A Framework for Asynchronous Change Awareness in Collaboratively-Constructed Documents

James Tam and Saul Greenberg

Department of Computer Science, University of Calgary
Calgary, Alberta, T2N 1N4 Canada
{tamj,saul}@cpsc.ucalgary.ca

Abstract. *Change awareness* is the ability of individuals to track the asynchronous changes made to a collaborative document or surface by other participants over time. We develop a framework that articulates what change awareness information is critical if people are to track and maintain change awareness. Information elements include: knowing *who* changed the artifact, *what* those changes involve, *where* changes occur, *when* changes were made, *how* things have changed, and *why* people made the changes. The framework also accounts for people's need to view these changes from different perspectives: an *artifact-based* view, a *person-based* view, and a *workspace-based view.*

1 Introduction

People often collaborate for the purpose of creating and developing a work artifact over time. This happens when people co-author papers, or iterate designs from conception to final form, or negotiate a plan through an evolving blueprint. The participation of all partners is vital, perhaps due to the group's particular combination of skills and expertise, or because all participants are interested stakeholders, or because there is too much work for one person to do by themselves, or because involvement is required if participants are to buy into the final outcome.

While many episodes of collaboration often occur in face to face meetings over the work artifact, others frequently occur asynchronously. Asynchronous collaboration and evolution of the artifact can happen in several ways.

- People may explicitly pass the artifact back and forth for comments and revisions (i.e., 'it is your turn, give it back to me after you have worked on it').
- Individuals may work on the artifact as time and opportunities arise, without explicitly coordinating this with the other participants.
- The group may drift in and out between collocated and asynchronous work (e.g., a group may begin work in an extended face to face meeting, but members may leave and return to the meeting over its course).

In same-time collaborations, we already know that people use *workspace awareness* not only to follow actions of others, but to understand and respond to any changes others make to the workspace artifact [5] (see Section 3). The problem is that when people interact asynchronously this awareness disappears; changes are only understood if one person tells the other what they have done (e.g., through prior coordinat-

G.-J. de Vreede et al. (Eds.): CRIWG 2004, LNCS 3198, pp. 67–83, 2004.

ing talk, on-going emails, or notes attached to the artifact), or if the person can somehow understand the changes made by inspecting the artifact. If people cannot understand what has changed, collaboration can quickly spiral out of control. Missed changes made by one person can unintentionally wreak havoc on the work of others or even the entire project.

Within this context, our research interest is on *asynchronous change awareness of artifacts* (which we call change awareness for short), defined as the ability of individuals to track the asynchronous changes made to a collaborative document or surface by other participants. Our research concentrates primarily on how people maintain change awareness by inspecting the changed document, rather than on how change awareness is transmitted by other communications e.g., verbal or textual dialog that occurs directly between people outside the confines of document.

Because change awareness is a broad subject, our immediate goal in this paper is to contribute a framework of the critical information people need if they are to maintain change awareness. First, we set the scene with several examples of failures that arise when people have inadequate change awareness, and then describe how current systems try to provide this information. Second, we summarize Gutwin's earlier framework for workspace awareness for real time interactions [5], for it acts as a theoretical precursor to our own work. Third, we introduce and describe in detail our framework for change awareness. We close by discussing several implications this framework has to practitioners and implementers of change awareness systems.

2 Motivations and Related Work

Several true incidents give an example of the consequences of missed changes.

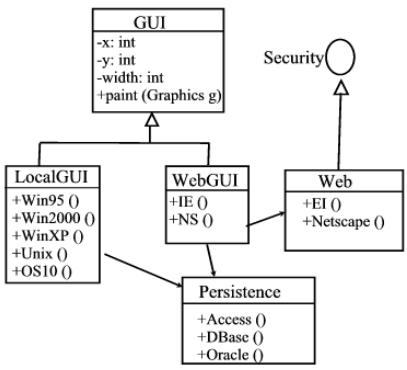
The Town of Canmore in Canada has an administrative office that oversees all subdivision plans submitted by developers. In this process, the developers and the administrative office negotiate the plan through many back and forth exchanges. The administration would ask for changes so that the subdivision fit the needs of the town, and the developers would respond by incorporating these (and perhaps other) changes in a modified plan of the subdivision development. In one on-going subdivision plan, the developers added a gate to the main road entrance, a security measure that inhibits 'outsiders' from entering the grounds. The administrative office did not notice this addition, and approved that particular version of the plan. This oversight was only seen after the subdivision was built with the gate in place. The gate generated widespread controversy, where it made the front page of the local newspaper and became a heated issue for the town council, the town population, and the developers. Because Canmore was a small town fostering community values, the townspeople felt that this gated community created a private enclave that violated the sense of community held by its population, and that it would also set a bad precedent for future developments. The developers, on the other hand, believed that they had followed the planning process correctly, and because the plan had been approved they had the right to implement their vision. The developers were adamant that they had not tried to 'slip in' this gate amongst the other changes in the plan, and indeed had told a staff member about it. The administrative staff said that the key staff members did not see the addition of

the gate; it was a visually small change in a complex plan that had many other changes within it. The town council stated that "they didn't know about the plan and wouldn't have approved it had they known" (reported in the Rocky Mountain Outlook, March 11, 2004). What happened was that the administrators lacked awareness of what had changed between documents, and thus missed a small but critical detail.

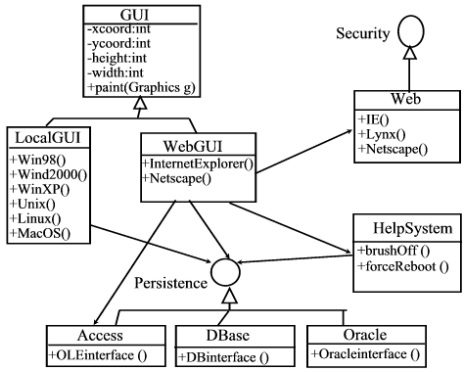
Another true example focuses on missed changes within TeamWave Workplace, a commercial room-based groupware environment [4]. Community members could enter a virtual room at any time and change its contents, i.e., a collection of persistent groupware applications and documents scattered on the wall of the room. The only difference between synchronous and asynchronous work was whether people happened to inhabit and make changes within the room at the same time or at different times. Independent evaluators of TeamWave found that its major usability problem was that people had difficulty maintaining awareness of whether anything had changed since the last visit, and what those changes concerned [16]. "Participants reported that they were unable to effectively and efficiently detect modifications to artifacts made by others in the rooms they were using. TeamWave users devised [email workarounds] to provide the information that they needed" (pp. 5). The problem was that TeamWave users had to resort to manual techniques to monitor changes. To see if anything had changed, they had to start and login to the application, navigate to a room, and visually inspect it. Because this is a heavyweight and error-prone process, people did not visit rooms often enough to notice if anything had changed, and when they did visit a room some changes were easily missed. In turn, this meant that people were increasingly reluctant to leave time-critical information in a room, for they knew that others were unlikely to notice it until too late. The usefulness of the entire system suffered as a consequence.

Our third example illustrates the effort people go through if a system does not explicitly provide a mechanism revealing change awareness information. A typical strategy is to compare the two document versions – by memory or visual inspection – and to try and spot changes and decipher their meaning. This is both a daunting and error prone task. For example, Figure 1 (a) and (b) show a before and after version of a UML class diagram respectively. Although a person may be able to eventually determine all the differences, it requires much time and effort to do so, and changes are easily missed. Try this for yourself: see how many differences you and your collaborators can spot in (say) 30 seconds. Have a first person try it when both images are side by side (for direct visual comparison). Have a second person try it when the images are on opposite sides of the same piece of paper (for short-term memory comparison). Have a third person look at the first image on the first day for 30 seconds, and then have them examine the second image on the second day for changes (for long-term memory comparisons).

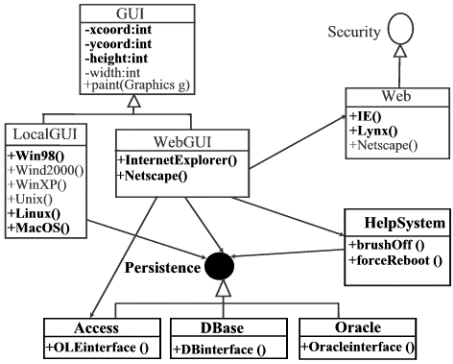
Of course, the document itself can help reveal changes. For example, the version of the UML diagram depicted in (c) highlights specific blocks of changed items by bolding them and by muting unchanged items (coloring would be preferred, but this is not possible in this black and white reproduction). While limited, even this simple technique clearly allows people to spot changes faster and with fewer errors than by using manual comparisons.



a) Before version of a UML diagram.



b) An after version of a UML diagram.



c) The after version, with changes now shown in bold, and unchanged parts muted.

Fig. 1. A UML diagram: a) original, b) changed version, and c) visually enhanced changes

These three examples deal with scenarios where people are already interested in collaborating within a shared workspace or document. [13] also describe the benefits of asynchronous awareness for potential future collaborations.

Developers are not blind to the importance of change awareness. Many commercial applications concerning sequential documents (e.g., collections of reports, papers, and source code) can detect and show change information between new and old versions. A sampling of standard techniques is listed below.

The first class of change awareness techniques contains several ways that differences between two versions of a document are visually displayed on the screen.

- *Sequential deltas,* as exemplified by the UNIX *Diff* system [7] inserts instructions after lines that are deemed different. Instructions describe how the line(s) in the previous document can be transformed into the new document, i.e., instructions to add or delete new lines, or instructions that transform an old line into a new form.
- *Annotations and markups* display differences as explanatory notes that points to the parts of the document that has changed. Microsoft Word's *track changes* capabilities, for example, optionally displays changes as margin notes [11]. Flexible Diff [14] accompanies the source text with multiple columns of annotations. The first and second columns contain the original and modified text. The third column

shows only the differences (where thresholds can be specified to mute small differences), while the fourth shows explanatory annotations added by the author. Another example is how Rational Rose shows changes in object-oriented source code [8]. In one of its views, it displays objects within a hierarchical class browser, and marks items that have changed by special symbols positioned next to the item.

- *Highlighting* displays the differences within the document itself by directly marking up the appearance of the document contents. One example is the UML diagram in Figure 1c. Another is Microsoft Word's ability to show inserted text in a contrasting color, and deleted text as crossed out (changed text is treated as a combination of deleted and new text) [11].
- *Overviews* communicate changes made to an entire document at a glance. They usually provide a graphical miniature of the document, and mark regions that have changed through highlighting. Attribute-mapped scroll bars [6] provide indications of edit-wear (i.e., the areas where people have repeatedly edited the document) as marks within a scroll bar. Seesoft provides a zoomed out view of source code scattered over several files [3]. Each column of the overview represents a file; each character is compressed into a pixel. Line coloring indicates change to the code, such as its age and the developer responsible for adding it. State TreeMaps uses a Treemap overview to emphasize changes made within multiple documents [12].
- *Graphical playback* replays changes over time by showing all fine-grained editing actions, or by storyboarding major changes into easily comparable scenes [9,17].

The second class of change awareness techniques describes how document versions are maintained or specified, and how changes within them are detected or tracked over time. Each technique may use one or more of the change display mechanisms described above to reveal its information to the end user.

- *File differencing* occurs when a user manually specifies two files and asks the system to compare them for differences [7].
- *Real time differencing* lets the author turn on a facility to continually track changes. The document itself stores these changes internally. Microsoft Word serves as a good example, where text is marked up as the author types [11]. All change information is kept until an author decides to accept or reject them.
- *Version control systems* automate and enhance the process of manually tracking progressive versions of documents and their changes [18,1,15,10]. The first version is typically created by freezing the current document. The frozen version can no longer be changed, and so editing it implicitly creates a new revision [18].
- *History systems* track all incremental changes made to a document, typically so they can be played back at a later time or so actions can be undone [9,17].

Most of the above are *ad hoc* solutions for sequential documents, and even then there is much they leave out. They typically neglect change awareness in two-dimensional graphical documents [17]: figures, photos, blueprints, concept maps, graphs, UML diagrams, and collaborative workspaces containing spatially scattered artifacts. While 2D documents are widespread, techniques for displaying change awareness within them are undeveloped and are likely non-trivial [17]; we do not even know what change information they should show to the end user. Our goal is to fill this void.

Table 1. Elements of WA relating to the present, from Gutwin [5]

Category	Element	Specific questions
Who	• Presence • Identity • Authorship	• Is anyone in the workspace? • Who is participating? • Who is that? • Who is doing that?
What	• Action • Intention • Artifact	• What are they doing? • What goal is that action part of? • What object are they working on?
Where	• Location • Gaze • View • Reach	• Where are they working? • Where are they looking? • Where can they see? • Where can they reach?

Table 2. Elements of WA relating to the past, from Gutwin [5]

Category	Element	Specific questions
How	• Action history • Artifact history	• How did that operation happen? • How did this artifact come to be in this state?
When	• Event history	• When did that event happen?
Who (past)	• Presence history	• Who was here, and when?
Where (past)	• Location history	• Where has a person been?
What (past)	• Action history	• What has a person been doing?

3 Theoretical Foundations

The prerequisite to understanding change awareness is to determine what information is necessary if people are to comprehend change in the collaborative workspace. These informational elements of knowledge verbalize, categorize and explain what information should be tracked and captured by an application as a change occurs and how this information could be useful to an end user. Once the informational elements have been specified, we can then design an interface that captures and display this information in a meaningful and useful fashion. The details of the first step - the information elements – are the focus of the remainder of this paper. Our theoretical foundations of these information elements have their roots in Gutwin's framework for workspace awareness [5], summarized in this section and in Tables 2 and 3.

3.1 Workspace Awareness for Real Time Interactions

Gutwin focused on *workspace awareness* within real time groupware environments. He was concerned with people's continuous maintenance of awareness of others in a visual workspace and how others were interacting with the artifacts held by that space. He articulated a broad set of awareness elements, where each element consists

of the information that people need to track events in the workspace as they occur. Table 1 shows Gutwin's elements of knowledge contained within a "who, what and where" category of questions asked about workspace events in the present.

For each of these categories of questions, there is a unique set of *informational elements* that provides answers to those questions. These informational elements are the specific pieces of information that a person would require in order to keep up with events as they occur in a collaborative real time workspace.

For example, knowledge of the 'who' category simply means that you know the number of people who are present in the workspace (if any), their identity, as well as being able to attribute a specific person to each action that is taking place (Table 1, first row). In the 'what' category (second row), awareness of action and intention means that you know what a person is doing both at a rudimentary level (e.g., typing some text) and at a more abstract level (e.g., creating a title). Awareness of artifact is knowing what object another person is currently working on. Location, gaze, view and reach are all inter-related in the 'where' category (third row): location refers to the part of the workspace where a person is currently working; gaze is the part of the workspace where a person is currently looking at; view is where they can potentially be looking (i.e., their field of vision), and reach includes the parts of the workspace that this person can potentially change [5].

3.2 Workspace Awareness for Past Interactions

Gutwin does mention elements for maintaining awareness of asynchronous changes in his framework, required for people to catch up with events that have already taken place in the workspace [5]. Table 2 lists this second collection as elements of knowledge contained within the "who, what, where, when and how" categories of questions that may be asked of workspace events in the past.

Determining 'how' the workspace has changed involves two elements: *action history* and *artifact history* (first row, Table 2). Action history describes the unfolding of events that changed the workspace. Artifact history includes details about the process of how an object was changed over time. Information about 'when' something has occurred (second row) is described by the *event history* of the workspace. This element indicates the time at which things occurred in the workspace. 'Who' provides a *presence history* of people in workspace, that is, of knowing who has visited a particular location and when this visit occurred (third row).

Although there are potentially many aspects to the 'where' category of questions e.g., where did events take place, where have artifacts been etc., Gutwin mentions only the *location history* of other people, which indicates where a person has been in the workspace. Finally the 'what' category of questions lists the *action history* of a person, which describes what actions have another person engaged in (last row).

Gutwin's framework is a good beginning. However, because he was mostly concerned with elements relating to the present, he did not elaborate on past elements beyond this initial list. The remainder of this paper tries to continue where he left off in developing a framework of change awareness.

4 Information Elements for Change Awareness

In this section, we extend and elaborate the elements Gutwin identified for workspace awareness to create a more comprehensive change awareness framework for different-time asynchronous work. In this situation, a person has been away from the workspace for a period of time (hours, days, weeks) and must be brought up-to-date on what has changed in the interim.

When trying to catch up with changes the first piece of information that a person needs to know is "Is anything different since I last looked at the work?" Obviously a change awareness system must provide the answer to this question in a very lightweight fashion. Afterwards, the person can then probe for further details by trying to find out the specifics of a change. The specific information that a person may require in order to track changes will vary from situation to situation. It will depend upon the task that is being performed, the person who is carrying out the task, as well as the surrounding environment.

In a manner similar to how Gutwin constructed his framework for workspace awareness, we can describe at a high level the questions that may be asked. This set of questions includes:

1. Where have changes been made?
2. Who has made the changes?
3. What changes were made?
4. How were things changed?
5. When did the changes take place?
6. Why were the changes made?

However, this does not suffice by itself. A change awareness framework must account for the fact that people may need to view aspects of the workspace in different ways at different times, i.e., from different perspectives. In particular, a person may query the workspace for changes in terms of:

- the artifacts that exist within it (*artifact-based view*),
- the people who work within it (*person-based view*), or
- the workspace may be viewed as one locale or as a collection of related locales (*workspace-based view*) where a person is interested in the changes and events that have taken place in one or more locales. This relates to [1]'s artifact metaphor.

The specific perspective that a person has of the workspace will have an impact on the information that he or she is interested in and the way that the information is requested and represented. In terms of the artifact-based view, the person will be interested in changes made as they relate to particular workspace artifacts, and will make various queries about those changes. Examples include: how has this item changed, and what has been done to this item? From the person-based view, an individual wishes to know about the changes that were made by another collaborator. Queries about changes will therefore be focused on this person e.g., what did he do while I was away? On the other hand, someone with a workspace-based view would be interested in and inquiring about the events that have taken place in a specific location e.g., what changes and events took place in this space? This location can

consist either of the workspace as a whole, or portions of the space that are somehow logically related e.g. a specific room in a room-based groupware system or a particular spatial region.

Of course there is a strong relation between the three workspace perspectives and the six categories of awareness questions. An individual that holds a person-based perspective will focus heavily on the 'who' category of questions. Someone that holds an artifact-based view of the workspace may focus instead on the 'what' category of questions and try to determine what changes were made to specific objects. Yet another person that holds a workspace-based view of the workspace may focus on the 'where' category of questions. Alternatively the person with a workspace-based view may focus on 'what' was done in the part of project that he or she holds an interest in. The main point is that the person's particular view of the workspace will influence the value that he or she attaches to each category of question. As will be seen, however, the specific example questions that are unique to each of the six high level categories awareness questions can be asked from any of the three workspace perspectives.

The following subsections will describe in detail the informational elements associated with each category of question as well providing some specific example questions that a person might ask about changes.

4.1 Where?

Location in a 2D graphical workspace could be a simple Cartesian spatial region, or a direct digital analogue of physical demarcations, e.g. the rooms in a room-based system such as TeamRooms [4] or locations may be more abstract in relating workspace entities to each other, e.g., the different logical or conceptual parts of a collaborative project. In all cases, the location of a change provides valuable clues regarding its context, which in turn guides people towards further exploration. For example, a person may ask if a given change was part of the project component that is undergoing extensive rework, that is, if the change occurred is in the same place where many other changes are also occurring.

Table 3 shows the specific questions that may be asked to learn 'where' changes have occurred with respect to each of the three workspace perspectives. As already mentioned, the difference is how queries about changes made to the workspace are formulated within each perspective. With the artifact-based view, the questions could be asked in terms of a specific object in the workspace. Where is it now? Where was it before? Where has it been since I have been away? From the person-based view, the example questions may be asked about a specific collaborator. Where has this person visited or looked in the workspace? Where did this person change things? From the workspace-based view, the questions asked would inquire about the different events that have taken place in the space since a person has been away. Where in the workspace have people visited? Where were the artifacts moved?

The informational elements that will answer the 'where' category of questions include Gutwin's location history (described in Section 3), and the new categories of gaze history and edit history. Location history refers to the parts of the workspace

that have been visited by a person. Gaze history includes the parts of the workspace that a person has looked at. The difference between location and gaze history is that while a person may have been present in a general location, he or she may not have actively attended to everything that went on there. Although location and gaze history do not directly provide information about changes that have been made, they do indicate the parts of the workspace that have been visited or viewed and the frequency of these visits [6]. This provides strong clues as to where one should look for changes. Edit history, on the other hand, explicitly deals with the changes that were made. Awareness of 'where' edits occurred is vital to routine project management as it provides strong clues as to the progress that has made towards satisfying project-level goals. By this very fact, the location of changes provides strong cues to the answers to other "who, what, why, and how" category of questions.

Table 3. Information elements and workspace questions related to 'where'

Where			
Information elements	Specific questions		
	Artifact based view	Person based view	Workspace view
Location history Gaze history Edit history	Where was this artifact (when I left)? Where is the artifact now? Where has this artifact been during the time that I have been away?	Where in the workspace has a person visited? Where in the workspace has a person looked at? Where in the workspace has a person made changes?	Where have people been in the workspace? Where were artifacts in the workspace? Which parts of the workspace have people looked at? Which parts of the workspace have people made changes in?

4.2 Who?

Answers to questions concerning 'who' are important. In collaborative environments, knowing who made a change becomes an opportunity to query that person directly for further information. Also, people may attend to changes differently depending upon who made them. For example Neuwirth et. al. [13] described how collaborators could be more interested in seeing changes made by co-authors that he or she has known for a long time and less interested in seeing changes made by less trustworthy co-authors.

In Table 4, we provide a more detailed breakdown of the 'who' questions from the different workspace views. To Gutwin's concept of *presence history*, we add *identity*, *readership history* and *authorship history*. While presence history tracks if anyone was present, identity indicates the specific individual associated with a change or event. Establishing identity is needed to provide the context for answering the person-based view of workspace changes. For example, a team member may be responsible

for auditing and upgrading different parts of the project. If his or her identity is associated with a particular change, it may better help other team members (who may prefer their original version) accept that change. Readership history carries identity even further by listing all the people who have viewed a particular artifact, or conversely, listing all the items that a particular person has seen. This is important as knowing that someone has viewed the changes without raising any objections or making any further changes suggests an implicit acceptance of the current work. By a similar vein, authorship history can list the people who have made changes to an artifact, or list all the artifacts that a particular person has changed. Tracking readership and authorship history can, for example, be used to gauge progress of the project through a process-oriented lifecycle. In such a case knowing who has seen an object and 'signed off' on it is an important part of workflow management and document routing.

Table 4. Information elements and specific workspace questions related to 'who'

Who			
Information Elements	Specific questions		
	Artifact based view	Person based view	Workspace view
Presence history Identity Readership history Authorship history	Who has looked at this artifact? Who has changed this artifact?	Who has this person interacted with? Who made changes with this person?	Who has been in the workspace? Who has looked at the workspace? Who has made changes to the workspace?

4.3 What?

The 'what' category of questions leads to answers that produce a picture of the *action history* of the workspace (Table 5). Gutwin described two ways that action history can answer these questions [5]. First, it can be used to track all low level actions that a person has done, e.g., creating, labeling and positioning a circle in a diagram. Knowledge about actions that people have engaged in while one was away is important. When people are asked to describe what is new in the workplace it is frequently in terms of the actions and events that have taken place. Sometimes there is, of course, a need to put all of these lower level actions in the context of the higher goals. So Gutwin also described action history from a higher-level perspective of the low level changes in a way that considers the goals that motivated these actions, e.g., the labeled circle was created in order to represent a new person in an organizational chart.

Yet the low-level questions and answers presented in Table 5 are often the only information that developers can hope to capture when they add change awareness support to an application. The problem is that it is difficult to ascertain and store the motives behind a series of low level changes. One could use spatial proximity (i.e.,

changes located near to each other) or temporal proximity (i.e. changes that occur at the same time) as predictors of inter-relatedness, but often these methods will fail because the sub-steps for achieving several different high level goals may often be interleaved. Thus, we will postpone discussing the higher-order view of changes until Section 4.6, and instead focus here on only the significance of low-level changes. It is important to point out, though, that people are able to relate and combine several low-level actions to derive a higher-level action if they are given enough contextual information to understand the relationships between these lower-lever actions.

The specific questions associated with the 'what' category varies depending upon the perspective that a person has of the workspace. These questions are shown in table 5. For the artifact-based view, inquires are made about the changes that have been made to a particular artifact. From the person-based view, the questions ask about what actions has a person undertaken. With the workspace-based view, the questions ask about the actions that were undertaken within the workspace or actions that were carried out on the artifacts in the workspace.

Table 5. Informational elements and specific workspace questions related to the 'what'

What			
Information Elements	Specific questions		
	Artifact based view	Person based view	Workspace view
Action history	What changes have been made to the artifact?	What artifacts has a person looked at? What artifacts has a person changed? What activities has a person engaged in?	What changes have occurred in the workspace? What artifacts were viewed? What artifacts were changed?

4.4 How?

The 'how' category asks how the current workspace differs from the way that it was before (Table 6). The answers to these questions can be integrated to derive one of two historical views of changes. The first is in terms of the *process history* of the workspace, which indicates incrementally how the workspace evolved from its previous state (i.e., the state that it was in when one last looked at it) to its present state. This is useful when a person is interested in the mechanical means (the intermediary steps) that produced a change or group of changes as well as the end result. Thus process history is tightly coupled with action history. The difference is that the action history consists of all the actions that have occurred while one was away, while process history relates and abstracts a subset of all actions into a series of steps in a process. The process view is important for, as Gutwin described, people may have trouble interpreting instantaneous changes [5]. Thus describing all the sub-steps involved in a change may help to clarify what happened. Also, the process-oriented view describes

the *evolutionary context* of changes (i.e., the specific details of the circumstances faced by the person who made the change at the time that the change occurred), and thus can provide valuable insight on 'how' and 'why' things came to be in their present state. Of course, a person may only be interested in the final result. This is the second historical view of 'how' a workspace has changed, i.e., the *outcome history*. The outcome history presents only a 'bottom line' understanding of a change where it highlights only those things that differ between the initial and the final state.

The choice of process *vs*. outcome history will depend largely upon the task at hand. For example, a graphic artist may be interested in the technique used to produce some visual effect. In this case, this person will want to see (and thus learn) the process history of the workspace. On the other hand, a newspaper editor reviewing an article submitted by a reporter is far too busy to be concerned with the rough drafts produced by this person, and would thus be interested only in the outcome history of the article. Consequently, it is important that software support for change awareness provide the ability to discover both the process and outcome history of a workspace.

Table 6. Informational elements and specific workspace questions related to 'how'

How			
Information Elements	Specific questions		
	Artifact based view	Person based view	Workspace view
Process history Outcome history	How has this artifact changed?	How has a person changed things?	How has this workspace changed?

Table 7. Informational elements and specific workspace questions related to 'when'

When			
Information elements	Specific questions		
	Artifact based view	Person based view	Workspace view
Event history	When was this artifact changed?	When did a person make changes?	When were changes made to the workspace?
	When was a particular change to this artifact made?	When did a person make a particular change?	When did a particular change in the workspace occur?
	In what order were changes made to this artifact?	In what order did this person make changes?	In what order did changes to the workspace occur?

4.5 When?

The timing and ordinality (sequential order) of changes is revealed by the answers to the questions of 'when' changes occurred as listed in Table 7. The time when a change occurred, particularly if it overrides an earlier change by another person, is often of great significance and affects the perceived importance of a change. For

example, a person may only be interested in recent workspace events, or a person may only be interested in changes that occurred within a specific period of time.

The timing and ordinality of changes constitute the *event history* of the workspace, and it provides the *chronological context* for understanding and interpreting changes giving clues to the 'where', 'who', 'what', 'how' and 'why' categories of questions.

4.6 Why?

Knowing the thought and motives behind a change can be important for accepting the changes that others have made. The questions that a person will ask to discover 'why' changes were made are summarized in Table 8. A historical view of 'why' changes were made includes both the *cognitive history* and the *motivational history*. Cognitive history describes the logic or reasoning that may be behind a change, which is a rational reconstruction of the person's goals and plans. Motivational history deals more with the impulses or desires that are the impetus for making a change, which is the actual reason why a person did something at a moment in time. The reason that they are separate elements is because a change may be based upon a well thought out and carefully conceived plan or it may be more of a spur of the moment thing as one reacts to the current situation. Also, some changes are completely unintended accidents.

Although it is not always needed, knowing 'why' a change was made is obviously an important step for coming to understand and accept it. For lower-level changes that are the parts of a grander higher-level change, the motivating factors may be painfully obvious. In this way providing the motivational history for simple changes may be too effortful (and distracting from the main task) to explain. Also, describing all the motives behind a change and detailing all the reasoning behind each event is extremely difficult for computers to do automatically. This is because understanding the 'why' often draws upon a person's accumulated technical expertise and implicit or 'hidden' cultural information relating to group priorities, work practices, and short and long term goals. Today's computing systems lack the ability to sense these technical and cultural factors that motivated a change. They also lack the intelligence needed to produce a truly comprehensive picture of the cognitive history of the workspace. Consequently, most 'why' information will likely be generated explicitly by authors, e.g., as annotations added to changes or as design rationales.

Table 8. Informational elements and specific workspace questions related to 'why'

Why			
Information Elements	Specific questions		
	Artifact based view	Person based view	Workspace view
Cognitive history Motivational history	Why was this artifact changed?	Why did a person make that change?	Why was that change made in the workspace?

5 Discussion

We have described in detail the information that can be used by a person to track changes made by others over time in a collaborative project. The informational elements were classified according to several categories of change awareness questions. These categories are inter-related and inter-dependent. When a person is tracking changes he or she may start with the highest-level question, 'Has anything changed?' From that point the person will make inquiries about changes from one or more of particular perspectives - artifact, person, or workspace-based - that make the most sense to him or her and the work context. The inquiries can be directed towards a specific collaborator, 'What has this person done?' Alternatively, the process of inquiry can take the form of an examination of a particular artifact, 'How does this object differ from before?' or it can take the form of an inspection of a select portion of the workspace, e.g., 'What has happened in this corner of the project?'

Within the bounds of the chosen perspective, a person can ask specific questions from these categories to probe for further details of changes. The specific category that a person begins with (where, who, what, how, when or why) is not fixed. As mentioned previously, it will be influenced by the workspace perspective that is held e.g., if a person is making inquiries from a person-based view then the 'who' category may be of the most pressing urgency. Also, as we have shown in the previous sections, queries made in one category of question are not isolated from the other categories. The process of inquiry can occur in parallel as someone delves for answers in more than one category at once. For example, take the case of a project manager who is reluctant to have certain parts of the project undergo anymore changes, or who has severe misgivings about the work of specific team members. When the manager returns to the project after a period of absence and discovers that many changes have been made, he may immediately try to determine exactly where changes were made and specifically who made those changes.

The answers to the questions from one category may also inspire additional inquiries in another category. For example, a person who is tracking the historical context of the changes to a software system may start by asking about 'when' most of the changes occurred. Upon discovering this information she notes that the code was most volatile during a port between operating systems. Since she knows that there is a methodological way to do this, her queries then focus on the process history of the software as she tries to determine exactly what the programmers did during the port.

Furthermore, the answers to the questions that a person asks about one category of question may directly provide him with further information. For example when a person knows exactly where changes occurred, she then knows who made those changes (because she knows who is responsible for which portions of the project). Or the person may be able to make predictions about the answers to the other categories of questions based upon the information that she gets from one category. When a person learns that a specific team member made a change, she can guess as to how the change was made. These guesses are based upon her personal knowledge of the person who made the changes and the techniques that he has employed in the past.

Although a single answer may provide information about multiple categories of questions, the main point of the framework is to ensure that designers of change awareness systems actually consider what change information should be captured if the system is to provide its end-users who are tracking changes with the information they need to answer these questions. At the very least, the designer can use this framework to priorize what change information is needed, and to focus on those with the highest priority. The framework ensures that the designer will not neglect certain categories out of ignorance.

6 Conclusions

In this paper we have introduced a theoretical framework that can be used in several ways. First, designers can use it as a high-level guide for determining what change information should be tracked and displayed to participants, and what perspectives of viewing this information are relevant to the end user. Of course, we don't claim that this determination is always easy, for the needs of the collaborators can be quite situation specific. Second, evaluators can use the framework to critique existing systems: even using it as a simple 'checklist' will quickly reveal what change information is captured and displayed by the system, and what is left out. We have already done this critique to reflect on a change awareness system we had created before forming the framework, and it revealed many oversights and deficiencies [17].

The next challenge is to use the framework to build an exemplar of how a two-dimensional graphical groupware application can support asynchronous change awareness. While the framework states what is needed, creating good interaction and display techniques for change awareness remains a significant challenge.

References

1. Berlage, T., and Sohlenkamp, M.: Visualizing Common Artefacts to Support Awareness in Computer Mediated Cooperation, CSCW Vol. 8 No. 3 (1999)
2. Brown, H.B.: *The Clear/Caster System*. Proc NATO Conf. Software Engineering (1970)
3. Eick, S., Steffen, J. and Sumner, E.: Seesoft—A Tool for Visualizing Line Oriented Software Statistics. In Card, S., MacKinlay, J. and Shneiderman, B., Ed., Readings in Information Visualization, Morgan Kaufmann Publishers Inc. (1992) 419–430
4. Greenberg S. and Roseman, M.: Using a Room Metaphor to Ease Transitions in Groupware. In M. Ackerman, V. Pipek, V. Wulf (Eds) Sharing Expertise: Beyond Knowledge Management, 203-256, January, Cambridge, MA, MIT Press (2003)
5. Gutwin, C.: Workspace Awareness in Real-Time Groupware Environments. Ph.D. thesis, Department of Computer Science, University of Calgary, (Calgary Canada), (1997)
6. Hill, W.C. and Hollan, J.D.: *Edit Wear and Read Wear*. Proc ACM CHI'92, (1992) 3–9
7. Hunt, J., W. and McIlroy, M.D.: An Algorithm for Differential File Comparison. Computing Science Technical Report No. 41, Bell Laboratories (1975)
8. IBM Corporation: Rational Rose. //www-306.ibm.com/software/rational/
9. Kurlander, D.: Graphical Editing by Example. Proc ACM CHI'93, (1993)

10. Magnusson, B. and Asklund, U.: Fine Grained Version Control of Configurations in COOP / Orm. Proc Symposium on Configuration Management, SCM6 (Berlin, Germany) (1996)
11. Microsoft Inc.: Microsoft Word, as included in Microsoft Office Professional 2003
12. Molli, P., Skaf-Molli, H. and Bouthier, C.: State Treemap: an Awareness Widget for Multi-Synchronous Groupware. 7th Intl Workshop on Groupware - CRIWG'2001 (2001)
13. Morán, Favela, Martínez-Enríquez, and Decouchant: (2002), Before Getting There: Potential and Actual Collaboration Proc Springer-Verlag CRIWG (2002)
14. Neuwirth, C.M., Chandhok, R., Kaufer, D.S., Erion, P., Morris, J. and Miller, D.: Flexible Diff-ing in a Collaborative Writing System. Proc ACM CSCW '92, (1992) 147–154
15. Rochkind, M.J.: The source code control system. IEEE Trans Software Engineering Vol. 1 No. 4, (1975) 364-370
16. Steves, M.P., Morse, E., Gutwin, C. and Greenberg, S.: A Comparison of Usage Evaluation and Inspection Methods for Assessing Groupware Usability. Proc ACM Group '01 (2001)
17. Tam, J.: Supporting Change Awareness in Visual Workspaces. Unpublished M.Sc. thesis, Department of Computer Science, University of Calgary, Alberta, February (2002)
18. Tichy, W. F.: RCS – A System for Version Control. Software – Practice and Experience, Vol. 15 No. 7, (1991) 637–654