When News Is More Than What Makes Headlines

by **_Kayre Hylton_**, **_Mary Beth Rosson_**, **_John Carroll_** and **_Craig Ganoe_**

Introduction:

In order for a collaborative environment to be successful, collaborators must maintain a certain awareness about how the projects they participate in are progressing and about their fellow collaborators. This awareness is described as **Activity Awareness** [**3**]. It answers the question of social awareness ("Who is around?") as well as the question of action awareness ("What is happening?").

The issue of maintaining awareness in a collaborative environment has often been addressed, and several different styles of notification systems have been developed to accomplish this maintenance. Much emphasis has been put on interface design. McCrickard et al. [**7**] describe notification system interfaces based on their balance of interruption to the primary task, reaction to the notification, and comprehension of information received. Should the interface always be present like Microsoft´s sideshow? [**2**] Should it be transparent? [**6**] Are interruptions helpful or harmful? When and how should interruptions occur, and how should a user be required to respond to them? [**4**, **8**].

Instead of being concerned with simply how to display activity related information, in this paper we focus on the *content* of the information. What sort of information do collaborators need to know about different types of objects? We explore sending different types of information about activity to collaborators through the use of an RSS feed. RSS (which stands for RDF Site Summary) is an existing technology used to send information on items. The feed may be made up of several items, in which each item

has information such as a title, a brief textual description, and a date associated with it. The feed may also contain a link to the actual object that the item represents. Users can subscribe to an RSS feed and periodically pull information from the feed to get updates about new or changed items. They can subscribe through a variety of different RSS clients, each with its own interface and notification styles. This procedure may seem similar to an advanced mailing list system but is actually superior because it avoids the cost of moderation and facilitates aggregation of information within and across feeds [5].

RSS technology is primarily used to send out and retrieve new news headlines with links to the related news stories. We believe this technology is vastly under-utilized, and we explore the use of RSS as a notification system to support activity awareness. We use RSS 1.0 to investigate. [1].

Scenario:

Basic Resources for Integrated Distributed Group Environments (BRIDGE) is a set of tools that supports collaborative work. The Virginia Tech´s Center for Human-Computer Interaction (CHCI) originally created BRIDGE, and the Computer-Supported Collaboration and Learning (CSCL) lab in Penn State´s Information Sciences and Technology (IST) department further developed BRIDGE. BRIDGE contains many objects such as Web Pages, Calendars, Folders, Discussion Boards, Charts, Maps, and User Lists. Many ongoing projects in the BRIDGE environment make use of different combinations of the listed objects along with other objects.

One of the CSCL lab's ongoing projects in the BRIDGE environment is Multiple-View Geo-Collaborative Visualizations to Build Common Ground in Teamwork. The project involves co-located and remote collaborators and involves the use of collaborative visualizations to support synchronous and distributed teamwork in geo-spatial planning tasks. This type of intensive collaborative effort could benefit from the use of a notification system. With a notification system, group members, if they wanted to maintain awareness of their collaborators and of the progress of the project, would not be required to log into the BRIDGE environment, manually check each object to determine what changes were made, and finally check the User List to determine who was available and active in the environment.

We will re-address this scenario once we discuss in detail the goals and design for creating and distributing the content for our RSS based notification system.

Goals:

The goal of our RSS notification system is twofold. The first goal is to provide an activity summary. Through concise but meaningful content, we hope to increase the overall awareness of a user to the general status of a project. Users should gain useful knowledge of activity on a high level. Secondly, our system should facilitate change inspection. The content provided should be enough to be able to indicate to a user whether the change to an object is of interest to them. A user should be able to access any such object and easily investigate the changes in detail.

Design:

For the initial prototype of this project, we decided to focus on three BRIDGE objects: 1) the User List, 2) the Calendar and 3) the Web Page. A BRIDGE User List displays information regarding which users are logged in, when they logged in, whether they are active or idle in the BRIDGE environment, how long they have been idle, and any status message they may have included to indicate their location or activity. Calendars contain Calendar items, which may be part of months, recurring item lists, or to-do lists. Web Pages can contain textual data, standard HTML tags, links to other Web Pages, and any embedded object (internal or external to BRIDGE). We chose these three objects because they are widely utilized in BRIDGE and because they are very different in nature from each other. Providing users with information about User Lists contributes to social awareness, while action awareness can be facilitated with knowledge of events from Calendars and of changes made to visual representations of information in Web Pages.

User Lists

We had to consider the main information users hope to obtain when they look at the User List, and how the information can be represented in a condensed textual form. Users need to know with whom they can collaborate and who is currently available for collaboration with others or contributing to a project. Therefore, we reasoned that for our initial design, including information such as whether another user has an active status should be more relevant than what time a user logged in. However, some sort of indication as to how long a user has been available or unavailable might be useful. A user could infer how much activity one user has participated in compared to others or the likelihood that a user will soon become available.

The description for the item that represents the User List includes a list of active users, in which users who have been logged in longest are listed first, followed by a list of "idle" users, in which the users who have been active most recently or "least idle" are listed first.

Calendars

For objects dealing with social awareness, such as User Lists, simply representing some aspects of their status might be sufficient, but for a collection of events like a Calendar, we must give more information to be useful. Listing all of the items in a Calendar could prove to be too long and overwhelm a user. The list might contain too much information, forcing a user to sift through all of the items to find which are relevant. A recent items list displays what interests a user instead of the entire Calendar's status.

If users wish to be aware of "recent" items, do they wish to know about the last X number of items that have been modified in the Calendar? Or do they wish to know about the changes that have been made in the last X number of days? If "recent" is based on the number of changes, as in the first choice, we have the advantage of always being aware of how and when a Calendar changed, despite how and when the change was made. If "recent" is based on a particular length of time, we have the advantage of catching any changes if more than X number happen to occur in rapid succession. We run the risk, however, of reporting too little or too much information to be helpful because we are not specifying a particular number of items.

To solve the information issues above, we decided on the changes in the last X days because we found that users´ main concern was in the current situation. If a Calendar is not updated very often, the first method might give too much information and become cumbersome to read. Prospective users commented on liking the fact that this method would explicitly indicate whether any activity occurred during the period of time. The most useful value for X depends on how Calendars are used in the environment in question, and based on BRIDGE use in our lab, we opted for the last seven days.

The next issue to be addressed with Calendars is exactly what information a user might want to know about recent items. We decided that the title of the item was quite descriptive and might be the most useful detail for a user to know. As activity involves both actions and people, we also included which user was involved in the creation or modification of the item.

Giving the associated date of the item would also be useful in terms of giving the user a sense of activity over time. We chose when an item was added to or last modified in a calendar. We felt that knowledge of what items were being added or changed and when and by whom the items were added or changed gave a sufficient picture of overall activity. After discussion with prospective users, we chose to also include the time that the event was scheduled to occur because events may be scheduled far in advance and only knowing the time of the event might make an item difficult to locate on a calendar. However, we chose to let the time field remain as the time the Calendar was last modified, meaning that an event is added to the description and an item for a Calendar is reported as being updated only when the event is created or changed, not when the event occurs; our primary concern is still the activity done to the Calendar and not activity that the Calendar's content spawns.

Web Pages

To represent in part the current status of a Web Page, as with User Lists, would not be very useful, similar to why it would not be useful for a Calendar. Although Web Pages are more comparable to Calendars than User Lists, in that knowledge of changes made to Web Pages contributes more explicitly to *action* awareness, we cannot represent recent changes with the same methods as with a Calendar. Although the two may be similar in some respects, they are very different in what they represent and must be treated differently. A Calendar is basically a collection of items. To say that a Calendar has been changed means that one of these items has been added or (less likely) changed. The change is, in essence, the item. A Web Page, on the other hand, is not a simple container for subparts. The Web Page holds content which can be changed in non-trivial ways. To say that a Web Page has been changed is to say that additions, removals, or modifications have been made to any or all of the pieces of information in its content. The change could be miniscule or enormous.

Therefore, recent changes cannot be summarized and aggregated; we must describe only one change at a time. This method makes more sense when thinking of the way that these objects are edited. To edit a Calendar, a user would take barely a minute to make the simple addition of an item. To edit a Web Page, a user may type on the order of minutes or even hours. Each "edit" can contain several changes. These changes are are what users are interested in and much more informative.

Our task was then to determine what main parts of a Web Page could be changed and

would affect the document. The most obvious part is text. Knowing how much text was added, removed, or changed would certainly be useful in representing how much a Web Page has changed. The less obvious part was determining a method for measuring changes in the text that would make the most sense to users.

Using pages or lines as units of measurement would produce something with which users should be able to identify, as we most often speak of documents in terms of the number of pages or lines they contain. The problem is that a page or a line has no meaning for Web Pages since differing monitor resolutions and browser window sizes make measuring pages and lines difficult. Using paragraphs would be problematic in that they could be vastly different in size, and knowing that one paragraph was changed does not let you know how much it was changed, potentially leading to information hiding. Using characters for units would leave no room for ambiguity and would allow us to make calculations like percentage change, but this method would be too low level to mean anything to most users. Percentage change itself may be misleading because if a page is rather long, a change that one might generally consider significant gets downplayed if it is a very small percentage of the Web Page.

We decided to use sentences as units, segmenting the document with sentence ending characters such as periods, question marks, exclamations signs, and new lines. This technique gives the best idea of how much was changed but presents a problem in that certain "sentence ending characters" may often arise in places other than at the end of sentences. We also felt that sentences seemed to map more to people´s mental models of how we create textual data. Along with the number of changed sentences, the first few changes are included with the intention of giving an indication of the nature of the set of changes and serving as a cue to users as to where to find the changes if they choose to look at the Web Page in question.

An issue that often arises with Web Pages is that they are sometimes changed very frequently without any significant changes actually being made. Users may often proofread after saving and continually go back to make slight edits, such as fixing capitalization errors, adding punctuation, or adding or removing white space. We did not feel that these were significant contributions to activity, and informing a user every time one of these changes occurred might serve to annoy rather than keep the user aware.

Although we can make difference detection ignore case, punctuation, and white space, what about fixing typos? We considered having some threshold for change, in that a

page would not be reported as different from the previous version unless changes were made to more than a certain number of words or sentences. This idea was not implemented because the size of a change is not always proportional to its significance. A few words could change the meaning of a sentence, paragraph, or idea, but would not be easily distinguished from adjusting a few typographical errors or introducing a slight rewording that does not affect meaning. We also considered a time threshold, where changes that occurred within a short time frame would not be considered. We chose not to implement this idea as it may take only a few seconds to delete a large chunk of a page or paste a large chunk of information into a page.

For the items representing Calendar and Web Page objects, we made use of the "creator" field that the RSS specification allows. We inserted the name of the user who last modified the object. Although not all RSS readers support displaying this property, we felt it would be useful for users if they could see who made a change before choosing to inspect it. This method also allows users to sort items depending on who authored their change.

We also worked with some prospective users on formatting the descriptions so that relevant information would be evident. To make the descriptions easy to read, we organized them with techniques such as bolding the measures of change for Web Pages and the names of the items added or modified for Calendars.

To gauge if the design for our RSS based notification system was achieving its goals, we again obtained input from potential users. They mostly felt that the descriptions were a reasonable high-level indicator of how much an object had changed. For Web Pages, the quantitative measure of text, links, and objects changed proved most useful to the users. They also indicated that the information in the descriptions would be enough to convey whether they needed to look at the actual object in detail. Some users relied on the number of sentences, links, and embedded objects that were changed, while others relied more on the first few changes to make the decision. They also found knowing whether a page was new or simply modified to be useful in determining the significance of a change as well. Some of the prospective users we spoke with felt that the descriptions helped them to find the changes on a Web Page. They found that knowing where the changes began and roughly how much text had changed to be useful. One user mentioned that Web Pages tend to be organized in sections with titles and bulleted lists. She also stated that knowing actual lines that changed pointed her to the correct sections. Most of the users, however, felt that

knowing what a change was did not make them any more likely to find the change. Interestingly, one of these users did resort to using the web browser´s search feature, so he did in fact make use of the changes presented in the description. We feel that in practice, familiarity with a particular Web Page will facilitate users in finding changes.

Implementation:

We implemented our RSS notification system and created an RSS feed to report changes on objects associated with the Geo-Collaborative Visualizations project mentioned in our scenario. The figure below shows how changes to Web Pages and Calendars appear in the feed using Safari´s built in RSS reader. The first description shows the changes made to a Calendar in the last seven days. For each change the title of the item, the time the event is scheduled for, the date that each change was made, and the user who made the change are shown. The second description shows the changes made to a Web Page. It includes the number of sentences that have been added, removed, or changed and the first three different sentences. The number of links and embedded objects added, removed, or changed and the first changed (not shown in the figure) are also shown, followed by the user who made the changes.
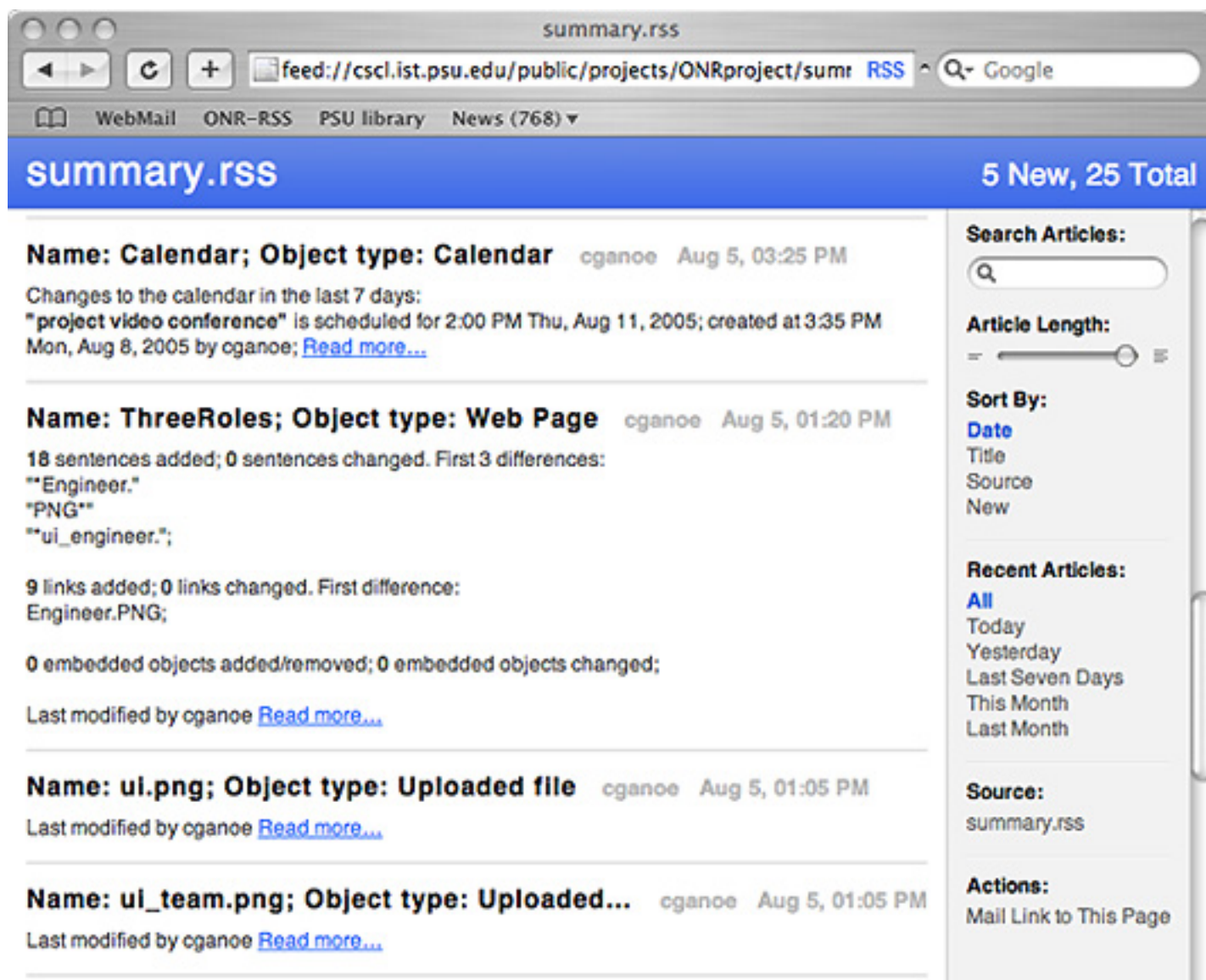
Figure 1: Items representing BRIDGE objects in an RSS feed, as shown in Safari´s built-in RSS reader.

Some additional issues came up that we did not address in the design for our initial version of the notification system. In practice, we noted that sometimes when meetings were held, one user would log in, but another user (or several users) would edit the pages. The RSS feed gets information from BRIDGE, which records the editor of the object as the one that logged in. We cannot edit the feed to act any differently, but perhaps BRIDGE could be changed to where a log in session could belong to multiple users. A simpler solution may be to create user accounts for groups.

Another issue that arose was the usefulness of the User List descriptions. Knowing who is online could indicate productivity or at least interest if we knew that they were viewing objects related to the project. The User List displays users who are logged in

anywhere in BRIDGE. Perhaps this arrangement could be changed so that we could determine if the user was viewing one of the files that our RSS feed was watching.

Multiple users said that they would like the ability to go to a Web Page and see all the changes highlighted. This feature is present in other software applications users are familiar with, for example MS Word. This feature would undoubtedly be useful but falls outside of the scope of this project. Perhaps the way BRIDGE works could be changed to allow for this feature, as it would be useful to use alongside the RSS technology.

Conclusion:

Through implementation and user comments, we have found our RSS notification system to be fairly successful at accomplishing its goals. The system proved very useful as an activity summary and let users know at a glance which objects were being changed and relatively how big a change was. This system is particularly useful to users in supervisory positions, as their main concern may be the high level activity and not the specifics of the changes. The system may be more limited in facilitating change inspection. The quantitative measures of change were valuable indicators to users in determining that a change to an object was significant and worthy of inspection. Having a direct link to the object in question was also useful and time saving. However, the RSS feed did not always prove helpful when trying to find the changes on a page.

As RSS use becomes more widespread, we hope that more people will take advantage of its capability to do more than just report news headlines. Due to the fact that the nature of our proposed use, reporting changes to objects containing data, differs so much from reporting new articles, to be most effective, the RSS notification system could be used along with other change notification systems, such as those which highlight edits made to a document.

Acknowledgements:

References:

1

Beged-Dov, G., D. Brickley, R. Dornfest, I. Davis, L. Dodds, J. Eisenzopf, David Galbraith, R.V. Guha, Ken MacLeod, Eric Miller, Aaron Swartz, and Eric van der Vlist. RSS 1.0 specification: RSS-DEV Group, 2000.

2

Cadiz, J.J., Venolia, G.D., Jancke, G., and Gupta, A. Designing and deploying an information awareness interface. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW 2002),* (Nov. 16-20, Luiziana, EUA) 2002, pp. 314-323.

3

Carroll, J.M., Neale, D.C., Isenhour, P.L., Rosson, M.B. and McCrickard, D.S. Notification and Awareness: Synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies.* 58, 5 (2003), 605-632.

4

Cutrell, E., Czerwinski, M. and Horvitz, E. Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance. In *Proceedings of IFIP Conference on Human-Computer Interaction (INTERACT 2001)* (July 9-13, Tokyo, Japan) 2001, pp. 263-269.

5

Gordon, T. F. An Open, Scalable and Distributed Platform for Public Discourse. *Informatik.* 2 (2003), 232-234.

6

Harrison, B., Ishii, H., Vicente, K., and Buxton, W. Transparent Layered User Interfaces: An Evaluation of a Display Design to Enhance Focused and Divided Attention. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1995),* (May 9-11, Denver, CO.)1995, pp. 317-324.

7

McCrickard, D.S., Catrambone, R., Chewar, C. M., and Stasko, J.T. (2003). Establishing Tradeoffs that Leverage Attention for Utility: Empirically Evaluating Information Display in Notification Systems. *International Journal of Human-Computer Studies.* 8, 5 (2003) 547-582.

8

McFarlane, D.C. (1999). Coordinating the interruption of people in human-computer interaction. In *Proceedings of IFIP Conference on Human-Computer Interaction (INTERACT 1999)* (Aug. 30- Sept. 3, Edinburgh, Scotland) 1999, pp. 295-303.

Biographies

Kayre Hylton (**khylton@andrew.cmu.edu**) received her BS in Computer Science from the Georgia Institute of Technology, and is currently pursuing her Masters in Human Computer Interaction from Carnegie Mellon University. In the summer between her degrees, she did research at Pennsylvania State University, in the Information Sciences and Technology's Laboratory for Computer Supported Collaboration and Learning.

Mary Beth Rosson (**mrosson@ist.psu.edu**) is a Professor of Information Sciences and Technology at Pennsylvania State University. She is a co-Director of the school's Laboratory for Computer Supported Collaboration and Learning.

John Carroll (**jcarroll@ist.psu.edu**) is the Edward M. Frymoyer Professor of Information Sciences and Technology at Pennsylvania State University. He is the other co-Director of the school's Laboratory for Computer Supported Collaboration and Learning.

Craig Ganoe (**cganoe@psu.edu**) is a Senior Research Associate at Pennsylvania State University, in the Information Sciences and Technology's Laboratory for Computer Supported Collaboration and Learning.