional attributes for data integration leaving the integration more accurate. The reputation system, can be also used in opposite way – for producing suggestions to sources in case of a primary inconsistency found. In this way, the reputation system provides a feed-back to sources, thus providing a new level of interaction between a searching engine and data sources.

6 Conclusion

The paper presents a vision of the next generation of the Internet as a global data repository together with a formalism for matrix representation of integrated sources. Built on well known integration approaches, the paper proposes a reputation system providing management of level of trust between sources as an additional attribute form better and more accurate integration. The reputation system can be also used for generating suggestions for sources in the case of the primary inconsistency found in the global repository, thus enabling a new level of duplex interaction between the search engine on the global repository and data sources.

References

1. Garfinkel, S.: Pgp: Pretty good privacy. O'Reilly & Associates, Inc. (1995)

2. Ellison, C.M., Frantz, B., Lampson, B., Rivest, R., Thomas, B.M., Ylonen, T.: Spki certificate theory. Internet RFC 2693 (October 1999)
3. Damiani, E., di Vimercati, S.D.C., Paraboschi, S., Samarati, P., Violante, F.: A reputation-based approach for choosing reliable resources in peer-to-peer networks. In Proceedings of ACM Conference on Computer and Communications Security (2002) 202216
4. Lee, S., Sherwood, R., et al.: Cooperative peer groups in nice. IEEE Infocom, San Francisco, USA (2003)
5. Kamvar, S., Schlosser, M., et al.: The eigentrust algorithm for reputation management in p2p networks. WWW, Budapest, Hungary (2003)
6. Bonatti, P., Samarati, P.: Regulating service access and information release on the web. In CCS 00: Proceedings of the 7th ACM conference on computer and communications security, ACM Press (2000) 134143
7. Li, N., Mitchell, J.: A role-based trust-management framework. In DARPA Information Survivability Conference and Exposition (DISCEX), Washington, D.C. (Apr. 2003)
8. Basney, J., Nejdl, W., Olmedilla, D., Welch, V., Winslett, M.: Negotiating trust on the grid. In 2nd WWW Workshop on Semantics in P2P and Grid Computing,New York, USA (may 2004)
9. Grandison, T., Sloman, M.: Survey of trust in internet applications. IEEE Communications Surveys **3(4)** (2000)
10. Gupta, M., Judge, P., et al.: A reputation system for peer-to-peer networks. Thirteenth ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video, Monterey, California. (2003)
11. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: Eigenrep: Reputation management in p2p networks. In Proceedings of 12th International WWW Conference (2003) 640651
12. Sabater, J., Sierra, C.: Regret: A reputation model for gregarious societies. 4th Workshop on Deception, Fraud and Trust in Agetn Societies, Montreal, Canada. (2001)
13. Pujol, J., Sanguesa, R., et al.: Extracting reputation in multi agent systems by means of social network topology. First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy. (2002)
14. Selvakkumaran, N., Karypis, G.: Multi-objective hypergraph partitioning algorithms for cut and maximum subdomain degree minimization. IEEE Transactions on Computer-aided design (2005)
15. Catalyurek, U.V., Aykanat, C.: Decomposing irregularly sparse matrices for parallel matrix-vector multiplication. dopnit **49(1)** (1994)
16. Catalyurek, U.V., Aykanat, C.: Hypergraph-partitioning based decomposition for parallel sparse-matrix vector multiplication. dopnit **49(1)** (1994)
17. imn, M.: Data structure estimation for rdf oriented repository building. cisis **0** (2007) 147–154
18. Golumbic, M.: Algorithmic graph theory and perfect graphs. Academic Press (1980)
19. Feldman, M., Lai, K., Stoica, I., Chuang, J.: Robust incentive techniques for peer-to-peer networks. ACM Press, New York, NY, USA (2004)
20. Marti, S., Garcia-Molina, H.: Limited reputation sharing in P2P systems. ACM Press, New York, NY, USA (2004)
21. Maniatis, P., Roussopoulos, M., Giuli, T., Rosenthal, D., Baker, M., Muliadi, Y.: Preserving peer replicas by rate-limited sampled voting. Technical Report arXiv:cs.CR/0303026, Stanford University (2003)

22. Lai, K., Feldman, M., Stoica, I., Chuang, J.: Incentives for cooperation in peer-to-peer networks. In Workshop on Economics of Peer-toPeer Systems (2003)