

Operational Transformation In Co-Operative Editing

Mandeep Kaur, Manpreet Singh, Harneet Kaur, Simran Kaur

ABSTRACT: Cooperative Editing Systems in real-time allows a virtual team to view and edit a shared document at the same time. The document shared must be synchronized in order to ensure consistency for all the participants. This paper describes the Operational Transformation, the evolution of its techniques, its various applications, major issues, and achievements. In addition, this paper will present working of a platform where two users can edit a code (programming file) at the same time.

Index Terms: CodeMirror, code sharing, consistency maintenance, Google Wave, Google Docs, group editors, operational transformation, real-time editing, Web Sockets

1 INTRODUCTION

Collaborative editing enables portable professionals to share and edit same documents at the same time. Elementally, it is based on operational transformation, that is, it regulates the order of operations according to the transformed execution order. The idea is to transform received update operation before its execution on a replica of the object. Operational Transformation consists of a centralized/decentralized integration procedure and a transformation function. Collaborative writing can be defined as making changes to one document by a number of persons. The writing can be organized into sub-tasks assigned to each group member, with the latter part of the tasks done before or after the previous part, or they might work together on each task. The write-up is planned, written, and revised, where a number of persons can be involved in more than one steps. It allows multiple users with editing privileges to edit the shared document simultaneously once user/users are authenticated for required access.

1.1 MAIN ISSUES RELATED TO REAL TIME CO-OPERATIVE EDITING

Three inconsistency problems

- Divergence: Contents at all sites in the final document are different.
- Causality-violation: Execution order is different from the cause-effect order.

E.g., $O1 \rightarrow O3$

- Intention-violation: The actual effect is different from the intended effect.

E.g., $O1 || O2$

One of the solution to overcome all the above three issues was invent of Operational Transformation.

2 OPERATIONAL TRANSFORMATION

A technique called Operational Transformation has been first introduced by Ellis and Gibbs in 1989 [1]. Operational transformation originated from research in the fields of computer-supported collaborative work (CSCW) and collaborative editing. The idea is that all participants work on local sets of data and only operations performed on these data sets are exchanged. Operational Transformation reorders the operations, handles the inconsistency between operations and guarantees same result on all nodes. It is about where and how operations are executed. It allows to build true real-time applications. An operational transformation system can be divided into two layers, the transformation control algorithm and the transformation functions [1]. The transformation control algorithm decides which transformation function has to be applied to which set of concurrent operations. The transformation functions, on the other hand, are used to transform concurrent operations. Two types of transformation functions have been proposed: inclusion and exclusion transformations. For algorithms that use a multi-dimensional data structure to keep track of operations in their original, intermediate, and executed forms only inclusion transformation is needed. For algorithms that use a one-dimensional history buffer to save operations in their executed form only apart from inclusion transformation, exclusion transformation is needed to recover operations' original and intermediate forms from their executed forms. The identification of proper

- Mandeep Kaur has completed graduation in Computer Science Engineering in Guru Gobind Singh Indraprastha University, India, PH-9818512789. E-mail: mndpkaur14@gmail.com
- Manpreet Singh has completed graduation in Computer Science Engineering in Guru Gobind Singh Indraprastha University, India, PH-8447344495. E-mail: manpreet22thesingh@gmail.com
- Harneet Kaur has completed graduation in Computer Science Engineering in Guru Gobind Singh Indraprastha University, India, PH- 9899936893. E-mail: harneetskaurz@gmail.com
- Simran Kaur has completed graduation in Computer Science Engineering in Guru Gobind Singh Indraprastha University, India, PH-9013875965. E-mail: simrankaur.adept@gmail.com

transformation post-conditions has played a crucial role in the design of both the generic transformation control algorithms and application dependent transformation functions. In essence, undo/redo can also be viewed as a kind of transformation, which is performed directly on the document states rather than on the operations.

2.1 WORKING OF OPERATIONAL TRANSFORMATION

For example we assume a list [a; b; c] that can be modified by two concurrent processes A and B. Let us assume that process A intends to delete the element C at index 2 (the list index starts at 0), which results in an operation $OPA = del(2)$. At the same time process B inserts a new element d at the beginning of the list (i.e. at index 0), resulting in an operation $OPB = ins(0; d)$. Both processes directly apply their own operation (OPA and OPB) to their own local list. Later on, they receive the operation sent by the other process. Applying the received operations without operational transformation would yield different lists for processes A and B (namely [d; a; b] for process A and [d; a; c] for process B) as shown in figure 1. This is because after process B inserted the element d, the index of the element that process A intended to delete shifted from 2 to 3.

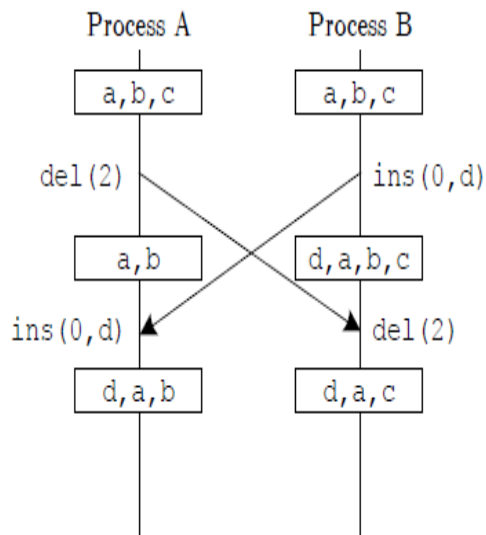


Figure 1. Without Operational Transformation

The operational transformation algorithm would thus modify the index of the delete operation in OP_A , to preserve the intention of process A. If process B applies the transformed operation OP_A with index 3, both processes result in the same list [d; a; b] as shown in figure 2. One of the main tasks of Operational Transformation is to maintain consistency in collaborative editing. Number of consistency model has been designed for consistency maintenance, for co-operative editing and for Operational Transformation. [2]

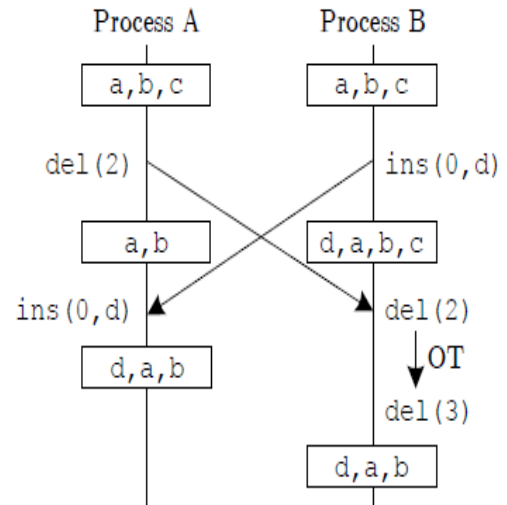


Figure 2. With Operational Transformation

Consistency maintenance is a fundamental issue in many areas of computing systems, including operating systems, databases systems, distributed shared memory systems, and groupware systems.

2.2 RELATED WORK

A number of prototype group editors have been built in the past by various research groups for testing the feasibility of transformation-based consistency maintenance algorithms, and for investigating system design and implementation issues. GROVE has been used in several real groups for a variety of design activities to evaluate the system from users' perspective and to gain usage experience [3], [4]. Moreover, the operational transformation has been found very useful in supporting user-initiated collaborative undo operations [5], [9]. Till now Google has produced various products using the concept of Operational Transformation which are renowned for their applicability. But none of the products available are open source. Various applications available online which are using the concept of Operational Transformation are: Writely (which later became Google Docs and finally named as Google Drive), Apache Wave (renamed to Google Wave), ZohoSuite, JotSpot, MobWrite, SynchroEdit, ShareJS, Coweb-jsoe, OpenCoweb, SubEthaEdit (code editor), EtherPad, and Mockingbird. Various frameworks for such type of applications are Racer, DerbyJS, OT.js, Differential Synchronization, Diff-Match-Patch, MobWrite, DriveSDK, beWeeVee and many more exists.

Apache Wave: It is a framework for real-time collaborative editing. Google originally developed it as Google Wave. Collaborative document editing means multiple editors are able to edit a shared document at the same time. It is live and concurrent where a user can see the changes another person is making, keystroke by keystroke [10]. Google Wave offers live concurrent editing of rich text documents. The result is that Google Wave provides a platform where you can see what the other person is typing, character by

character. Google Wave allows for more productive collaborative document editing environment, where people need not worry about stepping on each other toes and still use common word processor functionalities such as bold, fonts, italics, bullet points, and headings. To achieve these, Google Wave uses a concurrency control system based on Operational Transformation.

Google Docs: It is a web-based word processor part of a software offered by Google within its Google Drive, which provides the functionality of creating and editing documents online and collaborating them with users. Google Docs handles online edits in real time, and editors can see changes made by other collaborators. Docs can be shared, opened, and edited by a number of users simultaneously. A notification occurs whenever there is a comment or discussion or reply is made by a user. Sidebar chat allows users to discuss edits. One of the major features is that it works offline too. All changes are saved in a local cache and synched later to Google Docs once the user goes online [12].

3 LITERATURE REVIEW

3.1 WEB-SOCKETS

One of the most common hacks to create the illusion of a server initiated connection is called long polling. With the help of long polling, the client opens an HTTP connection to the server, which keeps it open until sending a response, which doesn't make them well suited for low latency applications. The main function of Web Socket is in web browsers and web servers though it can be used in a client-server application. The goal of this technology is to provide a mechanism for browser-based applications that need two-way communication with servers that does not rely on opening multiple HTTP connections [13]. They allow a long-held single TCP socket connection to be established between the client and server which allows for bi-directional, full duplex, messages to be instantly distributed with little overhead resulting in a very low latency connection.

The communications are done over TCP port number 80. The Web Socket protocol is supported by various browsers like Google Chrome [14], Firefox 6, Opera 12.10, and Internet Explorer 10. It requires web application on a server to support it. Server side implementation can be done in any language like Node.js, Java, Socket.IO, Ruby, Python, .NET and many more. Web Sockets are used in applications like Real-time Social streams, Chat Applications, Multiplayer online games.

API to create a new web socket object:

```
var Socket= new WebSocket( url, [protocol]);
```

3.2 CODEMIRROR

CodeMirror provides a code editor in the browser. It is specialized for editing code and comes with a number of language modes and add-ons that implement more advanced editing functionality. It is a functionality of JavaScript. It provides a number of features like auto indentation, add-ons for auto-completion, syntax highlighting, and broad programming API. The desktop version is supported by Firefox, Chrome, Safari opera and Internet Explorer [19].

4 PROPOSED WORK

The intention of this project is to provide a collaborative environment for editing the code document for which combination of web sockets, code mirror and ajax requests are used.

4.1 IMPLEMENTATION

In the application, to implement the concept of Operational Transformation, web sockets are used. Web Sockets are used to send data from one user to the other which helps in providing speed, security and removes lag, which may arise due to polling. For the editor purpose, we used CodeMirror which is based on JS lightweight. Ajax request gives 3 types of information, a particular location of the character inserted, an action performed and data. Action can be insert, delete, cut, paste. To access the application, the user needs to Login to the application. In case the user forgot or loses his password, a random pass-code will be sent to his GMAIL ID, after entering the code he can reset his password. For new users, Sign Up facility is integrated within the application. One aspect of the application is that it works online only, so any user who wants to code or edit file needs to be connected to the Internet. After logging into the system, user have two options:- to create a new file by providing file name, extension type(ex. .java, .js, .css) and the email id of a person with whom he wants to share the code and the other option is to edit the file previously created by him or shared with him. To remove the ambiguity of typing at same line location at the same time, we are displaying the line number of user at which he is working at. (in future we can use the concept of latency or bandwidth for maintaining consistency in data as it is used in Google Docs). The feature of emailing the code file is also incorporated into the application using the JAVAMAIL API which will send the document as an attachment to the email id being provided by the user who wants to mail it.

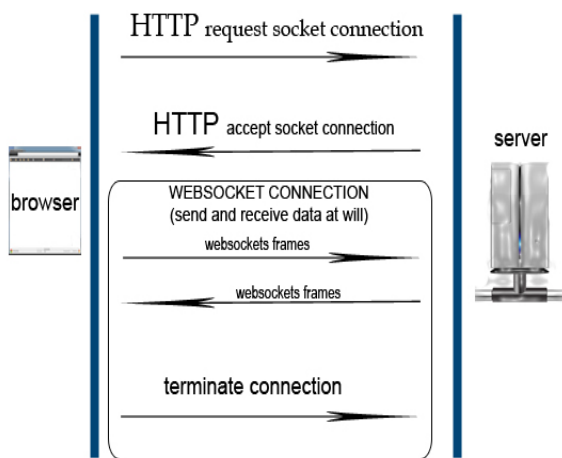


Figure 3. Life- cycle of web socket session

4.2 LIMITATIONS

- At present, the file can be shared between two users only but it can be further made available to multiple users to edit the same file.
- The user must provide GMAIL ID only for any purpose (login, sign-up, forgot password or while sharing document).
- Synchronization issues arises if one person is offline while other is editing the shared document, that is, if user A has created file and shared it with user B and user B joins user A later while user A has started editing text, then user B will be able to see contents which are written by user A after the instant user B came online. user B will be able to view the previous contents once user A saves the document.
- If both the persons are online and first one creates a file and share it with other, then the second person will have to refresh his page to view the shared document.
- Content synchronization problem also arises when a user performs cut operation on text in editor involving multiple lines. In that case the user which has performed cut operation will have normal file in which text will be shifted upwards but in other user's editor the code won't be shifted up and will remain on the same line number as it was previously, but the intermediate lines will be blank as the data is deleted.

5 FUTURE SCOPE

In future, the application can be extended to provide editing of a document by multiple users, that is, restriction of involvement of only two users can be removed. Secondly, right now user can mail the document but later a unique link can be created for each document which can be shared with other persons, who will be only able to read the file (no writing facility unless he login into the application). Third, this application can be extended to work as an Online Compiler which will be able to report online errors in the code. Lastly, notifications can be generated to let the other person know if any changes are done in the document and when are the modifications applied.

6 CONCLUSION

There are various methods to store the document. First, is storing the snapshots of the document. Second, includes storing the complete history of mutations applied to the document, which means there will be a complete history of every modification applied to the document. In this method, every time data model is rebuild from the mutation log which recreate data model using a minimal set of mutation from the mutation log. Google uses second method along with the concept of transformation to store the data, which further uses throttling algorithm. It says that prioritize the latency if the speed of typing is slow and prioritize the bandwidth if there is a high speed of typing. In our project, we are not using any transfer function, mutations or snapshot. We are saving the complete document and sending an AJAX request consisting of JSON which consists of three arguments: from, to and data.

ACKNOWLEDGMENT

This work was supported by Mr. Puneet Singh, Assistant Professor at Guru Tegh Bahadur Institute of Technology (Guru Gobind Singh Indraprastha University).

REFERENCES

- [1] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems," in Proceedings of the 1989 ACM SIGMOD international conference on Management of data, ser. SIGMOD '89. New York, NY, USA: ACM, 1989, pp. 399–407. [Online]. Available: <http://doi.acm.org/10.1145/67544.66963>
- [2] C. Boelmann, L. Schwittmann, T. Weis, "Deterministic Synchronization of Multi-Threaded Programs with Operational Transformation" Distributed Systems Group, University of Duisburg, Germany
- [3] C. A. Ellis, S. J. Gibbs, G.L. Rein: "Design and use of a group editor," In Engineering for Human-Computer Interaction. G. Cockton, Ed., North-Holland, Amsterdam, 1990, pp.13-25.
- [4] C. A. Ellis, S. J. Gibbs, and G. L. Rein: "Groupware: some issues and experiences," CACM 34(1), pp.39-58, Jan. 1991.
- [5] A. Prakash and M. Knister: "A framework for undoing actions in collaborative systems," ACM Transactions on Computer-Human Interaction, 4(1), pp.295-330, 1994.
- [6] 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 Trans. Comput.-Hum. Interact. 5:63{108, March 1998.
- [7] C. Sun and C. Ellis. Operational transformation in real-time group editors: issues, algorithms, and achievements. In Proceedings of the 1998 ACM conference on Computer supported cooperative work, CSCW '98, pages 59{68, New York, NY, USA, 1998. ACM. (Conference proceedings)
- [8] M. Ressel, D. Nitsche-Ruhland, and R. Gunzenbauser: "An integrating, transformation-oriented approach to concurrency control and undo in group editors," In Proc. of ACM Conference on Computer Supported Cooperative Work, pp 288-297, Nov. 1996. (Conference proceedings)
- [9] "Google + Quickoffice = get more done anytime, anywhere". Official Google Blog. Google. 5 June 2012. Retrieved 2013-09-12.
- [10] D. Wang, A. Mah, and S. Lassen. "Google Wave operational Transformation," July 2010. (White paper)

- [11] C. Sun, Y. Yang, Y. Zhang, and D. Chen. "A consistency model and supporting schemes for real-time cooperative editing systems," 1996.
- [12] Alexia Tsotsis (28 June 2012). "Google Docs now work offline". TechCrunch. (Preprint)
- [13] Ian Fette; Alexey Melnikov (December 2011). "Relationship to TCP and HTTP". RFC 6455 The WebSocket Protocol. IETF. sec. 1.7. RFC 6455.
- [14] D. Sheiko, "Persistent Full Duplex Client-Server Connection via Web Socket," 2010. Available: [http://dsheiko.com/weblog/persistent-full-duplex-client-ser-ver-connection-via-web-socket](http://dsheiko.com/weblog/persistent-full-duplex-client-server-connection-via-web-socket).
- [15] "Chromium Web Platform Status". Retrieved 2011-08-03.
- [16] Q. Liu, X. Sun, "Research of Web Real-Time Communication Based on Web Socket," 2012.
- [17] "Implementing a Syntax-Highlighting JavaScript Editor—in JavaScript". 2007-05-24.
- [18] CodeMirror list of language modes "CodeMirror list of language modes"
- [19] <http://codemirror.net/doc/releases.html>