Awareness Model to Overview Modifications in Collaborative Graphical Authoring Tools

Stavroula Papadopoulou
Department of Computer Science
ETH Zurich
CH-8092 Switzerland
papadopoulou@inf.ethz.ch

Claudia-Lavinia Ignat
INRIA Nancy Grand Est
Campus Scientifique
F-54506 France
ignatcla@loria.fr

Moira C. Norrie
Department of Computer Science
ETH Zurich
CH-8092 Switzerland
norrie@inf.ethz.ch

ABSTRACT

Although group awareness is an important factor in collaboration, there has been relatively little work to date on ways to compute and visualise summaries of changes made using graphical authoring tools. We propose an awareness model where changes can be presented at various document levels such as pages, layers, groups of objects and objects. We also present metrics and visualisation techniques suitable for large scene of objects. The underlying model is general and could be applied to authoring tools for any form of hierarchical documents including text as well as graphical documents.

Author Keywords

Change awareness, collaborative graphical editing, awareness model.

ACM Classification Keywords

C.2.4 Computer-Communication Networks: Distributed Systems—*Distributed applications*; D.2.2 Software Engineering: Design Tools and Techniques—*User interfaces*; H.5.3 Information Interfaces and Presentation: Group and Organization Interfaces—*Asynchronous interaction, Synchronous interaction, Computer-supported cooperative work*; H.1.2 Models and Principles: User/machine Systems—*Human factors*

INTRODUCTION

Many tasks in business and academia involve groups of individuals working together to achieve some common goal. Not surprisingly, the development of collaborative authoring tools therefore became a central topic of interest in the CSCW (Computer Supported Cooperative Work) community. According to the synchronicity of writing activities, collaborative authoring tools can be classified as either synchronous or asynchronous. Synchronous authoring tools, also referred to as real-time collaborative authoring systems, imply that changes performed by one user are immediately transmitted to other group members. Asynchronous authoring tools allow users to work in isolation and synchronise their changes at some later point in time.

Group awareness is an important factor of successful collaboration. In [2] awareness is defined as an "understanding of the activities of others which provides a context for your own activity". Various awareness mechanisms have been proposed for the coordination of user activities in collaborative environments, such as multi-user scrollbars [1], telepointers [13] and radar views [4]. Approaches for highlighting changes made by other participants over time, known as change awareness [10, 11] mechanisms, have also been proposed. Most change awareness approaches concentrate on collaborative text authoring, while less research focuses on tracking changes done in the domain of graphical authoring. An examination of existing change awareness approaches in graphical authoring tools, reveals that none of them attempts to compute overall information about user changes on a graphical scene of objects and visualise this information in a lightweight fashion.

For a collaborative system, it should be possible to provide users with awareness information customised according to their preferences. Additionally, awareness information should be provided at various document levels. For instance, it would be useful to have a quick overview of the amount of changes done at the level of pages and then have a detailed overview of the changes done on a particular page or on a certain component of a diagram. Awareness information should also be filtered according to specific types of actions such as formatting operations (colouring, resizing, etc.) or creation/deletion of objects to deliver to the users only the required information. Finally, an awareness mechanism should also include a filtering mechanism that enables the visualisation of modifications made by a specific user or set of users.

In [9], we proposed an awareness mechanism to visualise "hot areas" of a text document as well as who changed various parts of the document and, at the same time, provide quick access to the modified document parts. Metrics were proposed to represent the amount of changes on specific document levels such as paragraph, sentence, word or character.

In this paper, we propose an awareness mechanism that provides an overview of the amount of changes performed in collaborative graphical authoring system based on the same ideas that we used for text authoring tools. The structure of the document is key to supporting customisation of the awareness mechanism. We consider graphical documents as organised into pages, layers and groups forming a hierarchical structure as described by Figure 1. As in the case of text documents, different entities of the shared workspace are de-

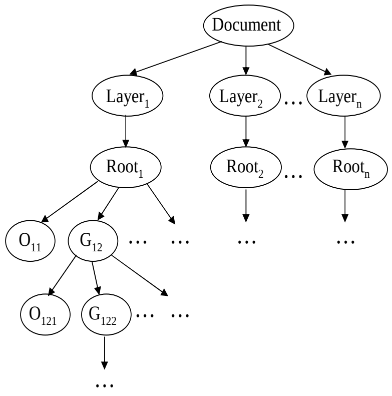

Figure 1. A hierarchically structured graphical document

fined at different levels and users can perform operations on these entities. Moreover, as in the case of text documents, we want to compute and represent the amount of changes associated with each document level in order to provide a change awareness overview at any document granularity. We propose a model for awareness and show its generality in terms of the types of the shared documents to which it can be applied, namely graphical and text, and in terms of the flexibility of adding new metrics and visualisation tools to provide the overviews of performed changes. Further, our approach is suitable for both synchronous and asynchronous collaboration.

In the next section, we start with a presentation of our model describing the computation and visualisation of awareness information in collaborative authoring tools. We then present our generic model and describe how the general concepts that can be applied to both text and graphical documents have been specialised for the case of graphical applications. After a section relating our work to existing awareness approaches, we finish with some concluding remarks.

AWARENESS FRAMEWORK

The awareness mechanism that we propose is an adapted version of the awareness framework that we already use for text collaborative authoring systems [9]. In this section, we present and discuss the framework's metamodel shown in Figure 2. It is composed of a core model shown in Figure 2(a), which is the same for both text and graphical applications, and the specialisation of its concepts for graphical authoring tools shown in Figure 2(b-d). The basic concepts are the *entity*, the *operation*, the *value*, the *metric* and the *visualisation tool*. Alongside our descripion of the concepts and how they are extended for graphical tools, we also discuss the similarities and differences between text and graphical authoring tools and some issues that emerge.

Our awareness mechanism uses the document hierarchy to compute and present, at different granularity levels, richer awareness information than existing solutions. Changes made to a document, in the form of operations, are tracked and their effect is attached to the object to which they are applied as well as to its ancestors in the document hierarchy. We describe this effect with the concept of a *value*, as presented in Figure 2. An object has a *value* for each type of operation applied to it. The numerical value of each *value* object further depends on different *metrics* applied. Using the document hierarchy, the *values* of an object's children can be summed and added to the object's *value*.

An ***entity*** represents any form of document, document part or document container that can be edited collaboratively. Different categories of *entities* can be defined for the various document types used in authoring systems, for instance text and graphical systems. *Entities* are linked together through parent-child relationships to build document structures. To allow for various document structures, we decided not to model the parent-child relationships as associations between *entities*, but rather through a separate concept, the ***structurer***.

A *structurer* holds all the information needed to build the document's structure when required. Its main functionalities are the addition of parent and child-*entities* to a given *entity*, as well as the retrieval of this information when required. The possibility for an *entity* to have any number of parents and children enables the creation of any document structure and renders our awareness framework applicable to a large number of authoring systems regardless of the document structure that they use. An *entity* is related to exactly one *structurer* through the association *hasStructurer*. Finally, a *structurer* keeps extra information about any application specific restrictions applied when creating the structure of the document. For instance, in a graphical authoring system with a hierarchy defined as in Figure 1, a layer's parent-*entity* is always a page, while its child-*entities* are always graphical objects.

Entities defined in Graphical Authoring tools: The concept of an *entity* is further specified for graphical documents as shown in Figure 2(b). The notions of a ***page***, a ***layer*** and a ***graphicalObject*** are defined as specialisations of an *entity*. A *graphicalObject* can either be a ***simpleObject*** or a ***group***. As *simpleObjects*, we define any of the objects that a graphical collaborative authoring tool supports, for instance an ***ellipse***, **a *rectangle*** or a ***textbox***. A *group* consists of a set of any *graphicalObjects*, as specified by the association *consistsOf*, thereby also allowing the creation of subgroups. Through the information kept in the structurer, the hierarchy of Figure 1 is created in a similar manner to the hierarchy of paragraphs–sentences–words–characters defined in text documents. Finally, attributes are assigned to each *simpleObject* as an ***attributeSet*** through the association *hasAttributeSet*. For example, an attribute can be the object's ***colour***, ***text*** in case of a textbox, ***annotation***, ***angle*** representing the rotation angle of an object, or the object's *xy* position in the 2D scene of objects, denoted as ***position***.

An ***operation*** represents any action done by a user. As presented below, we consider all the ways in which users can interact with a document that result in modifications of a document part's attributes or the document structure at any level of granularity. Every *operation* is related through the association *createdBy* to exactly one *user* that created the opera-

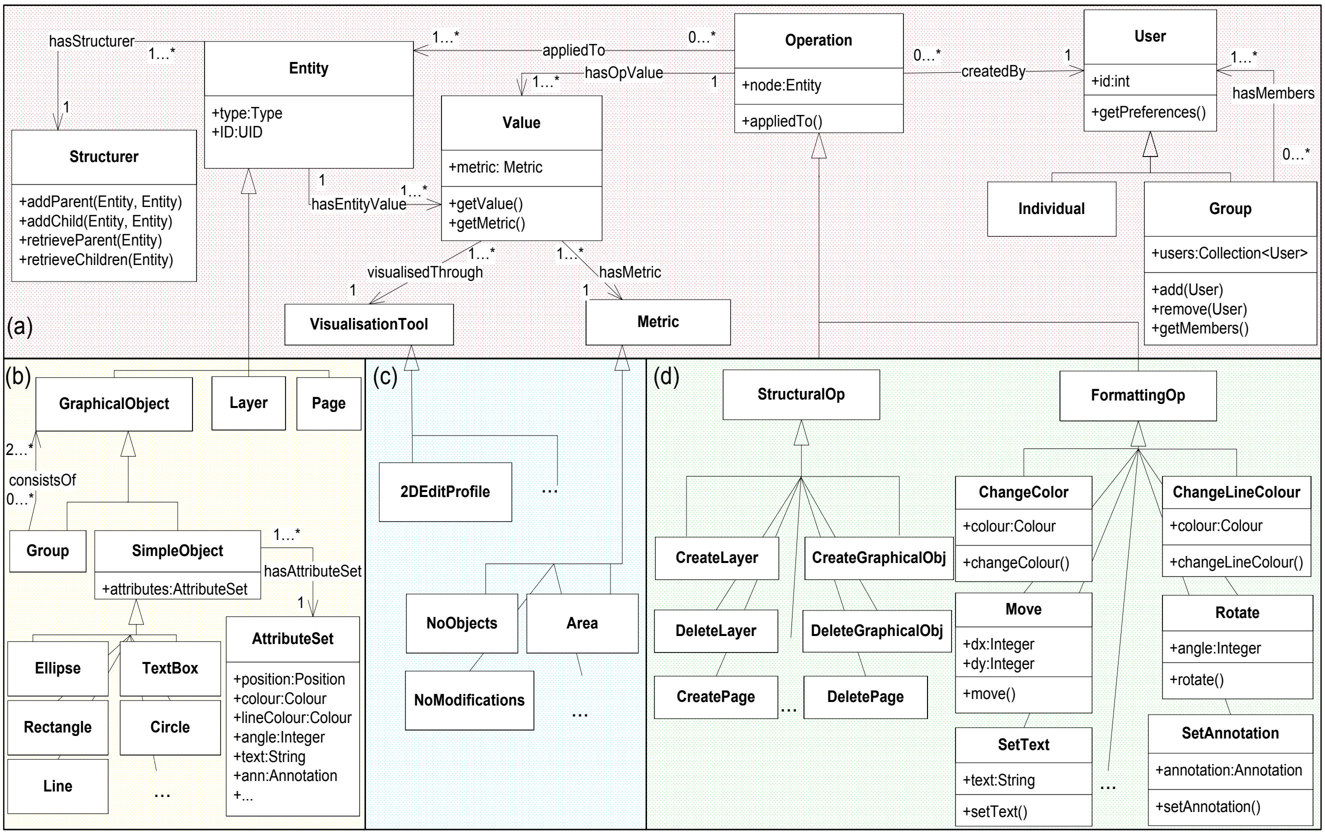

Figure 2. The awareness mechanism we propose. The core model (a) includes the key concepts. The specialisations of them for the case of a graphical collaborative tool are shown in b,c,d.

tion, i.e. made the document modification. A ***user*** can either be an ***individual*** or a ***group*** of *users*, allowing in this way the notion of subgroups as well. Finally, every *operation* is related through the association *appliedTo* to the entities where it will be applied.

Operations defined in Graphical Authoring tools: We classified the operations applied to document entities in graphical applications into ***structural*** and ***formatting*** operations as shown in Figure 2(d). *Structural* operations modify the document structure. Creation or deletion of an *entity* are typical examples of *structural* operations. Other *structural* operations could also be defined. *Formatting* operations modify an *entity's* attributes. ***Move***, ***changeColour***, ***rotate***, ***setAnnotation*** and ***setText*** are only some of many *formatting* operations defined in our model to describe collaborative activity in graphical authoring tools. The set of operations used in the text authoring tool that we enhanced with an awareness mechanism [9] consists only of structural operations that create and remove entities. The much richer set of operations defined for graphical documents in comparison to the set of operations defined for text documents, renders the computation and visualisation of awareness information more complex in graphical applications.

The concept of a ***metric*** is also included in the framework to serve as a means of measuring an *operation*'s effect, i.e. an *operation*'s ***value***. Since modifications are made to *entities* through *operations* applied to them in both text and graphical documents, the concept of *metrics* is defined for both document types in a similar manner. An *operation*'s *value* depends on the *metric* used. In text editors, for instance, a *deleteSentence* operation delivers different information, i.e. has different value, when a user is interested in the total number of deleted sentences or the total number of deleted characters in a document. Therefore, an *operation* is related through the *hasOpValue* association to multiple *value* objects equal in number with the number of *metrics* defined in the model. An *entity* is related through the association *hasEntityValue* to multiple *value* objects as well. Since *operations* are applied to *entities*, the total number of operations will describe the total number of modifications made to an *entity*. By summing the *values* of all the *operations* applied to an *entity*, the entity's own *value* is computed. Finally, the computed *values* can be filtered and only the interesting ones, as defined by the current user, will be visualised through the appropriate ***visualisation tool***. Note that an entity's value and an operation's value are both modeled under the same concept of a *value* to allow for a flexible implementation. However, we later present the definition of each of them separately as well, based on the metrics we propose.

Metrics and visualisation tools defined in Graphical Authoring tools: Our proposition of *metrics* and *visualisation tools* for graphical documents was influenced by some differences between text and graphical documents. While a hierarchy can be defined in both types of documents, in the case of text documents, the children of an *entity* are ordered. Therefore, any *entity* can be uniquely identified by a path in the document hierarchy. However, in graphical documents, *entities* are not ordered. For instance, there is no order between objects belonging to the same group of objects. Objects are therefore identified by unique identifiers. Taking this into account, as well as the need for a visualisation tool that offers an overview of the modifications made over the scene of objects, we chose a 2D visualisation tool. More details on the tool are given later.

Further, our proposal of a metric would be to count the number of modifications made at each *xy* position of the 2D scene. Since modifications to a document or document part are made in the form of operations applied to *graphicalObjects* and not to *xy* positions, the number of modifications at a specific *xy* position is identical to the number of modifications made to the *graphicalObjects* that occupy the corresponding position. Therefore, the metric is defined as the number of modifications made to a *graphicalObject*. This justifies the metric ***noModifications*** shown in Figure 2. Note however, that there exist various types of operations applied to *graphicalObjects*. To compute and present awareness information as required according to users, focus and granularity, we allow the computation of *entity values* for each of the operation types. As a result, a user interested in only a specific type of changes could visualise the corresponding *values*, withour being distracted by *entity values* that reflect other types of changes.

Definition 1. Based on the metric presented above, the value $opValue$ of an operation op of any type, is simply defined to be equal to 1 since it represents one modification. $opValue(op) = 1$

Definition 2. In a similar manner, the value $entityValue$ of an entity N for a given type of operation $opType$ is defined as the sum of the $opValues$ of the set of operations, denoted as $operations(N)$ applied to the entity, all of them being of type $opType$.

$$entityValue(N, opType) = \sum_{op_i \in operations(N)} opValue(op_i)$$

where $op_i(type) = opType$

The definition of appropriate *metrics* is a very demanding task due to its dependence on user preferences. We further intend to enrich the proposed metrics, possibly by conducting user studies and collecting the users' feedback. Examples of new possible metrics could be the number of objects affected by an **operation**, as well as the area in the 2D scene that will be modified. Both are shown in Figure 2(c) as ***noObjects*** and ***area***. It would also be interesting to investigate ways of combining the two metrics and their effectiveness. A further example could be the definition of metrics similar to the divergence metric approach proposed in [8] to measure the conflicts between concurrent operations. For instance, we could measure the number of conflicting operations applied to an object or to a region in the 2D scene of objects.

Finally, we present our proposition of a visualisation tool for awareness in graphical authoring tools. It is presented in Figure 3 and in Figure 2(b) as *2DEditProfile*. The proposed tool is based on the concepts of heat maps [3] to provide a highlighting mechanism suitable for large documents. We consider a 2D layer drawn upon the 2D scene of objects. The area around each *graphicalObject* is coloured according to its *entityValue* computed for a specific type of operation applied to it. If the user wishes to visualise the effect of many different kinds of modifications, i.e. different types of operations, then the sum of the object's *entityValues* is computed and the area around the object is coloured based on this value. Since an object's colour is one of its attributes and can be modified by users, the area inside a *graphicalObject* is not affected by the visualisation tool. The greater the number of modifications of each type made to the object, the greater the corresponding object's *entityValue* will be and the darker the colour used to highlight the area around the object. The tone of the colour is more intense right next to the object, while it gets less intense further from it.

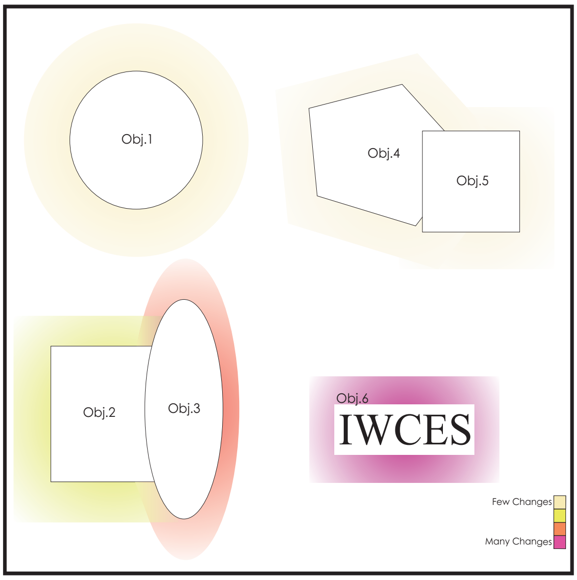

Figure 3. Different colour tones and intensities are used to denote different number of modifications made to objects. From less to more changes, the colours used are yellow, green, red and purple.

Consider, for instance, the example of Figure 3. Objects 1 and 6 have a different number of modifications made to them. Object 1 has fewer modifications, hence the colour around it is lighter than the one around object 6. Group objects are specially handled. If all objects in a group have undergone the same number of modifications, the colour around them is uniform as with objects 4 and 5. On the contrary, if the number of modifications of each of the group's objects differ, then different colours are used to highlight the areas around the objects. At the intersection of these areas, the colour is a combination of both colours. In this example, object 3 has undergone more modifications than object 2.

We have presented a visualisation tool that offers an overview of modifications made on a 2D scene of graphical objects. Most of the time, such a scene represents a page of a graphical document. We are also investigating how the tool should be extended to present awareness information about modifications made to objects in various layers, or to present more than one page at a time. Additionally, we intend to slightly change the use of colours in the visualisation tool by assigning specific colours to specific kinds of changes. This will help users easily distinguish the type of a change only based on the colour used for highlighting the modified object.

Finally, considering the implementation of the awareness framework we proposed, we intend to offer users the possibility of filtering the computed awareness information by selecting the type of modifications they wish to visualise. In a similar way to the implementation of this awareness mechanism for text documents, we intend to implement a flexible GUI that will enable users to select any combination of operation types to be visualised, at any granularity level. For instance, a user will be able to visualise only deletions of objects that belong to a specific layer of a specific page, or all of the rotated objects of a given set of pages.

RELATED WORK

In this section, we present existing change awareness approaches in the domain of graphical editing and compare them with our approach. Microsoft Word offers rich awareness support for text editing, while awareness about modifications of images is very limited. The model used for the representation of documents is a linear one, i.e. the document is seen as a sequence of characters and images. Although characters inserted or deleted are tracked, modifications of images are unfortunately ignored.

Rational Rose provides a UML editor for software engineering design and is another example of a two dimensional scene of objects. Even thought this UML editor is a 2D application, changes performed on the design are not marked on the graphical view but rather in a text representation showing the differences between the class hierarchies of two versions of the design.

Another mechanism for change awareness is the replaying of changes over time by showing the actions that represent the evolution of the document between two states such as in the Chimera system [6]. Unfortunately, such a mechanism can be proven time inefficient for users while it also lacks an overall presentation of the document modifications.

In [7], a multi-level coding method has been proposed as an awareness mechanism. An awareness strength function was defined between each pair of users to return a value reflecting how strongly users may be aware of each other. Each editing operation is encoded into n granularity levels. For instance, the levels of encoding of a graphics editing operation might be the text representation of the operation, a simple outline of the operation effects in black/white colour and degrading colour intensities of the operation effects. Depending on the strength awareness between a user and their collaborators, an operation performed by that user is sent to the collaborators in the corresponding encoded form. The approach is especially suitable for bitmap-based graphics where different collaborators can receive different data streams in different granularities.

PastDraw [11, 12] implements various display mechanisms for representing changes of a scene of objects: animated replays, storyboards, iconic displays and documentation methods. Animated replays employ playback animations of changes referring to a certain part of the scene of objects. The storyboard technique illustrates the changes that took place by capturing them with a series of still frames while the iconic display technique consists of attaching various icons corresponding to different types of changes. Finally, the documentation technique describes textually the changes that took place.

The approaches described above do not compute and visualise overall information about changes made by users to a document. To the best of our knowledge, PastDraw [12] is the only approach that offers an overview intended to communicate at a glance all of the changes made to a graphical document. In this approach, various colour intensities are associated with objects to illustrate the number of changes. However, this method was applied only for class diagrams, where classes do not have a background colour and could not be used for a general approach where objects have an associated colour. Furthermore, this approach is not suitable for large sized documents. The scene of objects has to be mapped to an overview and therefore objects will have a small size. Consequently, changing the intensity of the colour of an object is not a good visualisation highlight. Our visualisation approach based on the concept of heat maps is suitable for a document of any size. Moreover, PastDraw does not consider object groups and operations performed on the groups as well as an overview visualisation of these operations. Our approach can be applied to any scene of objects independently of its size and the colours associated to objects. We provide various awareness levels associated with the document: pages, layers, groups and objects. Moreover, our approach is general since it can be applied to other types of documents such as text documents.

CONCLUSION

In this paper we proposed an awareness model that computes and presents an overview of changes in collaborative graphical authoring tools. Awareness information can be customised according to user preferences: It can be computed and presented at different levels of granularity such as pages, layers, group of objects and objects and it can be filtered according to operation types and users. The proposed awareness model is fully general since it is adaptable for both text and graphical documents and various metrics and visualisation mechanisms can be used to provide an overview of modifications made to documents.

Our future work will include the implementation of the proposed awareness model over the collaborative graphical editor Draw-Together [5] that we developed. The representation of the graphical scene of objects used by this editor is based on a hierarchy and therefore awareness information can be provided at different granularity levels. Additionally, we plan to reuse the implementation of the core model from our awareness enhanced text authoring tool, which we expect will minimise the time and effort required. Finally, we plan to use the resulting graphical editor to perform user studies with various metrics and visualisation tools.

Another research direction we want to investigate in the future is to provide awareness for documents that contain both text and graphical objects. One of the big challenges we will face would be to find a suitable visualisation mechanism for overviewing changes over both text and graphical objects.

REFERENCES

1. R. M. Baecker, D. Nastos, I. R. Posner, and K. L. Mawby. The User-centered Iterative Design of Collaborative Writing Software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI'93*, pages 399–405, Amsterdam, Netherlands, 1993. ACM Press.

2. P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114, New York, NY, USA, 1992. ACM Press.

3. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the USA*, 95(25):14863–14868, 1998.

4. C. Gutwin, S. Greenberg, and M. Roseman. Workspace Awareness Support with Radar Views. In *Conference Companion on Human Factors in Computing Systems - CHI'96*, pages 210–211. ACM Press, April 1996.

5. C.-L. Ignat and M. C. Norrie. Draw-together: Graphical editor for collaborative drawing. In *Proceedings of the International Conference on Computer Supported Cooperative Work (CSCW'06)*, pages 269–278, Banff, Alberta, Canada, November 2006.

6. D. Kurlander. Graphical editing by example (abstract). In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, page 529, New York, NY, USA, 1993. ACM Press.

7. Y. Li, J.-J. Bu, C. Chen, and X.-H. Xu. Research on awareness model on real-time collaborative graphics editing system. In *International Conference on Machine Learning and Cybernetics : ICMLC 2003*, pages 2944– 2949, 2003.

8. P. Molli, H. Skaf-Molli, and G. Oster. Divergence awareness for virtual team through the web. In *Integrated Design and Process Technology, IDPT 2002*, Pasadena, CA, USA, June 2002.

9. S. Papadopoulou, C.-L. Ignat, G. Oster, and M. Norrie. Increasing Awareness in Collaborative Authoring through Edit Profiling. In *Proceedings of the IEEE Conference on Collaborative Computing: Networking, Applications and Worksharing - CollaborateCom 2006*, pages 1–10, Atlanta, Georgia, USA, November 2006. IEEE Computer Society.

10. R. Robbes and M. Lanza. A change-based approach to software evolution. *Electron. Notes Theor. Comput. Sci.*, 166:93–109, 2007.

11. J. Tam and S. Greenberg. A Framework for Asynchronous Change Awareness in Collaborative Documents and Workspaces. *International Journal of Human-Computer Studies - IJHCS*, 64(7):583–598, July 2006.

12. J. Tam, L. McCaffrey, F. Maurer, and S. Greenberg. Change Awareness in Software Engineering Using Two Dimensional Graphical Design and Development Tools. Research Report 2000-670-22, Department of Computer Science, University of Calgary, Alberta, Canada, October 2000.

13. S. Xia, D. Sun, C. Sun, and D. Chen. Collaborative object grouping in graphics editing systems. In *Proceedings of IEEE 2005 International Conference in Collaborative Computing (CollaborateCom'05)*, San Jose, California, USA, December 2005.