http://ieeexplore.ieee.org/document/6841963/

DOI: 10.1109/IACS.2014.6841963

2014 5th International Conference on Information and Communication Systems (ICICS)

Achieving consistency in collaborative image annotation systems

Hafed Zarzour*
Department of Computer Science
Mohamed Cherif Messaadia University
41000, Souk-Ahras, Algeria
hafed.zarzour@gmail.com

Mokhtar Sellami
LABGED, Department of Computer Science
Badji Mokhtar University
23000, Annaba, Algeria.
m.sellami@dgrsdt.dz

Abstract— **Collaborative image annotation is a useful strategy for assigning a set of labels, or keywords to an image, taking into account its content. While existing collaborative image annotation frameworks facilitate sharing, indexing, and understanding of huge number of images, newly-developed methods let group of users manage dynamically-updating data. Conflict-free Replicated Data Type method, used in several collaborative systems, offers to distributed participants through the web a real support to concurrently annotate copies of shared image file regardless of their order. A major benefit of this method, therefore, is to provide a good framework for representing semantic stores and a good mechanism for solving the concurrent updating problem without complex control in collaborative image annotation process. In this paper, we propose a novel optimistic replication approach for collaborative replicated image annotation stores that ensures eventual consistency. We describe how Open Annotation Collaboration can be extended to further support real collaboration between users allowing them to work together effectively to achieve common goals.**

Keywords- Semantic Web, Collaborative Image Annotation, Open Annotation Collaboration, Consistency, CRDT, Concurrent Annotation.

I. INTRODUCTION

Collaborative image annotation (CIA) is a useful strategy for assigning a set of labels, or keywords to an image, taking into account its content. This strategy is of great interest as it allows users to collaborate in order to interpret and pull meaning from an image. With CIA, distributed users can add, modify or remove information from an image file without modifying the file itself. The annotations can be thought of as a layer on top of the existing image, and this annotation layer is usually visible to other users who share the same annotation tool. The major benefits include indexing, retrieving, and understanding of large collections of image data. This helps in developing different viewpoints. When an image is shared and collaboratively annotated in distributed architecture, several factors should be considered and addressed. As such factors, we can cite all the factors relative to local annotation plus other due to sharing process in distributed environment. In this study, we only consider the problem managing concurrent updates on the collaborative image annotation system using semantic web technologies.

In the last few years, the CIA related to semantic web technologies have attracted a growing research interest [1]. The purpose of the Open Annotation Community Group [2] is to work towards a common, RDF(Resource Description Framework)-based, specification for annotating digital resources. As a result, the OAC (Open Annotation Collaboration) [3] seeks to facilitate the emergence of a Web and Resource-centric interoperable annotation environment that allows leveraging annotations across the boundaries of annotation clients, annotation servers, and content collections. Therefore, it would be interesting to develop a new generation of CIA that integrates the OAC concepts. A system of CIA based on OAC is considered as correct and sound if it preserves the CCI model [4, 5] that means Causality, Consistency, and Intention preservation defined as follows: (1) Causality: the execution order of all annotation operations is performed in the same way on each copy, (2) Convergence: when the system is idle, all annotation copies are identical. (3) Intention: the expected effect of a delete and insert annotation operations must be observed on all copies.

To achieving eventual consistency, CRDT (Conflict-free Replicated Data Types) [6], has been developed as a new class of methods to ensure convergence without any synchronization requirement. This approach states that all concurrent operations commute, allowing replicas to execute operations in different orders with the guarantee that the replicas will be identical at the end of collaborative session.

In this paper, we propose CIA-Store a new commutative replicated date type for collaborative annotating the images stores using OAC as basic data structure integrated with other parameters. This association will allow to use OAC not only for representing and storing annotations about images but also to support real collaboration between users allowing them to work together effectively to achieve common goals where all concurrent annotations commute. CIA-Store approach ensures also the CCI consistency model explained earlier.

The paper is structured as follows. Section 2 presents backgrounds and related works. Section 3 details our proposition model for collaborative video annotation based on CRDT concepts. Section 4 discusses our approach. Section 5 concludes the paper and points to future works.

II. BACKGROUNDS AND RELATED WORK

A. Open Annotation Collaboration

Open Annotation Collaboration is a set of architectures and specifications that aims to annotate any kind of web resources in an interoperable and semantic aspect. Therefore, OAC develops an interoperable vision for creating relationships between annotation pieces, related resources, using a tool that conforms to the World Wide Web vision. Any OAC-based application can easily be shared between different environments, with sufficient richness of expression satisfying complex requirements while remaining simple enough to also enable for the most common situations, such as attaching a piece of text to a single web resource. In OAC data model, annotations are expressed in an RDF and modeled as a set of connected URI-addressable resources, including one or more annotation target resources, and one or more annotation body resources, i.e. the annotation content or source. The Body and Target are identified by HTTP URIs unless they are embedded within the annotation.

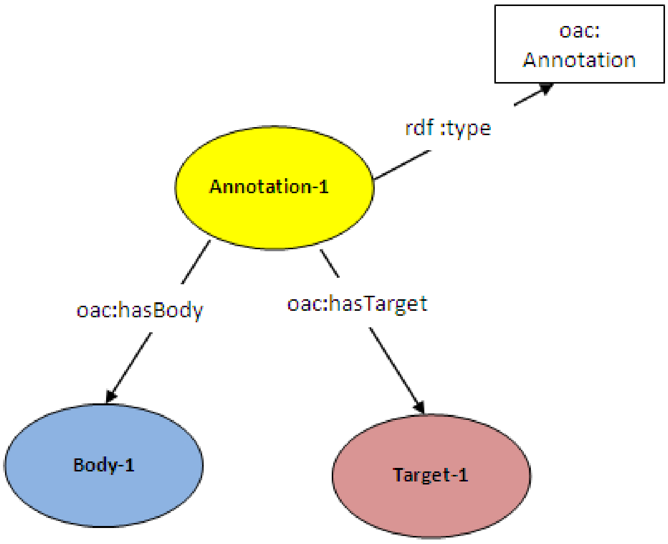

Figure 1. Baseline OAC Data model

Figure 1 illustrates OAC ontology that describes three classes of resources:

1) oac:Annotation: A document denoted by Annotation-1 and identified by an HTTP URI that includes the body and target resources involved in the annotation.

2) oac:Body: The body related to the annotation and denoted by Body-1. The body is somehow about the target resource. It is the information which is annotating the target.

3) oac:Target: The resource denoted by Target-1 that is being annotated. Like the body, target can be any URI identified resource.

The relationships oac:hasBody and oac:hasTarget are the annotation relationship between body and target respectively.

B. Collaborative image annotation systems

Many solutions have been developed in the image annotation research field [7, 8, 9, 10]. Most of these solutions define models for indexing, storing and sharing large collections of image data. However, synchronization of concurrent image annotations is not taken in consideration.

OntoELAN [11] inherits the porosities of ELAN [12], a linguistic annotation tool, for presenting a structured tool with an ontology based annotation method. OntoELAN can read and explore ontology written in OWL to create language profile and ontological layer. Anvil [13] is a tool that gives support for hierarchical multilayered annotations, visualization of waveform and pitch contour and offers an intuitive annotation board that shows color-coded elements on multiple tracks in time alignment.

In [14], authors detail a web-based media annotation suite that supports image, audio and video content. This suite provides a client interface for image annotation by selecting image parts and adding a marker with textual annotation. It also offers Semantic Web capabilities enabling users to augment existing images with related resources on the Web like resource derived from DBpedia[15].Users can select image segments adding descriptors and inserting textual annotation. VIA [16] is an annotation tool that allows participant to upload its descriptors from a given OWL ontology and to create a set of annotations of specific video regions and enable the captivation of movement trajectories. LabelMe [17], a web-based platform to collect user annotations in still images, is a significant example. However, LabelMe lacks intelligent mechanisms for quality control and integration of user annotations. In fact, the LabelME dataset, though being one of the largest datasets available, it is particularly inaccurate [18]. In [19], authors propose an image annotation system which exploits the semantic relationships between image tags. Each image tag has typically semantic information itself. This observation enables to find the tag contextualization by connecting the image tags. However, this system has several limitations, especially, if there are few user-assigned tags, the annotation cannot fully express the semantic of image contents.

Recently, Semantic Web-based annotation system [20] is presented that enables user annotations to form semantically structured knowledge at different levels of granularity. Annotation can be reused by external applications and mixed with Web of Data sources to enable 'serendipity,' the reuse of data produced for a specific task by different people and in different contexts from the one data originated from. PerLa [21] is another web-based platform for creating cooperatively object detection, tracking and recognition ground truth for big data which also integrates crowd sourcing methods for annotation integration.

C. Synchronizing semantic data without CRDT

RDFSync [22] is an algorithm for synchronizing a semantic data. Semantic data is defined as RDF graphs where each RDF graph is decomposed unequivocally into minimal subsets of triples and canonically represented by ordered lists of the identifiers. To ensure the synchronization, the difference is performed between the source and the target of the ordered list. However, it is not explicitly specified what happens in the case of concurrent updates on copies.

Delta [23] is an ontology designed for the distribution of differences between RDF graphs. It compares two RDF graphs by generating a sequence of differences then updates a graph from a sequence of differences. However, the presented algorithms do not explain how convergence strategy is applied in order to ensure eventual consistency in an efficient way.

Edutella [24] presents P2P platform for semantic data based on metadata. Its mechanism focuses on querying RDF metadata stored in distributed RDF stores. A replication service is proposed as complements local storage by replicating in additional peers to achieve metadata persistence / availability and workload balancing while maintaining metadata integrity and consistency. However, they do not mention how to replicate and synchronize metadata.

RDFGrowth [25] proposes a semantic data sharing environment where each peer can only update the shared data or read them. Concurrent operations are integrated by merge algorithms. However, the anatomy of RDFGrowth allows sharing of data but not collaborating. RDFPeers [26] is one of the first efforts for structured peer to- peer RDF stores. The key idea is to use a MAAN overlay [27] to index a triple three times, once based on the subject, another based on the predicate, and a final based on the object. However, it lacks the ability for supporting collaborative update operations on replicas [28].

D. Synchronizing semantic data with CRDT

CRDT [6] is a new framework where all concurrent updating operations must commute to ensure convergence. Initially, the algorithm has been successfully applied to different data representations types in scalable collaborative system for linear data type [29], tree document structure data type [30] and semi-structured data type [31].

Recently, many CRDT are proposed to support collaborative editing of semantic stores having set structure [32]. C-Set [33] is a data structure defined as CRDT for sets that can be integrated within a semantic store in order to provide P2P synchronization of autonomous semantic store. The main idea of C-set is to assign a counter to each triple of set for tracking how many times a triple t has been added or removed. To this end, four operations are defined on this set. The delete operation del() can performed locally and sends remote delete operation rdel() that is executed remotely. The ins() is an insert operation executed locally. It sends remote insert operation rins() that is executed remotely. However, they do not mention how to ensure the causality and preserve the intention of operations. Although c-set has been designed to ensure consistency, it violates the operations intentions especially when it comes to mutually execute remote delete operations on the same triples that locally have already been removed several times then reinserted.

In [34] authors present different set CRDTs, Grow Only Set (G-Set), Last Writer Wins Set (LWW-element-Set) and Observed Remove Set (OR-Set). In a G-Set, there is only an insertion operation where each element can be inserted and not deleted from the set. The reconciliation Principe is based on simple set union, since union is commutative. In a LWW-element-Set, A timestamp is attached to each element. If an element is not already exists, a local operation updates its timestamp and adds it to the set and cannot be scalable. In an Observed Remove Set (OR-Set) each element is associated to a set of unique tag. A local add creates a tag for the element and a local remove deletes all the tag of the element. However, G-Set ignores the intention of remove operations, LWW-element-Set is not enable to scale since it uses the tombstone mechanism and OR-Set requires transparent mechanism of unique tag generation between different sites.

SU-Set [35] presents a CRDT for RDF graphs based on OR-Set that supports the SPARQL 1.1 Upadate operation and guarantees consistency. SU-Set is designed to serve as base for an RDF-Store CRDT that could be implemented in an RDF engine. Since OR-Set considers only insertion and deletion of single elements, it is not possible to apply OR-Set directly to SPARQL Update. Therefore, SU-Set modifies the operations to send the relevant set of triples to affect one by one, but that could flood the network with traffic considering the potential size of an RDF-Graph. However, SU-Set relies on causal delivery of the underlying network, which is challenging and can pose problems in highly dynamic platforms.

III. CIA- STORE DEFINITION

One of the main powers of CIA-Store is the ability to use different order for the image annotation updating. This distinguishes CIA-Store from several other collaborative image annotation systems. As motivating example, consider two users on two remote sites (site 1 and site 2) who collaboratively annotate an image store through two instances for the same shared data: The image store is described as a set of RDF triples where each triple contains three components <subject, predicate, object>. Initially, Site 1 inserts an annotation triple <imgAnno1, hasBody, body1> by executing O_1=insAn(<imgAnno1, hasBody, body1>) then propagates it to Site 2. After that, Site 1 removes a certain annotation triple <imgAnno1, hasTarget2, target2> by executing O_3= delAn(<imgAnno1,hasTarget2,target2>) , at the same time Site 2 performs O_2= insAn(<imgAnno1,hasTarget2,target2>) in order to insert the same annotation triple but without deleting it back. After the mutual propagation, reception and re-execution of generated operations, the replicated annotation images stores diverge, this means that the eventual consistency is violated when the same operations are performed in different order. If two operations of annotation affect the same triple, they are potentially in conflict. To resolve this conflict, it must be decided which of the annotation operations is to be taken into account, while the other will be ignored or preserved in the same order of execution for all sites. (see Fig. 2).

Many collaborative image annotations such as this one combine concurrent annotation aspect and RDF data types, which motivates a general-purpose system. However the system must take into account the commutativity mechanism when concurrent annotation operations are performed, especially on the same annotation triple; for instance running the inserting and deleting operations in different order in different sites must not diverge. Therefore, It is obvious that the direct integration between the image annotation and the existing OAC specifications will not work and conduct to inconsistent results because running a delete operation before an insert operation will fail since the commutativity of operations does not supported in the manner in which the annotation operations are executed.

In CIA-Store, a new commutative replicated date type is defined for collaborative annotating the images stores using OAC as basic data structure integrated with other parameters. This association will allow to use OAC not only for representing and storing annotations about images but also to support real collaboration between users allowing them to work together effectively to achieve common goals where all concurrent annotations commute.

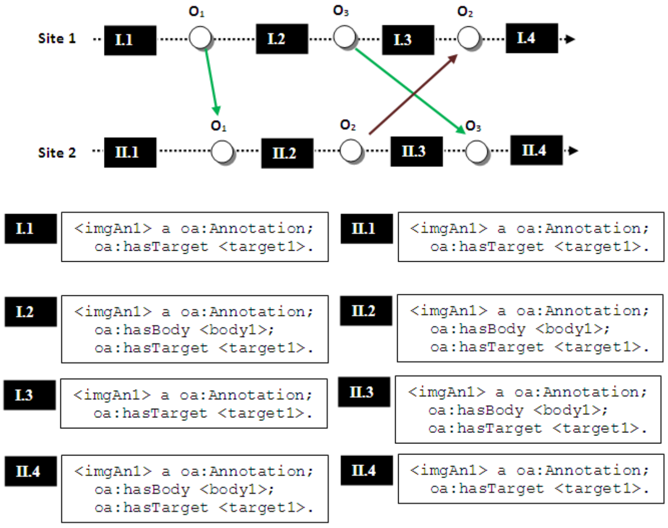

Figure 2. Divergence after integrating concurrent image annotations

A. *CIA-Store Data model.*

Our idea is to combine both advantages of semantic web technologies and collaborative image annotation. In CIA-Store vision, updating annotations generated locally are re-executed on remote copies without any requirement of total order on annotations, thus, concurrent annotations can be re-played in different orders. CIA-Store is based on CRDT for guaranteeing the consistency of all replicated copies when users execute the same sequence of annotations but in different orders. This demonstrates the importance of the commutativity propriety which assumes that changing the order of the annotations running does not change the final result.

To achieve commutativity on OAC structure for collaborative image annotation, our idea is to separate the added and deleted elements during the annotation process by using two additional repositories. The first one contains inserted annotations, whilst the second contains removed annotations. The consistent data will be obtained from the calculation of the difference between the multiplicities of both inserted and deleted repositories.

Definition 1: An OAC store is, denoted by Im, is a repository used for storing image annotations. It is a pair (T, M), where T is a set of annotation triples and M is a multiplicity function.

The notion of multiplicity M is important to be able to count the number of occurrences for a given annotation triple performed by a site. As OAC conveys in the Turtle RDF format adopted specifications, we use this format expressing annotations data in RDF data model. Therefore, the annotation image store can be formalized in Turtle RDF format as:

<S> a oa:Annotation ;

oc:P <O> < M>.

where S, A and O are subject, predicate and object respectively, and M is a multiplicity function.

Definition 2: An Insert OAC store, denoted by, iOAC-Store, is an OAC store which includes all added annotation triples inserted by the user along with the multiplicity values MA.

Definition 3: A remove OAC store, denoted by, rOAC-Store, is an OAC store which includes contains all annotation triples combined with the multiplicity MR removed by the user.

iOAC-Store and rOAC-Store have the ability to keep an event log about each added or deleted image annotation. Such repositories include a data collection used to provide the information necessary to compute the consistent OAC all along the annotating session.

Definition 4: Given an Insert OAC store iOAC-Store and a remove OAC store rOAC-Store. The coherent OAC store, denoted by cOAC-Store, is an obtained OAC that includes all annotation triples contained in iOAC-Store such that the multiplicity values of such annotation triples are greater than their corresponding in rOAC-Store in a way that annotation triples that their multiplicity values in rOAC-Store are greater than or equal to those of iOAC-Store are disregarded.

The coherent OAC store includes the same consistent result regardless of the kind of submitted operations. In addition, it will be identical in all on sites each time a given annotation is updated.

Definition 5: A final annotation store, denoted by fOAC-Store, is a triple < iOAC-Store, rOAC-Store, cOAC-Store>, where iOAC-Store is an insert OAC store, rOAC-Store is a remove OAC store and cOAC-Store is an coherent OAC store.

The fOAC-Store is replicated at each site for supporting not only a collaborative image annotation but also a concurrent updating regardless of the causal propagation. Finally, CIA-Store data model looks like it is shown in figure 3.

$$fOAC\text{-}Store=(iOAC\text{-}Store,\ rOAC\text{-}Store,\ cOAC\text{-}Store)=(<A_A,M_A>,\ <A_R,M_R>,A>)$$

:

$A_A=<S_A,P_A,O_A>$

$A_R=<S_R,P_R,O_R>$

$A=<S,P,O>$

Figure 3. CIA-Store data structure

Figure 4 shows CIA-Store following to RDF Turtle format used by the community of OAC in the specification phases.

Figure 5 shows how to construct the elements of cOAC-Store from iOAC-Store and rOAC-Store about collaborative annotation of Breast cancer images. All possible cases are presented in this sample, only the first, second, fourth and last annotation triples of iOAC-Store appear in cOAC-Store because they have multiplicity greater than the same annotation triples in rOAC-Store. Thus, the coherent OAC store includes

```
<S_A> a oa:Annotation ;      // iOAC-Store
  oa:P_A <O_A> < M_A> <M_A>.

<S_R> a oa:Annotation ;      // rOAC-Store
  oa:P_R <O_R> < M_R> <M_R>.

<S> a oa:Annotation ;        // cOAC-Store
  oa:P <O> .
```

Figure 4. CIA-Store data structure according to RDF Turtle format used in OAC specifications

<BreastTumor, hasBody, 'http://en.wikipedia.org/wiki/Breast_tumor'>, <BreastTumor, hasBody, http://www.cancercoachchris.com/?p=731'> <BreastTumor, hasTarget, 'http://www.flickr.com/photos/80666910@N03/7469353854/' >, <BreastTumor, hasTarget, 'http://www.flickr.com/photos/58937697@N00/228513113/'>. The second annotation triple does not get removed as its multiplicity is two in the iOAC-Store whilst it does not exist in the rOAC-Store. Meanwhile, the third has a multiplicity value in the iOAC-Store which is less than the one in the rOAC-Store. The manner in which the coherent OAC store is computed, support the commutativity between any generated operations and guarantee eventual consistency in any case. This mechanism of resulting RDF store construction ensures convergence and consistency in any case. As a result, all contributors should have the identical annotation image store after updating each annotation triple.

```
<BreastTumor> a oa:Annotation ;                                        // iOAC-Store
  oa:hasBody 'http://en.wikipedia.org/wiki/Breast_tumor' <2>;
  oa:hasBody 'http://www.cancercoachchris.com/?p=731' <3>;
  oa:hasTarget 'http://www.flickr.com/photos/ikayama/4441942736/' <4>;
  oa:hasTarget 'http://www.flickr.com/photos/80666910@N03/7469353854/' <2> ;
  oa:hasTarget 'http://www.flickr.com/photos/58937697@N00/228513113/' <1>.

<BreastTumor> a oa:Annotation ;                                        // rOAC-Store
  oa:hasBody 'http://en.wikipedia.org/wiki/Breast_tumor' <1>;
  oa:hasTarget 'http://www.flickr.com/photos/ikayama/4441942736/' <5>;
  oa:hasTarget 'http://www.flickr.com/photos/58937697@N00/228513113/' <1>.

<BreastTumor> a oa:Annotation ;                                        // cOAC-Store
  oa:hasBody 'http://en.wikipedia.org/wiki/Breast_tumor';
  oa:hasBody 'http://www.cancercoachchris.com/?p=731' ;
  oa:hasTarget 'http://www.flickr.com/photos/80666910@N03/7469353854/';
  oa:hasTarget 'http://www.flickr.com/photos/58937697@N00/228513113/'.
```

Figure 5. Annotation example in CIA-Store

B. Actions

An action realizing an image annotation can add information that can't be easily included in descriptions. It in general should highlight features, details, or points of interest within a shared image. In distributed collaborative annotation context, when a user modifies the local annotation replica, the site generates a corresponding operation that realizes the user's intention. Thereafter, this operation is immediately executed at the local replica and then it is broadcasted to all other users in order to be executed. Indeed, there are two basic annotating actions that affect an image OAC store: insert and delete. Meanwhile, the update operation can be considered or made equivalent as a delete of the existing value to be updated followed by an insert of the new value. The insert and delete functions are defined as follows:

1) *Insert action on t:* is an update operation in which the annotation triple t is added in the insert OAC-Store

2) *Delete action on t:* is an update operation in which the annotation triple t is added in the remove OAC store rOAC-Store.

Let us consider again the scenario presented in Figure 2. When the remove operation O_3 is retrieved and performed on Site 2, the multiplicity of the triple T in rOAC-Store is incremented to 2. When the insert operation O_2 is integrated Site 1, the multiplicity of the corresponding triple is incremented. The consistency between the rOAC-Stores on Site 1 and Site 2 is achieved. We can observe that after executing concurrent modifications, Site 1 and Site 2 now converge and the last rOAC-Stores are the same (see Fig 6).

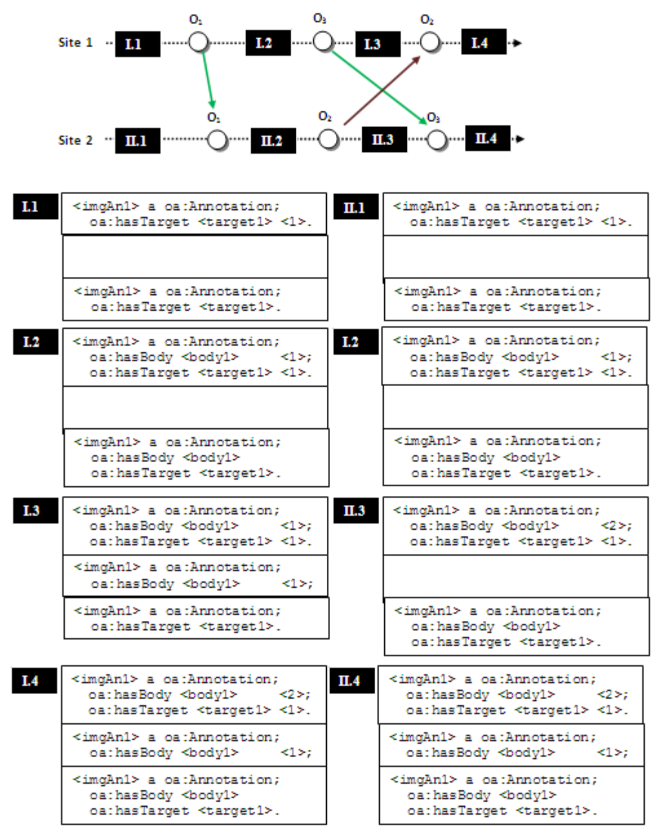

Figure 6. Convergence after integrating concurrent image annotations in CIA-Store

C. Specification

The optimistic replication philosophy used in CIA-Store aims to ensure that distributed sites can access annotation image store without priori synchronizations or order requirements. We present here the CIA-Store specification for supporting concurrent operations during image annotation process. To illustrate these, we describe the inserting and deleting functions designed using the final annotation store data structure. Figure 7 shows the specification of CIA-Store. CIA-Store specification has three main components that are payload, lookup and update. Payload corresponds to the data structure which holds the state of the different OAC stores. Lookup is a function which uses the payload in order return results about a given argument. Finely, update is another function that handles the insert and delete procedures. Initially, the payload is composed of set of triple S=(I,R, C) that has the same structure as the final annotation store fOAC-Store, I and R for including all removed and added annotation triples respectively, C containing a consistent result of annotation. The function lookup(e) checks if t has already been added in the insert OAC store iOAC-Store. Operation insert(t) listed in CIA-Store allows initially to test if the annotation triple already exists in the insert OAC store I. If it exists, its multiplicity is incremented, otherwise it is added to I with multiplicity one. At the end, the coherent OAC store C is computed automatically after each execution of local or remote operation. The operation remove(t) outlined has the same behavior as the previous operation except that the OAC store used is the remove OAC store R. It checks if there is a removed annotation triple t in D so that the multiplicity of t is incremented; otherwise, the element (t, 1) is inserted to D. After this, C is recomputed. The coherent OAC store C includes consistent annotation triples obtained after integrating concurrent modifications on each state at different sites.

payload fOAC-Store S=(I,R, C) --I:iAOC-Store; R:rAOC-Store; cAOC-Store
initial S=(∅,∅,∅)

query lookup (t) : boolean b
    let b =($\exists$ m|: (<t, m>) $\in$I)

update insert (t)
    if ($\exists$ m|: (t, m) $\in$I) **then**
        let m_1=m+1
        let I=I\{(t,m)} $\cup$ {(t,m_1)}
    else
        let m =1
        let I=I $\cup$ {(t,m)}
    if ($\exists$ m_I, m_R |: (t, m_I) $\in$I, (t, m_R) $\in$R, m_I> m_R) **then**
        C=C $\cup$ {t}
    else
        C=C\{t}

update delete(t)
    pre lookup(t)
    if ($\exists$ m|: (t, m) $\in$R) **then**
        let m_1=m+1
        let R=R\{(t,m)} $\cup$ {(t,m_1)}
    else
        let m =1
        let R=R $\cup$ {(t,m)}
    if ($\exists$ m_I, m_R |: (t, m_I) $\in$I, (t, m_R) $\in$R, m_I> m_R) **hen**
        C=C $\cup$ {t}
    else
C=C\{t}

Figure 7. Specification of CIA-Store

IV. DISCUSSION

Concerning the scalability, CIA-Store scales in terms of the number of replicas. The number of copies is not a factor in each component of CIA-Store structure. There is no total order on operations and no consensus process. Additionally to this, the CIA-Store philosophy is independent of the number of sites or replicas. The only requirement to maintain eventual consistency in CIA-Store is to perform the same set of annotation operations in all sites, the execution order of annotation operations within system is not important because all operations commute without any control since a delete annotation can be received before or after an insert. An important property of CIA-Store is its capability to integrate commutative replicated data type into OAC model for enabling participant annotations to form semantically structured content at different levels of complexity. This integration opens an interesting way to mix web semantic technologies advantages with the reconciliation and annotation approaches.

In addition, CIA-Store is an optimistic replication method that ensures CCI consistency for collaborative annotation of distributed OAC stores. To preserve consistency in the CIA-Store proposal, all concurrent annotation operations must commute. The generation of the consistent OAC store is done in two phases. In the first phase we increment the multiplicity of the given triple according to its situation in the insert or remove OAC store; then we update, in the second phase, the consistent OAC store following the obtained values by calculating the difference between the multiplicities of each triple in insert OAC store with its corresponding in remove OAC store. It is obvious that both realized operations in two phases correspond to simple addition and subtraction of the natural number. As addition and subtraction in N are commutative, all concurrent annotation operations in CIA-Store commute. Thus, CIA-Store preserves eventual consistency.

Unlike previous methods, causal propagation and reception are not required to ensure eventual consistency since the remove annotation can be executed before the start of the execution for insert annotation of the same triple. The use of the new concepts of insert and remove OAC stores defined as incremental counters where the effect of every annotation operation is observed in the main consistent OAC store by a multiplicity function associated with each annotation triple inserted or removed. Therefore, the intentions are preserved.

V. CONCLUSION

In this paper, we have presented the CIA-Store approach. CIA-Store is an optimistic replication approach that ensures CCI consistency model for collaborative image annotation based on OAC structures. One of the main powers of CIA-Store is the ability to use different order for the image annotation updating. This distinguishes CIA-Store from several other collaborative image annotation systems. CIA-Store combines both advantages of semantic web technologies and collaborative image annotation. In CIA-Store vision, updating annotations generated locally are re-executed on remote copies without any requirement of total order on annotations, thus, concurrent annotations can be re-played in different orders.

In the future, we plan to extend CIA-Store to manage more complex data. We are currently working on video and text annotation. We are also working on a group undo for the CIA-Store algorithm and implementing it in order to evaluate its overhead using real-world image annotation data.

REFERENCES

[1] C. Morbidoni, M. Grassi, M. Nucci, S. Fonda, and G. Ledda, "Introducing the Semlib Project: Semantic Web Tools for Digital Libraries". International workshop on semantic digital archives sustainable long-term curation perspectives of cultural heritage held as part of the 15th international conference on theory and practice of digital libraries, pp. 97-108, 2011.

[2] Open Annotation Community Group, http://www.w3.org/community/openannotation/ , accessed 02 March 2013

[3] Open Annotation Collaboration, http://www.openannotation.org/ , accessed 02 March 2013

[4] C. Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen, "Achieving Convergence, Causality Preservation, and Intention Preservation in Real-Time Cooperative Editing Systems", ACM Transactions on Computer-Human Interaction, vol. 5, no. 1, pp. 63-108, 1998.

[5] H. Zarzour, and M. Sellami, "B-Set: a synchronization method for distributed semantic stores", IEEE International Conference on Complex Systems, ICCS'12, Published on Xplore IEEE, Agadir, Maroc, November 5 - 6, 2012.

[6] N. M. Preguic J. M. Marques, M. Shapiro, and M. Letia, "A commutative replicated data type for cooperative editing", International Conference On Distributed Computing Systems, ICDCS, IEEE Computer Society, pp.395-403, 2009.

[7] F. Lopez, D. Nestor, and C. Oscar, "SMITag: A social network for semantic annotation of medical images", Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En. IEEE, 2012.

[8] M. Ameesh, V. Pavlovic, and S. Kumar, "A new baseline for image annotation", Computer Vision–ECCV 2008, Springer Berlin Heidelberg, pp.316-329, 2008.

[9] D. L. Rubin, C. Rodriguez, P. Shah, and C. Beaulieu, "ipad: Semantic annotation and markup of radiological images." AMIA Annual Symposium proceedings AMIA Symposium AMIA Symposium, pp. 626–30, 2008.

[10] C. Decurtins, M. C. Norrie, and B. Signer, 'Putting the gloss on paper: a frame-work for cross-media annotation', Hypermedia, vol.9, no.1, pp35-57, 2003.

[11] A. Chebotko al. "OntoELAN: an ontology-based linguistic multimedia annotator multimedia software engineering", In: Proceedings of the IEEE sixth international symposium on , pp. p. 329–36, 2004.

[12] P. Wittenburg et al. "ELAN: a professional framework for multimodality research", In: Proceedings of the 5th international conference on language resources and evaluation, LREC 2006, pp. 1556–9., 2006

[13] M. Kipp, "Anvil—a generic annotation tool for multimodal dialogue", In: Proceedings of the 7th European conference on speech communication and technology (Eurospeech), Aalborg, pp. 1367–70, 2001.

[14] B. Haslhofer, E. Momeni, M. Gay, and R. Simon, "Augmenting Europeana content with linked data resources", In Proceedings of the 6th international conference on semantic systems (I-Semantics), 2010.

[15] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, " DBpedia a crystallization point for the web of data", Web Semantics: Science, Services and Agents on the World Wide Web, Vol.7, no.3, pp. 154–165, 2009

[16] Video Image Annotiation Informatics and Telematics Insitute (CERTHITI), http://mklab.iti.gr/via/, accessed 18 April 2013

[17] B.C Russell, A. Torralba, K.P Murphy, W.T Freeman, "Labelme: a database and web-based tool for image annotation", Int J Comput Vis Vol.77, no.1–3, pp:157–173, 2008.

[18] F. Wang, C. Rabsch, and P. Liu, "Native web browser enabled SVG-based collaborative multimedia annotation for medical images", In : Data Engineering, ICDE 2008. IEEE 24th International Conference on. IEEE, p. 1219-1228, 2008.

[19] D. H. Im, and G. D. Park, "STAG: Semantic Image Annotation Using Relationships between Tags", Information Science and Applications (ICISA), 2013 International Conference on. IEEE, 2013.

[20] M. Grassi, C. Morbidoni, and M. Nucci, "A collaborative video annotation system based on semantic web technologies", Cogn Comput. Vol. 4, pp.497–514, 2012.

[21] R. Di Salvo, D. Giordano, and I. Kavasidis, "A crowdsourcing approach to support video annotation", In Proceedings of the International Workshop on Video and Image Ground Truth in Computer Vision Applications. ACM, 2013.

[22] B. Quilitz, and U. Leser, "Querying Distributed RDF Data Sources with SPARQL', Proc. European Semantic Web Conference on The Semantic Web: Research and Applications", pp. 524-538, 2008

[23] T. Berners-Lee, and D. Connolly, "Delta: an ontology for the distribution of differences between RDF graphs", http://www.w3.org/DesignIssues/Diff, 2004.

[24] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, "Edutella: a p2p networking infrastructure based on rdf", In 11th international conference on World Wide Web, pp. 604-615, 2002.

[25] G. Tummarello, C. Morbidoni, J. Petersson, P. Puliti, and F. Piazza, "Rdfgrowth, a p2p annotation exchange algorithm for scalable semantic web applications", In Proceedings of the MobiQuitous'04 Workshop on Peer-to-Peer Knowledge Management, 2004.

[26] M. Cai and M. R. Frank, "Rdfpeers: a scalable distributed rdf repository based on a structured peer-to-peer network", In WWW, pp. 650-657, 2004.

[27] M. Cai, M. R. Frank, J. Chen, and P. A. Szekely, "Maan: A multiattribute addressable network for grid information services", Journal of Grid Computing, vol. 2, no. 1, pp. 3-14, 2004.

[28] H. Zarzour, and M. Sellami, "p2pCoSU: A P2P Sparql/update for collaborative authoring of triple-stores", IEEE 11th International Symposium on Programming and Systems, ISPS'13, Published on Xplore IEEE, Algiers, Algeria, April 22-24, 2013.

[29] S. Weiss, P. Urso, and P. Molli, "Logoot-Undo: Distributed Collaborative Editing System on P2P Networks", IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 8, pp.1162-1174, 2010.

[30] M. Shapiro, et al. "Conflict-free replicated data types", Stabilization, Safety, and Security of Distributed Systems, Springer Berlin Heidelberg, 2011, pp.386-400.

[31] S. Martin, P. Urso, and S. Weiss, "Scalable XML Collaborative Editing with Undo", Proc. International Conference on Cooperative Information System, CoopIS, 2010.

[32] H. Zarzour, T. Abid, and M. Sellami, "Conflict-free collaborative decision-making over Mind-Mapping", 4th International Conference on Advanced Computing & Communication Technologies, ACCT'14, Rohtak, India, February 8-9, 2014.

[33] K. Aslan, H. Skaf-Molli, P. Molli, and S. Weiss, "C-set : a commutative replicated data type for semantic stores". In RED: Fourth International Workshop on Resource Discovery, At the 8th Extended Semantic Web Conference (ESWC), pp. 123-130, 2011.

[34] M. Shapiro, N. Preguica, C. Baquero, and M. Zawirski, "A comprehensive study of Convergent and Commutative Replicated Data Types", Research Report RR-7506, INRIA, January 2011.

[35] L. D. Ibáñez, H. Skaf-Molli, P. Molli, O. Corby, "Synchronizing semantic stores with commutative replicated data types", In Proceedings of the 21st international conference companion on World Wide Web, pp. 1091-1096, 2012.