Challenges in Real-time Collaborative Editing

Péter Szűcs

Collaborative software or groupware is a software that is designed to support a group of people in achieving a common goal [6]. This task typically includes viewing and editing shared media, which can be basically anything from a text document to a CAD model provided collaborative editing makes sense. A real-time collaborative editor allows its users who are connected by some communication network, like the internet, to view and edit this shared media in a parallel fashion regardless of their geographical locations. Consistency maintenance proved to be the most significant challenge in designing and implementing these kinds of systems. Due to this, collaborative editing has been research topic for over twenty years. This paper aims to go through some of the milestones of the past two decades, giving a brief introduction to a popular technique used in present-day implementations, called Operational Transformation, and also, to outline the concept of a framework that could support such systems.

The term Operational Transformation was introduced by S.J. Gibbs and C.A. Ellis in 1989 and the technique was first used in the GROVE (GRoup Outline Viewer Editor) system which is an outline editor that enables it's users to view and edit textual outlines simultaneously [1]. Since it was first introduced numerous research groups have contributed to the technique and created their own implementations. One of them is the Jupiter system [4] (developed at Xerox-PARC) which later lead to Google's wave protocol [5].

In Operational Transformation we take an edit-based approach, which means that we capture user actions and mirror them across the network to other users [2]. It's important note that in order to achieve a consistent state, all actions must be captured. The richness of modern user interfaces can make this rather problematic, since besides basic operations like textual insertion and deletion there are more complex actions that are much more difficult to handle, like automatic completion or drag & drop [2]. However, for simplicity, in this paper we will stick to basic textual editing that only involves insertion and deletion of text.

The basic idea is that changes to the document can be modeled as *operations*. When the user changes the document the changes are usually immediately applied locally, for fast response then an operation that represents this change is then sent to the other users. Since the others might have changed the document as well the incoming operation may be out of context. For example, if the incoming operation would insert character 'a' at position 2, but we deleted a character at position 0, the operation should be changed to refer to position 1 instead. In such cases, the incoming operation needs to be *transformed* first and then executed [3].

While it may seem simple, there are several difficulties to be handled in order to achieve a consistent document state at every user. In a complex scenario operations may arrive in different orders for every user, still we have to ensure that when no operations occur, the state of the edited document is the same for every user and that this state is consistent with what the users wanted to achieve. This isn't a trivial task even when we only deal with simple operations like textual insertion and deletion, not to mention more complex operations like drag & drop.

In this paper, we will see the basic concepts of Operational Transformation, some of the difficulties regarding consistency maintenance, solutions to them, as well as possible alternatives. In the contributions section we will also see a concept of a framework to support such systems. The framework is based on the Jupiter system and also uses ideas from the Google Wave white papers.

References

[1] C.A. Ellis, S.J. Gibbs. *Concurrency Control in Groupware Systems*, Proceedings of the 1989 ACM SIGMOD international conference on Management of data, 1989

[2] Neil Fraser. *Differential Synchronization*, Proceedings of the 9th ACM symposium on Document engineering, 2009

[3] Chengzheng Sun, Clearence Ellis. *Operational Transformation in Real-Time Group Editors: Issues, Algorithms, and Achievements*, Proceedings of the 1998 ACM conference on Computer, 1998

[4] David A. Nichols, Pavel Curtis, Michael Dixon, John Lamping. *High-Latency, Low-Bandwidth Windowing in the Jupiter Collaboration System*, Proceedings of the 8th annual ACM symposium on User interface and software technology, 1995

[5] David Wang, Alex Mah, Soren Lassen. *Google Wave Operational Transformation*, 2010

[6] *Collaborative software*, Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Collaborative_software, 2014