Real-Time Collaborative Annotation System Supporting Separation of Content and Annotation

Lili Gao[1(✉)], Tian Cheng[2], Liping Gao[2], and Dongbin Guo[2]

[1] College of Special and Preschool Education, Weifang University, Weifang 261061, China
snowgao@163.com
[2] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

Abstract. In earlier collaborative annotation system, the details of all comments were presented in the form of images and were marked directly on the contents. This approach may cause visual confusion no matter what style people choose to display annotations by overlaying contents or by overlapping other comments. This article introduces a new annotation display model to separate contents and annotations, and the MPSAC (Multi-processing and Separation of Annotation and Content) strategy was presented to achieve the consistency maintenance of different collaborative sites based on this model. With the foundation of controlling executive operations' effect and maintaining the consistency of annotations, the strategy discovers and resolves the collision problem among overlapped annotations and provide a better interactive experience for users. The feasibility and correctness of this strategy were verified by case analysis and CoNote model system at the end of this paper.

Keywords: Collaborative annotation · Conflict resolution · Consistency maintenance · MPSAC

1 Introduction

With the development of cooperative computing and network communication technology, teamwork has been broadly applied to handle complex transactions. In collaborative work, people have a higher demand for interactivity. Being a method to improve interactivity, collaborative annotation has been widely used in various fields such as decision making, product design, doctor consultation, online classroom discussion, etc. [1–11]. Texts, tables and other comments were displayed in the form of images and the previous collaborative annotation system distinguished comments by using different colors, increased accuracy by using a stylus and stored comments according to their classification to enhance the display effect of comments. However, none of these methods can absolutely solve the problem of comments' overlaying of the contents or the overlapping among themselves. For example, Eppler and Pfister [1, 2] has proposed using different colors to represent different identities and replacing different comments with different symbols during the discussion of a meeting. This method improves interactivity in some degree, but it can only be used in a small number of users with low mobility. The reasons are as follows: (1) The amounts of color limits the multi-level division of identities; (2) Too many symbols may not be quickly accepted by the users which lead to bad experience. Gorgan [3, 4] advanced using a stylus but not a mouse to select the part accurately which is to be annotated and to comment on the original image. But when there are too many comments, this method may affect the latecomers' view of the original content. Camba [6] described the collaborative annotations in the process of 3D design and proposed to store annotations according to the classifications of the comments but displayed them by hashing them in the original image. This raised a new problem—the overlapping among comments. Coustaty [11] described the process of taking electronic notes on the printed documents. Since different users use different sizes of the symbol when annotating the same object, they may be ambiguous when selecting the annotated object. Although there are varieties of collaborative annotation systems in the past, the user experience was not perfect. The reasons are as follows: (1) All comments are showed in the form of images. (2) All comments are stacked on the original image which leads to the overlapping among themselves.

This article introduces a new model to show annotations which separates contents and comments. And the content in a text form is taken as an example to be separated into keywords and invalid parts. The keywords are chosen and annotated by users as an annotation object and the consistency maintenance strategy MPSAC (Multi-processing and Separation of Annotation and Content) is proposed based on the new model that separates content and annotation. In MPSAC, each annotation added by the user will be uniquely identified and recorded by a timestamp and site priority. Only the valid annotation associated with a keyword which was generated earliest will be displayed and other relevant annotations will be hidden. In order to ensure that annotations from different sites are consistent and non-overlapping, algorithms like DeleteObject, UndoObject and SetAL are proposed in this paper. The DeleteObject and InsertObject algorithms ensure the consistency of all sites by controlling the executive effort of delete and insert operations and SetAL algorithm is used to determine and resolve the conflicts among annotations.

2 Annotation Document Model Supporting Separation of Content and Annotation

The participants of the collaborative annotations such as annotated objects, annotations, operations (Insert, Delete and Undo) and presentative annotations are re-modeled. The representation of annotation is a new concept which is put forward in this article and will be analyzed with text content in this article. The following gives the corresponding definitions.

Definition 1: Content Column(Wd): The content column is used to store various forms of content and the text content in different numbers of lines is presented according to Wd.

Definition 2: Annotation Column(Zd): The annotation column is used to store comments edited by users with a width of Zd. The annotation column provides a separated comment area for each line of the text content and each area only displays annotations of the corresponding line.

Definition 3: Annotated Object(AO): AOij represents the j-th annotated object in the i-th line. The line number and sequence number of one annotated object are unique, even if an annotation object group which contains keywords in different lines can only have one-line number.

Definition 4: Annotation(A): The annotation object is a square area containing descriptive content defined by users. It can be described by A = [Lu, Rd, C], where C is the content of the annotation, and Lu and Rd are both two-dimensional coordinates. Lu Indicates the annotation coordinate which is described as [Lu.x, Lu.y] located in the upper left and Rd indicates the annotation coordinate [Rd.x, Rd.y] located in the lower right. The position of the annotation can be judged by Lu and Rd. There are four conditions for the two annotation objects A1 and A2:

Condition 1(A1, A2): Indicates that there is an overlapping Angle between A1 and A2.
Condition 2(A1, A2): Indicates that two angles of A2 are contained in A1.
Condition3(A1, A2): Indicates that four angles of A2 are all contained in A1.
Condition4(A1, A2): Indicates A1 and A2 are not overlapped.

Definition 5. Operation Definition: The operations in this model is defined as: Ins(Ao, A, T, S), Del (AO, A, T, S), and Undo (O, S), where AO indicates the annotated object, A indicates the annotation, T indicates the generation time of the operation, and S indicates the status to record whether the execution effect of the operation is presented in the annotation column. It should be noted that when Del and Undo operate on the same annotated object after Ins, the Del operation has higher priority than the Undo operation, that is, the Del's status is stored in the operation sequence with the form of 1, and the Undo's is 0.

Definition 6. Active Width of the Annotation(Md): Based on the upper left and lower right vertices of the annotation, the left and right annotations which are the closest to the annotation but not overlapped with each other are found and indicated as A1 and A2. Then Md is defined as A2.Lu.x-A1.Rd.x. If there is no comment in the right of the annotation, that's is to say A2 doesn't exist, Md is defined as the distance from the right side of A1 to the right side of the annotation column, i.e. Md = Zd-A1.Rd.x.

Definition 7. Show Annotation(SA): Each annotated object AO has a series of comments. In which the earliest comment is generated as a comment presented to the user and this is the presentation annotation of the AO. There is at most one show annotation for each comment.

Definition 8. Comment Sequence(LA): The SA for each annotated object was held in the comment sequence. These comments are the earliest valid comments for the object being annotated. LAi is an array whose length is the same as the number of annotated objects in the i^{th} line of the content column. When LAi[k] is not empty, it indicates the show annotation of the k^{th} annotated object in the i^{th} line of the content column.

3 Consistency Maintenance Strategy

The new model puts forwards new requirement for the consistency of the collaborative annotation process. It not only requires the same annotations to be displayed at each site, but also requires annotations to be presented without overlapping. Therefore, the

MPSAC strategy divides the process of consistency maintenance of each site into two steps: 1. Control the execution process of various operations to ensure that annotation sequence of each station and the status of the operation in the history sequence are the same even if they are executed in different orders. 2. According to the same annotation sequence provided in step one, the SetAL algorithm will be used to discover and resolve the position conflict among annotations.

3.1 Processing of the Insert Operation

The process of the insert operation is relatively simple. When the operation O = Ins (AOij, A, T, S) comes, only the earliest valid comment will be found in the relevant annotation of the j^{th} in the i-th line: A1. If the generation time of A1 is later than that of A, A is selected as the show annotation of AOij and will be placed in LA, that is, LAi [j] = A. While if the generation of A1 is earlier than the generation of A, there is no effect on the A's comment list. Finally, if the operation O is placed in the HB in the order of generation time and placed in the HQ according to the execution time.

3.2 Processing of Delete Operation

When the delete operation O = Del(AOij, A, T, S) comes, the first two earliest valid annotations of the j-th annotated object in the i line should be found: A1 and A2. If A1 is the same as A2, it means that the show annotation is deleted, then we should adjust LAi[j] = A2 and set the flag S of the corresponding insert operation in HB to 0 at the same time. If A1 and A2 are different, we only need is to set the S of the corresponding operation to 0 and put the operation O into HQ. The corresponding algorithm is detailed as follows.

3.3 Processing of Undo Operation

When an Undo operation Undo(O, S) comes, the execution of it is determined by whether the target operation O is an Insert operation or a Delete operation. If O is an Insert operation, HB is traversed to find the corresponding insert operation O1 and the first two valid comments A1 and A2 which are corresponding to O.AOij. Then two conditions may occur: (1) If the corresponding insert operation is valid and the corresponding comment is a show annotation, that is, O1.S = 1 and O1.A = A1, the status of operation O is set to 0 and A2 is set to show annotation of AOij and be put in LAi[j]. After that, Undo(O, S) is stored in HB and HQ; (2) If the corresponding insert operation is valid but the corresponding comment is not a show annotation, ie O1.S = 1 and O1. A $\neq$ A1, change status of O1 to 0 and set HB and HQ. Then the operation O and Undo (O, S) will comprise a do-undo-pair; (3) If the concurrent operation is executed firstly which leads to the corresponding insertion operation to be invalid, that is, O1.S = 0. It only needs to maintain the relationships of the do-undo-pair in the HB and save the operation in the HQ. If O is a delete operation, it is necessary to find the corresponding delete operation del_1 and insert operation ins_1 in HB, where del_1 and ins_1 are operations aiming at annotation A in AOij, and the earliest valid annotation A1 which is corresponding to AOij will be obtained. (1) If the executed comment is the previous show annotation, ie A.T > A1.T, ins1.T is set to valid and the position of undo in HB is adjusted to guarantee the relationship of do-undo-pair and afterwards undo operation is stored in HQ. (2) Otherwise if A == A1 or AT < A1.T, ins1.T is set to valid and A is changed to the show annotation of AOij. At last the position of undo in HB is adjusted to ensure the relationship of do-undo-pair and save the undo operation in HQ.

3.4 Case Analysis

Now we use the above control algorithm to illustrate the execution effect of each site. When the HB and AL of all sites are the same, each site implements the consistency of the show annotation. Assume that the initial state of the i-th line of the comment column is ALi = {null, null …} and that O1 ~ O5 all process the keywords of the i-th line, where O1 = Ins(AOi1, A1, T1, 1), O2 = Ins(AOi1, A2, T2, 1), O3 = Ins(AOi1, A3, T3, 1), O4 = Del(AOi1, A1, T4, 1), O5 = Undo(O2, 1), and T1 < T2 < T3 < T4. The relationship among operations is shown in Fig. 1.

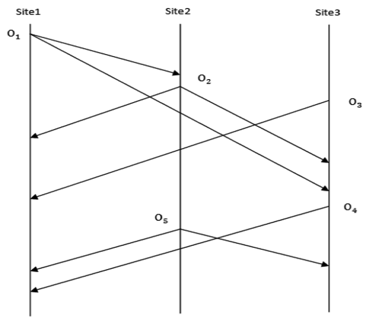


Fig. 1. Operation relationship

At Site 1, operation O1 is generated and executed immediately. Since A1 is the first comment of AOi1, A1 is placed into AL as a show annotation, that is, ALi[0] = A1. And O1 is placed into HB according to the order of generation time; When O2 arrives, since the generation time of A2 is later than the generation time of A1, O2 is directly put into HB in the order of generation time; the state and the execution mode of O3 and O2 are the same, so the operation sequence of the i-th line is HBi = [O1, O2, O3], and the comment sequence is ALi = [A1, null, …]; When O5 is executed, it is found that the comment carried by the operation is not the show annotation A1 of the annotated object AOi1, so it only needs to maintain the relationship between O5 and O2 of the do-undo-pair in HBi. At this time, HBi = [O1, O2, O5, O3], ALi = [A3, null, …]; The effect of O4 is to delete the show annotation of AOi1. Thus it is necessary to set the second earliest annotation A3 to show annotation, and set the status of O1 to 0 in the HBi to maintain the ins-del-pair relationship of O4 and O1. At this time in the annotation column HBi = [O1, O4, O2, O5, O3], ALi = [A3, null, …].

At site 3, the operation O3 is generated and executed immediately, so A3 is set as the show annotation of AOi1 and be put into the annotation sequence. When O2 arrives, since the generation time of the corresponding annotation A2 is earlier than A3, the show annotation of AOi1 is replaced by A2. Since A1 which carried by O1 is generated earlier than the valid annotations in the AOi1, when O1 arrives at station 1, A1 will be regarded as the show annotation and the operations in the HBi are maintained in chronological order; When O4 is generated, the two earliest comments A1 and A2, which contain the comment A1 that is to be deleted by O4 should be found. Therefore, it is necessary to set A2 as the show annotation of AOi1 and set the state of O1 to invalid and maintain the ins-del-pair relationship in HBi. The status of the comment column is HBi = [O1, O4, O2, O3], ALi = [A2, null, …]; When O5 arrives, the comment A2 carried by the operation to undo O2 is a show annotation, so the following valid annotation A3 is set to the show annotation, and the relationship of do-undo-pair is maintained. Finally, the status of the comment column is HBi = [O1, O4, O2, O5, O3], ALi = [A3, null, …].

By analyzing the execution status of the operations, it is found that each site has maintained the consistency of the annotations presented in the annotation column. The first part of the consistency in the collaborative annotation system is completed. The second part of the work will be introduced below.

4 Conflict Resolution

When the annotations of which are presented at each site need to be consistent, the follow-up of the collaborative annotation system is to present the annotations to the user in a friendly manner. The SetAL algorithm is introduced to resolve conflicts among show annotations to ensure that annotations are displayed in a non-overlapping way. In order to handle the overlap among annotations, there are two situations that need to be handled.

4.1 Conflict Analysis

4.1.1 Case 1: The Width of the Annotation Column Is Shorter Than Md

When the width of the annotation column is set as Zd = 945, there is an comment of AOi4, that is A1 = [[10, 10], [90, 90]] in the annotation column. At this time, there is an operation O1 which carriers the comment of AOi5, that is A2 = [[50, 50], [120, 120]]. Since A2 and A1 satisfy Condition1 (A2, A1), the result of directly executing O1 is that A2 and A1 are overlapped—as shown below. By calculation, A3.Rd.x-A3.Lu.x = 70, that is, the width of the annotation is 70 is shorter than Md, so simple translations can avoid overlapping of two annotations, as shown in Fig. 2. Here we resolve the conflict by translating the post-inserted comment horizontally to the right.

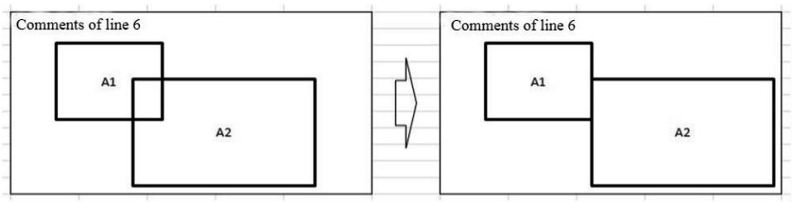


Fig. 2. Conflict resolution strategy in case that the width of the annotation column is shorter than Md

4.1.2 Case 2: The Width of the Annotation Column Is Longer Than Md

Suppose that the width of the annotation column is set to Zd = 945, and there are two annotations in the comment column: A1 and A2, which belong to different keywords, where A1 = [[10, 10], [90, 90]], A2 = [[150, 20], [230, 10]], there is a comment A3 = [[50, 50], [120, 120]] for ALi6. Since A3 and A1 satisfy Condition1 (A3, A1), we know that if A3 is inserted directly, overlap will be caused between A3 and A1. It is found by calculation that A3.Rd.x-A3.Lu.x = 70 > Md, so the lateral translation cannot eliminate the conflict. At this time the adjustment of width should be taken into consideration. In the adjust process, we have to ensure that the area of the annotation being adjusted does not change, so that it will not affect the content of the show annotation and eliminates the overlap problem among annotations (Fig. 3).

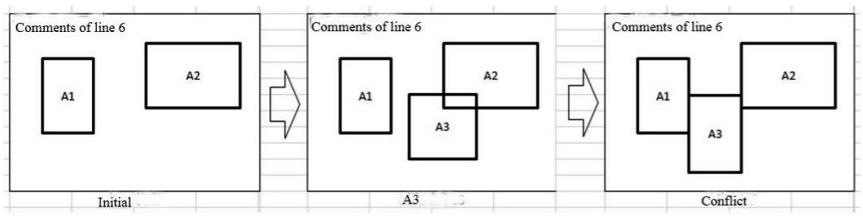


Fig. 3. Conflict resolution strategy in case that the width of the annotation column is longer than Md

4.2 Conflict Resolution Scheme

In order to avoid overlapping among annotations, the calculation of an annotation's Md should be done firstly before A is put into the annotation column. If Md is longer than the width of A and there is no overlap, A is placed directly; if the annotation placed previously has a position overlap with A and Md is longer than the width of A, the placement position of the annotation A will be adjusted; if the annotation placed previously has a position overlap with A and Md is shorter than the width of A, the shape of A is adjusted under the condition that the area of annotation A will not be unchanged. The corresponding implementation algorithm is as follows:

```
Function SetAL($AL_i$){
1  int Flag;
2  $A_l = A_r$ = null;
3  For(int i; i<$AL_i$.size; i++){
4      Md = GetMd($AL_i$[i],$AL_i$,$A_l$,$A_r$);
5      IF(A = = null &&$A_r$ = = null){
6        Set(AL[i]);
7      }ELSE{
8        Set(A` = TranAnnotation($AL_i$[i],Md));    //set transformed annotation A
```

```
    }
  }
}
//adjust the location and shape according to A's Md and conflict annotation
Function TranAnnotation(A,Wid,$A_l$,$A_r$){
  IF(Wid > (A.Rd.x - A.Lu.x)){
      IF($A_l$ = = null){
        int len =A.Rd.x - $A_r$.Lu.x;
        A.Rd.x += len;
        A.Lu.x +=len;
    }ELSE{
      int len = A.Lu.x - $A_l$.Rd.x;
      A.Rd.x += len;
      A.Lu.x += len;
    }
  }ELSE{  //Step 1. Change the shape
    int Area = (A.Rd.x - A.Lu.x)*(A.Lu.y - A.Rd.y);
    int high = Area/Wid + 1;
    A.Rd.x = A.Lu.x + Wid;
    A.Rd.y = A.Lu.y + high;  //Step 2. Change the location
    int len = A.Lu.x - $A_1$.Rd.x;
    A.Rd.x += len;
    A.Lu.x += len;
  }
  Return A;
}
//Find annotation A's Md
Function GetMd(A,$AL_i$,$A_l$,$A_r$){
  int left = 0;
  int right = Zd;
  For(int i; i < $AL_i$.size; i++){
    IF(A≠$AL_i$[i]){
    IF($AL_i$[i].Rd.x < A.Rd.x && $A_l$.Lu.x < $AL_i$[i].Lu.x){
      left = $AL_i$[i].Rd.x;
      $A_l$ = $AL_i$[i];
      IF($AL_i$[i].Lu.x > A.Lu.x && $A_r$.Lu.x > $AL_i$[i].Lu.x){
        right = $AL_i$[i].Lu.x;
        $A_r$ = $AL_i$[i];
        }
      }ELSE{
        Break;
      }
    }
  Return right - left;
  }
}
```

4.3 Control Algorithm

As mentioned above, the MPSAC policy requires that each operation is performed firstly. And then, under the premise of the consistency maintenance of the annotation sequence, the selected annotations are placed without overlap, and finally the sites are displayed identically.

```
Function Execute(O){
  IF(TypeOf(O) = = ins){
    InsertObject(O);
  }ELSE IF(TypeOf(O) = = del){
    DeleteObject(O);
  }ELSE{
    UndoObject(O);
  }
  SetAL[$AL_i$];
}
```

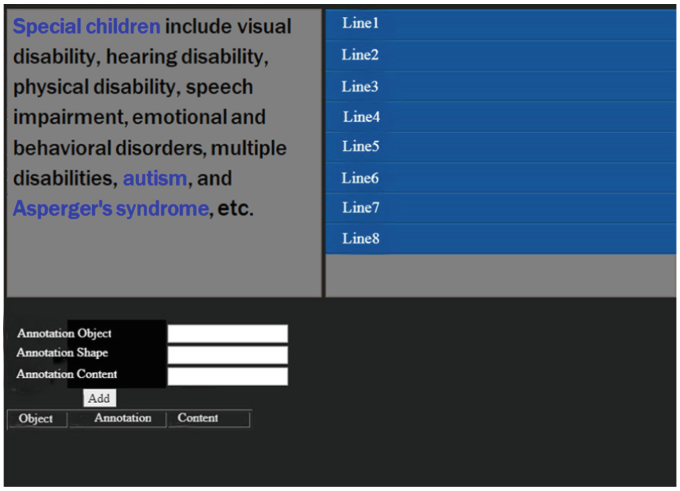
5 CoNote Model System

In order to verify the correctness and feasibility of the proposed algorithm, a collaborative annotation system CoNote that supports separation of content and annotation has been proposed. The goal of the system is to allow different users in distributed locations to operate comments on the text content in real time. After performing the same operation, the annotations of each site are not overlapped and displayed identically.

The CoNote system is based on the full-duplex communication WebSocket protocol and is developed by using HTML, JavaScript, CSS, and Java. The CoNote system is not a complete replication architecture. Each client in the CoNote system keeps a copy of the document, and the operations generated by each client are sent to the server. The server is regarded as a special client node, that is, the server will only synchronize and transmit operations sent by other clients without generating new operations.

On the client side, the system needs to complete three tasks. Firstly, it accepts operations generated by the user and execute them. Secondly, it establishes a connection to the server and receives the operation transmitted by the server and send to the local operation to achieve data synchronization. Thirdly, it performs operations synchronized from the server. The server serves as the end point of a data summary to synchronize the operations between the local and current connected clients' data.

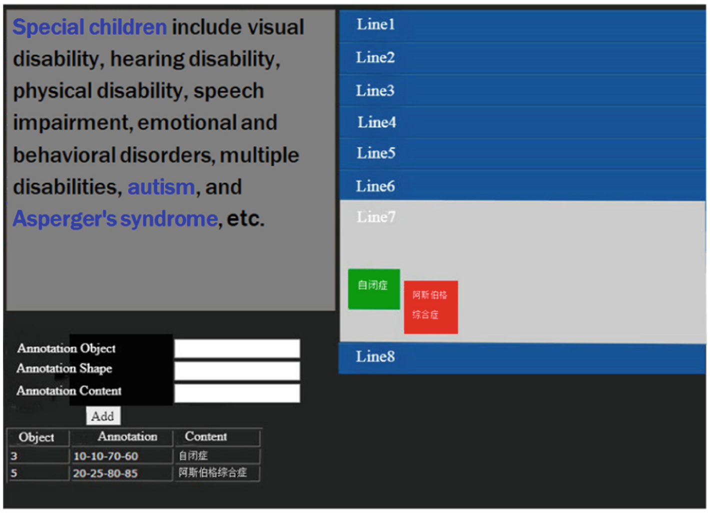
The following figure shows the user interface of the CoNote system. Figure 4 is a case where the user does not select the annotation space, where in the upper part is the display space and the lower part is the user operation space; Fig. 5 is a show annotation image in which the user selected the comment space.



Special children include visual disability, hearing disability, physical disability, speech impairment, emotional and behavioral disorders, multiple disabilities, autism, and Asperger's syndrome, etc.

Object	Annotation	Content

Fig. 4. System status before user select



Special children include visual disability, hearing disability, physical disability, speech impairment, emotional and behavioral disorders, multiple disabilities, autism, and Asperger's syndrome, etc.

Object	Annotation	Content
3	10-10-70-60	自闭症
5	20-25-80-85	阿斯伯格综合症

Fig. 5. System status when user select

6 Conclusion

There are a lot of collaborative annotation systems in the past, but they don't bring a good user experience. This is because these systems didn't notice the overlap problem between content and comments and overlap among comments. In addition, Undo functions are not supported by those collaborative systems. In response to the above three questions, the participants of collaborative annotation systems are re-modeled and separation between content and annotation is put forward as well as the consistency strategy MPSAC. To store content and annotation separately is to store content and comments in two separate spaces and divide the content into keywords and irrelevant content. The MPSAC strategy decomposes the process of the latter two problems into maintain annotation sequence consistency and resolve annotation conflict. Maintaining annotation consistency is ensuring the consistency of each site by preserve the consistency by using generation time, insert, delete and undo algorithms under the condition of only one show annotation of each annotation object is represent to the user. Ins-del-pair, do-undo-pair and their priorities' relationships are introduced in this paper to simplify the maintenance process of the operation sequence. With the new annotation model, the conflicts between annotations are mainly in two-dimensional space. The conflict resolution among annotations is to use the SetAL algorithm to handle the position conflict among annotations on the premise that the annotation sequences selected by each site are consistent.

In the collaborative annotation environment, the MPSAC strategy can achieve the site consistency within three types of operations and ensure the non-overlapping placement among the annotations. However, the SetAL algorithm proposed in this paper can only solve the problem of horizontal conflict among annotations. This method of resolution does not guarantee that the annotations are compactly placed in the comment column. In other words, there may be a vertical position between the annotations presented wasted.

Collaborative annotations have brought great convenience to users in online discussions, but there are still many problems in collaborative annotation. It is believed that with these problems optimized, the collaborative annotation system will play a more important role in people's work.

References

1. Eppler, M.J., Pfister, R.A.: Drawing conclusions: supporting decision making through collaborative graphic annotations. In: 14th International Conference Information Visualisation, London, UK, pp. 369–374. IEEE Computer Society (2010)
2. Eppler, M.J., Kernbach, S.: Dynagrams: enhancing design thinking through dynamic diagrams. Des. Stud. **47**(11), 91–117 (2016)
3. Gorgan, D., Stefanut, T., Gavrea, B.: Pen based graphical annotation in medical education. In: 12th IEEE International Symposium on Computer-Based Medical Systems, Maribor, Slovenia, pp. 681–686. IEEE (2007)
4. Smit, N., Hofstede, C.-W., et al.: The online anatomical human: web-based anatomy education. In: 37th Annual Conference of the European Association for Computer Graphics, Lisbon, Portugal, pp. 37–40. Eurographics Association, Aire-la-Ville (2016)
5. Gao, L., Xu, X.: A new algorithm for real-time collaborative graphical editing system based on CRDT. In: Sun, Y., Lu, T., Xie, X., Gao, L., Fan, H. (eds.) ChineseCSCW 2018. CCIS, vol. 917, pp. 201–212. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-3044-5_15
6. Camba, J.D., Contero, M., et al.: On the integration of model-based feature information in Product Lifecycle Management systems. Int. J. Inf. Manage. **37**(6), 611–621 (2017)

7. Coburn, J.Q., Salmon, J.L., Freeman, I.: Effectiveness of an immersive virtual environment for collaboration with gesture support using low-cost hardware. J. Mech. Des. **140**(4), 1–9 (2018)
8. Goy, A., Magro, D., Petrone, G., Picardi, C., Segnan, M.: Ontology-driven collaborative annotation in shared workspaces. Future Gener. Comput. Syst. **54**(1), 435–449 (2016)
9. CheeWyai, L., Cheah, W., Chowdhury, A.K., Gulden, C.: Engineering sustainable software: a case study from offline computer support collaborative annotation system. In: 9th Malaysian Software Engineering Conference (MySEC), Kuala Lumpur, pp. 272–277 (2015)
10. Poster, S.R.: Interactive and collaborative source code annotation. In: IEEE International Conference on Software Engineering, Florence, Italy, pp. 799–800. IEEE (2015)
11. Coustaty, M., Sidere, N., Ogier, J.: Augmented documents for research contact management. In: 4th International Forum on Research and Technology for Society and Industry (RTSI), Palermo, pp. 1–6. IEEE (2018)