Hybrid Consistency Framework for Distributed Annotation Records

[1,2]Ahmet Fatih Mustacoglu
[1]Community Grids Lab
[2]Department of Computer Science,
Indiana University
Bloomington, IN 47404, USA
+1 812 856 0753
amustaco@cs.indiana.edu

[1,2]Geoffrey C. Fox
[1]Community Grids Lab
[2]Department of Computer Science,
Indiana University
Bloomington, IN 47404, USA
+1 812 856 7977
gcf@indiana.edu

ABSTRACT
We describe a novel hybrid consistency framework that maintains consistency among Distributed Annotation Records kept at various web-based annotation tools. There are issues in semantics of annotation tools. Each annotation tool stores different metadata, has different rules for tags, and does not provide timing information for the updated records. As a result of these, documents can be updated inconsistently with unknown precise time stamps and spread around in existing annotation tools with different versions. Moreover, their communications with other annotation tools are also very limited through various forms and this also contributes to inconsistencies among Distributed Annotation Records. To deal with these major shortcomings, this paper introduces the notion of "hybrid-consistency framework", which maintains the consistency among distributed annotation records held at various annotation tools. We discuss the overall design, architecture and the components of the hybrid consistency framework, and provide a working prototype implementation and a roadmap of the future work in this research.

Categories and Subject Descriptors
H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *Collection, Dissemination*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Web-based services*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *Distributed systems*; H.2.4 [**Database Management**]: Systems – *Relational databases.*

General Terms
Design, Management.

Keywords
Hybrid consistency framework, Web 2.0, Annotation Tools, Distributed Annotation Records, Events, Tagging.

1. INTRODUCTION
Web 2.0 can be defined as the second generation of Web applications and network-enabled stateless services [1, 2]. It is generally characterized as building on the set of key concepts: (1) REST (Representational State Transfer) Services; (2) rich user interfaces via JavaScript, AJAX, and JASON; (3) online communities and social networks; (4) widgets, gadgets, and badges. There are various types of online community tools aiming at fostering online collaboration and sharing between users and communities. The most popular examples of these tools include Blogs (blogger.com, Google Blog), Wikis (Wikipedia, WikiWikiWeb, Wikitravel), Social Networking Tools (MySpace, LinkedIn), Social Bookmarking Tools (del.icio.us, Flickr, YouTube), Syndication Feed Aggregators (Netvibes, YourLiveWire) and other related tools.

The term "Web 2.0" is now getting more and more popular, and it is representing this wave of new Web-based tools and the use of technologies. This change is also very obvious in the domain of scientific research, with the recent creation of a number of online tools that enable the annotation and sharing of scientific content, such as CiteULike [3], Connotea [4], and Bibsonomy [5]. One of the famous annotation website is del.icio.us [6] and is sometimes referred as Delicious. It is a Web-based tool and it enables users to annotate and share URLs. There are other numbers of annotation tools and they support annotation and sharing of a variety of resources, such as videos (YouTube), goals (43things), photos (Flickr), and books (LibraryThing). Especially there exist Web-based online tools focusing on the annotation of scholarly publications such as Connotea, CiteULike, and Bibsonomys. The fundamental service provided by these Web-based annotation tools is the capability that allows users or communities to easily annotate their favorite resources (videos, photos, URLs, or citations) by using the keywords called tags and to share their tagged content with other users.

While the numbers of annotation tools are increasing rapidly, each of them having their own structure, design, interface, format of their holding and very few examples exist of any of these being able to communicate in some form with other annotation tools. These tools and services store annotations and metadata in their system. Users of these tools and services can update or modify descriptive fields of their entries such as title, description, or tag, etc. Today various online collaboration tools, peer to peer systems and internet have generated multiple sources of information about

the same data. These multiple sources of information are all dynamic, and each of them has value but no one has total value. As a result of this, multiple copies of a same object can be in different places, and users of these systems suffer from having multiple copies of a same record in different versions due to different metadata storage in each annotation tool. In addition, annotation tools do not provide timing information for updated records, and this can also lead to inconsistency for Distributed Annotation Record (DAR) once DARs get updated. In order to cope with these shortcomings, there is a need for architecture or a framework to reconcile these dynamic possibly inconsistent sources of metadata about the same DAR located at different annotation tools in a consistent manner.

We propose a hybrid consistency framework to maintain consistency for each DAR held on several annotation tools. The ideal approach to reconcile different sources of annotation and metadata for DARs is to have an event-based model [7] to keep track of changes to documents and metadata while providing our proposed consistency framework around it. In our proposed solution, we keep primary copy of each DAR with extra metadata fields in our relational database, and we provide a hybrid consistency framework to maintain consistency between all DARs stored at various annotation tools and a primary copy of each DAR.

This paper discusses the hybrid consistency framework for the DARs maintained at different annotation tools and expounds its implementation and integration with Semantic Research Grid [8] system. The rest of this paper is organized as follows: Section 2 provides a discussion of the consistency criteria. Section 3 describes the design philosophy. Section 4 gives the details of our proposed hybrid consistency framework architecture. Section 5 explains the architecture components. Section 6 presents a prototype implementation of our proposed framework. Section 7 discusses the future work in this research.

2. CONSISTENCY CRITERIA

The consistency enforcement issue has to do with ensuring that all copies of the same data to be the same. Some approaches to maintain consistency are discussed in [9-14]. Tanenbaum [13] differentiates consistency under two main category: (1) data-centric; and (2) client-centric. In data-centric approach, all copies of data are updated whether some clients is aware of those updates or not. In client-centric approach, consistency is maintained from a client's perspective. Client-centric consistency model allows copies of data to be inconsistent with each other as long as the consistency is ensured from a single client's point of view.

The implementation of the consistency models can be categorized as primary-based protocols (primary-copy approach) and replicated-write protocols [13]. In primary copy approach, updates are executed on a single location, and propagated replicas from there, while in the replicated-write approach; updates can be originated from multiple locations. For an example, techniques for maintaining consistency in P2P networks: (1) Push: Owner-initiated Consistency. In this model, messages are propagated through the P2P overlay in push approach; (2) Pull: Peer-initiated Consistency mechanism. Individual peers polls the owner to figure out if a file is stale or not; and (3) Hybrid Consistency mechanism. Our approach enhances the popular consistency techniques, which had been originally designed for the distributed replicated systems, to be applied to DARs to maintain consistency among web-based annotation tools.

3. DESIGN PHILOSOPHY

Annotation tools are one of the major Web 2.0 applications. They basically provide their users with ability to: (1) enter a new record; (2) delete an existing record; (3) modify an existing record; (4) tag their record; (5) share the content of their records with other users. The consistency concept arises when records get updated with unknown time stamp. Providing consistency maintenance is a fundamental issue [9], and our research focuses on how to design a consistency framework to maintain consistency for each DAR held on those annotation tools. The design of such an environment should consist of group of annotation tools intended to be consistent with each other, and a main system, where a primary copy of each document from each annotation tools are stored with additional metadata information into a relational database.

There are issues in semantics of annotation tools such as each annotation tool stores different metadata in their system and their rules for tag are also different from each other. Table 3-1 portrays the stored metadata comparison in Connotea, Citeulike, and Delicious annotation tools. One major problem with annotation tools is that they do not provide precise timestamps for the updated records. As a result of this, data can be updated inconsistently with unknown precise time. Another fundamental issue is that annotation tools are lack of services or mechanisms to provide their clients with notification services for deleted, modified or entered new entries into their system. Hence there is no way to realize any changes in those systems unless modifications are done through their interfaces. The only way to identify any change in those tools is having a mechanism to go and check them periodically. We have designed our hybrid consistency framework to be able to: (1) run all the time for consistency enforcement; (2) communicate with integrated annotation tools periodically; and (3) collect the differences between each DAR kept in each annotation tool and the primary copy of each DAR stored in a relational database. Eventually, if there are any changes to DARs in annotation tools, we can retrieve the latest updates by pulling them out from these tools, and apply them to update the primary copy of each DAR. Furthermore, users can collaborate on a primary copy of each DAR with each other by sharing the same document. And our hybrid consistency framework propagates updates made on a primary copy of a DAR to each annotation tool to reflect the changes in a consistent manner. As a result, we have designed to have a two way mechanism to maintain consistency among integrated annotation tools and a primary copy of each DAR. We are going to give the details of our proposed consistency framework in Section 4.

4. HYBRID CONSISTENCY FRAMEWORK ARCHITECTURE

Our hybrid consistency framework has been designed to maintain consistency between DARs kept at annotation tools and a primary copy of each DAR. The hybrid consistency framework is a data centric consistency model, and it is based on the primary copy based consistency protocol approach. In our proposed framework, update propagations are carried out through pull and push based

approaches. Push approach enforces strict consistency model on primary copies of DARs. In strict consistency model; whenever updates occurred on a primary copy of a DAR, they are being propagated immediately to each annotation tool to update DARs on their site. However, pull approach is a time-based consistency control approach [15]. We are periodically checking DARs from each annotation tool for any updates. If there is any, then we are pulling them out. Finally, we are applying them onto the primary copy of each DAR, which is stored in a relational database with additional metadata. We have also developed a rollback mechanism to ensure consistency. It basically allows users to rollback to a previous state at any time. We are going to explain rollback mechanism in detail in Section 5.4.2. Figure 4-1 represents the overall architecture of our proposed Hybrid Consistency Framework. Explanation of the architecture components are given in section 5 in detail.

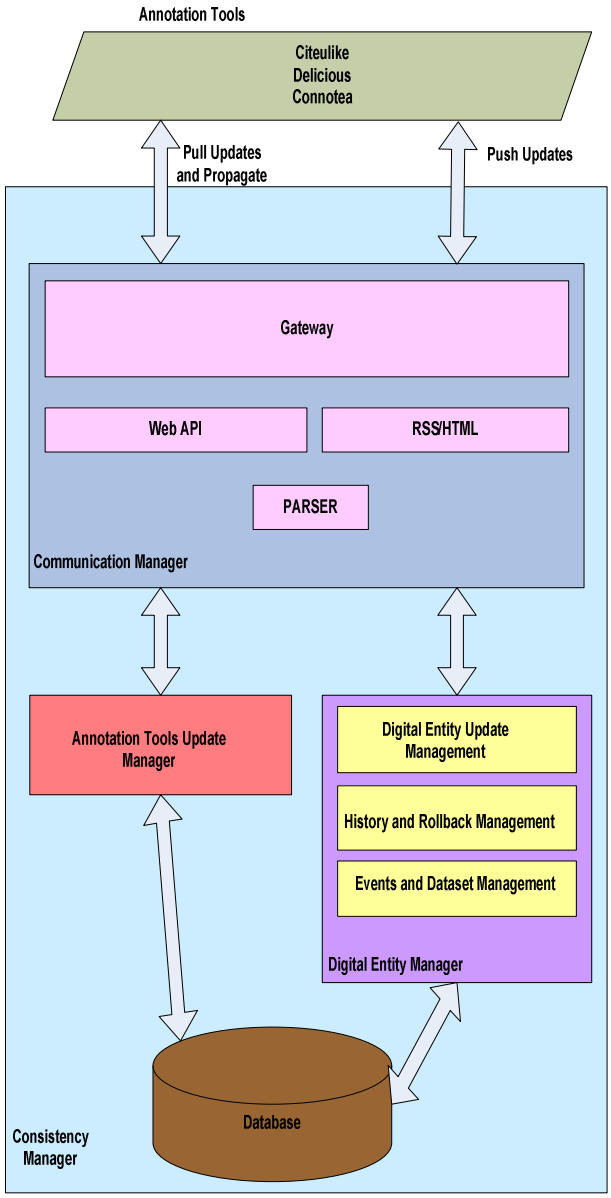

Figure 4-1: Hybrid Consistency Framework

Table 3-1: Stored Metadata Comparison in Annotation Tools

Stored Metadata	Citeulike	Connotea	Delicious
URL	✓	R	R
Title	R	✓	
DOI	✓	✓	
PMID		✓	
ISBN/ASIN		✓	
Reference Type	R	✓	
Authors	✓	✓	
Pub. Name		✓	
Volume No	✓	✓	
Issue No	✓	✓	
Chapter	✓		
Edition	✓		
Start& End Page	✓		
Pages		✓	
Year&Month&Day	✓		
Pub. Date		✓	
Date Other	✓		
Editors	✓		
Journal	✓		
Book Title	✓		
How Published	✓		
Institution	✓		
Organization	✓		
Publisher	✓		
Address	✓		
School	✓		
Series	✓		
Bibtex Key	✓		
Abstract	✓		
Display Title		✓	
Tags	†	R	✓
Tag Suggestions		✓	
Description		✓	R
My Work		✓	
Everyone's Tag	✓		
Privacy Settings	✓	✓	
Release date others		✓	
Priority of Records	✓		
Note	✓		✓
Comment		✓	

✓ = Supported, R = REQUIRED, † = Adds "no-tag"

5. OVERVIEW of the ARHITECTURE COMPONENTS

The detail explanation of the hybrid consistency framework architecture is given in the following sub sections respectively.

5.1 Annotation Tools

Annotation tools represent the integrated annotation tools into our proposed hybrid consistency framework. Our model, works around these Web 2.0 tools to reconcile DARs from each annotation tool in a consistent way. In the current implemented version, we have integrated Delicious, Citeulike, and Connotea into our prototype system called Semantic Research Grid (SRG) [8].

5.2 Communication Manager

Communication manager transports the data between the computing nodes. It is responsible for retrieving or posting data from/to annotation tools though their gateways. It retrieves updates from annotation tools via HTTPClient [16] native libraries by using: (1) Annotation tool's API and get the response in XML format. Updates are parsed by using DOM parser and XPATH [17]; or (2) HTTP GET, and POST method and get the response in RSS or HTML format. In RSS type responses, updates are parsed by using DOM parser and XPATH, and in HTML type responses, updates are parsed after cleaning faulty HTML by using JTidy [18] native libraries. Having retrieved and parsed updates, Communication Manager passes the organized data to Annotation Tools Update Manager explained in detail in 5.3. Updates are posted to annotation tools via: (1) annotation tools API; or (2) HTTP GET, POST methods through HTTPClient native library if an annotation tool does not provide an API. Its modules are explained in the following sections respectively.

5.2.1 Gateway

Gateway is an interface between hybrid consistency framework and an individual annotation tool. Our hybrid consistency model communicates with annotation tools through their gateways. The communications are carried out through HTTP methods by using HTTPClient native libraries [16]. An individual gateway is created for each interacting annotation tool, which has its own communication structures.

5.2.2 Parser

Parser is a native library used for parsing responses coming from annotation tools. There are several parsers to utilize in XML processing. DOM parser is the most widely used one. It reads and validates the XML documents. If the document is valid, then it returns a document object tree. We can randomly access any element since each element is entirely kept in memory. As a result, it provides a very efficient navigation mechanism over the parsed document. On the other hand, its drawback is that it requires large amount of memory in order to hold the whole parsed document. Most of the major annotation tools provide their Web API and responses are like in XML format. So users can communicate with their services easily. In our prototype implementation (described in Section 6), we have used JDOM [19] parser as our parsing library. In some annotation websites, they do not provide a Web API for their services, and then their responses in either in RSS or HTML format. In order to communicate with those annotation tools, we have used XPATH [17] to retrieve the desired element of the document and JTidy native library [18], which is used for cleaning faulty HTML and provide a DOM interface to the documents that is going to be parsed.

5.2.3 Web API

Web API (Application Programming Interface) is a service for accessing data on annotation tools. Most of the major annotation tools provide their Web API and RSS feeds for an easy access to their data. Their Web API and RSS feed return a document in XML format, which can be parsed easily by using a DOM parser in our prototype implementation, to the requester. Hence, data from annotation tools can be retrieved and modified via their Web API through HTTPClient tool by passing the necessary parameter to HTTPClient object.

5.3 Annotation Tools Update Manager

Annotation tools update manager is responsible for retrieving the updates from annotation tools periodically and applying the updates on the primary copy of each DAR. Its main duties are: (1) obtain the updates from annotation tools via Communication Manager; (2) applying each update on its primary copy stored in the relational database; (3) propagating the updates back to each annotation tools.

5.4 Digital Entity Manager

Digital Entity Manager is an umbrella name for a group of modules that contributes to a DAR management together. Its modules are: (1) Digital Entity Update Management; (2) History and Rollback Management; (3) Events and Dataset management. Details of each module are given in the following sections respectively.

5.4.1 Digital Entity Update Management

It deals with updates that are made directly on a primary copy of each DAR. Each update to a DAR consists of minor event(s) and dataset(s) [7]. Once an update made to a DAR, it becomes a minor event. Having dataset created from minor events, the changes are reflected in the database as events, which allow us to track the changes to a document. Furthermore, the updates are disseminated to annotation tools via the Communication Manager once they occurred.

5.4.2 History and Rollback Management

Using the mechanism described in [7], all the changes that have occurred to a DAR are stored in the user session as minor events [7]. They do not have any effect on the current value of the DAR unless minor events are used for creating a dataset. Once a dataset is created by using minor event(s), the dataset is applied to the DAR metadata during the latest DAR retrieval process.

To allow users to restore the state of the system to any previous state, we have implemented a module that allows users to view the history of each DAR and to undo any changes (rollback). In the history tool of the Digital Entity Manager, each DAR has an initial entry and a list of time-stamped datasets, which represents the changes made to the DAR if there is any. During the rollback

execution; first a user selects and applies a time-stamped dataset. Second, the selected state of the dataset compared with the latest metadata of the DAR. Finally, the DAR is rollbacked to the selected state by unrolling the related events from the current version of DAR. Further details can be obtained from [7].

5.4.3 Events and Dataset Management

In our framework, an event is defined as a time-stamped action on a DAR. Our hybrid consistency framework identifies the events: (1) Minor Events that encapsulates the changes to a DAR; (2) Major Events that are represent an entry of a new DAR into the system or deletion of an existing DAR. A dataset consists of collection of minor events. Further details can be found in [7].

5.5 Survey of Technologies

In our implementation of hybrid consistency framework, we have used various technologies. Summary of the technologies [16-21] are represented in Table 5-5-1.

Table 5-5-1: Technologies

API	Purpose
JDOM	For parsing XML documents
Jakarta Commons HTTP Client	For handling HTTP communication
XPATH	For querying an XML document object
JTidy	For parsing HTML documents
Apache Axis	For creating Java Web Services
JAVA	For implementing the framework

6. PROTOTYPE IMPLEMENTATION

We have applied our proposed Hybrid Consistency Framework to Semantic Research Grid (SRG) system described in detail [8]. Our proposed framework has been implemented using Web Service Technology, and services can be accessed via SOAP calls. The SRG system has been designed based on the Web 2.0 technologies and it consists of tools and services for supporting Cyberinfrastructure [22] based scientific research. The SRG system integrates a number of existing online research tools (social bookmarking, academic search, scientific databases, journal and conference content management systems) and aims to develop added-value community-building tools that leverage the semantic analysis of DARs. Running instance of the SRG system can be accessed from project demo website [23].

7. FUTURE WORK

In the current implementation, users can only track consistency updates in our system via the history tool. A desired future of the system would be a tool for logging the consistency updates. It will allow users to see automatically applied updates for consistency enforcement and their status as well. We intend to do this improvement by creating a database table for keeping consistency updates and retrieving the data whenever users request to access history of consistency updates.

Another desired future work will be conducting various scalability and performance tests of our proposed hybrid consistency framework.

8. CONCLUSION

In this paper we discussed Hybrid Consistency Framework for reconciling DARs stored at various Annotation Tools in Web 2.0 domain. We have also mentioned the implementation details of our proposed framework in SRG system. Furthermore, we described the current state of the development of the event-based hybrid consistency framework and outlined some directions for future work.

9. REFERENCES

[1] P. Graham, "Web 2.0," 2005. Available from http://www.paulgraham.com/web20.html.

[2] T. O'Reilly, "What is Web 2.0," 2005. Available from http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[3] CiteULike web site. http://www.citeulike.org

[4] Connotea web site. http://www.connotea.org

[5] Bibsonomy web site. http://www.bibsonomy.org

[6] Delicious web site. http://de.icio.us

[7] A. F. Mustacoglu, A. E. Topcu, A. Cami, and G. Fox, "A Novel Event-Based Consistency Model for Supporting Collaborative Cyberinfrastructure Based Scientific Research," in *Collaborative Technologies and Systems CTS 2007 in Technical Cooperation with The IEEE Computer Society*. Orlando, FL, USA, 2007.

[8] G. Fox, A. F. Mustacoglu, A. E. Topcu, and A. Cami, " SRG: A Digital Document-Enhanced Service Oriented Research Grid," presented at Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference, Las Vegas, NV, USA, 2007.

[9] S. Chengzheng and C. David, "Consistency maintenance in real-time collaborative graphics editing systems," *ACM Trans. Comput.-Hum. Interact.*, vol. 9, pp. 1-41, 2002.

[10] L. Jiang, L. Xiaotao, S. Prashant, and R. Krithi, "Consistency Maintenance In Peer-to-Peer File Sharing Networks," in *Proceedings of the The Third IEEE Workshop on Internet Applications*: IEEE Computer Society, 2003.

[11] R. Jonathan, F. Sarah, and V. Sankar, "Consistency management for distributed collaboration," *ACM Comput. Surv.*, vol. 31, pp. 13, 1999.

[12] V. Jurgen, V. JiRgen, G. Werner, C. Li-Te, and M. Michael, "Consistency Control for Synchronous and Asynchronous Collaboration Based on Shared Objects and Activities," *Comput. Supported Coop. Work*, vol. 13, pp. 573-602, 2004.

[13] A. S. Tanenbaum and M. V. Steen, *Distributed Ssytems Principles and Paradigms*, 2002.

[14] G. Werner, V. Jurgen, C. Li-Te, and M. Michael, "Supporting activity-centric collaboration through peer-to-peer shared objects," in *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*. Sanibel Island, Florida, USA: ACM Press, 2003.

[15] L. Rui, L. Du, and S. Chengzheng, "A Time Interval Based Consistency Control Algorithm for Interactive Groupware Applications," 2004.

[16] JAKARTA COMMONS HTTP CLIENT. Available from http://jakarta.apache.org/httpcomponents/httpclient-3.x/

[17] XML Path Language (XPATH). Available from http://www.w3.org/TR/xpath

[18] JTIDY. Available from http://jtidy.sourceforge.net

[19] JDOM. Available from http://www.jdom.org

[20] WebServices-Axis. Available from http://ws.apache.org/axis/

[21] JAVA Technology. Available from http://www.java.sun.com

[22] D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. Garcia-Molina, M. L. Klein, D. G. Messerschmitt, P. Messina, J. P. Ostriker, and M. H. Wright, "Revolutionizing Science and Engineering Through Cyberinfrastructure. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure," 2003.

[23] Semantic Research Grid (SRG) Project Website. Available from http://gf6.ucs.indiana.edu:48080/SRGrid