Intelligent Author Identification

Qing Wang[1] and René Noack[2]

[1] University of Otago, Dunedin, New Zealand
qing.wang@otago.ac.nz
[2] Christian-Albrechts-University Kiel, Germany
noack@is.informatik.uni-kiel.de

Abstract. This paper addresses a fundamental problem existing in the development of digitalising scientific contribution for individuals – the author identification problem. Instead of proposing an accurate and complete approach to identify authors in an open-world domain, which seems to be hardly found, we aim to develop the knowledge-based identification for authors by establishing an identity layer between a conceptual layer and a view layer. With the evolving knowledge acquired from different communities, a visual model built upon the conceptual and identity layers is adaptive such that the degree of accuracy and completeness on author identification can be improved over time.

1 Introduction

Author identification is a long-standing problem remaining in the area of institutional repositories, scientific communities, etc. In the past decades, a great number of working groups [9] have been setup at various levels – international, national or community-based – to explore possible solutions. Despite the vast amount of efforts, no satisfactory solution has yet been found. This severely hinders the capabilities of providing bibliometric analysis for scientific contribution of individuals, personalising content and services in social networks, integrating applications with assured quality, etc.

A common approach towards the author identification problem is to assign each author with a unique identifier [1,6]. This approach usually works well when application domains are small. In an open-world domain, this approach would however lead to the egg-and-chicken controversy – assigning a unique identifier to an author first or identifying an author uniquely first. An alternative, driven by library communities such as International Federation of Library Association and Institutions (IFLA), is to use authority files [3,5]. As this approach leaves users to select a desired authority record from potential matching results, the accurateness of author identification is a concern, particularly when disambiguating popular names such as Stephen Smith. Some repositories also prefer to directly contact authors and ask for their confirmation for the identity. The difficulty they meet is how to deal with people who do not reply or have been dead.

In light of various challenging issues surrounding the author identification problem, we intend to address the following questions in this paper.

J. Trujillo et al. (Eds.): ER 2010 Workshops, LNCS 6413, pp. 96–106, 2010.

- Since it is difficult to identify authors in an open-world domain, can we find a way of establishing the knowledge on author identification which will be more accurate and complete over time?
- As identifying authors often relies on community-based knowledge, can we find a way of efficiently and effectively sharing the knowledge on author identification across different communities?

In this paper we propose a multi-layer architecture to incorporate knowledge-based identification into a database system. The goal is to establish a more realistic and natural perspective on the modelling of authors – an author may be viewed differently in different communities at different times. A traditional perspective on data modelling differentiates levels of abstraction by considering a model at a conceptual, logical or physical layer. A conceptual layer emphasises on representing the real world; a logical layer describes the database schema and a physical layer implement the database schema. In addition to these, a view layer may be built upon a data model to provide customised information. To enable knowledge-based identification, we generalise this architecture by abstracting the knowledge on author identification into an independent identity layer and adding an adaptive visual layer lying between a conceptual layer and a view layer. More precisely, we have

- a conceptual layer models primitive objects in the print world and the like;
- an identity layer manages the knowledge in relation to the identity of authors;
- a visual layer presents dynamic objects by ultilising a flexible binding of knowledge at an identity layer and objects at a conceptual layer.

In the remainder of this paper we give the motivation in Section 2. Then we revise the definition of Higher-Order Entity-Relationship model in Section 3. In Section 4 an identity layer is introduced, which consists of a ground model and a set of refining relation tuples. Section 5 presents a visual layer built upon the conceptual and identity layers. We discuss the possible author identification services in section 6 and conclude the paper in Section 7.

2 Author Tagging in Open-World Domains

In an application domain every object is assumed to possess some properties that can uniquely distinguish itself from the others. These properties are often conceptualised by a set of key attributes in a data model. However, when an application domain is open (i.e., an open-world domain), the conceptualisation of objects is subject to the availability of information. It may lead to indiscernible objects whose properties do not suffice to uniquely identify themselves within an application domain. Therefore, the author identification problem we encounter has its roots in modelling authors in open-world domains with very limited author information.

Example 1. Let us consider the following publications and the question of whether the authors named Qing Wang are the same person. According to the similarity of publication titles, they might be the same person. By comparing their affiliations, they might be different persons. Since it is possible that an author changes his/her subjects and affiliations over time, it is unlikely to disambiguate the identity of authors with merely the information provided with publications.

- *FIXT: A Flexible Index for XML Transformation* by Jianchang Xiao, **Qing Wang**[1], Min Li and Aoying Zhou (1:Fudan University, China)
- *XML Database Transformations with Tree Updates* by **Qing Wang**[2], Klaus-Dieter Schewe and Bernhard Thalheim (2:Massey University, New Zealand)

Due to the ambiguousness of authorship, a dilemma with author tagging inevitably arise. When authors of different publications refer to the same person in the real world but are tagged into different objects in a data model, the problem of *incompleteness* occurs. As a converse, when the authors of different publications refer to different persons in the real world but are tagged into the same object in a data model, the problem of *inaccurateness* occurs.

Example 2. Let us look back the publications in Example 1 and the query "list all the papers by Qing Wang who wrote the paper named XML Database Transformations with Tree Updates". Assume that Qing Wang in these publications are different authors but tagged to the same author object. Then the query result would not be accurate as it includes the first paper which belongs to a different person. Similarly, assume that Qing Wang in these publications are the same author but taggged to two different author objects. Then the query result would not be complete because the first paper is not included.

Consequently, the incompleteness and inaccurateness of author tagging would affect the manageability, traceability, interoperability, quality of analysis, etc. of an application. More specifically, if the knowledge of making decisions on author tagging is not kept, it would be very difficult to detect and correct mistakes hidden in the system. Moreover, it would be impossible to trace back the reasons of making mistakes and thus to avoid them in the future. When exchanging the information across applications, mistakes can be easily spread around, which would rise the concern on the quality of service. Furthermore, without accurate information stored in the system, results of applying analytical tools would be diluted. The author level metric tools (e.g., citations, h-index, etc.) would be either overestimated or underestimated.

3 Conceptual Layer

In the common practice of conceptual modelling, objects are modelled with the *unique-key-value property* such that two objects sharing the same values on all key attributes are meant to be the same object. As exemplified in the previous section, it is inevitable to have objects existing in an open-world domain whose known properties are not sufficient to uniquely identify themselves, so we have to remove the unique-key-value property in conceptual modelling. Then the question of how to uniquely distinguish objects in an open-world domain arises. For this, a straightforward approach is to use object identifiers. Their representation is not important but the interrelationship among them does matter.

We revise the definition of Higher-order Entity-Relationship Model (HERM) [7,10]. There are four kinds of objects: entities, relationships, clusters and collections. Every object is associated with a unique identifier.

Definition 1. *Let* $\mathfrak{D} = \{D_i\}_{i \in I}$ *be a fixed family of basic domains,* $\mathcal{O}$ *be the universal set of identifiers,* $\mathcal{D} = \bigcup_{i \in I} D_i$ *and* $\mathcal{D} \cap \mathcal{O} = \emptyset$*.*

- *An* entity type τ_E *(or* relationship type *on level* 0*) consists of a finite non-empty set of attributes:* $attr(\tau_E) = \{A_1, \ldots, A_m\}$ *and a domain assignment:* $dom : attr(\tau_E) \rightarrow \mathfrak{D}$*. An* entity *of type* τ_E *is a pair* (i, e) *with an identifier* $i \in \mathcal{O}$*, and a mapping* $e : attr(\tau_E) \rightarrow \mathcal{D}$ *with* $e(A) \in dom(A)$ *for all* $A \in attr(\tau_E)$*.*

- *A* relationship type τ_R *on level* $k + 1$ *consists of a finite non-empty set* $comp(\tau_R)$ *of object types in which each has level at most* k *and at least one must have level exactly* k*, a finite set of attributes:* $attr(\tau_R) = \{A_1, \ldots, A_m\}$ *and a domain assignment* $dom : attr(\tau_R) \rightarrow \mathfrak{D}$*. A* relationship *of type* τ_R *is a pair* (i, r) *with an identifier* $i \in \mathcal{O}$ *and a mapping* $r : comp(\tau_R) \cup attr(\tau_R) \rightarrow \mathcal{O} \cup \mathcal{D}$ *with* $r(\tau) \in \mathcal{O}$ *for all* $\tau \in comp(\tau_R)$ *and* $r(A) \in dom(A)$ *for all* $A \in attr(\tau_R)$*.*

- *A* cluster type $\tau_C = \tau_1 \oplus \cdots \oplus \tau_n$ *consists of a finite, non- empty set of object types* $\tau_1, ..., \tau_n$*. A* cluster *of type* τ_C *is an object of type* τ_i *where* $i \in [1, n]$*.*

- *A* collection type τ_L *has a single object type* τ*. We denote a list-type by* $\tau_L = [\tau]$*, a set-type by* $\tau_L = \{\tau\}$ *and a bag-type by* $\tau_L = \langle\tau\rangle$*. A* collection *of type* τ_L *is a finite list (finite set, finite bag, respectively) of objects of type* τ*.*

A conceptual model M *is a pair* $\langle \mathfrak{U}, \mathfrak{O} \rangle$ *such that* $\mathfrak{U} = \mathcal{D} \cup \mathcal{O}$ *is the base set of* M *and* $\mathfrak{O} = \mathcal{E} \cup \mathcal{R} \cup \mathcal{C} \cup \mathcal{L}$ *is the set of objects in* M*, where* $\mathcal{E}, \mathcal{R}, \mathcal{C}$ *and* $\mathcal{L}$ *represent a finite set of all entities, relationships, clusters and collections in* M*, respectively.*

To serve our purpose effectively, the conceptual modelling process should obey the following principles. Firstly, data which may have variant expressions should be modelled as an object rather than a representation of a finite set of values. An object-based view on these data can empower us to establish a flexible abstraction level handling the identity of authors, whereas a value-based view cannot. For example, when modelling authors as objects, we may define an identity relation among author identifiers. In doing so, different objects modelled for an author are interconnected via the identity relation, in which each object is allowed to have a variant of representation for the author. In contrast with this, modelling an author in terms of a set of values leaves us an oversimplified choice which ignores the diversity of objects and thus cannot always be correct – either treating authors represented by the same set of values as the same person or treating them as being different.

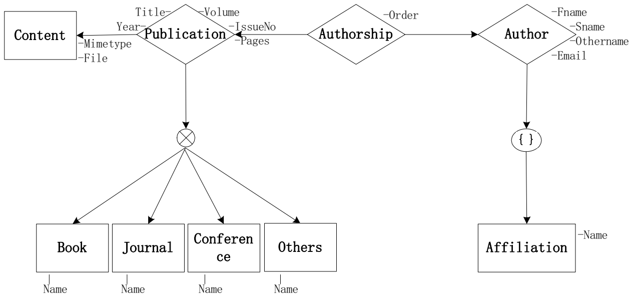

Fig. 1. A simple conceptual model

Example 3. Fig. 1 provides a conceptual model for publications, which is simplified for clarity. Publication meta-data such as affiliations, books, journals and conferences are modelled as objects because capturing their variants is of interest. Other publication meta-data that can be interpreted without ambiguousness such as page numbers, volumes, issues, etc. are modelled as values.

The second principle is the separation of concerns on modelling objects and identifying objects. Taking author tagging as an example, authors of publications which look like to be the same person still should be modelled as being different objects. Meanwhile, the knowledge of why they might be the same person should be captured at a different layer - identity layer, which will be discussed in Section 4. In doing so, a conceptual model may serve as a faithful reflection of the print world and the like, while all the knowledge on interrelationships of objects are managed at the identity layer to enable a flexible and continuous adaptation of knowledge that can cope with an evolving application domain.

4 Knowledge-Based Identification

The knowledge-based identification resides at an identity layer consisting of a ground model and a set of refining identity tuples. By the combined effects of a ground model and a set of refining identity tuples, the knowledge of author identification can be efficiently handled at two different levels: structural and instance-based levels.

Let $\mathcal{O}_\tau$ be a set of identifiers of type τ. Then all identity relations discussed in this section are defined as: $\mathcal{O}_\tau \times \mathcal{O}_\tau \Rightarrow \{true, false\}$. Furthermore, we will use Abstract State Machines (ASMs) [4] to establish models in this section.

4.1 Ground Model

The intuition behind a ground model is to establish an *approximate identity relation* E^A based on the general knowledge of finding the variants of an author. Each general knowledge is represented by a generic query returning pairs of identifiers. A *generic query* must respect the genericity principle [2,8] and only concerns about structural properties of author variants; otherwise, a query is said to be *non-generic.*

Definition 2. *A* ground model *consists of a set* $\{Q_1, ..., Q_k\}$ *of generic queries such that*

```
par
    forall $x$ with $x \in \mathcal{O}$
        do $E^A(x,x) := true$ enddo
    forall $x_1, x_2$ with $E^A(x_1, x_2)$
        do $E^A(x_2, x_1) := true$ enddo
    forall $x_1, x_2$ with $Q_1(x_1, x_2) \vee ... \vee Q_k(x_1, x_2)$
        do $E^A(x_1, x_2) := true$ enddo
par
```

Since generic queries may use the intermediate results in E^A, a ground model is defined after iterating the above rule until a fixpoint of E^A is reached.

Example 4. Suppose that we have the following relational schemata for objects Affiliation, AffiliationSet, Author and Authorship shown in Fig. 1:

- Affiliation = {ID, Name};
- AffiliationSet = {ID, {AffiliationID}};
- Author = {ID, FName, SName, OtherName, Email, AffiliationSetID};
- Authorship = {ID, Order, PublicationID, AuthorID},

and the generic queries Q_n, Q_c, Q_a and Q_e representing the following rules:

1. `Affiliation-rule`: two affiliations are identical if they have the same name.
   $Q_n(x, y) := \exists z.\textsc{Affiliation}(x, z) \wedge \textsc{Affiliation}(y, z)$
2. `AuthorCollaboration-rule`: two authors are identical if they have the same name and both have co-authored with at least one other author.
   $Q_c(x, y) := \exists x_1, x_2, y_1, y_2, z_1, z_2, z_3, z, z_1', z_2', z_3', z'.E^A(z, z')\wedge$
   $\textsc{Authorship}(x_1, x_2, z_3, x) \wedge \textsc{Authorship}(z_1, z_2, z_3, z)\wedge$
   $\textsc{Authorship}(y_1, y_2, z_3', y) \wedge \textsc{Authorship}(z_1', z_2', z_3', z')$
3. `AuthorAffiliation-rule`: two authors are identical if they have the same name and both are associated with the same affiliation.
   $Q_a(x, y) := \exists x_1, x_2, x_3, x_4, x_5, x_6, x_4', x_5', x_6', z, z'.$
   $\textsc{Author}(x, x_1, x_2, x_3, x_4, x_5) \wedge \textsc{AffiliationSet}(x_5, x_6)\wedge$
   $\textsc{Author}(y, x_1, x_2, x_3, x_4', x_5') \wedge \textsc{AffiliationSet}(x_5', x_6')\wedge$
   $\wedge z \in x_6 \wedge z' \in x_6' \wedge E^A(z, z')$
4. `AuthorEmailaddress-rule`: two authors are identical if they have the same email address.
   $Q_e(x, y) := \exists x_1, x_2, x_3, x_4, x_5, x_1', x_2', x_3', x_5'.$
   $\textsc{Author}(x, x_1, x_2, x_3, x_4, x_5) \wedge \textsc{Author}(y, x_1', x_2', x_3', x_4, x_5')$

The approximate identity relation E^A defined by this ground model would be

```
par
    ......
    forall $x, y$ with $Q_n(x, y) \vee Q_c(x, y) \vee Q_a(x, y) \vee Q_e(x, y)$
        do $E^A(x, y) := true$ enddo
par
```

A ground model abstracts the general knowledge on author identification, however, some accurate problems still exist. (i) *partial applicability*: e.g., people change their surnames after marriage or inconsistently use name abbreviations in their publications. (ii) *partial correctness*: e.g., two persons who have the same name work in the same affiliation or both co-author with another person.

4.2 Stepwise Refinement

As a ground model can not handle exceptional cases in author identification, we need an approach to refine it. The idea is to use two refining relations working towards opposite dimensions and capturing specific knowledge on author identification, called *positive and negative identity relations* and denoted as E^+ and E^-, respectively. A tuple $E^+(x_1, x_2)$ states that identifiers x_1 and x_2 are identical while a tuple $E^-(x_1, x_2)$ states that identifiers x_1 and x_2 are not identical.

Definition 3. *The refining relations E^+ and E^- are associated with the sets $\{Q_1^+, ..., Q_n^+\}$ and $\{Q_1^-, ..., Q_m^-\}$ of non-generic queries, respectively, such that*

```
par
    forall $x_1, x_2$ with $Q_1^+(x_1, x_2) \vee .... \vee Q_n^+(x_1, x_2)$
        do $E^+(x_1, x_2) := true$ enddo
    forall $x_1, x_2$ with $E^+(x_1, x_2)$
        do $E^+(x_2, x_1) := true$ enddo
    forall $x_1, x_2$ with $Q_1^-(x_1, x_2) \vee ... \vee Q_m^-(x_1, x_2)$
        do $E^-(x_1, x_2) := true$ enddo
    forall $x_1, x_2$ with $E^-(x_1, x_2)$
        do $E^-(x_2, x_1) := true$ enddo
par
```

Example 5. Suppose that an author named “Susan Lee” with identifier i_1 and an author named “Susan Maneth” with identifier i_2 refer to the same person in the real world because Susan Lee changed her surname to Maneth after marriage. For this, we can add a tuple $E^+(i_1, i_2)$ in E^+. Similarly, suppose we know that an author named “Susan Lee” with identifier i_1 is different from an author named “Susan Lee” with identifier i_3. For this, we can add a tuple $E^-(i_1, i_3)$ in E^-.

It is possible that the knowledge on identifying a specific author may conflict with each other. For instance, we might have both $E^+(i_1, i_2)$ and $E^-(i_1, i_2)$ in the refining relations. To solve such conflicting knowledge, it is important to automatically discover all the inconsistencies induced by two refining relations. To handle this, we may consider tuples in E^+ and E^- as propositions in a propositional logic. Meanwhile, two propositions $E^+(x_1, x_2)$ and $E^-(x_1, x_2)$ should always satisfy the axiom $E^+(x_1, x_2) \Leftrightarrow \neg E^-(x_1, x_2)$. In doing so, we can infer the consistency of two refining relations by using propositional tableaux for a proposition ϕ that is a conjunction of all tuples in E^+ and E^-. Let n and m be the numbers of tuples in E^+ and E^-, respectively. Then we have

$$\phi = E^+(x_1, x_1^{'}) \wedge ... \wedge E^+(x_n, x_n^{'}) \wedge E^-(y_1, y_1^{'}) \wedge ... \wedge E^-(y_m, y_m^{'}).$$

If the proposition ϕ is true, then the refining relations are consistent. When the refining relations are consistent, they can be utilised to fine-tune a ground model. We thus obtain an *exact identity relation* E^E which provides the decisive knowledge of author identification at the identity layer. That is,

```
par
    forall $x_1, x_2$ with $(E^A(x_1, x_2) \vee E^+(x_1, x_2)) \wedge \neg E^-(x_1, x_2)$
        do $E^E(x_1, x_2) := true$ enddo
    forall $x_1, x_3$ with $E^E(x_1, x_2) \wedge E^E(x_2, x_3)$
        do $E^E(x_1, x_3) := true$ enddo
par
```

Let S denote an identity layer containing E^E. Then any changes on the ground model and refining relations which consequently affect E^E can be captured by a finite set Δ of updates in the form of $(E^E(i_1, i_2), true)$ or $(E^E(i_1, i_2), false)$. In doing so, the knowledge at an identity layer can be continuingly stepwise refined via various learning processes such that $S_1 \rightarrow^{\Delta_1} S_2 \rightarrow^{\Delta_2} ... \rightarrow^{\Delta_{n-1}} S_n$.

Remark 1. The ASM methods [4] have been intensively used in system design and analysis, in which ground models are established for capturing requirements and then turned into executable code by stepwise refinements. At its core, the principle of substitutivity does not have to be obeyed. As our purpose of using the ground model and refinement methods is to specify the evolvement of knowledge in an application domain, the refinements of knowledge comply with the principle of consistency, instead of the principle of substitutivity.

5 Visual Layer

A visual layer is a federation of a conceptual model and an identity layer. By applying the knowledge of author identification stored at an identity layer over a conceptual model, authors referring to the same person are identified and represented as a unified object at a visual layer. Formally speaking, a set of equivalence classes of identifiers is defined in terms of E^E.

Definition 4. *An* equivalence class *of an identifier* $i \in \mathcal{O}$ *w.r.t.* E^E *is the subset of all identifiers in* $\mathcal{O}$ *which are equivalent to* i*, i.e.,* $[i] = \{i^{'} \in \mathcal{O} | E^E(i^{'}, i)\}$.

Following the convention, the set of all equivalence classes in $\mathcal{O}$ with respect to E^E is called the *quotient set* of $\mathcal{O}$ by E^E and denoted as $\mathcal{O}/E^E$.

Definition 5. *Let $M = (\mathfrak{U}, \mathfrak{O})$ for $\mathfrak{U} = \mathcal{D} \cup \mathcal{O}$ be a conceptual model and E^E be an exact identity relation defined by an identity layer. Then a* visual model *$K = (\mathfrak{U}^{'}, \mathfrak{O}^{'})$ for $\mathfrak{U}^{'} = \mathcal{D} \cup \mathcal{O}/E^E$ is defined by a homomorphism $h : M \to K$ such that (1) for each value $a \in \mathcal{D}$, $h(a) = a$; (2) for each identifier $i \in \mathcal{O}$, $h(i) = [i]$; (3) for each entity $(i, e) \in \mathfrak{O}$, $(h(i), e) \in \mathfrak{O}^{'}$ is an entity; (4) for each relationship $(i, r) \in \mathfrak{O}$, $(h(i), h(r)) \in \mathfrak{O}^{'}$ is a relationship, where $r : comp(\tau_R) \cup attr(\tau_R) \to \mathcal{O} \cup \mathcal{D}$ with $r(\tau) \in \mathcal{O}$ for all $\tau \in comp(\tau_R)$ and $r(A) \in dom(A)$ for all $A \in attr(\tau_R)$, and $h(r) : comp(\tau_R) \cup attr(\tau_R) \to \mathcal{O}/E^E \cup \mathcal{D}$ with $h(r)(\tau) \in \mathcal{O}/E^E$ for all $\tau \in comp(\tau_R)$ and $r(A) \in dom(A)$ for all $A \in attr(\tau_R)$; (5) for each cluster $u \in \mathfrak{O}$, $h(u) \in \mathfrak{O}^{'}$ is a cluster; (6) for each collection $[l_1, ..., l_n]$, $\{l_1, ..., l_n\}$ or $\langle l_1, ..., l_n \rangle \in \mathfrak{O}$, $[h(l_1), ..., h(l_n)]$, $\{h(l_1), ..., h(l_n)\}$ or $\langle h(l_1), ..., h(l_n) \rangle \in \mathfrak{O}^{'}$ is a collection.*

A visual layer is always in flux so as to reflect the evolving knowledge on author identification. By knowledge acquisition and reasoning across different application domains, the degree of precision and completeness on the identity of authors can be improved over time. The following figure presents an overall picture for this multi-layer architecture with knowledge-based identification.

$$\begin{array}{rccccccc}
\text{Conceptual Layer} & M & & M & & ... & & M \\
 & + & & + & & ... & & + \\
\text{Identity Layer} & S_1 & \rightarrow^{\Delta_1} & S_2 & \rightarrow^{\Delta_2} & ... & \rightarrow^{\Delta_{n-1}} & S_n \\
 & = & & = & & ... & & = \\
\text{Visual Layer} & K_1 & & K_2 & & ... & & K_n
\end{array}$$

6 Discussion

We discuss several author identification services built upon the knowledge-based identification in the proposed multi-layer architecture and an intelligent way of managing the knowledge acquired from different systems.

Knowledge Sharing. The knowledge of author identification can be shared via various forms of services. We may provide author profile services which contain all the information associated with individual authors such as affiliations, names, email addresses, publications, etc. These services can be published as data feeds providing regularly updates (e.g., HTML, Atom or RSS feeds), as widgets providing dynamically content embedded in other applications (e.g., arXiv myarticles widget), as Web APIs providing the capability for marshups (e.g., Scopus APIs), as interactive tools into social networks or other systems (e.g., Thomson Reutors ResearcherID Upload, arXiv Facebook application) and so on. We may also generate author authority files to help repositories control author names.

With the additional abstraction for knowledge-based identification, it would be easy to trace the knowledge of author identification such as who, when and why add a piece of specific or general knowledge into the identity layer. Thus, the reliability of services can be enhanced and well controlled. Moreover, we can treat the whole identity layer as a plug-and-play service applied on a conceptual model so as to reuse the successful knowledge and in the meantime rapidly deploy the knowledge into new systems.

Knowledge Acquisition. The way of acquiring knowledge within this multi-layer architecture involves two steps: (1) analyse and extract all explicit or implicit knowledge of author identification from external services provided by third-party providers; (2) store the knowledge of author identification and other primitive data relating to publications (including their citation metrics) into the identity and conceptual layers, respectively. One of the biggest advantages offered by this multi-layer architecture is to effectively reason about the knowledge of author identification integrated from different systems. Any conflicts between the added new knowledge and the existing knowledge can be automatically discovered in the integration process. It can thus help detect accidental or historical errors contained in the knowledge of author identification. In doing so, collective knowledge from different communities can be acquired after checking the quality of external services. In addition to this, this architecture provides great flexibility and efficiency for managing author tagging. For example, when new knowledge has been acquired after integrating a service provided by other scholarly communities such as Thomson Reutors ResearcherID Download, Scopus RSS feed, etc., this knowledge can be instantly applied on all the primitive data at the conceptual layer. Since there is no need for tagging authors individually, a vast amount of heavy work on author tagging, which currently happens in practice can be saved.

7 Conclusion

We proposed a multi-layer architecture to tackle the author identification problem. An additional layer for managing the knowledge of author identification has been established, lying between a conceptual layer and a view layer. In doing so, we can build an adaptive visual model by binding a conceptual model with knowledge-based identification to capture the identity of authors in an evolving application domain. In the future we will further investigate identity services to integrate collective knowledge from different communities.

References

1. Credit where credit is due. Nature 462(7275), 825 (December 2009)
2. Aho, A.V., Ullman, J.D.: Universality of data retrieval languages. In: Proceedings of Principles of programming languages, pp. 110–119. ACM Press, New York (1979)
3. Bennett, R., Hengel, C., Hickey, T., O'Neill, E., Tillett, B.: Virtual international authority file. In: ALA Annual Conference, New Orleans (2006)
4. Börger, E., Stärk, R.F.: Abstract State Machines: A Method for High-Level System Design and Analysis. Springer, Heidelberg (2003)
5. Bourdon, F., Webb, R.: International cooperation in the field of authority data: an analytical study with recommendations. KG Saur (1993)

6. Habibzadeh, F., Yadollahie, M.: The problem of who. International Information and Library Review 41(2), 61–62 (2009)
7. Hartmann, S., Link, S.: Collection type constructors in entity-relationship modeling. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 307–322. Springer, Heidelberg (2007)
8. Hull, R., Yap, C.K.: The format model: a theory of database organization. In: Proceedings of Principles of database systems, pp. 205–211. ACM Press, New York (1982)
9. Swan, A.: Author identification web page, `http://repinf.pbworks.com/Author-identification`
10. Thalheim, B.: Entity-Relationship Modeling: Foundations of Database Technology. Springer, Heidelberg (2000)