The Friendbo Attention Bar: Improving Engagement with Web Feeds on Social Networking Sites

Greg Bigelow

A senior thesis submitted in partial fulfillment of the requirements of the degree of Bachelor of Science With College Honors

Computer Science & Engineering
University of Washington

Abstract

I describe a method for providing a global overview of the content of a web feed by annotating its scrollbar. These annotations enable users to notice where activity is occurring throughout the feed and quickly navigate to places of interest. This provides a general, elegant, and extensible solution to many of the frustrations encountered in navigating dynamic feeds of information. I discuss techniques for measuring interaction with feed items and making that information actionable to users in the form of annotated scrollbars. I argue that annotated scrollbars offer a significant improvement over current ad-hoc techniques aimed at overcoming the limitations of feeds. We provide a demonstration of annotated scrollbars and supporting infrastructure in the form of Friendbo, a communication and social networking application of our own design.

Presentation of work given on June 1st, 2012

Thesis and presentation approved by:

Date:
Contents

1 Introduction ........................................................................................................................................ 3
2 Feed Layouts .................................................................................................................................. 3
3 Existing Solutions ............................................................................................................................ 6
   3.1 The Root of the Problem .............................................................................................................. 9
4 The Friendbo Attention Bar ............................................................................................................. 9
   4.1 Attention as a Common Currency ................................................................................................. 9
   4.2 Gathering Evidence of Attention ................................................................................................ 10
   4.3 Attention Bar Design .................................................................................................................. 11
5 Analytical Evaluation ...................................................................................................................... 15
   5.1 Suggestions for Empirical Evaluation ....................................................................................... 17
6 Conclusion ....................................................................................................................................... 18
Acknowledgements ............................................................................................................................ 18
References .......................................................................................................................................... 19
1 Introduction
A feed layout, in which content streams are displayed on a single page, is a popular display format for mobile and web communication applications. This is especially prevalent in social networking applications such as Facebook, Twitter, and LinkedIn. New content, in the form of feed items, is usually inserted into the top of the feed as it becomes available. Past items are made visible by scrolling down the feed. Despite the usefulness of the feed layout, it is limited by the fact that one can only have a small fraction of the overall feed scrolled into view at any one time.

Current solutions to this problem involve adding additional widgets into an application. These widgets allow for different views onto the content in the feed. One such widget is a list of notifications highlighting feed activity that is assumed interesting to users. Widgets like this are not necessary if we can provide users with a global overview of the feed.

I argue that an elegant and practical overview of the feed can be made by measuring the total amount of recent attention given to all items on the feed and then abstracting that measurement onto a modified scrollbar. Since this addresses the core problem with feeds, that we only ever have a local view onto feed content and activity, it suggests itself as a simultaneously more powerful and more general (and hence extensible) solution to a variety of specific challenges with feed views. The following thesis argues that this is the case and presents a working implementation of these ideas in the form of the Friendbo web application.

This argument and demonstration is divided into three parts. First, I describe four specific problems with feeds that are rooted in the lack of a global view. Existing solutions to these problems are examined and I suggest that they could be replaced with an annotated scrollbar. The existing solutions provide multiple local views onto the feed where a single global view would excel. Second, I present the details of our implementation of this view and discuss its theoretical foundations in past work on annotated scrollbars and social translucence. Finally, I demonstrate how our implementation provides specific solutions to the four identified problems with feeds.

2 Feed Layouts
In this section, I will describe the Friendbo web application and discuss the shortcomings of feed layouts. There are four specific challenges with feed layouts: finding new content, knowing who is listening, identifying the important stuff, and developing presence. These will be examined in turn.

2.1 The Friendbo Web Application
Friendbo is a web application that supports online asynchronous communication. Its central innovation is the use of a selector widget which allows for each message to be scoped to an access list of arbitrary size in a short period of time. Thus, each message can be made visible to a unique subset of people (see Figure 1). This approach stands in contrast to the more static scoping options available on social networks such as Facebook.
and Google+, where users are required to maintain lists and ‘circles’ (lists in disguise), respectively, that specify groups of people with different possible permissions.

Messages in Friendbo are displayed on a personalized feed for each user. This feed view shows all messages that the current user has access to. New messages are shown at the top of the feed and feed items are ordered chronologically. The feed can be explored in the vertical dimension by scrolling up and down the page.

Prominent examples of other applications with feed layouts are Google+, Facebook, and Twitter. The feed is especially well suited to displaying continuously updated streams of information. New content can be easily loaded into the top of the page. In particular, the feed is useful when the information being displayed does not require a response from the viewing user. Most Facebook status updates and tweets, for example, offer the possibility of interaction without implying that it is required. A feed allows a large amount of potentially useful information to be quickly browsed by the user. If a user finds something of interest, they can interact with that content without leaving the feed, a significant savings of time and effort when contrasted with the necessity of clicking through an index to access individual items of interest, such as with most email applications.
2.2 Problems with Feed Layouts
Despite the ubiquity and usefulness of the feed layout, it is not without significant shortcomings. Four important difficulties with feeds are described below.

Finding New Content
In a typical feed layout, Friendbo included, new content is loaded at the top of the feed. However, most applications want to support some form of interaction with feed content. For example, Friendbo users can reply to posts and have their replies shown below the original post. Facebook ‘likes’ are an example of a more lightweight interaction with feed content, insofar as it requires less effort by users.

The problem then becomes a matter of enabling awareness of new content, whether in the form of actual utterances or more subtle interactions with already existing feed items. It is not usually practical to simply move posts that were interacted with to the top of the feed. First, if there was a lot of interaction, updated old posts will crowd out any new content. The leading edge of a feed, usually the top, is a very small place compared to the total length of the feed. Only so much content can be displayed there at any one time. Second, continually reshuffling the order of the feed can be confusing. This prevents users from gaining a sense for the relative location of feed items. With little sense of order, successful navigation is often prevented. In experimenting with this approach in Friendbo, we found the shuffling significantly confusing even when the volume of posts was low. In contrast, a consistent ordering of posts, particularly if in chronological order, allows a user to more quickly browse for posts that they have seen before simply by knowing the general location of the post in the feed relative to other content.

Knowing Who Is Listening
Knowing who has received and processed content that you have created or shared is a problem for almost any online communication mechanism. It is particularly pronounced, however, in asynchronous communication systems employing a feed layout. There are two primary reasons for this. First, the asynchronous nature of the system prevents users from mutually acknowledging the other’s presence. With chat systems, in contrast, both users can generally see that the other is online. When a message is sent, there is thus a fairly good chance that it has been seen. And the synchronous nature of the chat medium allows for quick responses that can validate this hunch. Second, feed layouts are usually employed when content is multicast, or sent to many different users simultaneously. This is a problem because multicast systems usually result in a very large amount of non-priority messages available to be viewed by each user. As discussed above, this is a situation for which feed layouts are particularly beneficial.

Note that it is potentially useful to understand not just who is listening to one’s own messages, but who is listening to other people’s content as well. In the former case, users generally want to know that their messages have been received, for reasons of both practicality and pleasure. For example, an average user would like to know both that their idea for a project had been understood by other team-members as well as know that their joke had been enjoyed by their friends. In the latter case, it is helpful for users to gain a
sense of which content is well-received and enjoyed by a common audience. At a party or conference in the physical world, one can get a sense for the potential audience and what kind of content is generally well received by observing how people respond to the things said by others as well as the things said by oneself.

In general, having a sense of who is listening is a precondition for conversation [2, 9]. Indeed, “speech acts take effect by virtue of public declaration — by mutual knowledge of hearer and speaker that the act has been made. This is especially obvious in the case of declarations and expressives (e.g., an apology muttered but not heard is not an apology), but is equally true of the others” [13].

**Identifying the Important Stuff**

A central aspect of a feed is that all content goes into one place, namely, the feed itself. It is difficult to differentiate a particular item as especially important and in need of a response. As discussed above, a feed is generally used in situations in which there is a large amount of low to medium priority content that needs to be displayed. Since there is such a mass of content, any particular high-priority item stands a significant chance of being lost in the melee. A high priority item might be a question about work for which you require an answer. These are items for which one would generally send an email or make a phone call. However, this would deprive users of the particular advantages of using Friendbo and other web-based applications. Thus, such applications must devise a means of differentiating the high-priority items from the rest of the feed. Doing so in a manner that does not introduce large amounts of complexity into the system is a non-trivial task.

**Developing Presence**

In most communication applications with feeds, each user’s feed is unique. Individual feed items may be common across multiple feeds, as is usually the case, but the composition of any one feed is almost certainly unique. As such, the feed can seem a lonely place. There is no sense that other users are connected to a shared environment, as one might experience in a chat room. This problem is pronounced when communication is asynchronous [7].

Since so much feed content is shared across multiple users, it seems within reach, if perhaps difficult, to provide mechanisms to allow users a sensation of being in the same place at roughly the same time. Providing for this sensation is what is meant by developing a sense of presence.

### 3 Existing Solutions

Current solutions aimed at addressing these shortcomings are primarily ad-hoc additions separate from the feed itself. Facebook, for example, maintains several different lists, or miniature feeds, meant to augment the primary feed. Many of the techniques Facebook employs are common across different popular communication systems.
A popular solution to identifying new content in the feed is through the use of a notification list. In Facebook, for example, users are notified at a separate area at the top of the page whenever other users interact with their content or content that the user has also interacted with (see Figure 2a). The notification takes the form of a button that glows when there are unseen notifications. The interactions that caused the button to glow are then presented as a dropdown list.

This has the obvious limitation of not scaling well. The miniature feed that is the notification list is subject to the same limitations in how much information can successfully be displayed. In practice, this is generally not a problem because the number of notifications is limited. This is achieved, however, by significantly restricting the amount of new content that users are notified of. For example, you would not receive a notification of a vibrant conversation occurring on some feed item if you had not commented on that item yourself.

This particular limitation also has ramifications for discovering who is listening to other people’s contributions. While you receive notification when attention is located on one’s own content, you have no way of knowing, short of scrolling through the feed in search of it, how other people’s content is received.

A second list used in Facebook that aims to address some of these problems is the Facebook Ticker, which shows feed items in real-time (see Figure 2b). This is yet another miniature feed. While its purpose is not totally clear, it is presumably in part meant to encourage serendipitous discovery of other’s activity and engender a sense of real-time co-presence in a shared online space.

Most relevant to our discussion is that the Facebook Ticker provides information about where current activity is located on the feed. For example, if a post is receiving a large number of likes or replies, each of those likes and replies will likely show on the user’s ticker and the user can then navigate to the place on the feed that the conversation is taking place by clicking on the ticker item. This has the limitations that the ticker is just as space-constrained as the primary feed and that it is necessary for users to be present in real-time in order to reliably receive the indications of attention.

The Facebook Ticker is also interesting in so far as it provides a de facto list of who else is currently active on the site. This is in addition to the more explicit list of logged-in users shown in the chat widget (see Figure 2c). An almost identical chat system is used by both Gmail and Google+. Both the ticker and chat list work to encourage a sense of real-time co-presence on the system. They reveal that you are on the site at the same time as others. However, they are limited in two significant ways. First, they only reveal real-time information. A more challenging, and thus interesting, goal is engendering a sense of presence in communication systems with a blended-synchrony model [7]. Second, they are divorced from the feed itself. They do nothing to give a sense that the feed is a distinct place populated by many different users.
Figure 2a. The Facebook Notification list, upper-left, can be thought of as a filtered version of the primary feed meant to highlight a very few selected items.

Figure 2b. The Facebook Ticker, shown in the upper-right, is yet another miniature feed with a different view onto the primary feed. The main difference is that it is a real time version of the main feed and appears

Figure 2c. Facebook shows two separate lists of users currently on the site. The list on the right shows both images and names of users while the list on the lower-left consists solely of images.
3.1 The Root of the Problem
There is a common thread between the four identified difficulties with feeds. As will be elaborated upon throughout the remainder of this paper, the difficulties are inherent in the fact that the main view onto a feed is extremely local. To be clear, the only part of a feed visible is the part currently scrolled into view. Since feeds are generally long, or, in the case of ‘infinite feeds’, essentially never-ending, users only ever see a small part of the entire feed at any one time. In essence, the typical feed is a large territory without a map.

With no global view of the entire feed, it is inevitable that the previously mentioned difficulties arise. This is a case where “the difficulty of digital communication and collaboration stands in stark contrast to our ability to communicate and collaborate with one another in the physical world” [7]. In a living room in the physical world, it is quite easy to identify new content, understand how your utterances are received, separate critical messages from non-critical ones, and feel a sense of shared presence. The primary reason that this is the case is that people in the living room have some access to an image of the room as a whole. In other words, the room is socially translucent to the actions of others [7]. A quick glance around the living room might reveal three active clusters of people. One cluster seems particularly animated, indicated by large hand gestures and loud voices by some of the participants. You walk over to that cluster and your presence is recognized by brief sideways glances. You say something aloud, and the utterance is acknowledged with a few nods of the head. You might be overhead by people from a nearby cluster, and perhaps they will also join in the conversation.

A typical feed is like attending a party in a living room but only being able to look or hear a few inches in front of you. Existing solutions to this problem involve presenting users with lots of lists separate from the feed itself (lists of notifications, currently active users, real-time feed items). Their discussed limitations stem from the fact that they still only provide local views onto the feed. There is a proliferation of views in the form of list widgets, but still no ability to get a sense for the feed as a whole.

4 The Friendbo Attention Bar
If the problem is one of having only a local view onto the feed, the simplest solution is to provide a global view as well. The Friendbo Attention Bar is an attempt to provide such a view. The bar treats all communication activity as “attentional acts”. The attentional acts performed on a given post are accumulated and presented on the bar. Thus, the bar serves as an abstraction of the feed as a whole. It is a map of where attention has recently been located. As such, it provides a single solution to the aforementioned problems with feed communication systems. While the current implementation is specific to the Friendbo application, it would be usable on a variety of different feed communication systems with little modification.

4.1 Attention as a Common Currency
Before looking at the details of the Attention Bar, it helps to understand why we ought to posit attentional acts as the activity of interest. The most obvious alternative is to
consider only utterances. In the case of Friendbo, this would be specific posts and written replies to those posts. However, utterances are a small part of what makes a successful conversation. In fact, the term conversation is most useful when used “to indicate a coordinated sequence of acts that can be interpreted as having linguistic meaning. It need not be a spoken conversation, or even involve the use of ordinary language” [13]. In practice, large parts of both our online and physical conversations are acts that fall outside the scope of utterances. Facebook ‘likes’, nods of the head, turning away from the speaker – all are conversational acts that carry meaning.

If there are so many different ways to interact in a conversation, how can one hope to provide a simple representation of them? The key is understanding that all conversational acts are grounded in attention. All conversational acts, at some level, involve a giving or taking away of attention. This is not a particularly remarkable fact, as “an activity cannot be carried out without a sufficient allocation of attention” [6]. However, this does not mean that attention is trivial. In fact, as I will show, online conversations can be enhanced by providing more means of transmitting information about attention and making that information available at a global level. Attention measurements are useful both to those designing communication tools as well as those using them [12]. The Attention Bar makes these measurements available to Friendbo users in a simple, actionable way.

4.2 Gathering Evidence of Attention
To successfully show where attention is located on Friendbo, or any online communication system, we need to gather that information from users. Preferably, this information gathering will take place as a side-effect of the user’s normal behavior. In Friendbo, we have provided six different ways that users might give attention to a post.

1. Posting in the first place. This indicates attention spent by the poster and the hope that others give their utterance attention.

2. Being replied to with a written response.

3. Being read. This is currently indicated by readers clicking on a post.

4. Being ‘toasted’. This method is similar to a Facebook ‘like’, but allows for a wider range of possible reactions, positive, negative, or neutral. The key point is that this is a way for users to give attention to a post that is easier than writing a full reply but more substantial than simply indicating that they have read it. We suspect that the toast both provides information about attention allocation and encourages further gifts of attention by facilitating a gift-giving culture (in which the gift is one’s attention) [8].

5. People making themselves ‘active’. Each user with permission to view a post can be either active or inactive on that post. The precise semantics and provided features here are beyond the scope of this paper. The basic idea is that people can make themselves or others active on a post if they feel that the post is high-priority content. Users who are active within a thread receive special notifications when there are changes to the thread.
These notifications are built into the Attention Bar. In this way, users can easily stay current on their most important conversations. This will be discussed in more detail in the overview of the Attention Bar’s design.

6. People sharing with others. The current scope of a thread is visible to all users with access to that thread (see Figure 1) and each user can modify that scope, granting and revoking access at will. All access changes are recorded and displayed to users with access to the associated thread. If a user expands the scope of access so that more people can see a post, it is an indication that they found the content worth sharing with others.

For the purpose of simplicity, we treat each of these attentional acts as an integer score. More significant allocations of attention, writing a long reply, for example, are scored higher than less significant acts such as simply reading a post. The amount of attention that a post has received can therefore be expressed as a single integer. This integer is then further abstracted onto the Attention Bar.

4.3 Attention Bar Design
The Attention Bar is an implementation of “a technique called attribute-mapped scroll bars”, first described by Hill and Hollan [10]. In their implementation, every time a line of text was read or written, a kind of digital ‘wear’ accumulated for that line. The scroll bar for the text document is annotated to show that wear. “The width of individual wear marks is proportional to the largest magnitude of edits per line” [10]. Thus, a viewer could gain a quick sense of which parts of the document had been most heavily edited or most heavily read.

Rather than limit itself to just reading and writing, the Friendbo Attention Bar deals with the wide range of attentional acts described above. Importantly, the Attention Bar does not differentiate between utterances, lighter forms of replies, and reading. Instead, all such activity is grouped as attentional acts and is treated the same. Another key difference is that the target of attention is an entire post rather than individual lines. However, the basic mechanism of showing the allocation of attention as lines of various lengths is preserved.

The Attention Bar is a modified scroll bar (see Figure 3). Each post is represented as a line on the bar. The height of the line (the amount of space it occupies on the y-axis) is proportional to the amount of the feed it occupies. The width of the line is proportional to how much attention has been recently allocated to it. One of the benefits of annotating the scroll bar is that the map functionality and the navigation functionality is located in the same place. Specifically, attentional activity is immediately visible and it is “easy to get to those sections” on the feed by clicking on the bar itself [10]. Note that high-priority items (posts appearing in threads in which the user is marked ‘active’) are shown as black dots to the right of the mark for each post (see Figure 3). The dots are hidden when the high-priority item is marked as complete by the user. This is important because it provides a mechanism for highlighting high-priority items in the feed. This will be elaborated on further in evaluating the Attention Bar.
I wish I could make replies to threads that people are active on without them getting another email. Right now - turning everyone temporarily off, is way too cumbersome.

But there are cases where you do want people to receive emails. However, now that okays are becoming more prominent in the queue and bar, maybe emails are getting less and less necessary for those who are logging-in to Friendbo fairly often. So maybe the emails are starting to get in the way more than they’re helping? Not sure if the solution is to bail on them for now, but at least for me, their usefulness is fairly limited. What about for you?

A hidden solution: make it less cumbersome to turn people off. Why is it cumbersome?
- Cause you have to scroll up to the list of people, which is far away from the reply box at the bottom
- Cause they get a SECOND email when you turn them back on
Anything else?

We’ve talked about making the people scroll down with the page, storing the emails in the db so that people don’t get remailed.

Need some more heavy-duty data to test with. I remember you sent me a a copy of the main page data a while ago but I have since lost it.

Would it be possible to get that from either of you? Also, isn’t there some script you can run to read all the data in? What was that again?

Thanks!

Also - ok, yeah. Realize the okays are just implied from the data so they would show up anyway.

I just pushed a change where these wacky “advanced” features:
- okbar
- okqueue
- quickle creation

Figure 3. The Attention Bar can be seen on the left. Pink bars represent posts that the user made that have recently received attention but that the user has not seen the details of. The width of each bar is proportional to the amount of attention given its post. Black dots to the right of a mark indicate a high-priority item that the user has not yet marked as completed.
Recent Attention vs. Total Attention

Perhaps the biggest difference between the Attention Bar and the annotated scroll bar described in *Edit Wear and Read* is that the former only shows recent wear whereas the latter shows a total accumulation of wear over time. If we follow Hill and Hollan in drawing an analogy between wear in the physical world and the possibilities of wear in the digital world, then Friendbo’s metaphorical medium is sand. Wear accumulates but is erased by the wind and water of time. In practice, in computing a post’s attention score, we only consider attentional acts that occurred within the last N hours, where N might typically be 24 or 48.

This more closely models how awareness of attention in a physical location might take place. One either senses attention in real-time (see where someone’s eyes are pointed) or observes temporary signs of attentional activity (see lots of wine glasses left on the kitchen counter). The key is that one needs to be present in order to observe where attention has been recently allocated. A currently untested hypothesis is that configuring things in this way will increase a sense of shared presence on the feed, since seeing another person’s activity is an indication that you are there at roughly the same time. This hypothesis relies on the assumption that it is even possible to engender a sense of presence when communication is asynchronous. We suggest that an extended present is a suitable substitute for a discrete present. Drawing on the work of Donald Schoen, we can think about this extended present as an “action present, the zone of time in which action can still make a situation” [11]. To the extent that the attentional activity shown on the bar lies within this admittedly hazy ‘action present’, we suggest it is still capable of engendering a sense of co-presence among Friendbo participants.

Admittedly, there is a strong argument to be made that it would be beneficial to have the marks on the Attention Bar show the total accumulated attention over all time or to show both the total accumulated attention and recent attention. The latter could be accomplished by showing a second dimension on the Attention Bar. For example, the length of each mark could represent the total amount of attention its respective post had received while its level of opacity could represent the amount of recently allocated attention. This would be a trivial extension to our current implementation, and for some users, the extra information may be worth the added complexity. One might also introduce the ability for users to change N, the length of the considered ‘action present’, on the fly using a slider widget or dropdown menu.

These approaches are worth looking into. However, I did not feel them necessary because of the nature of feeds. As will be explained later, the Attention Bar already includes a mechanism for identifying high-priority posts, so those stand no risk of being lost. The other content includes a large amount of low and medium priority items. If a user did not log in to Friendbo for a few weeks, much of the attention given posts during that period of absence would not be visible on the Attention Bar upon their return. This is not concerning because the high-priority posts from that time period will be identified through a separate mechanism and there will most likely be a large number of more recent low and medium priority posts that will stand out on the Attention Bar. To experience the present state of attention, a user must themselves be present.
Showing the Details of Attentional Acts

The marks on the Attention Bar are abstractions of specific attentional acts. They are most useful when acting as signposts on a map. However, users generally would like to know the details of the acts being represented. Showing such details also provides users a means of learning what the length of each mark stands for. They come to associate the details of the attentional acts for each post with the length of the mark for that post. There are a variety of ways to present such details. While the specifics are not important for the scope of this paper, our current Friendbo implementation lists the recent attentional acts (those that are influencing the length of the mark) to the right side of each post in the feed (see Figure 4). A typical use case involves a user seeing an interesting mark on the Attention Bar, clicking on it to navigate to that point in the feed, and then seeing the detailed attentional acts that caused that post to stand out. To engender a sense of presence, the attentional acts are all described in the present tense.

Figure 4. The list of specific attentional acts is shown in the upper-right. It comes into view when users hover over a post.

4.4 Social Translucence and the Attention Bar

The development of the Attention Bar has an extensive conceptual debt to work on social translucence. A socially translucent system is one in which there is “visibility, awareness, and accountability” [7]. Importantly, social translucence is quite different from social transparence. A translucent system is one in which there is visibility without total visibility. Furthermore, translucence implies a governing physics that underlies what is visible – and the extent of that visibility [7]. For example, physical space is socially translucent insofar as you can hear people talking in a room, and it is far easier to hear people that you are physically closer to and impossible to hear people who are very far away. Erickson and Kellogg suggest that the high degree of social translucence in physical space helps enable the richness of in-person conversations. They further argue
that computer applications for communication can be improved by carefully increasing
the degree of social translucence in the presented communications space [7].

The Attention Bar provides a means of providing extra social translucence. Kellogg and
Erickson note:

*Conversation is social in two ways. First, people speak to an audience. Speakers
notice how their audience is reacting and steer their remarks appropriately: nods and
eye contact convey one message; questions and furrowed brows another; yawns and
fidgeting still another. Second, conversation is social in that people portray themselves
through conversation.*

The Attention Bar promotes these social properties of communication. First, it allows
posters to register input from their audience. This both confirms that there is in fact an
audience and gives the speaker feedback as to how their utterances were received.
Indeed, this feedback takes the place of indications of attention (nods and eye contact for
in-person conversation and ‘likes’, ‘reads’, ‘shares’ for online communication). Second,
providing greater awareness of attentional activities on the bar promotes the ability for
individuals to portray themselves through conversation to a greater extent than they could
with posting capabilities alone. Indeed, as previously argued, a large component of a
conversation is attentional acts that are not utterances. By providing more ways for
people to show attention to other people’s utterances and then making that attention
globally visible on the Attention Bar, users can develop images of themselves as active
participants in a wide variety of conversations - even if they are not writing actual words.

Granted, attentional acts are presented on the Attention Bar in a much more abstract
manner than they would be in physical reality. This follows an approach in which “social
information” is presented “in ways that are not closely tied to their physical analogs” [7].
However, a fair amount of research shows that abstractions of such information, such as
text, actually perform quite well at conveying social information [7]. The marks on the
Attention Bar can be thought of as “social signifiers” [10]. They are signals “in the
physical or social world that can be interpreted meaningfully” [10]. Importantly, and as
is the case in Friendbo, the signifiers can be accidental byproducts of the world. Users
never explicitly increase the length of the marks on the bar. Rather, they grow and shrink
as a function of time and attention.

5 Analytical Evaluation
We can get a sense for how well the Attention Bar addresses the originally identified core
problem with feed communication applications by analyzing its approach to resolving the
four main challenges.

Finding New Content
The main difference between the Attention Bar and notification solutions to the problem
of finding new content is that the Attention Bar provides implicit notifications for all new
content. In contrast, a notification system based around a list, or feed-in-miniature, is
limited by the same space constraints as the overall feed. The greater amount of information presented on the Attention Bar comes at the cost of a higher level of abstraction. Whereas a notification system notifies users of explicit kinds of interactions, the Attention Bar only shows attention in aggregate. To see specific items, users must navigate to the place in the feed that the attentional activity occurred. This extra navigational cost is assuaged by having the Attention Bar enable both the viewing of attention and instantly navigating to different parts of the feed.

A second strength of the Attention Bar in relation to finding new content is that it recognizes that attentional acts can be just as important as new utterances. For example, a user might be just as, if not more, interested that another user’s post received an extremely large number of reads than if that same post simply received another comment. By treating utterances and attentional cues in common, the Attention Bar succeeds in providing more information about where all users’ attention has been recently allocated.

**Knowing Who Is Listening**

The notifications on the Attention Bar provide a large amount of information related to who is listening (and thus participating in) the ongoing conversations. As just discussed, the abstractions on the Attention Bar are a suitable substitute for explicit notifications. However, the Attention Bar goes further in presenting more information regarding who is listening to who else. This information is valuable insofar as it allows users to gauge not just who is listening to them, but who is listening in general. And by observing the specific ways in which people are doing that listening, users can get a sense of what kind of, and how, conversations proceed within various social circles.

Furthermore, presenting this information in aggregate across the entire feed does a better job of showing where attention in general is located. Users can get an immediate, intuitive sense of where the entire conversation is. For example, if there are number of very wide marks on the Attention Bar a quarter of the way down the screen, users know that there is a lively set of conversations concentrated in that region. If they were looking to say something, they now know where there are likely to be listeners. Furthermore, coordination among conversational participants across threads and over longer periods of time requires the constant updating of a common ground [1]. By providing more information about what all participants have been attending to across many different feed items, we expect the Attention Bar to do a better job of providing common understanding of long-running conversations.

**Identifying the Important Stuff**

Differentiating between feed items that require a response and those that do not is a difficult task. One popular solution is to place high-priority conversations into separate channels. For example, a Facebook user might send a message or post to a specific person’s wall if they want to increase the chances that they will get a response from that person. However, this is limited insofar as it requires, at the minimum, an extra feed.

In contrast, the approach taken in Friendbo is to place high-priority messages in the main feed and then use the Attention Bar to mark the locations of those high-priority items.
using small dots (see Figure 3). Once a user has marked a high-priority item as taken care of, the dot disappears. This technique is applicable across a wide array of feed applications. Rather than creating separate feeds for high-priority items or creating yet another feed of notifications for those items, everything can be placed in a single feed and that feed can then be globally annotated using the scrollbar. Once there is a map of the feed, the number of ways it can be productively used is limitless.

**Developing Presence**

This is the most difficult challenge to evaluate because it is an attempt to measure the rather subtle feeling of being in the same place as others. We can, however, examine what capabilities the Friendbo Attention Bar provides that current feed systems do not, and explore the consequences of those capabilities. For one, the activity shown on the Attention Bar is related to the feed itself. In contrast, systems that simply show if another person is online are providing a sense of presence, but in a way that is totally divorced from the feed. These systems are also lacking in that they are unable to take into account recent asynchronous activity. This is unfortunate because asynchronous or blended-synchronous communication tools can often be more useful than their synchronous counterparts [5]. My hypothesis is that a sense of presence can be encouraged by making the feed feel more like a space. The proposed means of doing so is through the Friendbo Attention Bar. The strength of the bar is that it provides a global overview of an entire space and how attention is currently being allocated throughout that space.

A second reason I think the Attention Bar might increase a sense of presence is that it allows for a greater degree of serendipity in discovering the presence of others. While interactions on most communication feeds are inscribed permanently on the feed itself, the Attention Bar only takes into account recent activity. Users learn that marks on the bar indicate the presence, or near-presence, of people they know. By clicking on the bar to navigate to highly-occupied spaces on the feed, users can then serendipitously discover the real-time, or almost real-time, presence of others. In mirroring physical reality, one’s traces of presence are quickly erased over time. This leads to a smaller number of indications of presence, but increases the importance of each such indication. We conjecture that by more closely following physical reality in this regard, we might replicate some of the sense of co-habitation that arises in the physical world. An advantage of Friendbo’s design is that the Attention Bar can show only recent activity while the feed itself can permanently preserve utterances (posts and replies) for future reference. We have our cake (keep digital records) and eat it to (let attention act out its inherently fleeting nature).

### 5.1 Suggestions for Empirical Evaluation

Empirical evaluations of the Friendbo Attention Bar would be useful for providing further evidence of its efficacy, or lack thereof, and generating ideas for improvements to the design. I suggest two possible such evaluation methods as a means of motivating future work and provoking ideas for other possible techniques.

The first suggested study would measure the ability of the Attention Bar to help users find new content. Users would be divided into three sets. The control set would use...
Friendbo with no Attention Bar or notification mechanism. Another set would use Friendbo with a simple list-based notification system that alerted users to all new activity not visible at the top of the feed. And the final set of users would use Friendbo with the Attention Bar. The relative usefulness of the notification system and of the Attention Bar could be approximated by recording how often users navigate to lower areas of the feed and how much time they spend viewing the content there.

A second possible study would use the same three different groups to examine how well the Attention Bar provides feedback on how other people are responding to the user’s content. This time, however, the notification system would be modified to only list activity involving interaction with the user’s posts (‘toasts’, reads, replies, ‘activations’, and shares). A dimension of success is how much pleasure users experience when discovering how their content has been received. This can be approximately measured by recording how likely it is for users to repeat the actions that lead to the reward. To be clear, the action would be either scrolling through the feed to uncover responses, clicking a button to reveal the notification list, or clicking on the Attention Bar. The reward is the revelation of the details of the attention that was given to the user.

Of course, the success of the bar might vary significantly depending on the content of the feed. However, it is quite possible to test the bar on several different kinds of feeds. This is made practical by the fact that the bar is presenting information that is embedded in the feed itself. Thus, it is possible to write a series of browser extensions that applied Attention Bars to popular websites that utilize a feed layout, such as Facebook, Twitter, and Google+. They would parse the html composing the feeds, aggregate the information, and then draw that information on an inserted widget to the side of the page. The huge numbers of active users of these services suggests that getting even a small fraction of the users to try such a browser extension would result in abundant empirical data.

6 Conclusion
The Friendbo Attention Bar treats content feeds as large two-dimensional territories. By providing an omnipresent and dynamic map of that territory, several challenges inherent in feed communication systems can be addressed with a single, elegant solution. These ideas are applicable to many applications outside of Friendbo. Indeed, any feed system can potentially be enhanced through the use of attribute-mapped scroll bars. Our experience with Friendbo also suggests that these attribute-mapped bars can be made more useful by treating attention as the fundamental unit of conversational discourse and providing a variety of ways for users to express attention within the application.

Acknowledgements
I thank Michael Toomim, Travis Kriplean, and James Landay for so freely sharing their time, knowledge, and hugely unique perspectives. This work is made possible only by their ample support.
References


