Google Wave – Social Impact and the Creation of a Wave Gadget

JAMES HAU
and ANDERS WESTLING

KTH Computer Science
and Communication

Bachelor of Science Thesis
Stockholm, Sweden 2010

Google Wave – Social Impact and the Creation of a Wave Gadget

J A M E S H A U
a n d A N D E R S W E S T L I N G

Bachelor's Thesis in Computer Science (15 ECTS credits)
at the School of Computer Science and Engineering
Royal Institute of Technology year 2010
Supervisor at CSC was Henrik Eriksson
Examiner was Mads Dam

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2010/hau_james_OCH_westling_anders_K10002.pdf

Royal Institute of Technology
School of Computer Science and Communication

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

Abstract

This thesis gives a presentation of Google Wave and tries to evaluate it, as well as describes the development of the Guess Gadget for Wave. Google Wave is a new communications platform that is currently in development by Google. The thesis tries to evaluate the platform in its current state by comparing it to current forms of e-communication: email, instant messaging, and social networking sites. By looking at the current issues with these forms of e-communication, we will see if Google Wave improves on these issues, and is viable as a new form of e-communication.

Google Wave has several interesting features such as real-time collaboration and the possibility to add user-made extensions. Google Wave improves on many issues of e-communication but is too confusing for ordinary users to use. After trying out the platform we agree that it's not very simple for inexperienced computer users to use Wave. The features that Wave provides can be difficult to understand. There are also currently too few users to make this platform really efficient.

The Guess Gadget is an extension to Google Wave. It's a gadget where participants play a guessing game. One player is the King who sets a question and answers. Then it's up to the other participants to make the right guess to become the King for the next turn. To keep track of how well the participants are performing a high score list is implemented. The development of the gadget was a success. The gadget is written in HTML and JavaScript with the help of the Google Wave API. The knowledge of HTML and JavaScript required to make a gadget is low, thereby making the versatility of Google Wave greater.

Referat

Google Wave - Genomslagskraft och skapandet av en Wave gadget

Denna uppsats ger en presentation av Google Wave och försöka utvärdera den, samt beskriver utvecklingen av Guess Gadget till Wave. Google Wave är en ny kommunikationsplattform som för närvarande är under utveckling av Google. Uppsatsen försöker att utvärdera plattformen i dess nuvarande skick genom att jämföra den med nuvarande former av e-kommunikation: e-post, chatt och sociala nätverkswebplatser. Genom att studera aktuella problem med dessa typer av e-kommunikation kommer vi se om Google Wave förbättrar dessa problem, och om Wave är rimlig som en ny form av e-kommunikation.

Google Wave har flera intressanta funktioner såsom samarbete i realtid och möjlighet att använda tillägg skapade av användare. Google Wave gör förbättringar inom många områden i e-kommunikation, men inte inom alla. Efter att ha provat denna plattform är vi överens om att det inte är enkelt för ovana datoranvändare att använda Wave, och de egenskaper som Wave har kan vara svåra att förstå. Det finns också för närvarande för få användare för att göra plattformen effektiv.

Guess Gadget är ett tillägg till Google Wave. Det är en gadget där deltagarna spelar ett gissningsspel. En spelare är Kung som ställer en fråga och sätter svar. Sedan är det upp till de övriga deltagarna att göra rätt gissning för att bli Kung nästa omgång. För att hålla reda på hur det går för deltagarna finns det en poänglista. Utvecklingen av tillägget var en framgång. Detta tillägg är skrivet i HTML och JavaScript med hjälp av Google Wave API. De kunskaper i HTML och JavaScript som krävs för att göra ett tillägg är låga, vilket gör mångsidigheten hos Google Wave stor.

Contents

Chapter 1

Introduction

E-communication is becoming an everyday tool for people to converse with each other. Ordinary mail is getting replaced with email and social network communication, such as instant messaging or Facebook. All of these have several millions of people connected to each other and give an easy way for people to keep in touch with each other. Facebook especially is an enormously successful social networking website which a lot of people use every day.

Google is hoping to revolutionize online social networking with its Google Wave service. This essay will give a brief description of Google Wave architecture and how Google Wave fits into the world of e-communication. It will also outline the Google Wave API and document the creation of the Guess Gadget. We will discuss the issues of e-communication as of today and see if Google Wave improves on these issues. One of the questions to be considered is if Google Wave has the possibility to gain market shares from the already well-known established applications and websites.

Chapter 2

Technical background

2.1 About Google Wave

Google Wave is an Internet communications platform created by Google. Its purpose is to become a replacement for today's email, instant messaging and Internet forums, by incorporating all of the functionality into one single service [1]. Google Wave is what Google envisions that email would look like if it was invented today [2]. Currently, Google Wave is in a preview phase, so it's not a finished product yet, and changes can be made in the near future. Wave is a service that can be accessed through a web browser, like web based email. Everything that happens in Google Wave is stored on a central server, so users can access their account anywhere.

Figure 2.1. Google Wave is a very ambitious project.

To communicate with other Google Wave users, you use so called *waves*. A wave is a full conversation between people. A wave consists of a number of participants and a number of messages. When a new wave is created, there will only be one participant, the creator, and a single empty message, ready to be edited. Participants can add new participants to the wave from their contact list at any time. The participants can add new messages to the wave, they can reply to specific messages in a threaded structure, and edit existing messages made by any parti-

cipant. It is also possible to make private replies which are only visible to a subset of the participants of the original public wave. Also, all changes made to a wave are permanently stored. By clicking the "playback" button, a user can follow the whole history of a wave, from creation to present time. Playback can go through the wave's history step by step, showing in what order messages were created, the changes that were made, and by whom, making it easier for new participants to see what has actually transpired. Playback will only show those messages that the user in question is allowed to read.

Google Wave also incorporates real-time collaboration. If two participants of a wave are using it at the same time, the changes made by one of them will be visible to the other one in real-time. The transfer works on a character-by-character basis, though this can be disabled so that the user must confirm what is sent to the other participants. In this way, Google Wave can be used as an instant messaging service, and users could write documents together without worrying about ruining each others' changes.

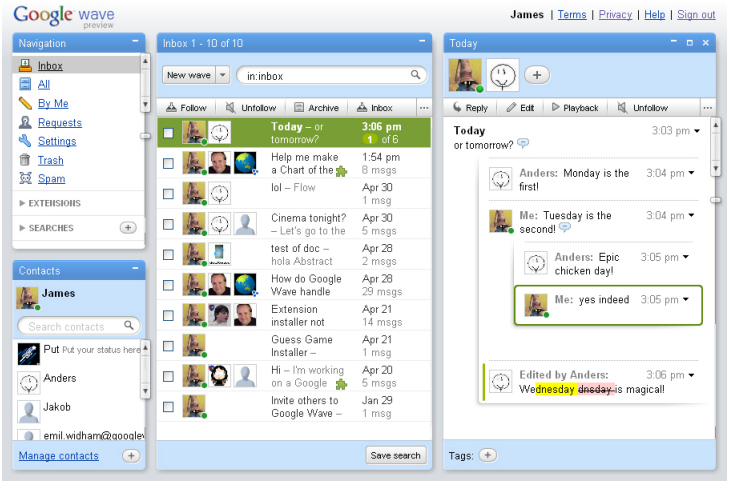

Figure 2.2. This is what Wave looks like.

In addition to these features, Wave also incorporates applications called extensions. There are already extensions made by Google, but it's also possible for users to create their own extensions using the provided API [3]. There are mainly two types of extensions, gadgets and robots.

Gadgets are applications that can be added to a wave message, written mainly in HTML and JavaScript. Users will then be able to interact with the gadget. An example of a built-in gadget is the Map Gadget. When added, the gadget will show a map of the creator's area, using Google Maps. Users will be able to navigate around the map, and add markers on locations for the other users to see. We have made our own gadget, the Guess Gadget, which can be read more about in section 3.2. Gadgets can be added by clicking the add extension button, and pasting an

URL of the gadget into the input box. However, a gadget can also be installed on an account, making it possible to add the gadget with the click of a button.

Robots are the other type of extension. A robot works as a participant in the wave, so a robot would be added just like any other participant, except that it will be a program instead of a person. The robot will be able to interact with the wave like a normal participant, making changes to it or adding new messages. Examples of robots are Flippy, which flips all text written by participants upside-down, or Wikify, which will read text of the format "<wikify topic>" and add a link to the Wikipedia page of the topic.

It is also possible to embed waves into webpages. This means that you can put your wave on a website, and other Wave users can interact with the wave through this website if they are logged in. Anonymous Access is also available, making it possible for non-Wave users to see the wave in its current state. They will not be able to interact with it though, as it will be read-only.

The Wave source code will be mostly open source, allowing third parties to create their own Wave service provider for public or private use [4]. Google also encourages these new service providers to "federate" the waves. That is, to share the waves with other Wave service providers, like how users of different email services are able to send emails to each other. The service provider will be identified by its Internet domain name.

2.1.1 The structure of a wave

A wave is essentially a collection of XML documents stored on the Wave service provider's server [5][6][7]. A user's waves will be accessible from any computer with an Internet connection, and will just require to log in. The basic principle of Wave is that every single change made to a wave is sent to the server, which applies it to the wave, and sends the change to the other participants.

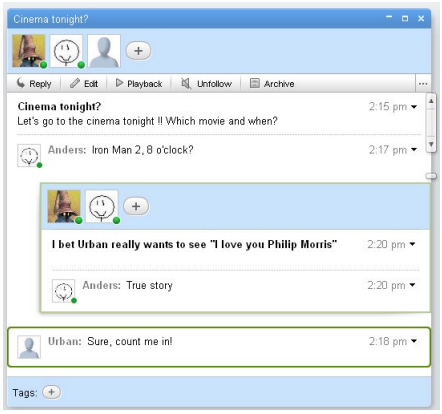

Figure 2.3. A wave with a few messages.

The wave itself consists of so-called "wavelets". A wavelet can be the whole conversation or just a part of the wave conversation, a branch of it. The wavelet contains information about who the participants are, and a number of documents. A private reply in a wave could become a new wavelet for example, if the participant list is different from the rest of the wave. The documents are even smaller pieces of the wave. There are two types of documents. The first type are the visible text documents called "blips". A blip is a single message in the wave. These blips are saved as XML documents (with some non-XML annotations first) in the wavelet with information about who created it and what the message is. The second type of document is the data document. It simply contains data relevant to the wavelet, but aren't visible to the users. This data can be used by extensions, for example. Every participant also has his own invisible wavelet, with private information such as what the user has and has not read in the wave.

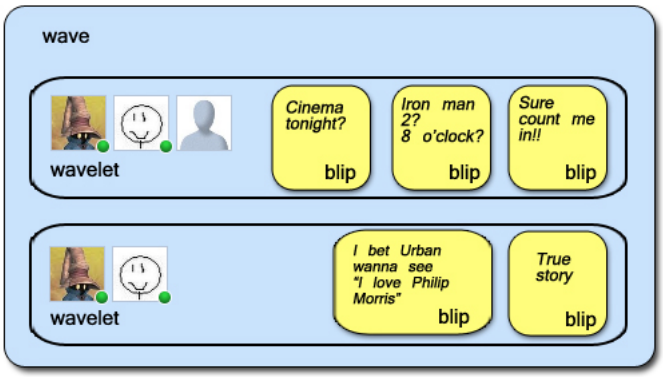

Figure 2.4. The same wave as in figure 4.1, showing how the wave is divided into wavelets and blips.

When a user changes a wavelet, he directly performs operations on his copy of the wavelet before sending them to the server. This is called Optimistic User Interface. The server then receives the operations, which are applied to its own version of the wavelet, the definite version. When this is done, the operations will be sent to all active copies of the wavelet, making sure every wavelet copy is identical. However, there is a potential problem. Say that a wave message consists of the word "Hello". Bob now removes the first letter from it, making the word "ello". At the same time, John adds the character "J" as the first character in the word, making the word "JHello". They apply the changes to their own versions of the wavelet, and sends the operations to the server, which applies it to the real wavelet, and sends the operations to the other person. John's wavelet will now perform Bob's operation, removing the first letter, resulting in the word "Hello" again. Bob will perform John's operation, ending up with the word "Jello". Now the waves aren't the same, and one of them will not be identical to the server's version. This must not happen, and so there is a solution to this, as Google Wave uses Operational Transformation

[8] to make sure every wavelet is always identical. The Operational Transformation is quite complex, and will not be described in detail, just in a short version.

The idea behind Operational Transformation is to transform operations made by other users at the same time as the user made an operation. In the above situation, Bob removed the first character of a word at the same time as John added a character first to the word. Let's say that the server got Bob's operation first, and then John's. This would mean that the final word is "Jello". For Bob, who was first, John's operation won't pose a problem, as the word will match the server's version. In John's version however, performing Bob's operation after his, will cause a mismatch, ending up with the word "Hello". To solve this, John will transform Bob's operation, compensating for his own operation. This means that Bob's operation, removing the first letter, will be transformed into an operation removing the second letter, to compensate for John's own addition.

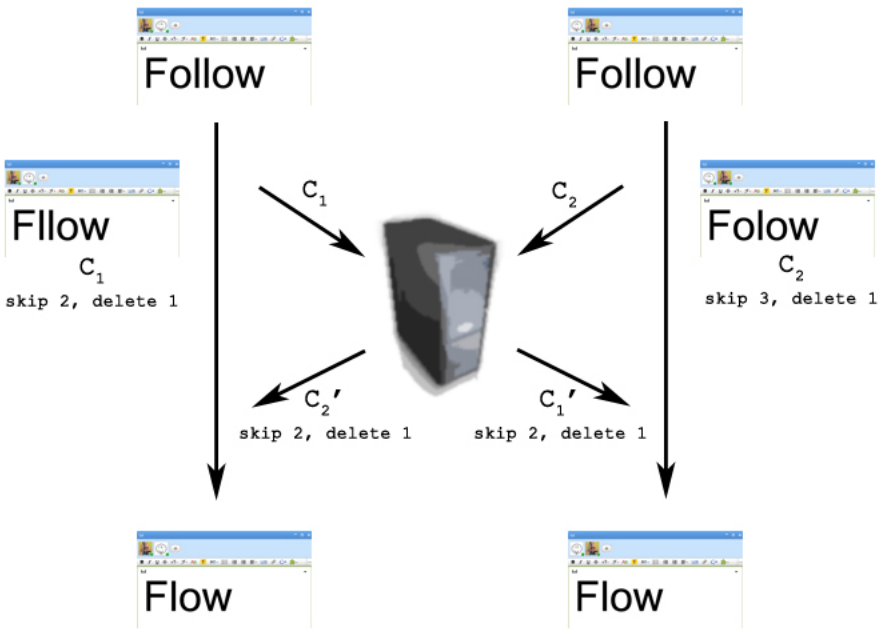

Figure 2.5. An example of Operational Transformation between two clients and a server.

Operational Transformations demands that the operations sent between the client and the server are ordered. The sequences of operations sent between them are called deltas. When a client makes changes to the wavelet, the client will send a delta with the operations to the server, along with a version number of the wavelet. The client will then wait for an acknowledgment from the server before sending a new delta. Operations made before acknowledgment will not be sent until acknow-

ledgment has been received and verified. The server will then apply the changes, and send the acknowledgment, which consists of a version number, and a history-hash. The history-hash will be a rolling hash over all operations made between the first version and the provided version number. The client will have to compare this hash with its own hash and if they do not match, the client will have to recover and make sure that all operations were applied in the correct order.

When users of different Wave service providers are communicating, where will the data be stored? The answer is that wavelets will be stored on different servers depending on who created it. If bob@wave1.com starts a wave and invites john@wave2.com, the wave's only wavelet will be stored on wave1's server, since Bob created the wavelet. He is the host. On the wave2 server, a copy of the wavelet will be made, and the servers will handle the communication between each other and their respective users. However, if John invites steve@wave2.com to the wave, and makes a private reply with just the two of them, a new wavelet will be created, and since John created it, the real wavelet will be stored on the wave2 server instead. In fact, this wavelet will never be copied to the wave1 server unless at least one participant is a user of that provider. All info exchange between servers will be on a need-to-know basis, just like communication between users of a single Wave service provider.

2.2 Internet Communication

2.2.1 Email

Definition and explanation

Email is a fast way to transmit electronic messages from a sender to a receiver. It takes less than a minute to send an email between communicators if they are connected to the same network [Bälter 1995]. Email requires the accessibility of an email server which helps the sender's email to get to the right email account. This is often done through the Internet but it can also be done through a local area network. Although a connection to the network for the sender is required, email is asynchronous communication. If the email has been sent from the senders computer, the email is stored at an email server. The receiver has to connect to the email server and checks his inbox for new email. If there are new emails, the receiver will retrieve these new emails and be able to read them.
An email address is of the following format:

```
travis.bickle@taxi.se
```

The part before the "@" identifies a specific user of the email service. The @ is followed by a domain address. "taxi" is often the company that is providing the email service followed by a dot and the country code [9].

Usage

Email is used all over the world. There are about 1.8 billion email users and over 2.9 billion email accounts [The Radicati Group 2010]. These email accounts are mainly free accounts where 75% are used by individuals, and 25% are corporate users. As having an email account becomes more necessary and the accessibility to the Internet increases the number of email accounts are expected to grow. The expected increase of email accounts over four years are 0.9 billion to a total of 3.8 billion.

Issues

Emails are not instantaneous

Emails are asynchronous communication and were never meant to be instant. But in modern days the need of quick and instant communication gets more and more necessary.

Email delivery is not guaranteed

There is no assurance that the mail has been sent or read by the intended receiver. One example is when AOL deleted acceptance letters from Harvard; the emails didn't get through in the spam filter [Boston Globe 2002].

Emails can be dangerous

Many viruses and Trojans are spread through emails. In some cases you only have to view the emails to become infected with a virus. The Love Letter virus [10] spread all over the world in a few hours and did tremendous damage to the society. Several government servers were shut down until they fixed the problem.

Emails can be insecure

The administrator of the email server has full accessibility to the emails. They can read them at any time. Recently more and more supervisors read their employees' emails.

Problems caused by spam

Spam or junk mail are very common as they fill the users inbox with unwanted emails. Even with the best spam filters around the problem cannot be completely resolved. Because the email system is designed as it is, this cannot be changed.

Problems caused by large amounts of incoming email

If there is a large flow of incoming email the user has to prioritize, read and reply to them. This takes a lot of time. The user might not also be sure which emails he should prioritize.

Problems with file sharing

The files might be dangerous, you cannot organize the files and there is a size limit to the file since the file has to be stored on the email server.

Problems with interface

Since a lot of email applications look different this might confuse the user. The user may not use the full functionality of the email system just because he does not know about it.

Problems to access via application

Users nowadays access their email from different locations, home, work or at the university. To have an application installed at all these locations can be a nuisance. Therefore web based applications are getting more popular.

2.2.2 Instant Messaging

Definition and explanation

Instant Messaging (IM) is a technology which allows user to have a real-time conversation through a network. Users can instantly see who is available or not and send them a message [Nardi, Whittaker, Bradner 2000]. Normally the users don't have the need to know the recipients email address; instead there are contact lists which they can interact with. IM is similar to online chat rooms although you only send messages to a specific person or people who have been invited to the conversation [11]. IM is very similar to a phone call whereas email is like a fax or mail.

Usage

Since more and more people gain access to the Internet the usage of IM has increased. There are about 2.4 billion IM accounts [The Radicati Group 2010].

Issues

IM requires users to be online

You cannot always send a message to a user who is offline, though some applications let you send messages to users who are offline. They will then receive the messages when they log in to the application.

Problems with having application based IM

Just as the issue with accessing emails through an application, there are issues with having application based IM. Many of the Instant Messaging services use an application which you have to have installed on your computer. Web based IM is not as popular as web based email platforms.

Problems with prioritizing different messages

As more and more messages are being sent to a user, often along with a sound or a blinking banner to show that a new message has been received, it becomes difficult to prioritize correctly which message is the most important one to answer since the warnings look the same.

IM has no message delivery insurance

Many of the IM applications don't have any kind of assurance that the message got through to the receiver.

Problems with message history

It is not always possible to keep a history of your conversations. In some applications there is an option if you want to save down the conversation when you close the application, but there is no possibility to browse through all the history at any computer.

Problems with creating group conversations

If there are several members in a group conversation it can get hard to get a clear overview of the conversations. Several messages will be sent back and forth and the user might get confused if several different topics are being discussed in the same group conversation.

IM has no standards for file sharing

Different applications use different ports for internet communication. There might be conflicts with the firewall or with the anti-virus program. Web based IM don't generally have the file sharing feature.

2.2.3 Social Network Sites

Definition and explanation

A social network site, or service, can be defined as web-based service where users of a particular interest can form a community [Boyd & Ellison 2007][12]. The users can get to know new people to build their social networks or just have public discussions related to the subject of their interest. These interests can be of very different forms,

from employees of a company discussing their job, to singles looking to find dates. Most of today's social network sites have a profile page. A kind of private webpage where they present themselves and keep users up to date on what happens in the user's life. It may contain personal information as well.

Usage

It began, even before the Internet, with the Bulletin Board [13] System, where users could use software to connect to a server. When connected to this server they could write messages viewable to other connected users. These systems were often dedicated to a certain subject. As the Internet became popular, the more modern Internet forum became popular. Users could through a web site make threads with a topic, and other users could post in these threads. Forums are often dedicated to a certain topic, and are still popular today. However, social networking reached its big boom in 1995 with the site classmates.com. Visitors could create accounts and make searches for old classmates who had also signed up to the site. The site became very popular, and still has over 40 million users [14]. Today, Facebook is the most popular social network service, with over 400 million users [15]. Originally intended for college students to keep contact, Facebook quickly became very popular and was made available to everyone. Registered users can search for people they know, send messages to friends, become members of groups with a common interest, play games with contacts, and much more.

Issues

Privacy

Probably the largest issue with social networks today is privacy. Users of social networking sites, especially teenagers, post personal information about themselves and their friends on the sites [Barnes 2006][Dwyer, Hiltz & Passerini 2007]. This data can be collected by governments and third parties. A common example is bosses checking their employee's Facebook page, or sexual predators finding personal information about new victims. Users can generally make their profiles and comments private, but not everyone does that, and friends can still post private information about a user.

Chapter 3

Google Wave

3.1 Google Wave API

There are mainly three things that can be done with the provided API: Embed a wave into a web page, make a robot, or make a gadget. To embed a wave into a webpage, only some extra HTML and JavaScript code will be required. A JavaScript-file written by Google must first be imported for relevant functions. Then you can use the functions to load a wave using its URL, customize the user interface, and add the wave to an HTML div.

Robots are programs hosted on Google's App Engine. Google has currently provided libraries to write Robots in Java 6 or in Python 2.5 or higher. The creator will have to provide some user info for the Robot, such as name and profile picture. The robot can then be added to a wave as a participant. It may be programmed to react to different events occurring in the wave, such as new blips being made or the participant list changing by e.g. making a new blip with some clever comment. When the program is done, it can be uploaded to the App Engine server, and added to a wave for usage.

Gadgets are small applications that can be put inside a wave. However, gadgets are not exclusive to Wave. Gadgets have been used before Wave [16], and can be put in a website for example, or on the desktop if using Google Desktop. These old gadgets are functional in Wave, but will not take advantage of some of Wave's gadget features, such as finding the user identity of a viewer, or making use of the playback function. The basic file of a gadget is an XML-file. In it specifications can be put such as title, gadget size and libraries that will be imported. Afterwards you will write your own code, which will be in HTML5, JavaScript 1.8.3 or later, and by using AJAX all types of server-languages can be utilized. It can be pretty much anything that can be put in a normal webpage, and with libraries provided, you can for example dynamically change the height of the gadget, add tabs, or embed flash files.

Specifically for Wave, a gadget can also get access to all the participant info of a wavelet, their account name, their name, and their profile picture. It can also make

use of states. The gadget will have a state, a key-value map, that can save values shared by all the participants. Both the key and the value are of string type. This means that state changes made by one user can affect what happens in the other participants' gadgets, making it possible for the gadgets to be used together with all participants. The gadget can be told to run specific JavaScript functions when the state or when the participant list changes. These are called callback functions. The total available state for a gadget is 100 kilobytes of memory.

Google has a testing version of Wave, called Wave Sandbox [17]. Wave Sandbox is a separate entity from Google Wave, and is meant to be used to test extensions made by users. Sandbox isn't able to interact with regular Wave, and data stored might be lost, so it is just for testing purpose. A debug feature is available in Sandbox, making it possible to see the state changes and log messages made by extensions. When an extension is ready, it can be "deployed" to regular Wave, making it available for use via an extension installer.

3.2 The Guess Gadget

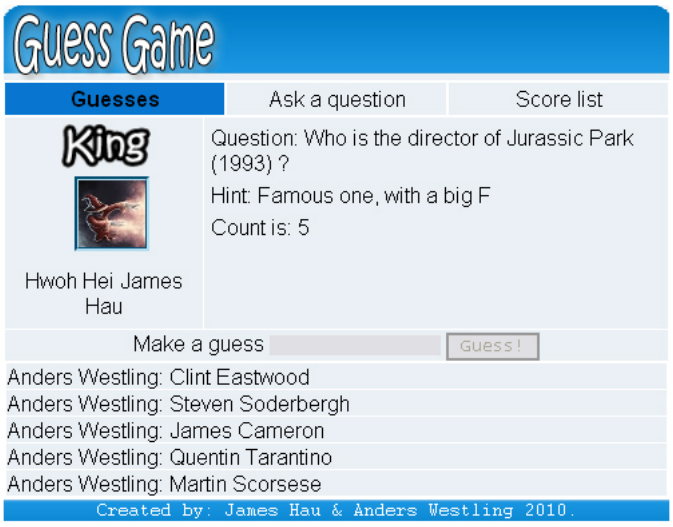

Figure 3.1. A picture of the Guess Gadget.

As part of this thesis we decided to make our own gadget using the API provided by Google. Our goal was to make use of the participant and state-related functions. If we didn't, it would pretty much be a webpage put into a wave, which isn't very exciting. We decided to make a guessing game gadget. The idea comes from how it's popular on Internet forums for users to play a game where one user asks a question, and the other users try to answer it. The one who comes up with the correct answer first gets to ask the next question. The gadget would have a participant to ask a

question and to set some answers. The other participants can then write guesses into an input box, and if the guess matches an answer, this participant would get a point, and gets to ask a new question. This would make use of the Wave-specific functions of a gadget, since we use the state to save statistics and questions, and the participant list can be used to make sure that only one participant can ask a question at a time, plus keeping track of the score.

The idea is to have one user be the King. The King is the person who asks a question, sets answers, and give a hint if needed. We use the TabSet provided by the API to give three different views (HTML divs) accessible by clicking on their corresponding tabs. One will be the main guessing view, where users can see the question and the five last guesses made, and an input box to make their own guesses. Then there will be a question view, where the king can set the question, between one and six different answers, and add a hint. Users who are not the king can select this tab, but they will not be able to view the settings which the King has set. Finally, there's a tab which brings up a score list. A user gets a point if he makes a correct guess. The one who makes a correct guess will then become the new King.

The state is used to keep track of the King, the five last guesses, everyone's score, the questions, the answers and the hint. For the scores, each user ID will be the key, and the value will be the score. The five last guesses will be put together in a single state value. There is a limitation to a state value size, so five guesses seemed like a reasonable amount, though more or less would be possible depending on the length of the guesses. The answers are handled in a similar way. Guesses will be compared to the answers, ignoring case. Since the API makes it possible to see who is using the gadget, we can make sure that participants can't do what they aren't meant to do, like a non-King asking a question, or the King answering his own question and getting points.

Figure 3.2. The Guess Gadget extension installer.

Chapter 4

Social Aspects

4.1 Was the creation of Google Wave necessary?

Google asked themselves what email would look like if it was created today. Google Wave became the product of those thoughts and it became available for public testing fall 2009 [2]. When Google decided to develop Google Wave it was a good idea as social networking is becoming rapidly popular. Google Wave can be accessed through an internet browser and therefore doesn't require any specific applications to be installed. As more and more people get connected to the Internet more people joins social networks. With a growing network the chance of getting more users to Google Wave increases. Facebook, for example, started as a university project but became so popular that it was made public. As more and more people signed up for it a chain reaction started. More people got conscious of the social network and more people decided to get connected to it. The question is: where do you start such a reaction?

4.2 Is Google Wave the State of the Art?

By having waves, wavelets and blips the users can collaborate and directly reply to a new message input if they are online. Since the messages are all stored on the wave server you get a guarantee that the message has been sent and received by the server. When the receiver logs on and synchronizes with the server he gets all the new updates.

Waves are ongoing conversations, you can write replies directly in them and even branch off in different wavelets. If the user gets lost in the wave, he has the possibility to use the playback function. The playback function plays up all the changes made to a wave, one by one. It works like a video player. By having these features it makes it easier to keep track of which waves are the most important to view.

By giving the developers the possibility to create their own gadgets or robots, Google Wave versatility becomes enormous. There is at the moment not really any

restrictions to what the gadgets or robots are allowed to do. The largest limitation lies in the programming languages. Google recommend the gadgets or robots to be hosted at their Project Hosting for open source projects [18].

You are allowed to attach files up to a size limit of 20 megabytes. All the files are, like files attached to a web based mail, scanned by their anti-virus programs. By having the possibility to view images in waves that are hosted on other web pages the user may store those images wherever they want. You can also import video clips from YouTube through a gadget. Only a gadget is needed to be able to import videos from other video sites.
According to Google Wave Privacy Policy [19] there are only a few circumstances where Google will access private data on the wave servers:

1. To be able to provide the best possible service.
2. When Google is required to do so by law.
3. When there might be a violation of the Terms of Service, Google's policies or for security or technical issues.

While this applies to Google's Wave service, other future Wave providers might not be as private with their wave information. Companies having their own Wave service could still read their employees waves.

4.3 Analysis

4.3.1 Reception

When Google Wave was released for developers and not yet open for public use it was such a big hype the testing accounts and invites were even sold on Ebay [20]. After a while though the hype settled down and the conclusion from the testers were: Google Wave was a disappointment. Google Wave had great potential but it might be too early for the public to comprehend the usefulness of the techniques [21].

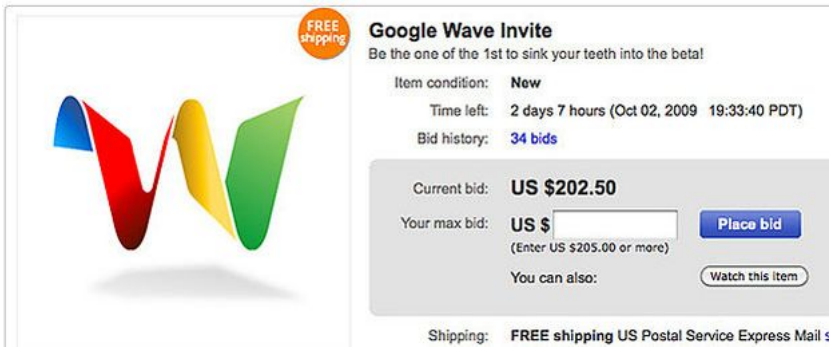

Figure 4.1. An invite to the Wave preview being sold in Ebay.

Google Wave is still in development and there are quite a few features which

are still missing, for example the ability to access your Google mail account from the Google Wave platform. There are at the moment not enough members to show the worth of Google Wave. According to Metcalfe's Law [22] the worth of a telecommunication network is proportional to the square of the number of users. With Wave preview not having nearly enough users to compete with email, it's difficult to say if it can become a viable communications platform, since everyone isn't reachable from Wave yet. Perhaps when Google Wave gets close to being finished, Google will try to launch their new social network platform in a larger scale, making it a viable choice for communication.

4.3.2 Different kinds of users have different needs

The needs of an ordinary user are far different from a corporate user. For many ordinary users asynchronous communication is enough, they don't need the real-time collaboration. As it seems today, ordinary email is good enough. Whenever they need to contact another person quickly, they usually just make a call. To be able to move the users from already established sites Google Wave has to make a tremendous increase of popularity. This is hard to do but certainly not impossible.

Google Wave has a great value for enterprises. You can collaborate and at the same time communicate. There are very few applications who offer both these features. To be able to work in the same document and at the same time being able to have a conversation with your collaborators is a great feature along with being able to see the live typing. If the enterprise needs any special features they have the possibility to look for a solution at the Google Wave App Store. It's not yet implemented but Google has confirmed that there will be one [23]. Another solution is to have someone program an extension with their required features.

Students will also have great use of Google Wave. They have great benefits from using a collaboration tool which is both asynchronous and instant. All the data is stored on a server and thereby always accessible, both from home and from school. Google Wave is free and lots of gadgets are free to use as well. The question is rather: is it too confusing to use? Students with computer experience might not have any issues using Google Wave but those who are completely new to Wave might not understand all of its features. Johanna Eidergren has made an analysis of Google Wave as a CSCL-tool (Computer Supported Collaborative Learning)[Eidergren 2009]. She concluded that Google Wave has several of the needed features of a CSCL-tool [Wegerif 2005]. A few of these are speed and automaticity, to be able to quickly make changes, feed-back and response and the possibility be able to integrate a variety of different communication tools. Eidergren's own personal opinion of Google Wave is that it has the possibility to make a great social impact as the new email, but it will take time for it to become big.

According to her it depends on the users, users have always had problems with new features. But there are a quite a few teachers who are having thoughts about using Google Wave as a CSCL-tool.

4.3.3 Overall strengths

The combination of being able to have an asynchronous conversation as well as being able to instantly reply are the greatest features of Google Wave. Since the wave is stored on a Wave server it is certain that the message has been sent if the sender sees the update in the Wave, otherwise Wave will send a message saying something went wrong, and the changes will not be made. Since a wave consists of a full conversation that is constantly expanded, a long conversation will not take up more space in the inbox compared to a short conversation. This is an improvement over email where many exchanged emails between two users quickly fill a whole page in the inbox, since every response will become a new email. This can make it easier to keep track of which waves are important, since it's always in the same wave.

Google Wave is web based. Therefore making it almost always accessible for users. It can be accessed through any up-to-date Internet browser.

Google Wave is based on the open source Google Wave Federation Protocol [4]. This might be to increase interest and have developers create their own platforms and weave in the services from Google. In this era of programming open source is becoming a viability and nowadays there are a lots of great open source projects.

As a developer you can develop gadgets and robots. There are close to no restrictions in what can be created and thereby the versatility is enormous. There are a wide range of gadgets, from game gadgets to poll gadgets. There is also a gadget which edits documents in Google Documents.

4.3.4 Overall weaknesses

There are a few issues which has been presented in section 2.2 that haven't been solved. File sharing through Google Wave is doable. It's very similar to ordinary email attachments. Since Google Wave is more of a collaboration and communication platform, file sharing hasn't been one of the focuses. But sharing files, between students as well as corporate users, has become more troublesome with larger files. The interface is clear and simple but there are new terms like wave and extensions. It can make it confusing for users without much technical experience.

There is no simple solution to large amounts of incoming messages. This issue is very similar to having large amounts of incoming email. The best solution at the moment is probably to have messages of the same subject in the same wave, and tag them with descriptive tags. Since the waves are just built upon, they can quickly become very large. If you just add messages to the bottom of the wave, the user will soon have to scroll down the wave to read the newest message. This can of course be solved by removing messages, but that is a tedious task. Google Wave can take a bit of time to load a wave. Compared to email, which is pretty much instant, except for loading external pictures, large waves can require a few seconds of loading before a user can use it. The playback feature takes even longer loading, and isn't as seamless to use as it should be.

As mentioned Google Wave is still early in development. There are features

missing and the security is not yet fully implemented. As an example, there was a post in the API Help forum in Wave. A person who replied added a gadget that automatically redirected the user to a Twitter site. This spread fast but has been solved by Google. As more issues come forth Google Wave's development team will deal with these. In this case Pamela Fox, a developer at Google Wave, mentioned that their current plan is to have Gadgets only load on a button click [24]. At the moment all gadgets are loaded as you enter the wave.

More users are getting connected to the network, but there are too few to make it really efficient in reaching other people. If you use a communication tool you will prefer to be able to contact everyone, and Wave doesn't have a large userbase at the moment.

Social Network sites nearly always have one thing in common, a private profile site. Google Wave does not have this and that makes it less useful for socializing. Google Wave is more useful for private communication, like email and Instant Messaging.

4.3.5 Strength of the Google trademark

The Google trademark is very strong. Google Search [25] is the world's most used search engine. By releasing Google Mail and Google Documents, the Google trademark became stronger and is now a very trusted trademark. Google is also the owner of YouTube which is one of the biggest video sharing services [26]. With a very strong trademark more people are leaned towards using their products, which speaks in favor of Wave.

Chapter 5

Conclusions

5.1 The Guess Gadget

We successfully developed the Guess Gadget. The gadget works as we originally intended it to do, and we didn't have to skip any features that we originally intended to implement. The creation of it went smoothly without many problems, even though we weren't very experienced with JavaScript when we began. This confirms that it is easy to make a gadget for Wave. The biggest problem with making our gadget was probably the wave state. Since everything that is saved in the state is of the string form, it needs a bit of work if you for example want to store multiple values together. Still, developing a wave gadget is really easy, and as long as there will not be high demands on the state, developing a wave gadget should never pose much trouble for someone familiar with HTML and JavaScript.

While the basic functionality works, there are a couple of improvements that could be made to the gadget. First, the wave sandbox doesn't let users leave a wave. They can just move it to the trash bin, but will still remain a participant. In the preview version, it is possible to remove yourself as a participant completely. This would pose a problem if a participant removes himself while he is king, as he would remain king until someone gets the question right, if there even is a question. This can be solved by having a function that is called when the participant list changes, as this is a callback available by the API. This function could then go through the participants and check if the king is still a participant. The function could also remove the scores from removed participants. Another improvement could be to let users dethrone a king, if he for example asks an impossible question and isn't active. A new king could then be chosen either by implementing a "I want to be the king" button, or by randomizing from the participants list. The gadget is Open Source under GNU General Public License v3 [27]. The gadget source code can be found at *http://code.google.com/p/guess-gadget/*.

5.2 Google Wave

Google Wave uses some interesting technology. The idea behind everything is good, which the users seem to agree about. The interaction possibilities between users are great, and Wave's ability to interact with other systems is also noteworthy. However, a problem that both we and others have experienced is the difficulty to use Wave. It's not always obvious what you should or can do, as the interface is filled with buttons and menus. The design will probably have to be streamlined, especially if Wave is to appeal to the average computer user. For example, a message should be edited by double-clicking on it, but now the user will have to click the message, and select the edit option in a pop-up menu. Since inexperienced users can find email clients confusing, Wave would quickly become a nightmare in its current state.

The big question is probably: Is the functionality that Wave offers needed? Right now, there are three major forms of e-communication: email, Instant Messaging, and social network sites. Wave will not replace the social network sites, since the functionality is very different. Wave is for communicating mainly with people you know the address of, while social networking focuses more on getting in touch with people you don't already know or rarely meet. Public waves can be used to find new contacts, but it probably will not be as effective as for example Facebook or a website dedicated to a subject. Wave does however implement most of the functionality of instant messaging and email. Wave is definitely an improvement of email for users who are logged in at the same time and want to communicate, both because it saves time, and because communication will be easier with the real-time collaboration. Can Wave replace Instant Messaging though? To mix email type waves with chat type waves could be confusing, and sorting them could be time-consuming if the user communicates with many other users. Conversations can quickly become long, and scrolling to the bottom of the wave each time you continue the conversation, or just delete old messages doesn't sound appealing. There are gadgets however that makes chatting easier.

There is the possibility that Google will listen to the criticism. Google Wave is considered "overhyped" and the interest is faltering for the public. Google might focus on launching the platform for enterprise collaboration and communication instead of public use. After they have established Wave in the enterprise business and made some improvements they might launch it for ordinary users. The Google Wave features are too early for its time.

With a big name as Google, Wave might have a possibility to become big, but for that to happen, improvements would be needed. Wave must become more user-friendly, Wave must get more popular so that it can become a viable form of communication, and time will be needed for people to adapt to a new type of communication. Email has been a popular form of communication for many years now, and changing standards is never easy, even if you are Google.

Chapter 6

References

Barnes, S.B. 2006: *A privacy paradox: Social networking in the United States*. First Monday, volume 11, number 9.

Boston Globe 2002: *AOL junks Harvard acceptance e-mails*. St. Petersburg Times, Jan 2 2002.

Boyd D.M. & Ellison N.B. 2007: *Social Network sites: Definition, history, and scholarship*. Journal of Computer-Mediated Communication, 13(1).

Bälter O. 1995: *Electronic mail from a user perspective: Problems and remedies*. TRITA-NA-9510. NADA, Royal Institute of Technology, Stockholm, Sweden.

Dwyer C., Hiltz S.R. & Passerini K. 2007: *Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace*. AMCIS 2007 proceedings.

Eidergren J. 2009: *Google Wave: Revolutionerande CSCL-verktyg eller överdriven hype* (Swedish). Introduktion till kommunikation, lärande och IT, TIA037 H09. University of Gothenburg, Sweden.

Nardi B.A., Whittaker S., Bradner E. 2000: *Interaction and outeraction: instant messaging in action*. CSCW 2000. Philadelphia.

The Radicati Group, Inc. 2010: *Email Statistics Report 2010*. Palo Alto, Canada.

Wegerif R. 2006: *Towards a dialogic understanding of the relationship between cscl and teaching thinking skills*. International Journal of Computer-Supported Collaborative Learning, 1.

Websites, last visited 2010-05-01

[1]: http://wave.google.com/about.html
[2]: http://www.youtube.com/watch?v=v_UyVmITiYQ
[3]: http://code.google.com/apis/wave/guide.html
[4]: http://www.waveprotocol.org/whitepapers/google-wave-architecture
[5]: http://www.waveprotocol.org/whitepapers/internal-client-server-protocol
[6]: http://www.waveprotocol.org/draft-protocol-specs/wave-conversation-model
[7]: http://www.waveprotocol.org/draft-protocol-specs/draft-protocol-spec
[8]: http://www.waveprotocol.org/whitepapers/operational-transform
[9]: http://www.nada.kth.se/dataterm/rek.html#a13
[10]: http://news.zdnet.com/2100-9595_22-107318.html?legacy=zdnn
[11]: http://www.uen.org/core/edtech/glossary.shtml
[12]: http://www.answers.com/topic/social-networking-site
[13]: http://www.britannica.com/EBchecked/topic/183745/bulletin-board-system
[14]: http://www.digitaltrends.com/features/the-history-of-social-networking/
[15]: http://www.facebook.com/press/info.php?statistics
[16]: http://code.google.com/apis/gadgets/docs/gs.html
[17]: http://wave.google.com/a/wavesandbox.com/
[18]: http://code.google.com/hosting/
[19]: http://wave.google.com/privacy.html
[20]: http://www.svd.se/naringsliv/it/google-wave-invites-saljs-pa-ebay-for-tusentals-kronor_3590881.svd
[21]: http://www.linux-mag.com/id/7653
[22]: http://spectrum.ieee.org/computing/networks/metcalfes-law-is-wrong
[23]: http://thenextweb.com/apps/2009/10/27/google-wave-app-store/
[24]: https://wave.google.com/wave/?pli=1#restored:wave:googlewave.com!w%252BUAr62fzcA
[25]: http://www.google.com
[26]: http://www.msnbc.msn.com/id/15196982/
[27]: http://www.gnu.org/licenses/gpl.html