

A look ahead at Google's exciting Wave project

CATCH A WAVE

Thomas Del Amo, 123RF

Part email, part instant messenger, part bulletin board, and part wiki, Google's new communication platform promises to transform correspondence into conversation. **BY MARTIN STREICHER**

Throughout the history of the Internet, a series of presentation and communication formats have occupied the spotlight. Early on, maintaining a homepage was all the rage. Next came the rush to email on free services, such as Yahoo!, Hotmail, and pobox.com, quickly followed by instant messaging, forums, vanity sites, MySpace, webcasts, and feeds. Recently, blogs and video were stars, at least until Facebook and Twitter upstaged the show. Interestingly, few of these forms are defunct; like print, radio, film, the telephone, television, and cable before them, each medium has been assimilated into mainstream use.

So what's next? According to Google, the answer is Wave. Part email, part instant messenger, part bulletin board, and more, Wave promises to transform correspondence into conversation. Moreover, because any number of users can participate in a discussion – what Google calls a *wave* – Wave promotes conversation into collaboration.

This article offers a quick look at the many facets of Google Wave. Wave provides components for end-users, site owners, and application developers, and given its lineage, one can expect something of a Twitter-like cottage industry to pop up to push Wave in all sorts of directions. As this story goes to press, Wave is available only to developers. If you're a software developer, you can download all the protocol and API documentation, and you can petition for your own account on a development box. Otherwise, open (albeit limited) public access should be available by the end of October – perhaps even as you read this. Public, unfettered access is expected some time in 2010.

Surf's Up!

Figure 1 shows Google's Wave client, which runs in any browser that supports HTML 5. At a glance, the features of Google Wave might remind you of the email and chat applications you already use. Indeed, Wave has many features in

common with those applications, but as you will learn in this article, the Wave interface offers some options you won't find in the average chat or mail tool.

The main window is divided into four regions:

- The organizer (top left) – Much like a desktop or web email client, the Google Wave application can save and collect waves. You can create a simple folder and amass content manually or create a folder to populate automatically via search terms.
- The address book (bottom left) – Like email and instant messenger clients, Google Wave maintains a list of your contacts. The superimposed green dot indicates the person is online and active in Wave. This presence cue indicates you can converse in real time. However, if any person is idle or otherwise engaged in another wave, your new wave is added to his or her inbox and marked to indicate new activity.
- Your waves (middle). The middle pane is a list of all your waves. In this case,

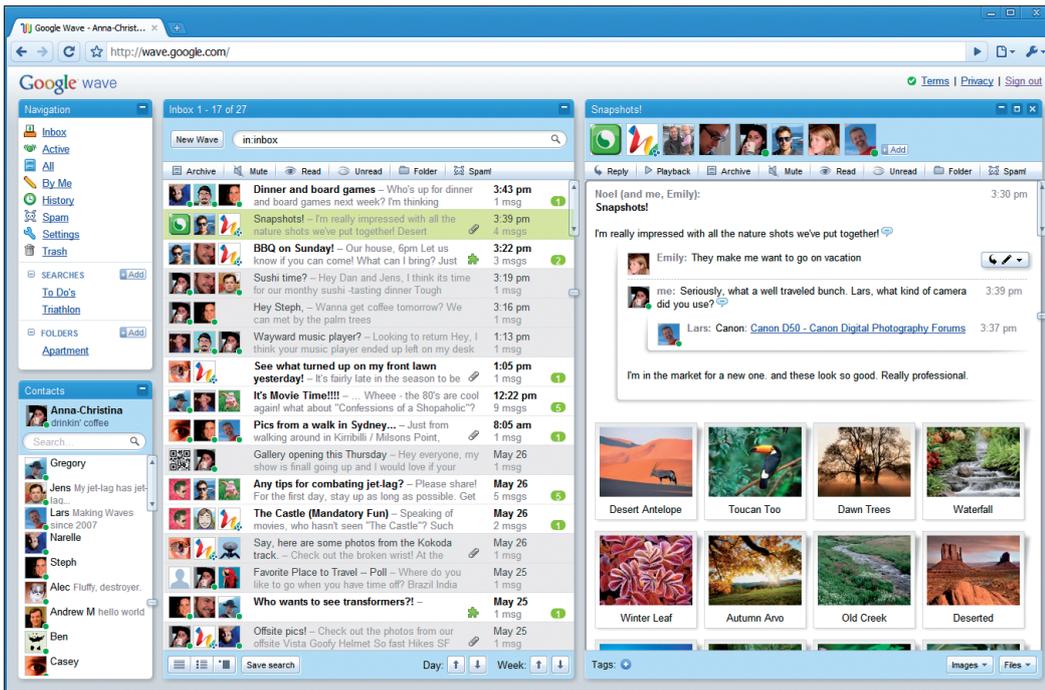


Figure 1: Google's Wave client running in an HTML 5-capable web browser.

the search term *in:inbox* in the filter box up top limits the list to waves in your Inbox. Of course, you can type any search phrase and produce a new matching list in an instant. The controls for *Day* and *Week* at the bottom add additional filters for the day or week the wave was started. Much like the "From" line of an email message, each wave shows a summary of its participants. And much like an instant messenger client (say, Adium) new activity on a wave is highlighted in green. The number shown indicates how many new, unread messages have been posted to the wave since it was last opened.

- Your current wave (right). The right-most pane is the currently selected wave. The list of people in the conversation is shown at the top. You can add another contact at any time simply by clicking on *Add*. Unlike an email message, a wave is a living document that can be amended by any participant at any time. You can drag-and-drop media into a wave, and you can chat back and forth at any position in the wave. A wave starts out like email, but also operates like a forum and can be edited collaboratively like a Google Doc. Figure 2 shows another wave with the latter additional features. The user Dan injected a question into the original post. Both Greg-

ory and Casey responded with individual answers to Dan's inquiry. Others also added comments at the bottom. Unless you specify the contrary, all content in a wave is exchanged with all other participants in real time, as you type. To send a message only after you have proofed it, enable draft mode until you are ready, then click *Done*.

Like other Web 2.0 content, you can tag a wave to organize content according to your own taxonomy. To add a tag, click the button at the bottom of the pane. Unlike other Web 2.0 content – and unique to Wave – you can also playback the creation of a wave to witness its genesis and evolution. Because a wave can be edited simultaneously by all participants, *Playback* is an essential feature.

Wave is as easy to use as any email or instant messenger client. Just click *New*, choose your contacts, and start typing.

Under the hood, Wave maintains a rich hierarchy of data to bundle messages into conversations, control access, and embed files and images. Wave comprises four essential models: the *wave*, the *wavelet*, the *blip*, and the *document*.

A wave must have at least two participants, although one can be a robot. (I'll describe robots later in this article.)

Each wave is composed of one or more wavelets. You can think of a wavelet as a thread of the conversation. Each wavelet is composed of individual messages called blips. A blip might be the sentence "Who wants Thai

for lunch today?" or a file attachment. Blips can also contain other blips, which allows for a collection of photos or a tangential series of comments to be managed as an atomic unit. The contents of a blip, a sentence or an attachment, is stored in the blip's *document*.

Counter to what you might expect, a wave does not enforce access controls. A wave is merely a container for wavelets.



Figure 2: A Google wave can act as forum, chat, and live document at the same time.

Instead, each wavelet maintains its own permissions. Thus, otherwise widely accessible content in a wave – say, a discussion held between all your friends – can embed private sidebars shared by just you and another.

Figure 3 shows the relationships between the four data structures. A wave has one or more wavelets. Each wavelet has two or more participants and at least one blip. And each blip contains a document, which can include text, binary data, and a hierarchy of other blips.

The Wave Extensions

In addition to text, images, other media, and human correspondents, a wave can also include *robots* and *gadgets*. Both the robot and the gadget are *extensions* to Wave, and each serves a special purpose. A robot is an automated participant in a wave. A gadget is a small application that's shared via a wave.

For example, Emoticony is a robot that replaces the text representation of an emoticon, such as the familiar :-), with its corresponding smiley image. Another robot is Rssybot. If you add it to a wave and enter the address of an RSS feed, the robot updates the wave each time a new entry is added to the feed. Rssybot turns a wave into an impromptu RSS reader.

A robot's purpose is to modify information in a wave or interact with its

human and electronic counterparts. A robot can also import and export content, connecting the wave with other online sources. Like any other contact, a robot has its own contact information and must be invited to a wave to engage in the conversation. Once invited, the robot begins to function and work autonomously. (Future releases of Google Wave will likely support automatic robot invites on the basis of some criterion.)

To compare a robot to a gadget, consider PlusOne. You can attach this gadget to any message in a wave and allow viewers to vote “thumbs up” or “thumbs down.” The gadget tracks the number of positive and negative votes. Checky the Checklist is another gadget. It records a list of tasks.

A gadget's purpose is collaboration. Once the gadget is embedded within a wave, all participants can access it and make changes to its content. However, a gadget cannot affect the other content found in the wave. Think of a gadget as a facilitator: A gadget improves the ongoing conversation. Gadgets that are available for use include a shared whiteboard for ad lib sketches, several turn-

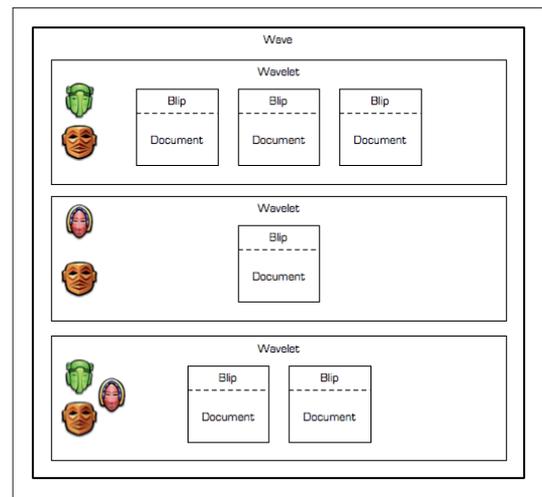


Figure 3: A wave contains wavelets that consist of two or more participants and at least one blip.

based games, and a team Sudoku challenge.

In addition to the functional differences between a robot and a gadget, the two extensions run in different environs. A robot runs on a server (for now, only Google App Engine); a gadget runs within the Wave client.

Waves, Waves Everywhere

Google Wave is compelling as it is, running in its own client; however, the goals for Wave are far more ambitious. With the *Wave Embed API*, a website can include a wave in situ, so that any site can use Wave for collaboration. These far-flung waves are all available from your Wave client, and in this way, you can track all your conversations from one place.

Listing 1 is a simple example of embedding a wave into a web page. You can find this code and additional documentation on the Google Wave Embed API project page.

Most of the HTML and code in Listing 1 is standard fare found in any page. The page loads the Embed JavaScript library (*embed.js*) and defines the function *initialize()*, which creates a view for the wave. The function creates a viewport for the wave (*new WavePanel(...)*), loads the wave from the Wave server (*loadWave(...)*), and assigns the wave to a region in the page. The string *waveID* is a placeholder. You must replace it with a true wave ID pre-assigned to you via your developer account. When the document finishes loading, the wave appears in the *div* element with ID *waveframe*.

Listing 1: Embedding a Wave in a Website

```
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
02 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
03 <html xmlns="http://www.w3.org/1999/xhtml"
04 xmlns:v="urn:schemas-microsoft-com:vml">
05 <head>
06 <meta http-equiv="content-type" content="text/html; charset=macintosh" />
07 <title>Google Wave Embed API Example: Simple Wave</title>
08 <script src="http://wave-api.appspot.com/public/embed.js" type="text/
09 javascript"></script>
10 <script type="text/javascript">
11 function initialize() {
12     var wavePanel = new WavePanel('http://wave.google.com/a/wavesandbox.
13 com/');
14     wavePanel.loadWave('wavesandbox.com!w+waveID');
15     wavePanel.init(document.getElementById('waveframe'));
16 }
17 </script>
18 </head>
19 <body onload="initialize()">
20 <div id="waveframe" style="width: 500px; height: 100%"></div>
21 </body>
22 </html>
```

At the moment, the Embed API is somewhat limited. Specifically, an embedded wave either allows everyone to participate or restricts access to a list of users defined *when you are issued a wave ID*. The Wave team hopes to address this restriction in the future. Nonetheless, an embedded wave as it functions now is well-suited to both private collaborations and public forums.

Wave Servers Everywhere

Google Wave encompasses the features of email and instant messaging. Wave is a suitable substitute for wikis and forums; it can be syndicated to any website; via robots, it can interoperate with virtually any external online service. What more could there possibly be?

How about making Wave open source?

That's the plan, and soon any organization will be able to (pardon the pun) make waves. Google Wave will remain one wave provider, but not the only provider. If Google follows through on its plans, individuals will be able to host a wave service for private use, corporations will generate waves for internal purposes, and web-based providers will have the option to offer Wave as a service, much like email and storage providers offer its services now. By convention and design, a wave shared among users from disparate providers must be syndicated among all providers. In other words, providers "exchange" waves and updates to keep all participants up to date.

This interoperability is defined in the *Google Wave Federation Protocol*, which is available now as an open specification. In a sense, the Federation Protocol is not unlike the Net News Transfer Protocol (NNTP), which powers the long-standing Usenet newsgroups. In a nutshell, a wave service provides local storage for waves, connects to other wave providers to exchange waves and updates, enforces local access controls, and display waves to its local subscribers. Local changes are propagated "upstream" to remote providers, and remote changes propagate "downstream" to the local provider.

The Federation Protocol describes a strategy to preclude the loss of data and an encryption scheme to keep data safe in transit. Unlike so many instant messaging services today, the Federation Protocol guarantees interoperability no matter who your wave provider is.

A New Wave

Since the announcement of Google Wave at the 2009 Google I/O conference, developers have rushed to gain early access to the platform. By the time you read this, some 100,000 early adopters will have accounts. Some companies already have announced product plans, whereas others are taking a wait-and-see approach. All are excited to put Wave through its paces.

Admittedly, a communication platform that combines email, instant messaging, forums, dropboxes, and more could overwhelm many users – the same users who are already drowning in data. However, Wave holds the promise to combine many sources into one. Data will still stream (very) quickly, but Google might just have us riding the wave instead of getting crushed by it. ■

MISSING LINUX MAGAZINE?



Ever have problems finding Linux Magazine on the newsstand? Just ask your local newsagent to reserve a copy of Linux Magazine for you!

Simply download our Just Ask! order form at www.linux-magazine.com/JustAsk, complete it, and take it to your local newsagent, who will reserve your copy of Linux Magazine.

Some newsagents even offer home delivery, making it even easier to ensure you don't miss an issue of Linux Magazine.



**SPECIAL SERVICE
FOR OUR UK READERS!**

www.linux-magazine.com/JustAsk