

Using commutative replicated data type for collaborative video annotation

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 video annotation system is groupware system which enables a virtual community of participants to share and annotate the same digital video file from geographically dispersed nodes interconnected via the network. The video annotation process allows participants to browse videos, add, delete or update annotations. However, the existing systems are mainly concerned with the indexing, annotating, storing and sharing of video data. Hence, they only provide a basis for implementing simple way for video annotation and do not treat concurrent annotation aspects during the collaborative work. In this paper, we describe AV-Store an original method that combines both advantages of collaborative video annotation and semantic web technologies, and complies with eventual consistency condition when concurrent annotations are performed. The main idea of this approach is to define a new data type where all concurrent annotations commute. The commutativity aims at assuring the consistency of all replicas if participants perform the same sequence of annotations in different orders.

Keywords—Collaborative Video Annotation; Open Annotation Collaboration; Concurrent Annotation; CRDT; Semantic Web.

I. INTRODUCTION

Collaborative video annotation system (CVAS) is groupware system which enables a virtual community of participants to share and annotate the same digital video file from geographically dispersed nodes interconnected via the network. The video annotation process allows participants to browse videos, add, delete or update annotations, selecting concepts from a sequence of video data.

In collaborative environment, annotating a video is very important task and it is triggered by inserting subjective explanation, notes, comments and external remarks which can be related to a part of video. This does not require actually modifying the video file. When participants at remote sites retrieve a video, they can also retrieve the attached annotations from a site who initiates it in order to share their contributions on the particular video or to insert, delete or update their own annotations. CVAS is of great interest to the educational, defense, scientific, cultural, medical and media organizations [1]. However, Existing CVASs do not provide a flexible support for the annotations representation and exchange over these organizations.

Nowadays, new techniques have being developed based on Semantic Web technologies in order to improve the interoperability and provide annotations with clear data structures. Indeed, CVAS related to semantic web technologies have attracted a growing research interest [2]. The purpose of the Open Annotation Community Group [3] is to work towards a common, RDF-based, specification for annotating digital resources. As a result, the OAC (Open Annotation Collaboration) [4] 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 CVAS that integrates the OAC concepts.

A CVAS based on OAC is considered as correct and sound if it preserves the CCI model [5, 6] 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.

Recently, CRDT (Commutative replicated data type CRDT) [7, 8, 9], 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 describe a new specific data type based on OAC for collaborative video annotation called AV-Store that can be applied for a video annotation store and we prove that this type ensures CCI model criteria. With the existing solution, the concurrent aspect in annotation process does not treated since the insert and delete annotations do not commute in OAC data model. AV-Store is designed as CRDT to handle OAC representation and the concurrent video annotations.

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. Existing Frameworks for Collaborative Video Annotation

There are several collaborative video annotation frameworks, such as [10], [11], [12], [13] and [14]. However, they are mainly concerned with the indexing, annotating, storing and sharing of video data [15]. Hence, they only provide a basis for implementing simple way for video annotation and do not treat concurrent annotation aspects during the collaborative work.

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. In the same way, SVAT [17] allows participant to enable film analysts to efficiently annotate video footage. It also integrates a plug-in for object recognition and search tool based on Difference-of-Gaussian local key points.

In [18], authors present a web-based media annotation suite that enables to extend existing bibliographic information about digital contents like videos and images. This suite offers a client interface for annotating videos. It also offers Semantic Web capabilities enabling users to augment existing videos with related resources on the Web like resource derived from DBpedia[19]. Users can select video segments adding descriptors and inserting textual annotation.

YouTube [20] is the most well-known video annotation website which provides the possibility to upload videos, to add textual annotations in the form of notes, also highlighting part of the screen and to make these annotations visible to all YouTube visitors when the video is invoked. VideoAnt [21] is a Web-based application that is also based on YouTube videos as source, which enables to insert annotations in video timeline and to associate textual description. The obtained annotations can be shared by e-mail to be used also by other participants.

Hyper-Hitchcock [22] devolves a concept called detail-on-demand, which is a specialized version of a hypervideo. A major goal is to reduce the negative effects of hypermedia on users like disorientation. It restricts the amount of information by only allowing one outgoing link of a video at a time. Project Pad [23] is a project to build a web-based system for media annotation and collaboration for teaching, learning and scholarly applications. ToolClips [24] implements and validates the concept that, compared to text, videos support the learning processes better and that they help users to understand program functionalities much easier.

Recently, Semantic Web-based annotation system [25] 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 [26] is another web-based platform for creating cooperatively object detection, tracking and recognition ground truth for lengthy videos which also integrates crowd sourcing methods for annotation integration.

B. Consistency in Web Semantic Technologies

RDFGrowth [27] 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 [28] is one of the first efforts for structured peer-to-peer RDF stores. The key idea is to use a MAAN overlay [29] 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.

RDFSyn [30] 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.

Edutella [31] presents peer-to-peer 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.

C. CRDT-Based Systems

CRDT [7] 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 (text document) [32], tree document structure data type [33], semi-structured data type [34] and Mind-Mapping Structure [35].

Recently, many CRDT are proposed to support collaborative editing of semantic stores having set structure. C-Set [36] 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 [37] 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 principle 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 [38] presents a CRDT for RDF graphs based on OR-Set that supports the SPARQL 1.1 Update 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. AV-STORE APPROACH

AV-Store is a collaborative annotation approach that considers a set of distributed sites, where each site owns a replica of the shared video and a local copy of the video annotation store that contains all performed annotations about the shared video. The shared video is a sequence of frames which are displayed in order and represent scenes in motion. It is assumed here that the video annotation store can only be updated by insert and delete operations. Each site can concurrently modify its replica of the video annotation store by annotating the shared video. Its modifications are executed immediately against the local replica, then podcasted to other sites through the network. When a remote site receives an annotation operation, it is integrated to their local replica in the same way as a local update. Since the shared video annotation store is replicated, each site owns a local state of its copy which is updated only by actions performed locally or remotely. The initial state of the shared video annotation store is identical for all sites. Annotations video data can be stored in RDF repositories and the video can be uploaded or imported from any host by copying its URL.

A. A data model for AV-Store

The data model of AV-Store is based on the CRDT [7] framework for collaborative video annotating. The main idea of this approach is to define a new data type where all concurrent annotations commute. The commutativity aims at assuring the consistency of all replicas if participants perform the same sequence of annotations in different orders. In other words, a binary annotation is commutative if changing the order of the annotation execution does not change the result.

The following definitions detail the data structure of our solution:

Definition 1: An annotation A is an RDF triple consisting of a subject, a predicate and an object.

Definition 2: An arithmetic progression is a sequence of numbers such that the difference between the consecutive terms is constant.

We denote by d the common difference. For instance, the sequence 5, 7, 9, 11, 13, 15 ... is an arithmetic progression with common difference of 2.

Definition 3: An incremental arithmetic progression, denoted by X , is an arithmetic progression where the common difference d equals to 1. The n -th term of an incremental arithmetic progression is obtained by: $X_n = X_{n-1} + 1$.

Definition 4: An Insert incremental arithmetic progression, denoted by A , is the incremental arithmetic progression that represents the number of insert operations performed on the annotation triple, it defined as: $A_n = A_{n-1} + 1$.

Definition 5: A delete incremental arithmetic progression, denoted by B , is the incremental arithmetic progression that represents the number of delete operations performed on the annotation triple, it defined as: $B_m = B_{m-1} + 1$.

A delete and insert incremental arithmetic progression are sequences in which each term is obtained by the addition of a constant number to the preceding term, a constant number equals 1 as 1, 2, 3, 4, 5, 6, 7, 8.

Definition 6: Given an insert incremental arithmetic progressions A_n , a delete incremental arithmetic progressions B_m at the n -th and m -th terms respectively. The logical difference, denoted by D , is a boolean that can be *True* if $A_n > B_m$ or false elsewhere.

The logical difference is a boolean variable that represents if the annotation is visible or not. Annotations are never really removed they are just masked.

Definition 7: A consistent association, denoted by C , is a triple $\langle A_n, B_m, D \rangle$, where A_n is an Insert incremental arithmetic progression, B_m is a delete incremental arithmetic progression and D is a logical difference.

The consistent association is a relationship between the three elements that allows to maintain the consistency when concurrent annotation are performed.

Definition 8: A video annotation store, denoted by V , is a repository used for storing annotations. It is a pair (T, C) , where T is a set of annotation triples and C is a consistent annotation.

The video annotation store V is a repository that includes all annotations done by distributed users, each one from his site. V describes, in detail, the properties of each triple added or deleted that serve as a support for a flexible collaborative annotation mechanism taking into account not only a collaborative aspect but a concurrent annotation aspect on network. The repository V can be detailed as follows : $V = (T, C) = (\langle S, P, O \rangle, \langle A_n, B_m, D \rangle)$, where S , A and O are subject ,

predicate and object respectively, and $\langle A_n, B_m, D \rangle$ is a consistent association. As OAC conveys in the Turtle RDF format adopted specifications, we use this format for expressing annotations data in RDF data model. Therefore, the video annotation store can be formalized in Turtle RDF format as:

$\langle S \rangle a oa:Annotation ;$
 $oa:P \langle O \rangle \langle A_n, B_m, D \rangle .$

An annotation statement represents information using triples, each of which consists of a subject S , a predicate P , and an object S coupled with a consistent association $\langle A_n, B_m, D \rangle$.

$(\langle anno1, hasBody, body1 \rangle \langle A_{n1}, B_{m1}, D_1 \rangle)$
 $(\langle anno1, hasBody, body2 \rangle \langle A_{n2}, B_{m2}, D_2 \rangle)$
 $(\langle anno1, hasBody, target1 \rangle \langle A_{n3}, B_{m3}, D_3 \rangle)$
 $(\langle anno1, hasBody, target2 \rangle \langle A_{n4}, B_{m4}, D_4 \rangle)$

↓

$\langle anno1 \rangle a oa:Annotation ;$
 $oa:hasBody \langle body1 \rangle \langle A_{n1}, B_{m1}, D_1 \rangle ;$
 $oa:hasBody \langle body2 \rangle \langle A_{n2}, B_{m2}, D_2 \rangle ;$
 $oa:hasTarget \langle target1 \rangle \langle A_{n3}, B_{m3}, D_3 \rangle ;$
 $oa:hasTarget \langle target2 \rangle \langle A_{n4}, B_{m4}, D_4 \rangle .$

Figure 1. Passage from basic annotation for RDF representation to Turtle RDF format after integrating AV-Store model

Figure 2 shows an example of video annotation using the Turtle RDF format based on OAC standardizations about the open source e-learning platform called Moodle. Figure 3 shows the same example presented in figure 2 but according to our data model. In this case, it is mentioned how to compute the visibility of each annotation, about the platform Moodle, from an insert and delete incremental arithmetic progressions. All possible cases are presented in this sample, only the first, second, third and last annotation appear to end user because they have an insert incremental arithmetic progressions greater than a delete incremental arithmetic progressions, thus the visible video annotation store contains $\langle Moodle, hasBody, 'http://fr.slideshare.net/ialja/moodle-presentation-version-20-1203736' \rangle$, $\langle Moodle, hasBody, 'http://en.wikipedia.org/wiki/Moodle' \rangle$, $\langle Moodle, hasTarget, 'https://www.youtube.com/watch?v=WvCfv5KCbeE' \rangle$ and $\langle Moodle, hasTarget, 'https://www.youtube.com/watch?v=9P9kGWkT00U' \rangle$.

$\langle Moodle \rangle a oa:Annotation ;$
 $oa:hasBody 'http://fr.slideshare.net/ialja/moodle-presentation-version-20-1203736' ;$
 $oa:hasBody 'http://en.wikipedia.org/wiki/Moodle' ;$
 $oa:hasTarget 'https://www.youtube.com/watch?v=WvCfv5KCbeE' ;$
 $oa:hasTarget 'https://www.youtube.com/watch?v=W4uipnBkj-4' ;$
 $oa:hasTarget 'https://www.youtube.com/watch?v=9P9kGWkT00U' .$

Figure 2. Annotation example using OAC specifications

$\langle Moodle \rangle a oa:Annotation ;$
 $oa:hasBody 'http://fr.slideshare.net/ialja/moodle-presentation-version-20-1203736' \langle 1, 0, True \rangle ;$
 $oa:hasBody 'http://en.wikipedia.org/wiki/Moodle' \langle 3, 2, True \rangle ;$
 $oa:hasTarget 'https://www.youtube.com/watch?v=WvCfv5KCbeE' \langle 1, 0, True \rangle ;$
 $oa:hasTarget 'https://www.youtube.com/watch?v=W4uipnBkj-4' \langle 2, 3, False \rangle ;$
 $oa:hasTarget 'https://www.youtube.com/watch?v=9P9kGWkT00U' \langle 5, 3, True \rangle .$

Figure 3. Annotation example according to AV-Store data model

B. Annotation Operations

Video annotation is a set of information pieces that participants attach to resources by inserting some information that is not explicit in the video itself. In fact, the annotation process consists of adding, deleting, and updating contents of the video annotation store. To motivate participant not only to watch video but also to collaborate in generating new data, the annotation mechanism should be simple and support basic operations. Therefore, to annotate any data video type, the following both operations are required: 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 two primitive operations used in the annotation process are: $insAn(A)$ and $delAn(A)$, where $insAn(A)$ is used to insert an annotation A in the video annotation store, $delAn(A)$ is used to remove the annotation A from the video annotation store. An update action consists of two operations, including an annotation to be removed and an annotation to be inserted. In other words, the execution of an updating action, changing an annotation A_1 to a new value of A_2 , consists of the sequential execution of $delAn(A_1)$ followed by $insAn(A_2)$. After each execution of local add or remove operation of the annotation A , the value of logical difference is computed, and the corresponding remote operation is broadcast to all other sites in order to be re-executed. More formally, we define the main operation used for modifying and annotating a given video as follows:

Adding an annotation: $insAn(A)$ performs the union between a video annotation store V and an annotation A to be created or added inline by the participant.

Removing an annotation: $delAn(A)$ performs the set difference between video annotation store V and an annotation A to be removed from V inline by the participant.

Updating an annotation: $insAn(A_1)$ followed by $delAn(A_2)$ perform, firstly, the set difference between video annotation store V and an annotation A_1 to be removed from V followed by the union between V and an annotation A_2 to be inserted inline by the participant.

The Concurrency video annotation issues are described in Figure 4, where an existing video annotation store without our solution diverges when concurrent operations are performed on the same annotation since the $insert(T)$ and $remove(T)$ do not commute. Consider two sites sharing the same video in order to annotate it and starting from the same video annotation store that is not initially empty. The first site deletes a certain annotation triple $\langle anno1, hasTarget2, target2 \rangle$ by executing $Op_1 = delAn(\langle anno1, hasTarget2, target2 \rangle)$, at the same time a second site performs $Op_2 = delAn(\langle anno1, hasTarget2, target2 \rangle)$ in order to insert the same annotation triple but without deleting it back. After the mutual propagation and re-execution of generated operations, the replicated video annotation stores diverge, this means that the eventual consistency is violated when the same operations are performed in different order (see fig. 4). Let us consider again the scenario of the figure 4 but with our solution (see fig. 5), after mutual generation and execution of sequence of operations on both replicated video annotation stores, an identical result is obtained in both sites since the delete and insert annotations commute.

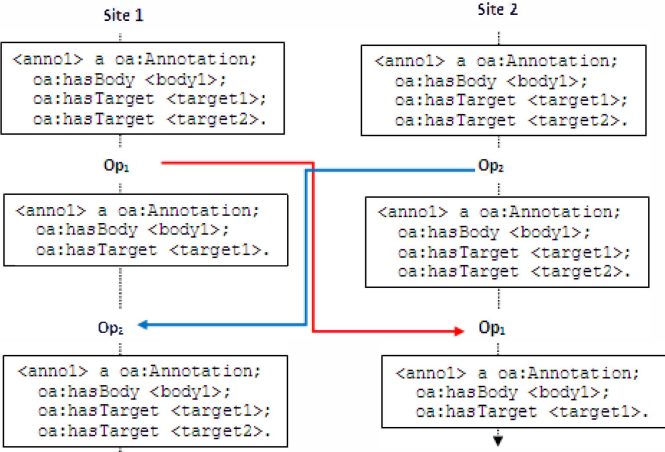


Figure 4. Concurrency video annotation issues, divergence after exciting concurrent operations

C. Specifications

We now present clean specifications of our model designed for collaborative video annotation in distributed environment. To ensure that all concurrent users' annotations commute, specifications are defined in all possible combination cases. In fact, they describe the execution comportment of any local or remote annotation operation that can be invoked by users in different order. Figure 6 specifies our solution. The payload is composed of set of pair (T, C), T for including all annotation triples added or removed and C for preserving a consistent annotation criteria. Operation lookup(e) checks that t has been added and exists in the couple (T, C).

To add or remove an annotation, two counters are used to implement a video annotation store. The first one corresponds to the insert incremental arithmetic progression whilst the second corresponds to the delete incremental arithmetic progression. Both counters are associated to every annotation, the values of these counters represent the number of delete and

insert occurrences of the triple in the repository of video annotation store. During the insert operation, the counter of the annotation triple to be inserted is incremented and the logical difference is computed if it is already in the video annotation store. Elsewhere, a new triple is created $\langle t; 1; 0; True \rangle$, where 1 and 0 are initial values of insert and delete incremental arithmetic progressions functions respectively. The default visibility value is true and the annotation triple will appear to the user. In the case of the remove operation, if the annotation triple to be deleted was already inserted in the video annotation store, the counter that represents the delete incremental arithmetic progression is incremented, then the visibility value is computed. When a user executes a remove operation of an annotation triple that does not exist or has not been inserted, the corresponding counter will be equal to one. Thus, this annotation triple will be inserted in the video annotation store but it is hidden.

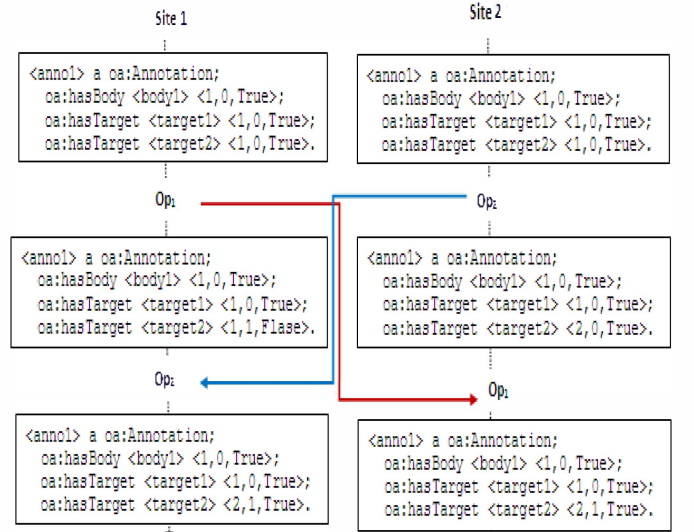


Figure 5. Solution of Concurrency video annotation issues in AV-Store approach, convergence after executing concurrent operations

IV. DISCUSSION

AV-Store is the first CRDT data type designed for supporting the scalable collaborative video annotation in distributed architecture with different levels of complexity and abstraction. Unlike previous approaches, AV-Store combines both advantages of semantic web technology and collaborative video annotation complies with the CCI consistency model criteria. The causality assumes that all pairs of operations are ordered by a precedence relation, this involves executing operations in the same order on every replica. To guarantee causality, there are many scalable causal broadcasts such as [20]. A probabilistic causal broadcast [36] integrated with causal barriers [37], whose size is lower than vector clocks. However, an interesting property of AV-Store is that the causality preservation can be removed since the causal delivery or receive does not required to ensure eventual consistency. Another interesting property of AV-Store is its ability to perform a garbage collection by removing locally masked elements on one site without interaction with others sites when

logical differences associated to sequence annotation triples are equal to false.

```

payload set  $S = (T, C) = (\langle S, P, O \rangle, \langle A_n, B_m, D \rangle)$ 
initial  $S = (\emptyset, \emptyset)$ 

query lookup  $(t) : \text{boolean } b$ 
    let  $b = (\forall t, \exists A_n, B_m, D : (\langle t, , \langle A_n, B_m, D \rangle) \in S)$ 

update add  $(t)$ 
    if  $(\exists A_n, B_m, D : (\langle t, , \langle A_n, B_m, D \rangle) \in S)$ 
    then
        let  $A_{n+1} = A_n + 1$ 
        if  $A_{n+1} > B_m$  then
            let  $D_1 = \text{True}$ 
        else
            let  $D_1 = \text{False}$ 
         $S = S - \{(t, \langle A_n, B_m, D \rangle)\} \cup \{(t, \langle A_{n+1}, B_m, D_1 \rangle)\}$ 
    else
        let  $A_n = 1$ 
        let  $B_m = 0$ 
        let  $D = \text{True}$ 
         $S = S \cup \{(t, \langle A_n, B_m, D \rangle)\}$ 

update remove  $(t)$ 
    pre lookup  $(t)$ 
    if  $(\exists A_n, B_m, D : (\langle t, , \langle A_n, B_m, D \rangle) \in S)$ 
    then
        let  $B_{m+1} = B_m + 1$ 
        if  $A_n > B_{m+1}$  then
            let  $D_2 = \text{True}$ 
        else
            let  $D_2 = \text{False}$ 
         $S = S - \{(t, \langle A_n, B_m, D \rangle)\} \cup \{(t, \langle A_n, B_{m+1}, D_2 \rangle)\}$ 
    else
        let  $A_n = 0$ 
        let  $B_m = 1$ 
        let  $D = \text{False}$ 
         $S = S \cup \{(t, \langle A_n, B_m, D \rangle)\}$ 

```

Figure 6. AV-Store Specifications

To ensure that AV-Store preserves consistency, we must demonstrate that all pairs of concurrent annotation operations commute. On each site AV-Store generates a set of add and remove operations, where a new value of logical difference is calculated in terms of insert and delete incremental arithmetic progressions. As illustrated in the above specifications, the calculation procedure of the logical difference depends on an incrementation of counters followed by subtraction between obtained variables values to decide appropriate boolean value. Since addition and subtraction are two commutative operations in \mathbb{Z} , so all operations couples of AV-Store commute.

According to [6, 39], AV-Store ensures eventual consistency. The proof of the intention preservation is straightforward. The reason is that the effect of each generated annotation operation about any video is preserved in local or remote site by introducing of the consistence association concept related to each added or removed annotation triple. Thus, this association allows to maintain the intention of the user who initiates the operation.

V. CONCLUSION

This paper has presented the details of the AV-Store, its data model, operations and the specifications that constitute it. AV-Store is an original CRDT data type designed for supporting the scalable collaborative video annotation in distributed architecture with different levels of complexity and abstraction. It combines both advantages of collaborative video annotation and semantic web technologies, and complies with CCI data model criteria. Unlike previous approaches, AV-Store is more efficient and treats concurrent annotation aspects during the collaborative session.

Future works will include the test of AV-Store for the specific domain of news multimedia. From these results, we plan to enhance our simulations with more authentic models of clients' annotating behavior and another document types.

REFERENCES

- [1] I. Kavasidis, S. Palazzo, R. Di Salvo, D. Giordano, and C. Spampinato, "An innovative web-based collaborative platform for video annotation", *Multimedia Tools and Applications*, pp. 1-20, 2013.
- [2] 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.
- [3] Open Annotation Community Group, <http://www.w3.org/community/openannotation/>, accessed 02 March 2013
- [4] Open Annotation Collaboration, <http://www.openannotation.org/>, accessed 02 March 2013
- [5] 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.
- [6] 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.
- [7] N. M. Prego, 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.
- [8] H. Zarzour, and M. Sellami, "srCE: a collaborative editing of scalable semantic stores on P2P networks", *International Journal of Computer Applications in Technology*, vol. 48, no.1, pp. 1-13, 2013.
- [9] 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.
- [10] 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. 1367-1370, 2001.

- [11] T. Schmidt, "Transcribing and annotating spoken language with EXMARaLDA", In Proceedings of LREC 2004 Workshop on XML-based Richly Annotated Corpora, 2004.
- [12] J. Yi, Y. Peng, and J. Xiao, "Exploiting semantic and visual context for effective video annotation", IEEE Transactions on Multimedia, IEEE Computer Society, Vol.15, no.6, pp. 1-1, 2013
- [13] P. Wittenburg et al, "ELAN: a professional framework for multimodality research", In Proceedings of the 5th international conference on language resources and evaluation, LREC. pp. 1556-1559, 2006.
- [14] A. Chebotko A et al, "OntoELAN: an ontology-based linguistic multimedia annotator multimedia software engineering", In Proceedings of the IEEE 6th international symposium on, IEEE Computer Society, pp. 329-36, 2004..
- [15] M. Bertini, A. Del Bimbo, A. Ferracani, and D. Pezzatini, D, "A social network for video annotation and discovery based on semantic profiling". In Proceedings of the 21st international conference companion on World Wide Web, pp. 317-320, 2012.
- [16] Video Image Annotation Informatics and Telematics Institute (CERTH-ITI), <http://mklab.iti.gr/via/>, accessed 18 April 2013
- [17] SVAT Semantic Video Annotation Tool by JOANNEUM RESEARCH, <http://www.joanneum.at/en/fb2.html>, accessed 18 April 2013
- [18] 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.
- [19] 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
- [20] YouTube Video Annotations, http://www.youtube.com/it/annotations_about, accessed 18 April 2013.
- [21] VideoANT, <http://ant.umn.edu/>, accessed 18 April 2013.
- [22] A. Girgensohn, F. Shipman, and L. Wilcox, "Hyper-Hitchcock: Towards the Easy Authoring of Interactive Video". Human-Computer Interaction INTERACT, IOS Press, pp. 33-40, 2003.
- [23] Project Pad, <http://dewey.at.northwestern.edu/ppad2/>, accessed 18 April 2013.
- [24] T. Grossman, and G. Fitzmaurice, "ToolClips: an investigation of contextual video assistance for functionality understanding", In Proceedings of the 28th international conference on Human factors in computing systems, ACM Press, pp. 1515-1524, 2010.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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
- [31] 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.
- [32] 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.
- [33] M. Shapiro, et al. "Conflict-free replicated data types", Stabilization, Safety, and Security of Distributed Systems, Springer Berlin Heidelberg, 2011, pp.386-400.
- [34] S. Martin, P. Urso, and S. Weiss, "Scalable XML Collaborative Editing with Undo", Proc. International Conference on Cooperative Information System, CoopIS, 2010.
- [35] 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.
- [36] 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.
- [37] 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.
- [38] L. D. Ibáñez, H. Skaf-Molli, P. Molli, and 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.
- [39] H. Zarzour, and M. Sellami, "Achieving consistency in collaborative image annotation systems", 5th International Conference on Information and Communication Systems, ICICS'14, Irbid, Jordan, April 1-3, 2014.