MCPS2: Intention Maintenance of Structure Document Based MCPS Under Mobile Platform

Dan Wang[1(✉)], Sizheng Zhu[1], Liping Gao[2], Shanshan Wang[1],
Xiaofang Xu[2], and Changqing Tao[2]

[1] Computer Center, University of Shanghai for Science and Technology,
Shanghai 200093, China
wangdandjxy@sina.com

[2] School of Optical-Electrical Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Abstract. In order to meet the intentions of multi-user to the maximum extent possible, and enhance the meaning of collaborative works. We improve an intention maintenance algorithm called MCPS2, which based on the MCPS algorithm in the last paper. The MCPS2 algorithm uses the P2P network connection method to construct a cooperative customer network, and all cooperative clients participate in a collaborative editing according to requesting structured documents by partial replication strategy. At the same time, combined with user liveness, giving master attribute to collaborative sites, and add the Title Creator (TC) attribute to title-node, reasonable arrangements for new nodes, editing, request and other rights. Finally, we support permission transfer dynamic control mechanism to ensure the user's immediate demand.

Keywords: P2P · Partial replication architecture · Structured document
Liveness · Permissions allocation

1 Introduction

With the vigorous developing of the computer industry, promote and affect people's daily work and life, especially the emergence of the Internet, help them across the area and culture to implement online and real-time communication. In recent years, the increasingly scale of work in some industry, is difficult to handle a huge job on individual efficiently and qualitatively. Therefore, a new working mode called CSCW [1] (Computer Supported Cooperative Work) have been improved.

The appearance of CSCW significantly changes the traditional way of working, highly improves the efficiency. It supports the users to deal with the same collaborative work at the same time by using different equipment even in the different areas.

After that, multiple collaborative algorithms, systems and applications [2–5] related to CSCW be supported, and in the research and development process, also emerged a

© Springer Nature Singapore Pte Ltd. 2019
Y. Sun et al. (Eds.): ChineseCSCW 2018, CCIS 917, pp. 20–31, 2019.
https://doi.org/10.1007/978-981-13-3044-5_2

series of conflict and consistency problems [2]. Later, researchers put forward and improved a variety of algorithms around conflict resolution and consistency maintenance. It is important to point out that the research of consistency maintenance mainly solves the problem of differences within documents between the collaborative users, finally confirm each collaborator holds the same copy; Conflict resolution mostly intended to maintain the intentions of every collaborative user and reflect the significance of collaborative work. At present, according to many researchers deep work, the achievements about consistency maintenance algorithms and methods is relatively mature, but there are relatively few studies involved in the intention maintenance. In one collaborative work, how to maintain the intention of each cooperative user to the maximum level is important to improve the significance of collaborative work.

In recent years, the emerge of mobile devices and the fast development of mobile terminals technology, platforms and applications in endlessly, urge the traditional collaborative applications and algorithms supporting PC side gradually transplant to mobile applications in order to adapting to the trend of technology development. The differentiation of network, storage and device with PC side in the process of transplantation are the key problems to be resolved urgently.

Structured documents, we can simply understand it as a context structured by text content and hierarchical organization [7]. For example, the word document editing templates that we frequently use at work, headlines, tables and pictures have strict style and hierarchy, in which the title level can clearly reflect the structure of the content in the document. Combine the structured documents with collaborative editing work, and store related context in a structured way. On the one hand, it refines the integral conflict to part conflicts, improves the efficiency of collaboration; on the other hand, we give corresponding authority to content of different levels, as much as possible to ensure all collaborative users' intention; moreover, the structured storage mode more suitable for mobile terminals to solve the shortcomings of limited storage space. Therefore, how to optimize the storage mode of the structure document at the mobile end, how to rationally design the user's editing authority, and ultimately improve the cooperation efficiency, is of great significance.

The rest parts of this paper contains: First, overview the MCPS algorithm [6], and define the document structure, node attributes, node type, and the network connection mode. Next, we present the MCPS2 algorithm and give the detail algorithm description of the initialization of documents, the mode of arbitration, the transfer of authority and so on. Then, analyze the complexity of the core algorithm and present a motivating example for implementing the MCPS2's processing mechanism, to verify its accuracy. Finally, we give the conclusion of this work and discuss future work.

2 Background Knowledge

In 1988, Sun [2] supplemented the standard of intention maintenance based on the consistency maintenance model CCI (Convergence Causality Intention) that proposed in the original dPOT [1] (distributed Operational Transformation) algorithm, emphasized the importance of collaborators' intention in a collaborative work. In terms of consistency maintenance, the most representative algorithms are roughly divided into two classes, the OT [1] (Operation Transformation) algorithm based on operation transformation and the AST [4] (Address Space Transformation) algorithm based on address space transformation. So far, many concurrency control algorithm and scheme based OT or AST derived, as GOTO [1], SCOT4 [8], ABST [9], TIBOT [10] and so on. At the same time, these two kinds of technology are widely used in the development of collaborative applications and systems, such as the distributed sketching system called SketchPad [11] supports multi user collaborative editing and the Co-CAD [12] collaborative system plunks for product design. In the aspect of intention maintenance, some conflict resolution strategies had been proposed to solve the problems appear in editing of different types of documents, like the multi version strategy [13] to solve the intention conflict of text, and the semantic conflict resolution strategy based on the two-dimensional graphics editing proposed by Ignat [14]. However, most of these studies are based on fully replication architecture [2], and rely on the state of optimism, it is necessary to ensure that the network is continuous and communication is unhindered, and the storage space is sufficient and so on. Once the scale of document becomes lager or the number of operations is rising, even the network interrupted suddenly, if we cannot provide appropriate resolved scheme, it must lead to failure of collaborative working and collaborative strategy. However, limited storage and network instability are the characteristics of mobile terminals. In response to this problem, in 2014, Huanhuan Xia [15] et al. proposed the partial replication architecture, which not only saves storage space, but also saves the update time of the copy and is more suitable for mobile terminals than the full replication architecture. Then, Sun proposed the CSOT [16] (Cloud storage Operational Transformation) in 2016, first transplanted the consistency maintenance ability of OT algorithm into the shared space of cloud storage, and achieved an ideal combination of concurrent operation and effect, which lays a foundation for the development of cloud based collaborative technology.

Based on the research of Huanhuan Xia and Sun, we put forward a conflict resolution algorithm in last article to resolve the problem of structured document collaborative editing based on the partial replication architecture in cloud platform called MCPS algorithm. We summarize several conflicts may occur in the collaborative editing of structured document, and give each of them the appropriate algorithm of resolution. Innovatively proposed two concepts of TLV (Tree liveness) and NLV (Node liveness) to solve conflicts of collaborative users' conflicts dynamically, to some extent, the intention of cooperative users maintained objectively. Nevertheless, the entire collaborative system of structured document still needs to be refined, such as how to control the user's entry and exit, as well as how to distribute the privileges of different cooperative users. In previous intention maintenance algorithms, more of them solve the conflicts by comparing priority of cooperative sites or comparing the timestamp of conflict operations or use the version control method and so on. In this paper, we modify the mode of network connection, add the attributes of title-master and site-master, give permission to all collaborators in the work. In order to realize the characteristics of the user that comes and go, we design a reasonable control mechanism of authority transferring according to the user's editing liveness in the shared document. We propose an intention maintenance algorithm of structured document based on collaborators' liveness, which is called MCPS2 algorithm.

3 Preparatory Works

3.1 Review of MCPS Algorithm

1. CIB: InsertBefore()_InsertBefore(), when the cloud server receives two operations of InsertBefore, they point to the same parent node, and the value of refId is the same.
2. CII: Insert()_Insert(), like the condition of CIB, it may get distinct results if the operations executed in different orders.
3. CD: Delete(), when a client wants to delete a title.
4. CDI: Delete()_InsertBefore(),when a delete operation is allowed, the cloud server receives a InsertBefore operation at the same time, they have a common parent node, and the refId of the deleted node is the same as the refId of the InsertBefore operation.
5. CUN: UpdateName()_UpdateName(), when the two operations change the name of the same node at the same time.
6. CUD: UpdateData()_UpdateData(), similar to CUN, when two operations want to change the text content of the same node, different executed sequence may occur conflicts.

In the Fig. 1, we give the basic definition involved in the node properties designed by the MCPS algorithm, in which the node set in the dashed box is the subtree of the StrTree. We define it as skeleton tree that the collaborative clients can request from the server. The detailed replica request policy of partial replication architecture can refer to the article [15]. In order to maintain the collaborative users' intention to the maximum extent, we propose the definitions of NLV: Node liveness and TLV: Tree liveness to record and update the editing liveness of each collaborative users at all title nodes dynamically. Once the editing conflict appearing, we judge the priority sequence of operation execution by comparing the liveness of the conflict sites. This method is more objective and fair than the previous way of judging operation order by judging the priority and timestamp of the conflict sites. This paper will improve and refine the communication mode, collaborative editing process, user permission, definition of structured document, and method of intention maintenance based on the previous MCPS algorithm, so as to improve the availability of the algorithm in the system development.

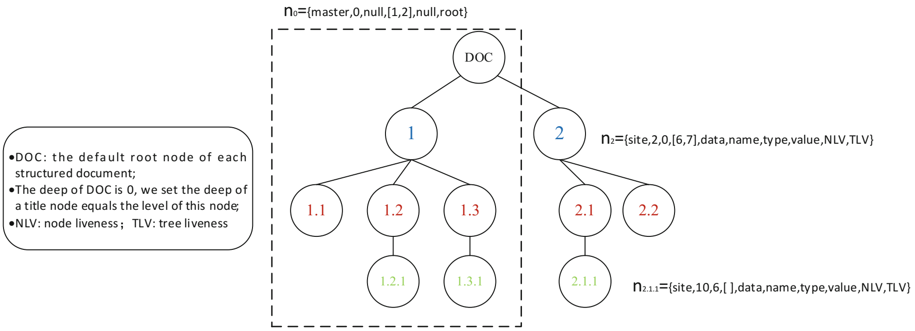

n0={master,0,null,[1,2],null,root}
•DOC: the default root node of each structured document;
•The deep of DOC is 0, we set the deep of a title node equals the level of this node;
•NLV: node liveness ; TLV: tree liveness
DOC
1
2
n2={site,2,0,[6,7],data,name,type,value,NLV,TLV}
1.1
1.2
1.3
2.1
2.2
1.2.1
1.3.1
2.1.1
n2.1.1={site,10,6,[ ],data,name,type,value,NLV,TLV}

Fig. 1. A figure introduces the Structured Document Tree (StrTree) and skeleton tree.

3.2 Basic Definition

Mode of Network Connection

This article does not follow the connection mode of MCPS, instead of selecting the connection mode using P2P. We set up a master for each collaboration team. When the collaboration work begins, master is the sponsor of the collaborative editing of the structure documents. Once the master is withdrawn from the collaborative work, a new master will be selected from the collaboration team according to the membership liveness, and the detailed selection process will be given in the next design of the algorithm (Fig. 2).

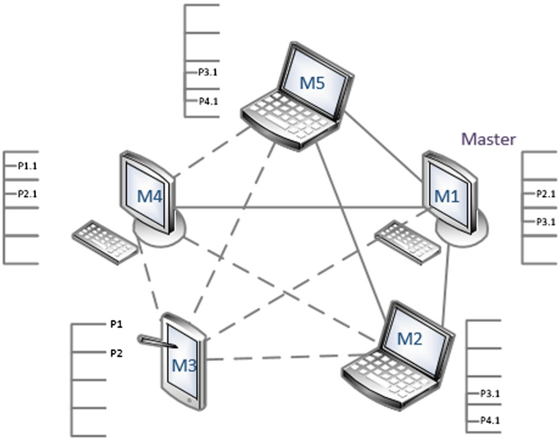


Fig. 2. The P2P connection of collaborative sites

The main work of Master is to initiate the structured document to the cooperative users at the beginning of the collaborative work, and have the authority to merge the structured documents. In this system, the combination of document is a timing triggered process, that is, requests and merges the data before the intercepting point of each cooperative user automatically after a specific time interval. If other collaborative users (is not the master) want to merge the structured document, he must post merging request to the master, the merging operation can be executed only after the master passing the request.

The Definition of Document and Node

The definition of document structure follows the definition in the MCPS algorithm. The main replica is stored as a tree structure, and the sub parent relationship in the tree represents the hierarchy of the title nodes.

The Definition of Node Types

1. Title-Root-Node: we call it TRN, as the parent node of all first class titles, stored to constructs the tree storage structure, it can be null or store the name of the structured document.
2. Title-Node: called TN, it is the request node of collaborative client.
3. Str-Parent-Node: called SPN, it is the parent node of request node of collaborative client, and the parent node doesn't be requested by collaborative client, it just maintains the tree structure of copy and doesn't contain any text content.
4. Str-Brother-Node: simply called SBN, it as the brother node of request node of collaborative client, and the brother node doesn't be requested by collaborative client, it likes Str-Parent-Node does not contain detail text content.

The Definition of Node Attributes

We add two new node attributes based on the basic definition of node attribute in the last paper:

1. CE: Current Editors Numbers, the number of collaborative users currently participating in document editing under the title node.
2. TC: Title Creator, it is in each title node to record the creator of this node, in order to be used in the process of requesting judge and combining the document.

The Definition of Operation

1. Append (N): Locally incrementally requesting data for each node's Title Creator in the node set N.
2. Delete (id, site): collaborative site delete the one node which has specify id.
3. InsertTitle (parentId, id, site, data, name): insert a new title node after the tail of child node which the parent id is parentId.

4. InsertTitleBefore (parentId, refId, id, site, data, name): insert one new node that id is id, title name is name and text content is data front of the refId which is in the child node queue of parentId.
5. UpdateName (id, newName, oldName, site): updating the oldName of node which identifier is id to newName.
6. UpdateData(id, newData, oldData, site): updating the old text content oldData of node which identifier is id to newData.

4 The MCPS2 Algorithm

4.1 The Request Node Process of Collaborative Sites

We provide a node attribute called TC (the creator of the title node) to mark the collaborative site who insert this title node. Suppose in Fig. 3, collaborative site M2 requests for editing the title node set P = [p3.1, p4.1], the TC value of title node p3.1 and p4.1 are the same as M1, so post request node to site M1, the detail operating process please refer to Algorithm 1. After receive the request node operation at site M1, determine whether the requested title node has reached the upper limit of the number of editors. If so, refuse the request, otherwise the request has to wait for the judgment of all collaborative editors of this node. And the TC has a vetoed right, if TC pass the request, the other sites in the CE has to arbitrate this request, and we give a time limit in the arbitrate process, if overtime, the site gives up the arbitrate right, judgement standard is: comparing with the sum of NLV (node liveness) at agree sites and refuse sites, the result of high node liveness is effective, in summary, more contribution higher arbitration power. The detail arbitration process refer to Algorithm 2.

```
Algorithm 1. RequesetNodes(P)
for $i \leftarrow 0$ to P.length do
    request for site of P[i].TC
end for
```

Algorithm 2. Arbitration(id):n /*this site is M，the node set is N */

```
for i←0 to N.length do
  if (N[i].id==id) then
    n←N[i]
  end if
end for
if (n.CE.length>=MaxCE) then
    post refuse to the request site
    return n←null
else
  if (site M agree)
    for (j←0 to n.CE.length) do
      Ask for arbitration from collaborative sites in n.CE[j]
      if (n.CE[j] agree) then
        NLVa←n.CE[j].NLV
      else if (n.CE[j] refuse) then
        NLVr←n.CE[j].NLV
      else
        j←j+1
      end if
    end for
  else
      post refuse to the request site
      return n←null
  end if
end if
/*comparing of node liveness*/
if (NLVa>= NLVr) then
  n.CE.add(M)
  post node n to the request site
else
  post refuse to the request site
  n←null
end if
return n
```

4.2 Permission Transfer of Master

Before master withdraws, he needs to decide to transfer the master permission to which site, the basis of selection of the next master is: “In a tree structured document, the site with the highest liveness and the largest contribution will be the next master”, that is, which site has the highest tree liveness TLV on the title root node DOC is selected as the next master.

```
Algorithm 3. masterChange($M_1$ , Str): M
sites ← Str.DOC.CE
result ← -1
if master exsit then
  for i ← 0 to sites.length do
      result ← Max(result , Str.DOC.sites[i].TLV)
  end for
  for j ← 0 to sites.length do
      if (Str.DOC.sites[j].TLV==result) then
        M ← sites[j]
        break;
      end if
   end for
end if
grant privilege to site M
```

5 An Example

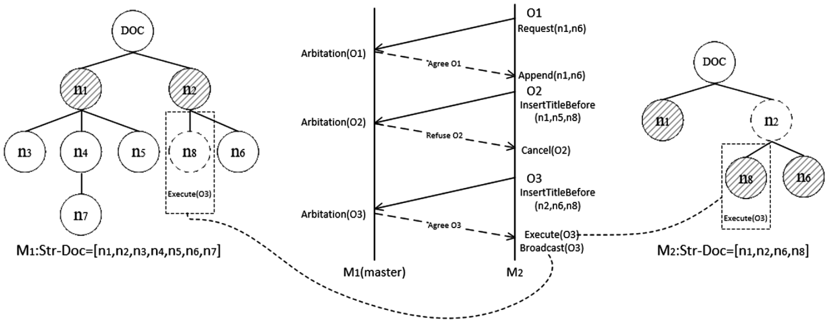


Fig. 3. The example of requesting nodes at M1 and M2

In Fig. 3 contains two collaborative sites M1 and M2, and the M1 sponsor the collaborative editing work of structured document as the master, it provide the initial copy Str-Doc = [n1, n2, n3, n4, n5, n6, n7], M1 is the creator of all nodes in Str-Doc, so the value of TC of the nodes in Str-Doc is M1. The operation O1 generated at site M1 only request to edit the title node n1 and n2, so n1.CE = [M1], n2.CE = [M1].

M2 Requests to Execute O1 = Request (n1, n6): M2 participates in the collaborative work, and request node set N = [n1, n6] from master, because the TC of n1 and n6 is M1, M1 arbitrates O1 directly, the detail arbitration process as follows:

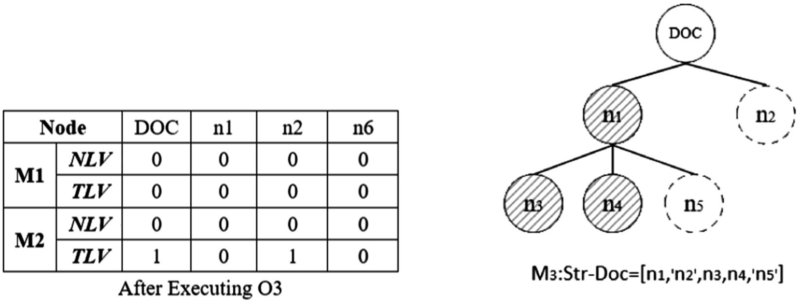
Step 1: Determining whether the number of edited users in the CE of each of the two nodes reaches the upper limit, if not, accepting arbitration of M1, M1 agrees to send node data to M2.
Step 2: M1 agrees and request for arbitration of the site in CE of n1 and n6, because n1.CE = [M1], n6.CE = [], the arbitration is passed by default.
Step 3: After executing the step 2, M1 modifies the attributes of n1 and n6 locally, as n1.CE = [M1, M2], n6.CE = [M2].
Step 4: According to the partial replicated architecture, M1 will return the node set N' = [DOC, n1, n2, n6] to M2 in order to guarantee the tree structure of replica, and the n2 just contain the parameters of storage structure.

M2 requests to execute O2 = InsertTitleBefore(n1, n5, n8, M2, data, name): The detail execute process refer to follow steps:

Step 1: Firstly find the parent node of n5 is n1, so n1.TC = M1
Step 2: Post InsertTitleBefore request to M1
Step 3: M1 refuse the request and return the result to M2
Step 4: O2 become a waste operation

M2 requests to execute O3 = InsertTitleBefore(n2, n6, n8, M2, data, name):

Step 1: Find the parent node of n6 is n2, so n2.TC = M1
Step 2: post InsertTitleBefore request to M1
Step 3: M1 agrees and post arbitration request to other collaborative editors (the sites in n2.CE) who join in the n2 collaborative editing, that is request to M1, the arbitration is passed by default;
Step 4: O3 is executed at M1, and Str-Doc = [n1, n2, n3, n4, n5, n6, n7, n8], n8. TC = [M2], n8.CE = [M2] and update the value of NLV and TLV at these nodes, the value of NLV and TLV in changed node record as Fig. 4.



Node		DOC	n1	n2	n6
M1	*NLV*	0	0	0	0
	TLV	0	0	0	0
M2	*NLV*	0	0	0	0
	TLV	1	0	1	0

After Executing O3

Fig. 4. The TLV and NLV of at M1 and M2

6 Conclusion and Future Work

In this paper, we support a new intention maintenance algorithm of structure document based MCPS under mobile platform called MCPS2, uses the P2P network connection mode to construct the collaborative work sites. Creatively put forward definitions of site's master and title creator. The reasonable distribution of permissions about node editing, request operation, to maximum maintain the intention of multi-user. In order to satisfy the needs of collaborators' coming and going, a dynamic control mechanism of permission transfer is given, which ultimately enhances the significance of the completely collaborative work.

The following research work includes: (1) optimize the efficiency of the algorithm, improve the permission transfer control process; (2) combine the MCPS2 algorithm with the MCPS algorithm, develop a collaborative editing APP for supporting structured document's editing work; (3) introducing the editing content of image [17, 18], form and text in the application.

Acknowledgement. We would like to thank the reviewers, whose valuable critique and comments helped to improve this paper. Moreover, this work is supported by the Innovation and entrepreneurship project of University of Shanghai for Science and Technology under Grant No. XJ2018371, National Science Foundation of China (NSFC) under Grant No. 61202376 and 61572325, Shanghai Natural Science Foundation under Grant No. 17ZR1419100, the Open Project Program of Shanghai Key Laboratory of Data Science (No. 201609060003).

References

1. Ellis, C.A., Gibbs, S.J.: Concurrency control in groupware systems. In: ACM SIGMOD Conference on Management of Data, New York, pp. 399–407 (1989)
2. Sun, C.Z., Ellis, C.: Operational transformation in real-time group editors: issue, algorithms, and achievements. In: ACM Conference on Computer Supported Cooperative Work, Seattle, pp. 59–68 (1998)
3. Shi, M., Xiang, Y., Yang, G.: The Theory and Application of Computer Supported Cooperative Work. Publishing House of Electronics Industry, Beijing (2000)
4. Gu, N., Yang, J.M., Zhang, Q.W.: Consistency maintenance based on the Mark & Retrace technique in groupware systems. In: ACM SIGGROUP Conference on Supporting Group Work, Sanibel Island, pp. 264–273 (2005)
5. Gao, L.P., Fu, Q.Q., Gao, L.L., et al.: Research on consistency maintenance of right management operations in real time collaboration environments. Int. J. Database Theory Appl. **7**(3), 255–264 (2014)
6. Wang, D., Gao, L., Zhu, S.: Research on consistency maintenance in three-stage network supporting operation re-transferring. J. Chin. Comput. Syst. **2018**(1), 128–133 (2018)
7. Wang, D., Zhu, S., Gao, L.: Conflict resolution of structured document collaborative editing based on the partial replication architecture in cloud platform. In: Chinese Conference on Computer Supported Cooperative Work and Social Computing, pp. 221–224. ACM, Chongqing (2017)
8. Vidot, N., Cart, M., Ferrie, J., et al.: Copies convergence in a distributed real-time collaborative environment. In: Computer Supported Cooperative Work, New York, pp. 171–180 (2000)

9. Shao, B., Li, D., Gu, N.: A fast operational transformation algorithm for mobile and asynchronous collaboration. IEEE Trans. Parallel Distrib. Syst. **21**(12), 1707–1720 (2010)
10. Li, R., Li, D., Sun, C.Z.: A time interval based consistency control algorithm for interactive groupware applications. In: the 10th International Conference on Parallel and Distributed Systems, Newport Beach, pp. 429–436 (2004)
11. SketchPad. http://www.dynamicgeometry.com/
12. Zheng, Y., Shen, H.F., Sun, C.Z.: Leveraging single-user AutoCAD for collaboration by transparent adaptation. In: the 13th International Conference on Computer Supported Cooperative Work in Design, pp. 78–83 (2009)
13. Sun, C., Chen, D.: Consistency maintenance in real-time collaborative graphics editing systems. ACM Trans. Comput. Hum. Interact. (TOCHI) **9**(1), 1–41 (2002)
14. Ignat, C.-L., Norrie, M.C.: Draw-together: graphical editor for collaborative drawing. In: Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work, Canada, pp. 269–278 (2006)
15. Xia, H.H., Lu, T., Shao, B.: A partial replication approach for anywhere anytime mobile commenting. In: The CSCW 2014, Baltimore, Maryland, USA, pp. 15–19 (2014)
16. Ng, A., Sun, C.: Operational transformation for real-time synchronization of shared workspace in cloud storage. In: Proceedings of the 19th International Conference on Supporting Group Work, pp. 61–70 (2016)
17. He, F., Han, S.: A method and tool for human-human interaction and instant collaboration in CSCW-based CAD. Comput. Ind. **57**(8), 740–751 (2006)
18. Cheng, Y., He, F., Cai, X., et al.: A group undo/redo method in 3D collaborative modeling systems with performance evaluation. J. Netw. Comput. Appl. **36**(6), 1512–1522 (2013)