

Conflicts and Resolutions in Computer Supported Collaborative Work Applications

B

Mamdouh Babi

Department of Electrical and Computer Engineering, Cleveland State University, USA

Wenbing Zhao

Department of Electrical and Computer Engineering, Cleveland State University, USA

INTRODUCTION

Computer Supported Collaborative Work (CSCW) refers to collaborative work of multiple users using computers or a network to share the same application. Computers are networked together linking people to create “social network.” Computer supported social networks (CSSNS) sustain strong, inter-mediate and weak ties that provide information and social support in both specialized and broadly-based relationships. CSSNS connect users within and between one or multiple dispersed geographical location (Wellman, 1996). Many CSCW applications have been developed to support different infrastructures (Conklin, 1988). Some applications are used for sharing editing, others used for Computer Drafting and Design using 2D and 3D models. One of the important applications used is CSCW is Collaborative Editing. Collaborative Editing is a real time Computer Supported application that allows geographically dispersed multiple users to view, edit, or share documents at the same instance of time over the Internet. There are three facts we must know about CSCW applications:

- Real time CSCW applications are applications not different from the ordinary applications with conventional functionality.
- Real time CSCW applications are different from the typical applications you normally use because they are able to support multiple users to share the same application at the same time.

- In CSCW applications, human users are part of the system; therefore conflict may have a negative impact on the system.

Many implementations have been developed taken into consideration the performance issues of Distributed Shared Memories systems (DSMs). Implementations developed to take into account the portability of programs and API, for example (Geva, M; Wiseman, Y, 2007). Such developments cannot help to resolve conflicts among users.

Conflict is a common phenomenon in CSCW and it may have both positive and negative impact on the system. Therefore, it is very important to come up with some types of conflict management systems that will help in reducing conflicts among users. Such systems will not only control the negative aspects of conflicts but also the positive aspects.

Generally, conflicts can be either system conflict or users' conflicts or may be both. While systems conflicts are typically reflect the system operation, delay, and bandwidth, users' conflicts are determined when users have different views and opinions on how to deal with shared document. Users may have different meaning and point of view even with the same concept and that is due to users relying on different resource of information.

Conflict can occur when users simultaneously change the content of the shared document (Agustina et al., 2012; Mills, 1996; Alemes et al., 1999). We can create some type of prevention to maintain consistency by prohibiting users from doing concurrent work, and therefore, preventing conflict.

DOI: 10.4018/978-1-4666-5888-2.ch054

In this Article, we will discuss the methods of Conflict Detection in CSCW applications. We will state our opinion leading us to our research in the area of conflict and resolutions in CSCW applications. Pessimistic Approaches, Optimistic Approaches, and Serialization as methods of conflicts resolutions as well as the advantages and disadvantages of each method will be discussed.

BACKGROUND

CSCW involves two or more users to work collaboratively on the same application. Typically, CSCW application allows multiple users to be part of a team without the concern of geographical restraints. Although, CSCW working environments have many great advantages such as, teams members working together asynchronously allowing members to participate with the luxury to contribute whenever they want, they are certainly facing many challenges. In this article, we will discuss the challenges related to multiple users' conflicts in CSCW applications and focus on some of the resolution of such challenges.

DETECTING CONFLICTS IN CSCW APPLICATIONS

Detecting conflicts in CSCW is a little tedious and can sometimes be complex, especially when we have more than two users using the same application. To understand conflict detection better, let us assume that each location maintains a log (L). The log will include all operations ($O1, O2, O3...On$) executed. We also need to have some type of mechanism for capturing the concurrency relationship between operations, such as the state and context (Chen et al., 1998; Sun & Sun, 2009). This type of infrastructure is required by consistency maintenance, or it undoes techniques in Collaborative Editing applications (Augustina et al., 2012; Begole et al., 1999; Kanawati, 1997; Xu et al., 2004).

To understand how we can detect conflict operations, we assume that casually dependency operations (O) are executed in casual order. On the other hand,

concurrent operations can be executed in any order (Chen et al., 1998). Therefore; for each casually ready operation, we need to detect other operations that might have an impact on the execution of such operation. The most common way to do this is to search through the log (L) and search for set of operations that conflict with O . This solution works only for simplistic approaches.

To illustrate, let us consider the following examples for simplistic scenario shown in Figure (1) with two participants with two conflicts operations. For this scenario, the following execution is to take place: Participant 1 at site 1 performing $O1 = Update (idx\ Color, Magenta)$ to change the color of the object idx to *Magenta* and Participant 2 at site 2 performing $O2 = Update (idx, color, Yellow)$ to change the color of the object idx to *Yellow*. Assume that $O1 = Update (idx\ Color, Magenta)$ is executed before $O2 = Update (idx, color, Yellow)$. After executing $O1$, the color changes to *Magenta* and $L = \{O1\}$. When $O2$ arrives a detection of the operation will occur and the synthesized color *Red* is created. Now let us assume that $O2$ is executed first and then $O1$ is executed.

However, things become more complicated when more than two participants and their operation are not mutually conflicted. Let us extend our pervious example to three users the three users will perform their operation $O1, O2$, and $O3$. The dependency relations are causal dependency: $O1 \parallel (O2 \rightarrow O3)$. Suppose that user 3 perform the operation $O3 = Update (idx, Color, Green)$, and both $O1$ and $O2$ have the same operations previously done in the previous scenario Figure (2) below.

The relationship between $O1$ and $O2$ is $O1 \otimes O2$ (\otimes conflict relation). However, $O2$ and $O3$ are not conflicting, because they are not concurrent.

- At site 1, when Operation $O1$ is executed the object color changed to *Magenta* and $L = \{O1\}$. When operation $O2$ is executed, the object color changed to *Red* and $L = \{O2\}$. When operation $O3$ is executed, the object color changed to *Blue* and $L = \{O3\}$.
- At site 2, when $O2$ and $O3$ is executed in sequence ($O2$ then $O3$), the object color *Magenta* and then to *Cyan* and $L \{O2, O3\}$. Then Operation for $O1$ arrives for execution. At this

7 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/conflicts-and-resolutions-in-computer-supported-collaborative-work-applications/112370

Related Content

The Potential Role of the Software Industry in Supporting Economic Development

Sherif H. Kamel (2018). *Encyclopedia of Information Science and Technology, Fourth Edition* (pp. 7259-7269).

www.irma-international.org/chapter/the-potential-role-of-the-software-industry-in-supporting-economic-development/184422

A Primer on Q-Method and the Study of Technology

Stéphanie Gauttier (2021). *Encyclopedia of Information Science and Technology, Fifth Edition* (pp. 1746-1756).

www.irma-international.org/chapter/a-primer-on-q-method-and-the-study-of-technology/260303

Factors Impacting Defect Density in Software Development Projects

Niharika Dayyala, Kent A. Walstrom, Kallol K. Bagchi and Godwin Udo (2022). *International Journal of Information Technologies and Systems Approach* (pp. 1-23).

www.irma-international.org/article/factors-impacting-defect-density-in-software-development-projects/304813

IoT Setup for Co-measurement of Water Level and Temperature

Sujaya Das Gupta, M.S. Zambare and A.D. Shaligram (2017). *International Journal of Rough Sets and Data Analysis* (pp. 33-54).

www.irma-international.org/article/iot-setup-for-co-measurement-of-water-level-and-temperature/182290

Cloud Computing

Gavin O'Donnell, Nigel McKelvey and Kevin Curran (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 1033-1038).

www.irma-international.org/chapter/cloud-computing/112498