Exploring Data Fusion under the Image Retrieval Domain

Nádia P. Kozievitch[1], Carmem Satie Hara[2], Jaqueline Nande[2] and Ricardo da S. Torres[3]
[1]*Dept. of Informatics, Federal University of Technology, Curitiba, Brazil*
[2]*Dept. of Informatics, Federal University of Paraná, Curitiba, Brazil*
[3]*Institute of Computing, University of Campinas, Campinas, Brazil*

Keywords: Content-based Image Retrieval, Data Fusion, XML, Metadata.

Abstract: Advanced services in data compression, data storage, and data transmission have been developed and are widely used to address the required capabilities of an assortment of systems across diverse application domains. In order to reuse, integrate, unify, manage, and support heterogeneous resources, a number of works and concepts have emerged with the aim of facilitating aggregation of content and helping system developers. In particular, images, along with existing Content-Based Image Retrieval services, have the potential to play a key role in information systems, due to the large availability of images and the need to integrate them with existing collections, metadata, and available image manipulation softwares and applications. In this work, we explore a data fusion approach for solving data value conflicts in the context of image retrieval domain. In particular, we target the process of solving value conflicts resulted from different features integrating the data resulted from the Content-Based Image Retrieval process, along with the image metadata, provided from a number of sources and applications. Our approach reduces the need of human intervention for keeping a clean and integrated view of an image repository when new data sources are added to an image management system.

1 INTRODUCTION

The process of organizing new information, and integrate them with data acquired from external sources are usually very time consuming tasks. Users involved in managing these data are often looking for ways to improve their productivity, and it is thus important to provide them with effective tools to reuse and aggregate content.

Motivated by the need for integration and interoperability, a number of works have proposed the aggregation of different information combined together to compose a single logical object. The resulting object has been denoted as *Aggregation* (Williams and Suleman, 2003), *Component-Based Object* (Santanchè and Medeiros, 2007; Santanchè et al., 2007), *Complex Object* (Nelson et al., 2001), and *Compound Object* (Awre, 2009). In particular, images are a representative example of a data source which is generally integrated and combined with different components, such as metadata, links, videos, and image manipulation softwares and applications.

One common strategy used to support image searches in large datasets is called Content-Based Image Retrieval (CBIR) (Torres et al., 2006). Roughly, the process can be divided in three steps: (1) feature vectors that represent image visual properties (such as color, texture, and shape) are extracted; (2) the similarity between images are computed based on the distance between their feature vectors; and (3) the most similar collection images are returned as the search result.

In fact, there are a number of works (Akbar et al., 2008; Nanni et al., 2011) that propose the integration or parallel use of several feature vectors, but few that propose the integration of a general purpose data fusion system in the context of CBIR, considering both image and metadata. A fusion process involves both entity resolution and cleaning. Entity resolution refers to the problem of identifying overlapping data in different sources. This problem has been the subject of extensive research on relational (Lim et al., 1996), entity-relationship (Menestrina et al., 2006), and XML (Poggi and Abiteboul, 2005) data models. Cleaning refers to the process of solving attribute value conflicts. In particular, several issues can be cited regarding integration of different sources in the CBIR domain: (i) existence of duplicate images, (ii) the use of different descriptors, (iii) the existence of images with different transformations (crop, resolu-

P. Kozievitch N., Satie Hara C., Nande J. and da S. Torres R..
Exploring Data Fusion under the Image Retrieval Domain.
DOI: 10.5220/0004869901710178
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 171-178
ISBN: 978-989-758-027-7

tion, etc.), (iv) definition of different ranked lists (e.g., defined in terms of different descriptors), (v) different sizes of ranked lists, among others.

Most of existing systems for data fusion consider data structured on relational format. Nevertheless, given that XML has become the *de facto* standard for data exchange on the Web, it is natural to also consider this format in the fusion process.

In this paper we report on a system that combines a general purpose data fusion system in the context of CBIR. In particular, we target the process of solving value conflicts resulted from different features integrating the data resulted from the CBIR process, along with the image metadata, provided from a number of sources and applications. The main novelty resides in automatically solving conflicts in CBIR without user intervention, using rules to integrate images, metadata, and sources along the fusion process. Nevertheless, these formalized rules can also be extended to compare, and highlight the differences among different multimedia resources.

1.1 A Motivating Example

Consider an infrastructure (shown in Figure 1) which has as input images and metadata and as output a XML file, aggregating the input data with resulting resources from the CBIR process. For instance, consider **Source 1**, which stores the CBIR information for Figure 2-a (a parasite image), within an XML file. The XML file is composed of a feature vector **FV1** (processed by a color descriptor – BIC), the similarity scores **SM1**, an image **IM1**, and the respective metadata **M1** (with attributes such as name, path, size, and resolution).

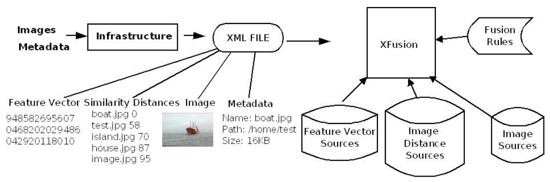


Figure 1: Fusion process integrating CBIR sources.

Now consider **Source 2**, where the same image (shown in Figure 2-b) was processed by another descriptor (SASI texture descriptor). The resulting XML file would include the feature vector **FV2**, with the respective similarity scores **SM2**, and metadata **M2**. Assume that both sources need to be integrated and stored in a repository. That demands that existing conflicts among entities need to be solved. The resulting XML file should automatically integrate the CBIR information, and metadata on **Source 1** and **Source 2** within one XML tree structure.

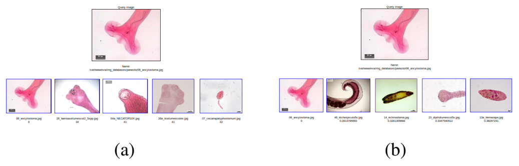


Figure 2: CBIR for image *01_ancylostoma.jpg*, using the (a) BIC (Stehling et al., 2002) descriptor and (b) SASI (Carkacioglu and Yarman-vural, 2001) descriptor.

In XFusion (Cecchin et al., 2010), the user defines a set of rules for solving these conflicts after merging the information imported from several data sources. In this paper, using basic services from XML, we explore XFusion in order to aggregate different CBIR-related data. The novelty resides on the use of fusion rules to automate the decision process regarding conflicted values, considering aggregation of metadata, images, and CBIR-related data.

1.2 Organization

The remainder of this paper is organized as follows. Section 2 contains a description of related work. An overview of our solution is described in Section 3. A case study is presented in Section 4. Finally, we present our conclusions and draw future work in Section 5.

2 RELATED WORK

2.1 Integration of Resources

Multiple definitions have been used (Kozievitch et al., 2011a; Nelson et al., 2001) to name the integration of resources into a single digital object as *Aggregation* (Williams and Suleman, 2003), a *Component-Based Object* (Santanchè and Medeiros, 2007; Santanchè et al., 2007), a *Complex Object* (Nelson et al., 2001; Lagoze et al., 2006), or a *Compound Object* (Awre, 2009).

Several integration formats arise from different communities (Nelson and de Sompel, 2006; Nelson et al., 2001; Fox and France, 1997; Karpovich et al., 1994; Burnett et al., 2006). In particular, Santanchè used the idea of integration within the field of software reuse and exchange (Santanchè and Medeiros, 2007; Santanchè et al., 2007), in a component-based technology named Digital Content Component (DCC). A DCC is composed of four distinct subdivisions: **(a) content**, **(b) structure**,**(c) interfaces**, and **(d) metadata**, used to manage different layers of the object aggregations.

2.2 Images and Applications

If we consider image data, new challenges have emerged to handle the Content-Based Image Retrieval services. A typical solution for this service requires the construction of **image descriptors**, which are characterized by: (i) an *extraction algorithm* to encode image features into *feature vectors*; and (ii) a *similarity measure* to compare two images based on the distance between their corresponding feature vectors. The similarity measure is a *matching function*, which gives the degree of similarity for a given pair of images represented by their feature vectors, often defined as a function of the distance (e.g., Euclidean), that is, the larger the distance value, the less similar the images.

There are several applications which support services based on image content, allowing integration in distinct domains (Murthy et al., 2010; Achananuparp et al., 2007). And along with these applications, images can also be explored to integrate annotation support for image digital libraries such as (Jochum et al., 2007). In particular, consider for the following examples a recently developed component-based CBIR infrastructure aiming to process and encapsulate images and related data (Kozievitch et al., 2012; Kozievitch et al., 2011b), presented in Figure 3. The bottom layer contains the data sources: the descriptor library, the image collection, and the database. The second layer contains DCCs, which encapsulate, provide access, and manage several parts of the CBIR process: the descriptors, the descriptor library, the images, the retrieval from the database, and finally a manager, for setting up the entire process. The third layer comprises the applications which manipulate the image aggregations, by accessing the CBIR process or the data sources.

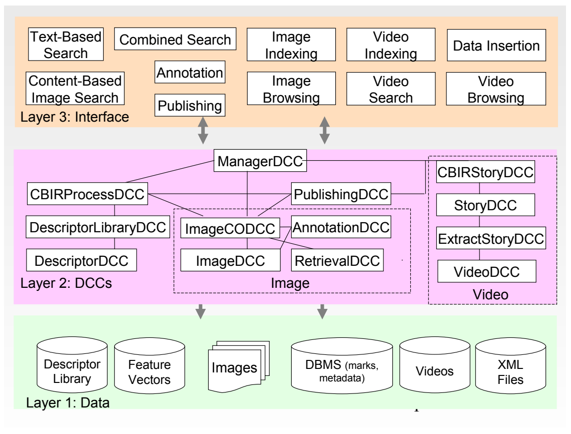


Figure 3: Management Layers (Kozievitch et al., 2012).

2.3 Data Integration and Cleaning

Data fusion and cleaning have been studied extensively by the database community (Bhattacharya and Getoor, 2006; Bleiholder and Naumann, 2008). Most of previous works consider data on relational format, but recently it has been stressed the need for investigating the problem of solving conflicts on semi-structured data. XClean (Weis and Manolescu, 2007) is a system that allows declarative and modular specification of a cleaning process. It consists of a declarative language with operators that cover not only the fusion process, but also entity identification and combination of values that refer to the same object. The main goal is to provide a modular system that can be easily extended with new operators. Potter's Wheel (Raman and Hellerstein, 2001) follows a cleaning strategy based on a set of operations to transform data, such as *format, drop, copy, merge, split, divide, fold* and *select*. However, instead of storing the result of a data transformation, the sources are stored along with the definition of the transformation. The transformation is applied on-the-fly whenever a consistent and clean information is required.

Several systems have been proposed in the literature on (i) the fusion process (such as Hummer (Bilke et al., 2005) and Fusionplex (Motro and Anokhin, 2006)), (ii) sharing structured data (Orchestra (Ives et al., 2008)), and (iii) update data-oriented workflow (Panda (Ikeda and Widom, 2010)).

There are a number of strategies for data fusion proposed in the literature (Yin et al., 2008; Dong et al., 2010; Cao et al., 2013; Fan et al., 2013), and a survey can be found in (Bleiholder and Naumann, 2008). The approach presented within this paper allows strategies to be reapplied in subsequent integration processes, and also keeps provenance information for tracing back the origin of the data stored in the repository.

3 OVERVIEW OF OUR SOLUTION

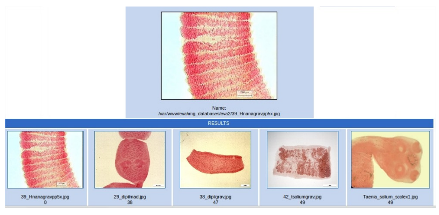
From a formal perspective, an aggregation of Content-Based Image Retrieval components comprises an structure that aggregates the image, feature vector, and similarity scores (Kozievitch et al., 2011a). In addition, each digital object might have metadata (such as file name, file size, among others).

In this section, we outline the proposed two-step solution, namely the information extraction from images, described in Section 3.1 and the fusion process, detailed in Section 3.2.

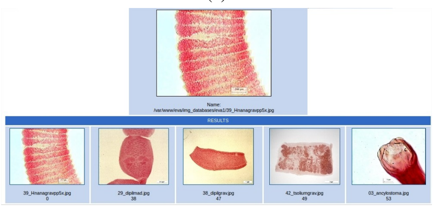
3.1 Gathering the Information from the CBIR Process

Suppose that within the parasite domain, a researcher has metadata, several images, and their respective CBIR information for several species (Kozievitch et al., 2010), from different sources. Recall that the CBIR process is responsible for creating the respective image feature vector and similarity scores among the images.

Consider now the CBIR infrastructure presented in Figure 3. As input, consider image *35_hnanagravpp5x.jpg* and respective metadata. Within **Source 1**, the image is processed by the BIC descriptor (Stehling et al., 2002), showed in Figure 4. The CBIR infrastructure aggregates all the related information within an XML file. Within **Source 2**, the same image processed by the same descriptor. In this case, however, **Source 2** considers a different image collection, i.e., images found within **Source 2** are not necessarily the same managed within **Source 1**. Note that, comparing the two XML files, besides metadata conflicts that might appear (such as different file paths), both images also present a different ranked lists (the 5th image is different in both ranked lists, as shown in Figure 8).



(a)



(b)

Figure 4: CBIR for image *35_hnanagravpp5x.jpg*, within (a) **Source 1** and (b) **Source 2**.

3.2 Fusion of Image-related Data

The data fusion problem refers to the merging of data provided from two or more sources. In particular, the CBIR information presented in Figure 4-a and Figure 4-b could be summarized within a XML tree representation, as shown in Figure 7. Basically each image (identified by a name) is processed by an image descriptor (identified by a name), which thereby provides a feature vector and a ranked list of similar images.

```
<db>
<image>
  <image_name>39_Hnanagravpp5x.jpg</image_name>
  <image_path>/tmp/39_Hnanagravpp5x.jpg</image_path>
  <image_feature_vector_name>/tmp/fv/39_Hnanagravpp5x.jpg
  </image_feature_vector_name>
  <image_descriptor><descriptor>Bic</descriptor>
   <distance><rank>1</rank>
     <imageId>39_Hnanagravpp5x.jpg</imageId>
     <image_dist_value>0</image_dist_value></distance>
   <distance><rank>2</rank>
     <imageId>29_dipilmad.jpg</imageId>
     <image_dist_value>38</image_dist_value></distance>
   <distance><rank>3</rank>
     <imageId>38_dipilgrav.jpg</imageId>
     <image_dist_value>47</image_dist_value></distance>
   <distance><rank>4</rank>
     <imageId>42_tsoliumgrav.jpg</imageId>
     <image_dist_value>49</image_dist_value></distance>
   <distance><rank>5</rank>
     <imageId>Taenia_solium_scolex1.jpg</imageId>
     <image_dist_value>49</image_dist_value></distance>
  </image_descriptor>
 </image>
</db>
```

Figure 5: XML file for Source 1, representing Figure 4-a.

```
<db><image>
  <image_name>39_Hnanagravpp5x.jpg</image_name>
  <image_path>/home/nadiapk/data/uploads/39_Hnanagravpp5x.jpg</image_path>
 <image_feature_vector_name>/data/fv/39_Hnanagravpp5x.jpg
  </image_feature_vector_name>
  <image_descriptor>
  <descriptor>Bic</descriptor>
   <distance> <rank>1</rank>
     <imageId>39_Hnanagravpp5x.jpg</imageId>
     <image_dist_value>0</image_dist_value></distance>
   <distance> <rank>2</rank>
     <imageId>29_dipilmad.jpg</imageId>
     <image_dist_value>38</image_dist_value> </distance>
   <distance><rank>3</rank>
     <imageId>38_dipilgrav.jpg</imageId>
     <image_dist_value>47</image_dist_value> </distance>
   <distance> <rank>4</rank>
     <imageId>42_tsoliumgrav.jpg</imageId>
     <image_dist_value>49</image_dist_value></distance>
    <distance><rank>5</rank>
     <imageId>03_ancylostoma.jpg</imageId>
     <image_dist_value>53</image_dist_value></distance>
  </image_descriptor>
 </image>
</db>
```

Figure 6: XML file for Source 2, representing Figure 4-b.

Consider now the XML documents presented in Figures 5 and 6. Each element provided by Source 1 has a corresponding one in Source 2. However, some of the values associated with elements disagree, such as the elements `image_path`, `image_feature_vector_name`, and the fifth element of the ranked list (`imageId` and `image_dist_value`), as illustrated in Figure 8. Note that there are two types of conflicts: *metadata conflicts* (such as the first two listed above) and *CBIR conflicts* (such as elements within the ranked list).

A cleaning strategy for solving metadata conflicts may determine that whenever a data item provided from **Source 1** disagrees with any other source, we should choose **Source 1**'s value over the others, for example. As a result of applying this strategy, the data repository keeps a single consistent value for all subelements.

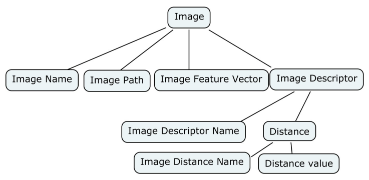


Figure 7: XML tree representation for CBIR-related data.

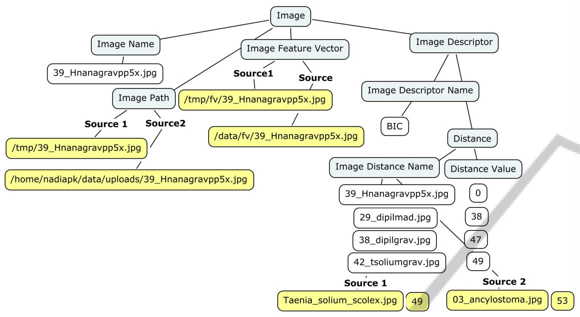


Figure 8: XML tree representation with conflicting elements (in yellow) from Source 1 and Source 2.

A cleaning strategy for CBIR conflicts may also determine that all conflicting values should be kept within the repository. As a result of applying this strategy, the union of the ranked lists are kept within the XML representation after cleaning, as shown in Figure 9.

XFusion is a system which provides the functionality described within this paper. It relies on the concepts of XML keys (Buneman et al., 2002) for determining which elements should be merged when integrating one or more data sources, and a set of cleaning rules for generating a consistent data repository.

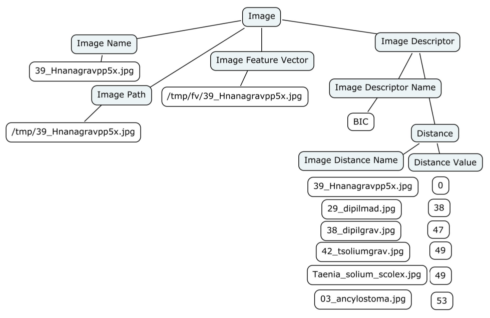


Figure 9: XML tree representation after cleaning.

3.2.1 XML Keys

XML keys are used within our context to express the result of the entity matching process. Thus, whenever objects from distinct data sources agree on their keys, they are merged into a single object in the repository. When the values of non-key objects differ we say that a *conflict* has been detected.

An XML key is defined as *(context-path, (target-path,* { *key-paths*})), where the values of the *key-paths* uniquely identify nodes reached following a *target-path* in the context of each subtree defined by the *context-path*. In order to generate the merge document in Figure 9, keys were defined within the CBIR context, such as:

- *(ε, (image,* {*image_name*}*)*: in the context of the entire document, (ε denotes the root), an image is identified by its name;
- *(image, (image_descriptor,* {*descriptor*}*))*: in the context of the each subtree rooted at *image*, the *image_descriptor* is identified by *descriptor*;
- *(image/image_descriptor, (distance,* {*rank*}*))*: in the context of the each subtree rooted at *image/image_descriptor*, their elements *distance* are identified by their *rank*;
- *(image, (image path,* {}*))*: in the context of any subtree rooted at image node, there is at most one *image_path*.

Although here we have presented keys following the syntax proposed in (Buneman et al., 2002), in (Cecchin et al., 2010) the key definitions are stored within an XML format. Note that within CBIR domain, additional keys can be used to define if how many descriptors can be used do characterize the visual properties of an image, how many images will be available at the ranked list fusion, etc.

3.2.2 Rules

Rules define high-level strategies for deciding how value conflicts should be solved. The context of a rule is defined by a path expression, and a list of strategies for solving a conflict. There are a number of strategies proposed in the literature for solving value conflicts. The following subset of those proposed by (Bleiholder and Naumann, 2008) is adopted by XFusion:

- *Trust Your Friends*. This strategy is based on a reliability criterion: the value provided by the source with the highest confidence rate assigned by the user is chosen to be stored in the repository;
- *Meet In The Middle*. This strategy mediate the conflict by generating a new value which represents an average among all conflicting values;
- *Cry With The Wolves*. This strategy is defined by choosing the value reported by the majority of data sources;
- *Roll The Dice*. A random value is choose among the conflicting ones; and

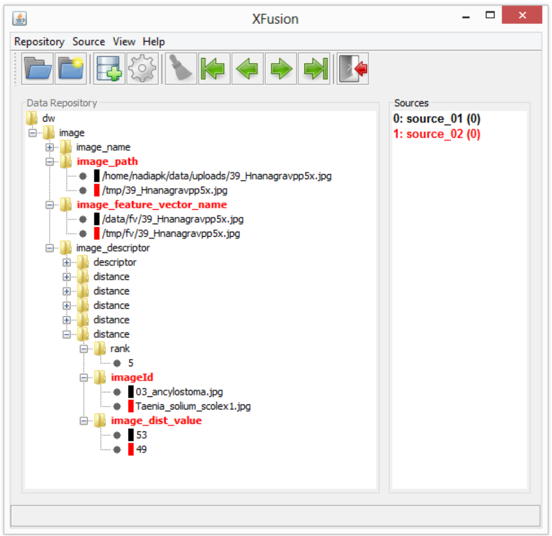
- *Pass It On*. All the conflicting values are kept in the repository.

As an example, consider the following fusion rule: *(/image[image_name="39_Hnanagravpp5x.jpg"] /image_path, [Trust your Friends, Pass it On])*
The rule determines that whenever image `''39_Hnanagravpp5x.jpg''` conflicts on the `image_path` element, the conflict is solved by first applying strategy *Trust your Friends*, followed by strategy *Pass it On*. Considering that the user has assigned a higher confidence rate to **Source 1** over **Source 2**, the result of the rule's application is illustrated in Figure 9.

Note that rules can be defined on larger contexts than on single elements. Suppose, for example, that the same strategy described above should be applied to any conflict on `image_path` elements. This rule can be expressed as:
(/image/image_path, [Trust your Friends, Pass it On])
With such a rule, future conflicts on this element do not require any user intervention.

3.2.3 XFusion Usage

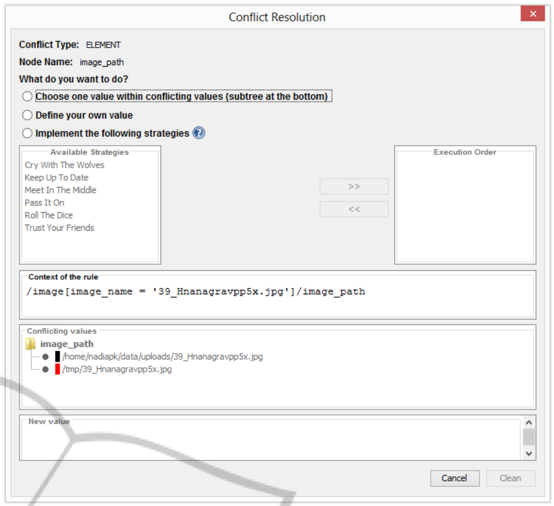
XFusion has a graphical interface that allows the user to load data sources into the repository and perform cleaning operations through the definition of fusion rules. The tool's main window is shown in Figure 10, with the main screen after both Sources 1 and 2, presented in Figures 5 and 6, have been uploaded. Note that the right window shows all uploaded sources, associating a different color with each of them, and showing their confidence rate inside the parenthesis.



```
dw
  image
    image_name
    image_path
      ● ▮/home/nadiapk/data/uploads/39_Hnanagravpp5x.jpg
      ● ▮/tmp/39_Hnanagravpp5x.jpg
    image_feature_vector_name
      ● ▮/data/fv/39_Hnanagravpp5x.jpg
      ● ▮/tmp/fv/39_Hnanagravpp5x.jpg
    image_descriptor
      descriptor
      distance
      distance
      distance
      distance
      distance
        rank
          ● 5
        imageId
          ● ▮03_ancylostoma.jpg
          ● ▮Taenia_solium_scolex1.jpg
        image_dist_value
          ● ▮53
          ● ▮49
```

0: source_01 (0)
1: source_02 (0)

Figure 10: XFusion main screen.



Conflict Type: ELEMENT
Node Name: image_path
What do you want to do?
- ○ Choose one value within conflicting values (subtree at the bottom)
- ○ Define your own value
- ○ Implement the following strategies

Available Strategies: Cry With The Wolves, Keep Up To Date, Meet In The Middle, Pass It On, Roll The Dice, Trust Your Friends

Context of the rule

```
/image[image_name = '39_Hnanagravpp5x.jpg']/image_path
```

Conflicting values

```
image_path
  ● ▮/home/nadiapk/data/uploads/39_Hnanagravpp5x.jpg
  ● ▮/tmp/39_Hnanagravpp5x.jpg
```

New value

Figure 11: XFusion conflict resolution screen.

The merged document is shown in the main window, with value conflicts identified with the sources that provide the conflicting values, represented by the colored squares that precede each value.

When the user selects an existing conflict and clicks on the resolve button, the screen depicted in Figure 11 is shown. Note that the user has three main options for solving a conflict: choose one among the conflicting values, manually insert a new value, or apply some of the available strategies (described in Section 3.2.2). Strategies are chosen by clicking on the direction buttons in the middle of the screen, determining the order in which they should be considered.

Below the strategy boxes, the context of the rule is presented. This path is originally set to uniquely identify the conflicting element or attribute, according to the XML keys defined on the repository. Nevertheless, the user can edit the path for applying the list of strategies on larger contexts. Finally, when the user clicks on the *Clean* button, the new rule is inserted into the policy base and its execution propagates the chosen value to the repository. XFusion allows the user to define rules incrementally. The user can then check whether the strategy has been effective for solving all conflicts within the rules context. If not, she may decide to extend the rule by defining additional strategies to be applied.

In particular, within the context of the presented parasite domain (Kozievitch et al., 2010), the fusion of different sources centralizes the data, providing more information about CBIR services and metadata. This strategy can provide parasite experts with an integrated view on available datasets, which can foster knowledge sharing. The same approach could be explored within other CBIR-related applications.

4 CONCLUSIONS

Many digital library implementations and applications demand additional and advanced services to effectively specify, reuse, describe and aggregate different resources. Examples of commonly required services include those related to the support of images and related CBIR tasks.

In this paper, we address the integration of image retrieval with XFusion, a rule-based cleaning tool that stores curated data in an integrated repository. A metamodel is proposed in order to specify the components of the CBIR related tasks, validated through a case study within the parasite domain. The main novelty resides in automatically solving conflicts in CBIR without user intervention, using rules to integrate images and associated metadata.

A straightforward future work consists in the use of rules to guide the design and implementation of image digital libraries that integrate different (and possibly distributed) image collections. One starting point relies on the use of applications, like those proposed in (Gonçalves and Fox, 2002; Zhu et al., 2003).

ACKNOWLEDGEMENTS

We would like to thank CAPES, CNPq, FAPESP, AMD, Microsoft Research, and Fundação Araucária.

REFERENCES

Achananuparp, P., McCain, K. W., and Allen, R. B. (2007). Supporting student collaboration for image indexing. In *ICADL'07*, pages 24–34, Berlin, Heidelberg. Springer-Verlag.

Akbar, S., Kung, J., and Wagner, R. (2008). Multishape-features and text-feature integration on 3d model similarity retrieval. *Int. J. Innov. Comput. Appl.*, 1(3):171–184.

Awre, C. (2009). Managing compound objects within Fedora, Enhanced E-theses Project Deliverable 9, available at http://igitur-archive.library.uu.nl/DARLIN/2010-0526-200241/UUindex.html. *Knowledge Exchange Group*.

Bhattacharya, I. and Getoor, L. (2006). Collective entity resolution in relational data. *IEEE Data Engineering Bulletin*, 29(2):4–12.

Bilke, A., Bleiholder, J., Naumann, F., Böhm, C., and Weis, M. (2005). Automatic data fusion with hummer. In *Proc. of the 31st VLDB Conference*, pages 1251–1254.

Bleiholder, J. and Naumann, F. (2008). Data fusion. *ACM Comput. Surv.*, 41(1):1:1–1:41.

Buneman, P., Davidson, S., Fan, W., Hara, C., and Tan, W.-C. (2002). Keys for XML. *Computer Networks*, 39(5):473–487.

Burnett, I. S., Pereira, F., de Walle, R. V., and Koenen, R. (2006). *The MPEG-21 Book.* John Wiley & Sons.

Cao, Y., Fan, W., and Yu, W. (2013). Determining the relative accuracy of attributes. In *SIGMOD'13: Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 565–576.

Carkacioglu, A. and Yarman-vural, F. (2001). Sasi: A new texture descriptor for content based image retrieval. *IEEE International Conference on Image Processing*, 2:137–140.

Cecchin, F., Ciferri, C. D. A., and Hara, C. (2010). XML Data Fusion. In *International Conference on Data Warehousing and Knowledge Discovery (DaWaK'2010).*

Dong, X., Berti-Equille, L., Hu, Y., and Srivastava, D. (2010). SOLOMON: Seeking the truth via copying detection. *PVLDB*, 3(2):1617–1620.

Fan, W., Geerts, F., Tang, N., and Yu, W. (2013). Inferring data currency and consistency for conflict resolution. In *ICDE'13: Proceedings of the IEEE International Conference on Data Engineering*, pages 470–481.

Fox, E. A. and France, R. K. (1997). Architecture of an expert system for composite document analysis, representation, and retrieval. In *Readings in Information Retrieval*, pages 400–412. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Gonçalves, M. A. and Fox, E. A. (2002). 5SL: A Language for Declarative Specification and Generation of Digital Libraries. In *JCDL '02*, pages 263–272, New York, NY, USA. ACM.

Ikeda, R. and Widom, J. (2010). Panda: A system for provenance and data. *IEEE Data Engineering Bulletin*, 33(3):42–49.

Ives, Z. G., Green, T. J., Karvounarakis, G., Taylor, N. E., Tannen, V., Talukdar, P. P., Jacob, M., and Pereira, F. (2008). The Orchestra collaborative data sharing system. *SIGMOD Record*, 37(3):26–32.

Jochum, W., Kaiser, M., Schellner, K., and Wirl, F. (2007). Living memory annotation tool — image annotations for digital libraries. In *Proc. of the 11th European conference on Research and Advanced Technology for Digital Libraries*, ECDL '07, pages 549–550, Berlin, Heidelberg. Springer-Verlag.

Karpovich, J. F., Grimshaw, A. S., and French, J. C. (1994). Extensible file system (elfs): an object-oriented approach to high performance file i/o. *ACM SIGPLAN Notices*, 29(10):191–204.

Kozievitch, N. P., Almeida, J., da S. Torres, R., Santanchè, A., Leite, N. J., Murthy, U., and Fox, E. A. (2012). Reusing a compound-based infrastructure for searching and annotating video stories. *International Journal of Multimedia Technology*, 2:89–97.

Kozievitch, N. P., Almeida, J., Torres, R. S., Leite, N. A., Gonçalves, M. A., Murthy, U., and Fox, E. A. (2011a). Towards a Formal Theory for Complex Objects and Content-Based Image Retrieval. *JIDM*, 2(3):321–336.

Kozievitch, N. P., da S. Torres, R., Santanchè, A., Pedronette, D. C. G., Calumby, R. T., and Fox, E. A.

(2011b). An infrastructure for searching and harvesting complex image objects. *The Information - Interaction - Intelligence (I3) Journal*, 11(2):39–68.

Kozievitch, N. P., Torres, R. d. S., Andrade, F., Murthy, U., Fox, E., and Hallerman, E. (2010). A teaching tool for parasitology: enhancing learning with annotation and image retrieval. In *ECDL'10*, pages 466–469, Berlin, Heidelberg. Springer-Verlag.

Lagoze, C., Payette, S., Shin, E., and Wilper, C. (2006). Fedora: an architecture for complex objects and their relationships. *Int. J. Digit. Libr.*, 6:124–138.

Lim, E., Srivastava, J., Prabhakar, S., and Richardson, J. (1996). Entity identification in database integration. *Information Sciences*, 89(1).

Menestrina, D., Benjelloun, O., and Garcia-Molina, H. (2006). Generic entity resolution with data confidences. In *Proc. of VLDB Work. on Clean Databases*.

Motro, A. and Anokhin, P. (2006). Fusionplex: resolution of data inconsistencies in the integration of heterogeneous information sources. *Information Fusion*, 7(2):176–196.

Murthy, U., Kozievitch, N. P., Leidig, J., da S. Torres, R., Yang, S., Goncalves, M., Delcambre, L., Archer, D., and Fox, E. A. (2010). Extending the 5S Framework of Digital Libraries to support Complex Objects, Superimposed Information, and Content-Based Image Retrieval Services. Technical Report TR-10-05, Virginia Tech, Department of Computer Science.

Nanni, L., Brahnam, S., and Lumini, A. (2011). Combining different local binary pattern variants to boost performance. *Expert Syst. Appl.*, 38(5):6209–6216.

Nelson, L. and de Sompel, H. V. (2006). IJDL special issue on complex digital objects: Guest editors' introduction. *International Journal of Digital Libraries*, 6(2):113–114.

Nelson, M. L., Argue, B., Efron, M., Denn, S., and Pattuelli, M. C. (2001). A survey of complex object technologies for digital libraries. Technical report, NASA/TM-2001-211426.

Poggi, A. and Abiteboul, S. (2005). XML data integration with identification. In *Proc. of DBPL*, pages 106–121.

Raman, V. and Hellerstein, J. M. (2001). Potter's wheel: An interactive data cleaning system. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 381–390.

Santanchè, A. and Medeiros, C. B. (2007). A Component Model and Infrastructure for a Fluid Web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):324–341.

Santanchè, A., Medeiros, C. B., and Pastorello Jr, G. Z. (2007). User-author centered multimedia building blocks. *Multimedia Systems*, 12(4):403–421.

Stehling, R. O., Nascimento, M. A., and Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM '02*, pages 102–109, New York, NY, USA. ACM.

Torres, R. d. S., Medeiros, C. B., Gonçalves, M., and Fox, E. A. (2006). A Digital Library Framework for Biodiversity Information Systems. *International Journal on Digital Libraries*, 6(1):3–17.

Weis, M. and Manolescu, I. (2007). Declarative XML data cleaning with XClean. In *International Conf. on Advanced Information Systems Engineering (CaiSE)*, pages 96–110.

Williams, K. and Suleman, H. (2003). A survey of digital library aggregation services. In *Scholarship at Penn Libraries, available at http://works.bepress.com/martha_brogan/10*.

Yin, X., Han, J., and Yu, P. S. (2008). Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, 20(6):796–808.

Zhu, Q., Gonçalves, M. A., and Fox, E. A. (2003). 5SGraph demo: a graphical modeling tool for digital libraries. JCDL '03, pages 385–385, Washington, DC, USA. IEEE Computer Society.