A Collaborative Visual Database

by

Imed Adel

A document submitted in partial fulfillment of the requirements for the degree of

Technical Report

at

MISKATONIC UNIVERSITY

I Conceptual study

The deciding factor for the success of one application and the failure of another is the architecture. In order to achieve the requirements we set in the previous chapter, along with solving the issues faced by the existing solutions, we need to carefully design our systems.

In this chapter, we will first dive into the realm of real-time collaboration and read into the existing theoretical research in the field and its practical applications in the real world. Then, we will explore the different architectures for building and scaling our application, taking into consideration our choice of collaboration algorithms. Finally, we will present the sequence and class diagrams to better define the implementation of our ideas and the trajectory of our work in the following chapter.

1.1 Real-time collaboration

Real-time collaboration is a type of collaboration used in editors and web applications with the goal of enabling multiple users on different computers or mobile devices to modify the same document with automatic and nearly instantaneous merging of their edits. The document could either be a computer file, stored locally, or a cloud-stored data shared over the internet, such as an online spreadsheet, a word processing document, a database, or a presentation.

Multiple web applications support real-time collaboration under various names. Microsoft, for example, refers to it as "co-authoring" and offers it as part of its Microsoft Office bundle, including Word, Excel, and PowerPoint [3]. Google Docs is another notorious contender in the space of collaborative editing, with products such as Google Docs and Google Sheets.

The interest in collaborative software has seen a resurgence since 2020, mainly due to the move to remote work, with companies like Microsoft offering ready-to-use APIs to enable this feature.

Real-time collaboration is different from other offline or delayed collaborative approaches, such as Git. While real-time editing performs automatic, frequent, or even instantaneous synchronization of data between all the connected users, offline editing requires manual submission, merging, and resolution of editing conflicts.

1.1.1 History

In 1968, Douglas Engelbart introduced the first collaborative real-time editor in a presentation named "The Mother of All Demos", which also demonstrated many other fundamental elements of modern personal computing including windows, hypertext, graphics, video conferencing, the computer mouse, word processing, and revision control [4].

It took many decades for collaborative software to become mainstream. One of the first editors that offered collaborative capabilities was Writely, a collaborative word processor launched in 2005 [2]. It was later acquired by Google and renamed to Google Docs [6]. Another Google-backed product from the same era was Google Wave, a collaborative email software. However, less than a year later, it was discontinued due to a lack of users [5].

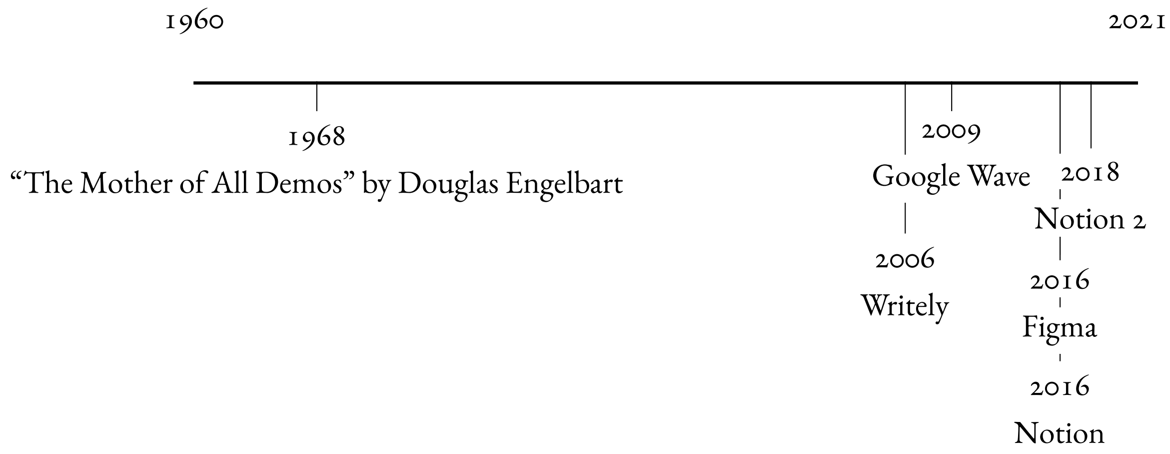

Figure 1.1: Timeline of real-time collaboration

Since 2010, the number of web-based collaborative software has been exploding with successful examples such as Figma and Notion. Figure 1.1 shows the resurgence of products with real-time collaboration.

1.1.2 Theoretical grounds

The main challenge of real-time collaboration is keeping multiple clients in sync. An algorithm has to be employed to determine how to apply—often conflicting—edits from the different remote users. Network latency is the main culprit for such a dilemma as modifications can reach the other clients with a certain amount of delay. Figure 1.2 shows a common example of conflicting changes by two users caused mainly by network latency.

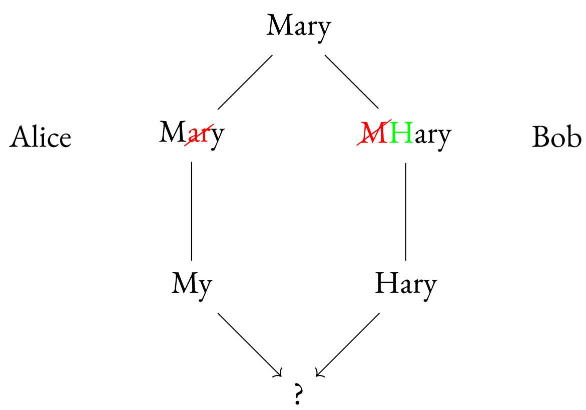

Figure 1.2: Example of a common problem in real-time collaboration

Over the years, two main algorithms emerged for dealing with real-time collaboration, Conflict-free Replicated Data Type (CRDT) and Operational Transformation (OT). Both have their benefits and drawbacks, and each one has numerous variations and implementations.

Operational Transformation (OT)

Operational Transformation was invented for supporting real-time co-editors in the late 1980s and has evolved to become a collection of core techniques widely used in today's working co-editors and adopted in major industrial products [10]. Google Docs has been using OT since at least 2009 [13].

The model of Operational Transformation works by calculating all possible transformations for a text and using them to transform received changes according to the local state, thus eliminating any possible mistakes of synchronization. Figures 1.3 and 1.4 explain the algorithm followed by OT for resolving conflicts. Figure 1.5 illustrates the solution proposed by OT for the problem proposed in 1.2.

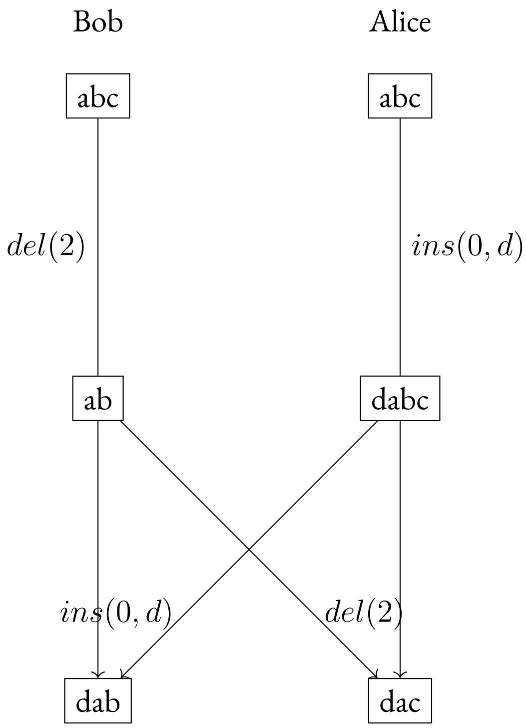

Figure 1.3: Without OT

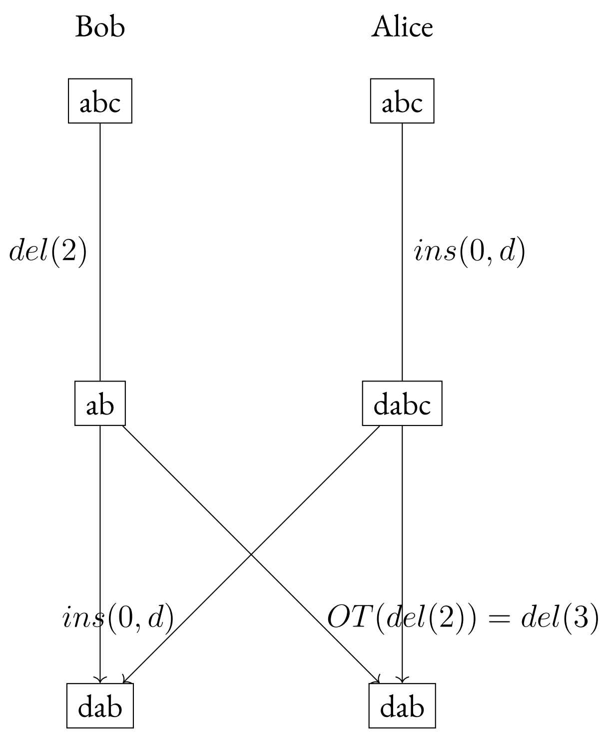

Figure 1.4: With OT

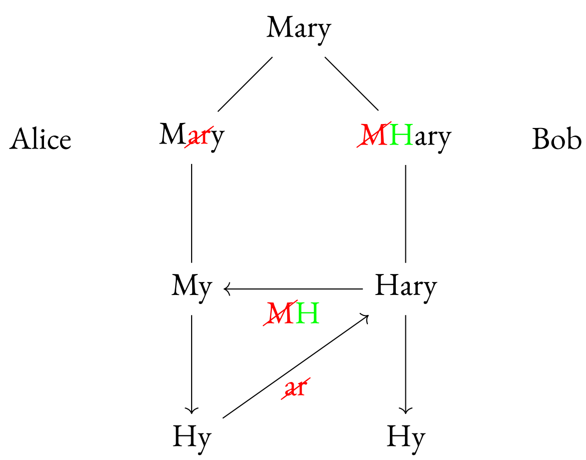

Figure 1.5: OT's solution for the problem in figure 1.2

As seen in the previous examples, OT can be hard and costly to set up, even for text-based documents. When it comes to more complicated and nested data structures, OT is rarely the first choice.

CRDT

Conflict-free Replicated Data Type is a data structure that was first proposed around 2006 under the name of WithOut Operational Transformation (WOOT) [10]. It has the benefit of being able to be duplicated over numerous computers in a network, where the replicas can be updated independently and concurrently without requiring coordination, and where inconsistencies can always be resolved mathematically [9].

In 2011, CRDT was formally defined [9], and while it was initially developed for collaborative text editing, it was eventually adopted for multiple other use cases such as online chat systems and distributed databases.

CRDT is further subdivided into two types based on its implementation—Commutative Replicated Data Type (CmRDT) and Convergent Replicated Data Type (CvRDT). Both of them offer the same real-time capabilities but differ in their design and approach to the concept.

CmRDT Commutative Replicated Data Type or Operation-based CRDT transmits the local state only by sending the update operations required to reach that state. Upon receiving these operations, remote clients must apply them to become in sync with the sender. The transmission server has to avoid duplicating the operations as this algorithm is not idempotent.

CvRDT Convergent Replicated Data Type or State-based CRDT sends over the whole local state to be merged with the receiving clients' states. This method tends to be slow due to the need of sending large amounts of data instantaneously over the network. However, the infrastructure does not have to deal with deduplication as it is the case with CmRDTs.

1.1.3 Practical examples

In practice, web applications do not adhere to one algorithm. Instead, they draw from different concepts to establish a solution that perfectly fits their needs.

Figma

Figure 1.6: Figma logo

Figma is one of the first web design tools with real-time collaboration. It was released in 2016 and since then it has been hailed as one of the best examples of performant and collaborative software [11].

The engineering team behind Figma avoided following the same footsteps of Google Docs in using OTs, which, while performant and required less memory, were too complicated to implement and scale [12]. Instead, they decided to implement their real-time collaboration features using a system inspired by CRDT. Since they already had a centralized server and a database acting as the source of truth, they were able to remove much of the complexity of CRDTs while maintaining their original concept [12]. In particular, the server decides which changes are applied and which ones are discarded based on the timing of these changes and their order. This mode of operation can be viewed as a modified version of CmRDT.

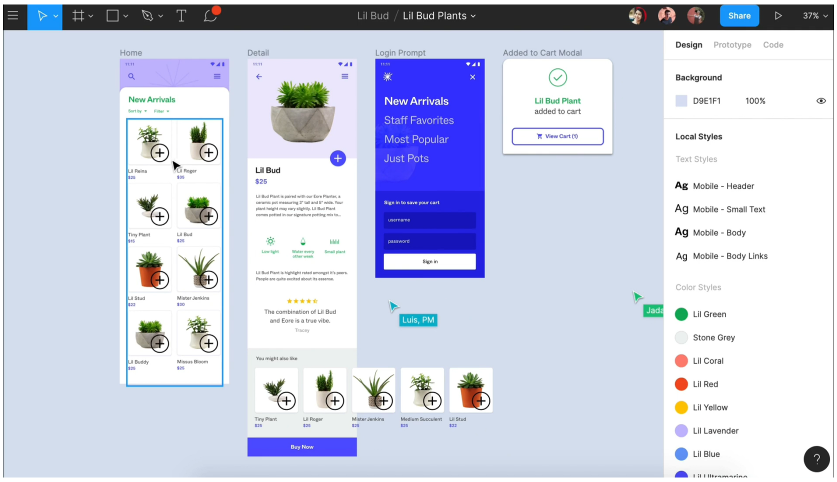

Figure 1.7: Figma collaborative interface

Excalidraw

Figure 1.8: Excalidraw logo

Excalidraw is an open-source virtual collaborative whiteboard tool that lets you easily sketch diagrams that have a hand-drawn feel to them [8]. Contrary to Figma, Excalidraw is Peer-To-Peer (P2P), that is, the server is not the source of truth and used merely for connecting the different clients and propagating changes [1]. On top of that, the exchanged data is encrypted end-to-end, meaning that the server has no access to clients' changes. Excalidraw uses a variant of Convergent Replicated Data Type (CvRDT) with the whole state being sent over and compared to the local state before applying changes.

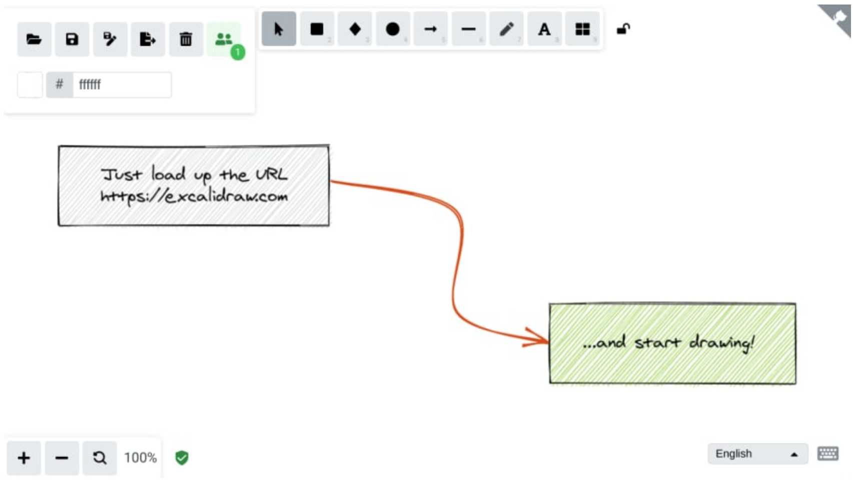

Figure 1.9: Excalidraw collaborative interface

What is peculiar about Excalidraw's example is the way they handle concurrent changes of the same element. Each element has a version attached to it and a hash of that version, which are compared to know which changes to apply and which ones to discard. If two clients edit the same element at the same time, Excalidraw does not try to merge those changes, instead, it picks whichever change came first [1]. Figure 1.10 illustrates the algorithm implemented by Excalidraw for merging conflicting states.

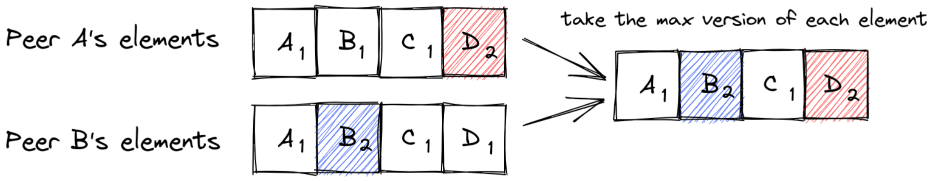

Figure 1.10: Excalidraw collaboration algorithm as explained on their blog [1]

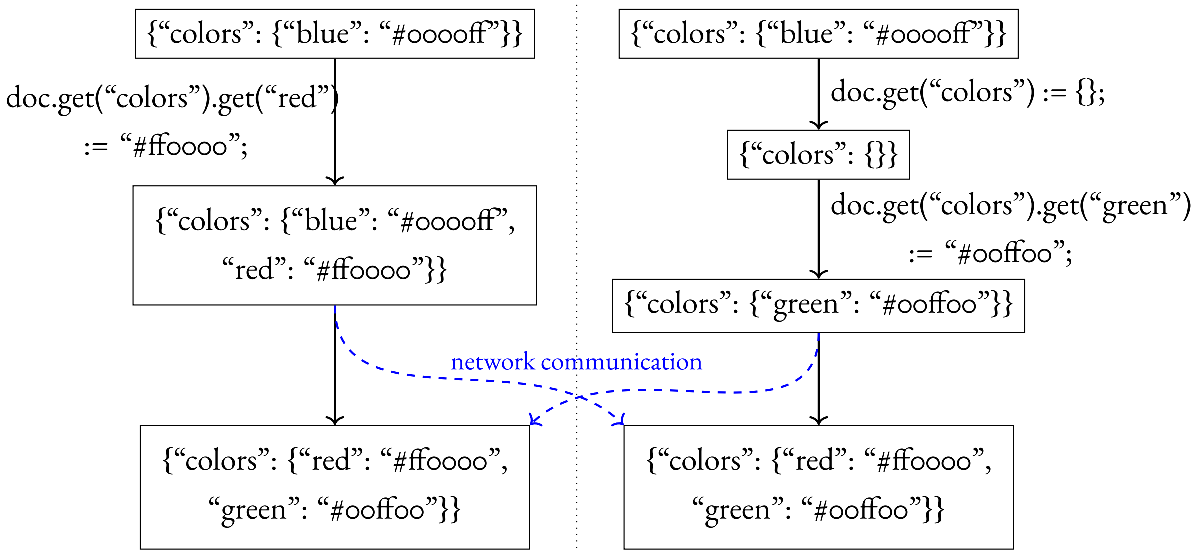

Figure 1.11: Example of Automerge algorithms

Automerge

Automerge is a set of algorithms for applying CRDT concepts to a JavaScript Object Notation (JSON) data structure without having prior knowledge of the inner working of the data structure or the infrastructure used to transmit changes. It works by registering all the changes applied locally to the state and sending these operations to remote clients, while employing a variety of CRDT algorithms for resolving conflicts [7]. Automerge also comes with a ready-to-use JavaScript library. Figure 1.11 illustrates one of the algorithms provided by Automerge.

Compared to Figma, Automerge does not require any central source of truth. On top of that, it tries to resolve conflicts rather than picking one of the changes based on the version or timestamp, as it is the case in Figma and Excalidraw.

Centige

Centige is a discontinued collaborative web app builder by the author. It was mainly a research project, started in late 2019 and developed throughout 2020, around the same time as Excalidraw. The web application used a collaboration model very similar to that of Figma, as it relied on a centralized server acting as the source of truth and used Operation-based CRDT for transmitting changes to the other clients. Each client had a Conflict-free Replicated Data Type (CRDT) and on ever change, the new local state would be compared with the older one, in a model similar to that of State-based CRDT, before generating a set of operations that are sent to the server. The latter would compare the operations' hash before propagating them to rest of the connected clients. All of these computations happened within milliseconds and supported offline editing to some extent.

1.1.4 Our choice

After reviewing the literature on real-time collaboration and having researched the different implementations of the feature in varying products, it is time to choose a fitting design pattern for real-time collaboration in Merebase. It is important to keep in mind that the design of the collaborative aspect of our application greatly influences the rest of the architecture. Considering the ease-of-use of Conflict-free Replicated Data Type (CRDT) and the sheer volume of the scientific literature around it, coupled with the community around it and our knowledge of the algorithm, we chose to implement a variant of CRDT.

In particular, we are going to implement Operation-based CRDT (CmRDT) in approach similar to that of Excalidraw with each element having a version attached to it. However, in contrast to Excalidraw's implementation, we are going to rely on a server a sourth of truth. This choice eliminates any complexities with P2P technology and keeps the performance smooth as we do not have to perform any comparisons or calculations on the client's device. Therefore, we achieve one of our non-functional requirements—speed.

1.2 Design patterns

1.2.1 Common patterns

1.2.2 Comparison

1.2.3 Conclusion

1.3 Detailed architecture

1.4 Class diagrams

1.5 Sequence diagrams

1.6 Activity diagrams

1.7 Conclusion

2 Implementation

Researching the current solutions led us to formulate a set of requirements to ensure that Merebase offers the best experience.

Acronyms

CmRDT Commutative Replicated Data Type

CMS Content Management Software

CRDT Conflict-free Replicated Data Type

CvRDT Convergent Replicated Data Type

FDD Feature-Driven Development

JSON JavaScript Object Notation

OT Operational Transformation

P2P Peer-To-Peer

RBAC Role-Based Access Control

SaaS Software as a Service

UI User Interface

UX User Experience

WOOT WithOut Operational Transformation

Bibliography

[1] *Building Excalidraw's P2P Collaboration Feature | Excalidraw Blog*. en. URL: `https://blog.excalidraw.com/building-excalidraw-p2p-collaboration-feature/` (visited on 06/05/2021).

[2] E. Chang. *eHub Interviews Writely*. English. Oct. 2005. URL: `https://web.archive.org/web/20110722190058/http://emilychang.com/ehub/app/ehub-interviews-writely/` (visited on 06/04/2021).

[3] *Document collaboration and co-authoring*. en-US. URL: `https://support.microsoft.com/en-us/office/document-collaboration-and-co-authoring-ee1509b4-1f6e-401e-b04a-782d26f564a4` (visited on 06/04/2021).

[4] *Firsts: The Demo - Doug Engelbart Institute*. URL: `https://www.dougengelbart.org/content/view/209/448/` (visited on 06/04/2021).

[5] I. Fried. *Google pulls plug on Google Wave*. en. URL: `https://www.cnet.com/news/google-pulls-plug-on-google-wave/` (visited on 06/04/2021).

[6] *Google acquires online word processor, Writely*. en-US. Mar. 2006. URL: `https://venturebeat.com/2006/03/09/google-acquires-online-word-processor-writely/` (visited on 06/04/2021).

[7] M. Kleppmann and A. R. Beresford. "A Conflict-Free Replicated JSON Datatype." In: *IEEE Transactions on Parallel and Distributed Systems* 28.10 (Oct. 2017), pp. 2733–2746. ISSN: 1558-2183. DOI: 10.1109/TPDS.2017.2697382.

[8] *Rethinking Virtual Whiteboard | Excalidraw Blog*. en. URL: https://blog.excalidraw.com/rethinking-virtual-whiteboard/ (visited on 06/05/2021).

[9] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. "Conflict-Free Replicated Data Types." en. In: *Stabilization, Safety, and Security of Distributed Systems*. Ed. by X. Défago, F. Petit, and V. Villain. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2011, pp. 386–400. ISBN: 9783642245503. DOI: 10.1007/978-3-642-24550-3_29.

[10] C. Sun, D. Sun, A. Ng, W. Cai, and B. Cho. "Real Differences between OT and CRDT under a General Transformation Framework for Consistency Maintenance in Co-Editors." en. In: *Proceedings of the ACM on Human-Computer Interaction* 4.GROUP (Jan. 2020), pp. 1–26. ISSN: 2573-0142. DOI: 10.1145/3375186. URL: https://dl.acm.org/doi/10.1145/3375186 (visited on 06/04/2021).

[11] U. X. Tools. *2020 Tools Survey Results*. en-us. URL: https://uxtools.co/survey-2020/ (visited on 06/05/2021).

[12] E. Wallace. *How Figma's multiplayer technology works*. en-US. URL: https://www.figma.com/blog/how-figmas-multiplayer-technology-works/ (visited on 06/05/2021).

[13] *What's different about the new Google Docs: Conflict resolution*. en. URL: https://drive.googleblog.com/2010/09/whats-different-about-new-google-docs_22.html (visited on 06/04/2021).