Fusion Tables
New Ways To Collaborate On Structured Data

by

Jonathan Goldberg Kidon

BSc CS, MIT, (2009)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2010

Author .....
Department of Electrical Engineering and Computer Science
May 17, 2010

Certified by .....
Professor Alon Halevy
Professor of Computer Science and Engineering
Google / 6A Thesis Supervisor

Certified by .....
Professor David Karger
Professor of Computer Science
MIT Thesis Supervisor

Approved by .....
Dr. Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Fusion Tables

New Ways To Collaborate On Structured Data

by

Jonathan Goldberg Kidon

Submitted to the Department of Electrical Engineering and Computer Science
on May 17, 2010, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Fusion Tables allows data collaborators to create, merge, navigate and set access control permissions on structured data. This thesis focuses on the collaboration tools that were added to Googles Fusion Tables. The collaboration tools provided additional functionality: first, the ability to view, sort and filter all the threaded discussions on the different granularities of the data set; second, the ability to take Snaps, dynamic state bookmarking that allows collaborators to save queries and visualizations and share them with other users. In addition, this thesis initiates a discussion about data collaboration on different platforms outside the Data Management System (DMS), and the implementation of the Fusion Table - Google Wave gadget that provides this functionality.
To evaluate these added features, we conducted a user survey based on three sources: Google Analytics, field study of experienced Fusion Tables users, and a user study to evaluate the UI and the collaboration tools. The results showed that approximately 40% of the visitors to the site use the collaboration features . Based on the user study, it appears that UI improvements can increase exposure to these features, and some additional functionality can be added to improve the collaboration features and provide a better collaboration system.

Google / 6A Thesis Supervisor: Professor Alon Halevy
Title: Professor of Computer Science and Engineering

MIT Thesis Supervisor: Professor David Karger
Title: Professor of Computer Science

Dr. Christopher J. Terman
Title: Chairman, Department Committee on Graduate Theses

Background

This thesis is the required written part of my 6-A thesis assignment; it summarizes my main contributions during the last two summers in the Fusion Tables team under Professor Halevy at Google Research.
During the summer of 2008, Adam Sadovsky and I were responsible for building the first prototype of Fusion Tables. Our first implementation had basic functionalities that allowed loading data sets, merging (fusing), and adding general comments to the data sets. After the first internship ended Professor Halevy formed a team to implement a full scale of our prototype. This product was launched in May 2009 - http://tables.googlelabs.com/.

When I returned to Google for a second internship in June 2009, I was thrilled to meet the team, consisting of three engineers and a program manager who had implemented the full scale framework of my earlier prototype. Not only was I excited that I was going to work on a live product with real users, but this new experience would also add technical challenges to every feature that I was about to develop. I had to make sure that my code integrated properly with the existing framework and that it was modular and could scale. During the second internship, I focused mainly on the new collaboration tools for Fusion Tables, which are described in this thesis. Like any typical software engineer, I spent considerable amount of time implementing and improving general features of Fusion Tables: featurs that were important for the experience of the Fusion Table's users or that were required for the collaboration features (See appendix A for more information on these additional features).

Acknowledgments

This thesis concludes the amazing learning experience that I had at MIT.
These last five years were not always an easy ride and therefore I would like to dedicate this work to those who supported and believed in me.

To Alon Halevy, who invented the wonderful idea of Fusion Tables, for being a great mentor since our first phone call, and most importantly for teaching me about the real value of a good coffee.

To David Karger, who I was fortunate to have him as my advisor and to learn from his advice and wisdom.

To the Fusion Tables Team, whom I enjoyed working with and learning from: Rebecca, Jayant, Anu, and Hector.

To my wonderful family, who are always there for me, and whom I love and miss the most - Irit, Moshe, Daniel, Alona, Savta Magda and Aviva, Saba Pali and Shoni.

To my Rachel, my true soulmate, with whom I have shared my MIT experience and hopefully also the future.

To No6 and to all the great friendships that I have made here that will always remind me of the good times at MIT - Tal, Itai, Anton, Leo, Vane, Constantinos, Evros, Lauri, Tom, and Angi.

And to everyone else who should have been on this list but I am just too tired to remember...

Contents

List of Figures

Chapter 1

Introduction

The recent developments in cloud computing have introduced new ways to easily create and share data between collaborators. One of the key challenges of data collaboration is how to utilize the fact that the data is stored in a cloud Database Management System (DMS), a system package that controls the creation and maintenance of a database (DB), to provide better online collaboration tools.

This thesis focuses on the collaboration tools within Google Fusion Tables [1]. Fusion Tables (see tables.googlelabs.com) is a DMS in the cloud that was launched in June 2009 and has since seen significant use. Fusion Tables has three core requirements. First, to support collaboration between multiple users and multiple organizations. Second, to attract the majority of data collaborators, who by their nature are not computer experts and do not know how to use a complex DB. Third, to allow easy integration between data collection and the presentation and visualization of data on the web. Therefore, much of the effort in the design and development has been geared toward simplifying the most common DB actions and making them available to a larger set of audiences, organizations and communities of users who want to make their data available online, and use it for better collaboration.
The Fusion Tables infrastructure is based on five elements: First, it supports the uploading of large data set files (up to 100 MB of CSV). Compared to other data cloud applications (Google Docs) this is almost a x100 scale jump . Second, Fusion Tables

provides tools to manage the data. Once the data is uploaded, it is possible, using an easy-to-use user interface, to filter, sort, and aggregate the data (see figure 1-1). This interface replaces the need to write complex SQL queries as in regular databases. Fusion Tables also allows the table owner to define customized access level controls between viewers and collaborators, and to make data public (see figure 1-2).

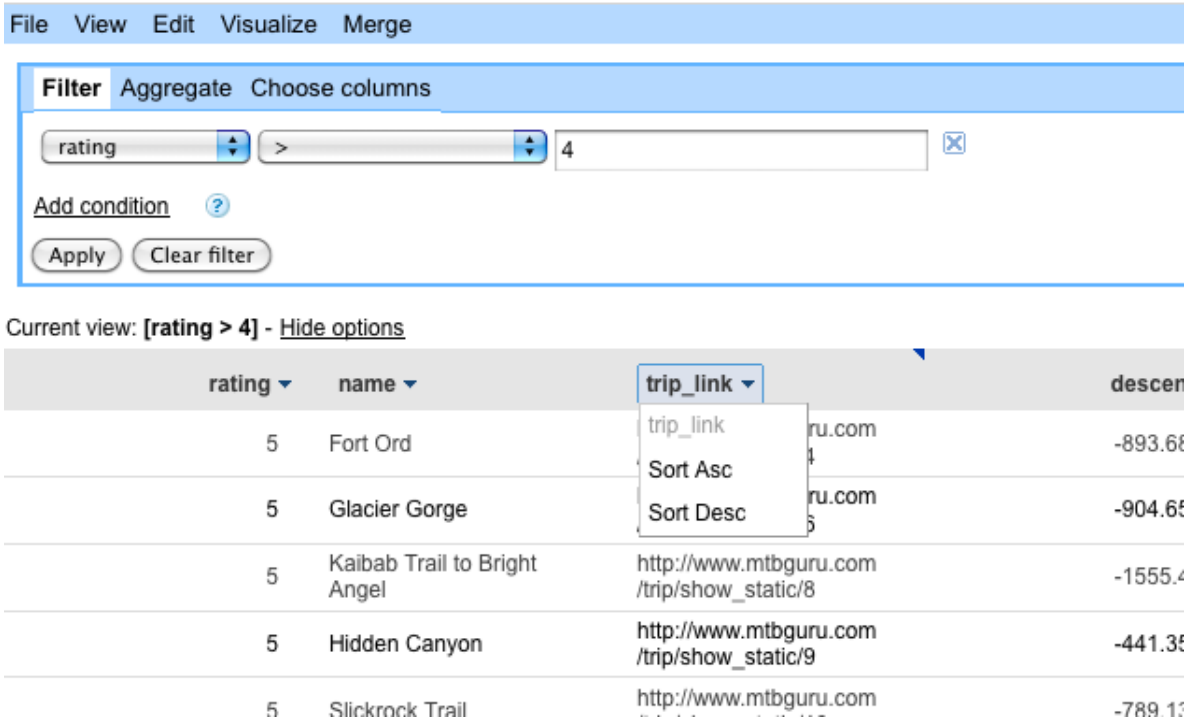

Figure 1-1: Fusion Tables provides easy-to-use tools to sort, filter, and aggregate data.

Third, it supports data integration from multiple data sets. Users can easily combine different data tables into one merged table, even if the tables have different owners. The new merged table is based on a join made on a column containing both columns (see figure 1-3). Even though a table could be based on a meaning of different tables, Fusion Tables allows a user to specify the attribution of each data sets origin and keep track of it.

Fourth, it provides various visualizations, such as a map, that can guess the data type, analyze the underlying schema, and present it correctly (see figure 1-4). The fifth feature of Fusion Tables is discussion tools for collaboration, which are the subject of this thesis. These tools allow data collaborators to better discuss and express their

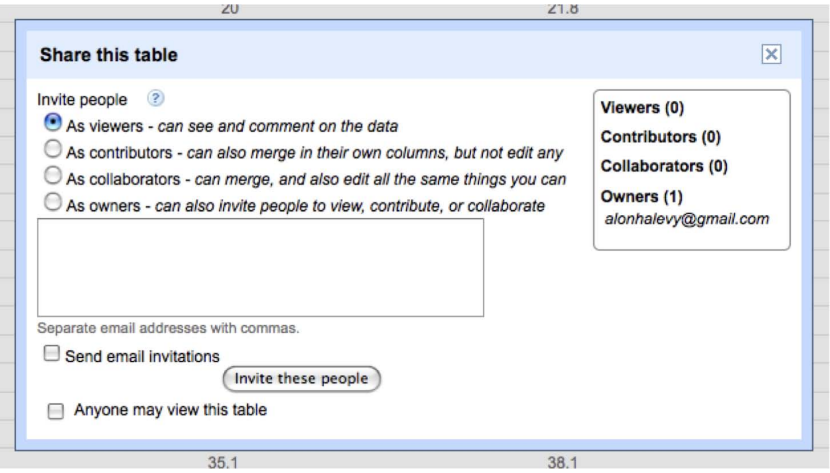

Figure 1-2: Fusion Tables provides a set of access control permissions that allow assigning users to become collaborators or viewers or to set the data as public

ideas, to share their thoughts, raise questions, point out outliers or incorrect data, or find new assumptions about the shared data. Users would likely enjoy having the ability to point to an interesting subset of data and to have the tools to give their inputs on this particular data set, whether it is a graph, a table or just a cell.

This thesis focuses on the implementation of collaboration tools. The first tool is discussions. Discussions are threads of comments (see figure 1-5) that can be added to different levels of data (cells, columns, rows, and tables). This level of granularity is important in large data sets because otherwise it may be impossible to keep track of the specific context of the comments. The discussion tools are geared toward helping the user easily identify activities in the data, browse through the comments (i.e. filter them), see their comments' threads, and view the history changes of the cell values. To better monitor and browse the comments, a discussion listing panel was introduced, where all the comments are aggregated and it is possible to sort and filter them. Another new feature that this thesis introduces is the ability to create "snaps" (see figure 1-6). Snaps allow users to define a specific query and visualization setting, which they can save for future reference within the data set or share with other users. For example, assume that Dan, a biology researcher, is working on a 100K row data set. He would like to be able to filter the data set, sort it, and even choose which visualization is most useful for analyzing this subset. Then, if he would like to load this

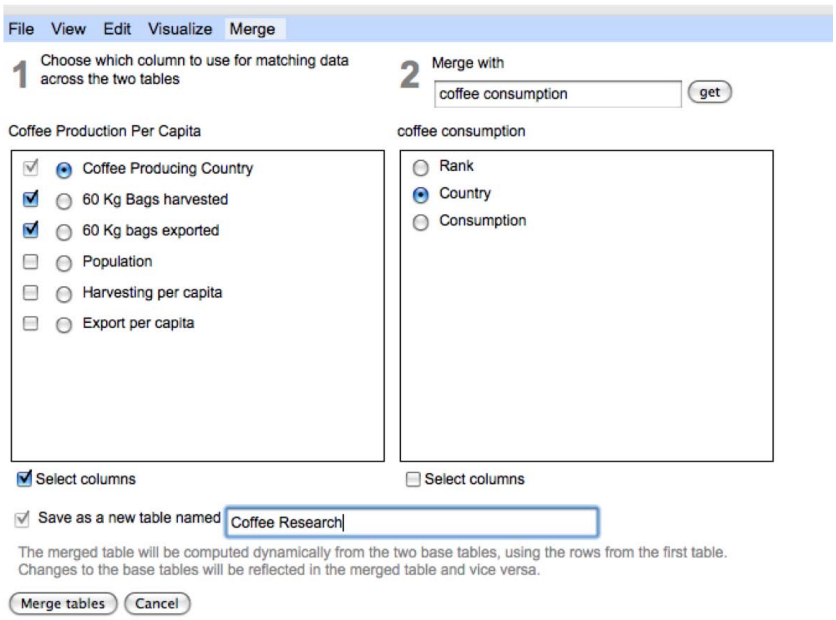

Figure 1-3: Fusion Tables allows merging of data from different sources, by merging them based on a column that contains the same values

filtered subset in the future or to collaborate on it with other researchers, he can easily save the current status of the screen and share it with his collaborators. We named this process "snap" because it represents a quick and easy way to creating a reference for the query and visualization. Unlike web browser bookmarks, snaps are unique in that they are shared and visible to all of the collaborators who can view the table. Snaps allow saving low level states of a web application, unlike bookmarking a document's URL in Google Doc, which only brings the user to the initial document screen.

The last part of this thesis defines a new model that might transform the typical collaboration on cloud DMS and will allow the discussion domain to be separated from it. Our first implementation of this approach is based on Google's Wave platform. It is a gadget that provides users with the ability to have an interactive discussion about the data without manipulating the DMS directly, only through the gadget (see figure 5-1). The gadget allows using SQL queries to show visualizations based on the shared data sets. One unique feature is that the gadget enables the users to tell a "data story." A data story can be told by using different visualizations from the same

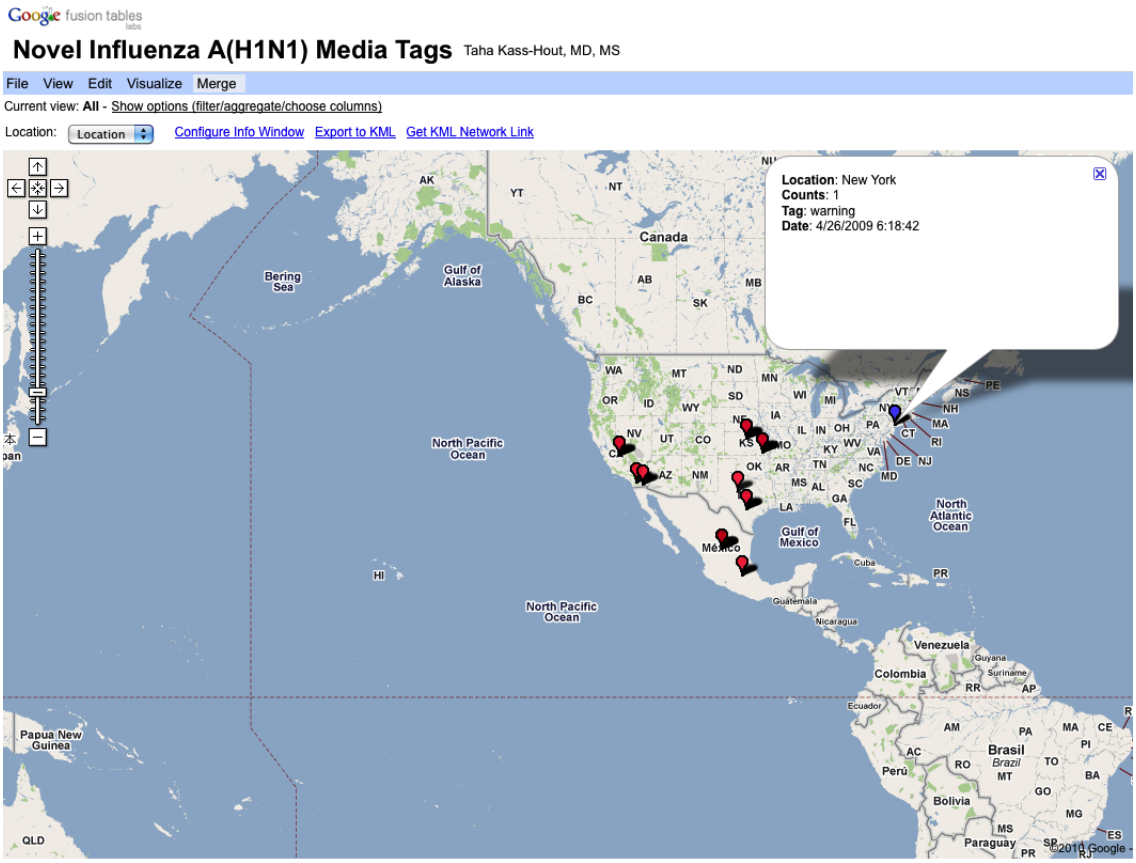

Figure 1-4: Fusion Tables allows easy mapping of geographical data points on a map. In this example: a mapping of H1N1 cases in the US and Mexico.

or from multiple data sets, and by adding comments to each of the visualizations. By using the wave platform, it is possible to later use the playback function, which allows users to browse through the discussion as if it were a slide show.

The implementation of the Fusion Tables-Wave gadget lays the ground work for future work that can enable more DMS and discussion space separation. This separation allows users to discuss their data on the platform of their choice, and may provide collaborators with more freedom to find new relations and facts. These discussions can later be used by the DMS to get a better understanding of the data set. For example, a blog or a wave that uses two different data sets in the same post can act as an indicator that these two data sets may have something in common, and then the DMS system may recommend that users view both data sets.

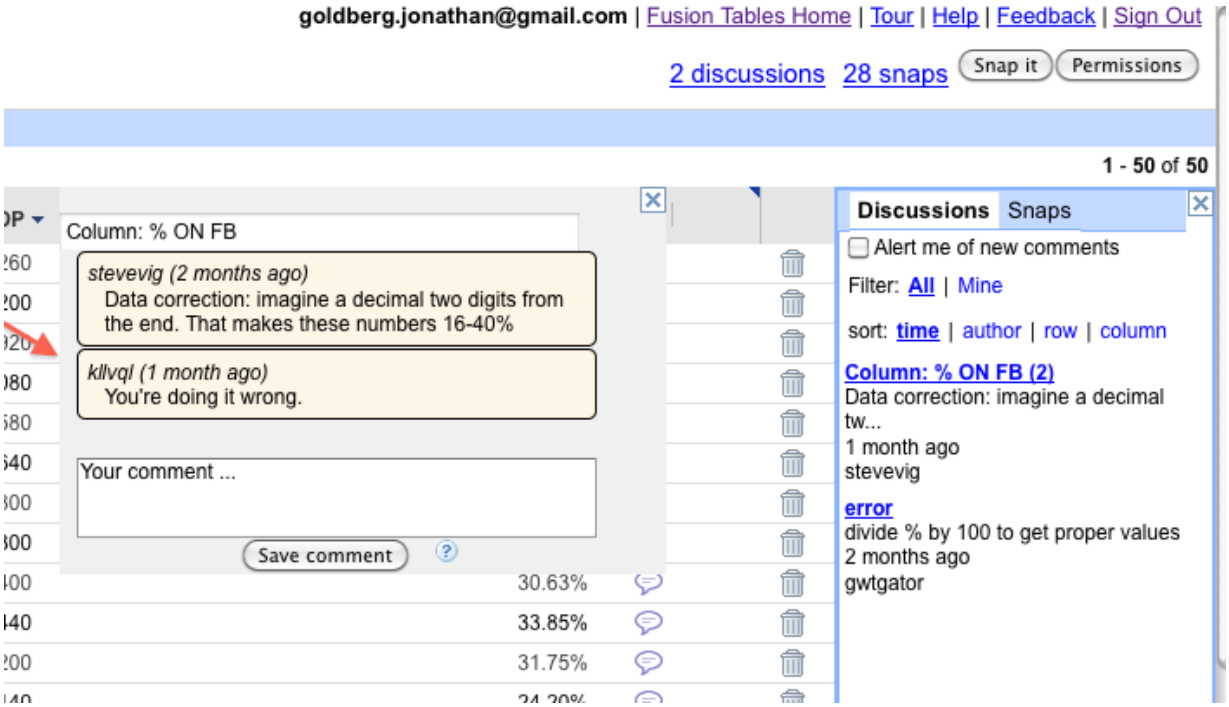

Figure 1-5: Discussion threads allow users to leave multiple comments on different levels (Cell/Row/Column) of the data set.

Figure 1-6: Clicking on "snap" permits saving the current query and visualization for future reference.

This thesis presents my three main contributions to Fusion Tables:

1. Providing the ability to easily view, sort, filter and set notifications of discussions.

2. Providing the ability to take snaps and later share them as a link or within the data set.

3. Laying the groundwork, with the Fusion Tables - Wave gadget, to promote discussion and collaboration outside of the DMS.

Currently there are no cloud DMS systems that are widely used for collaboration, and I suggest that Fusion Tables, with its combination set of tools, may provide the

right solution for this use, In order to evaluate these assumptions, I have conducted user experiments, which are elaborated in chapter 6.

1.1 User Scenario Example

Before implementing the discussion features of this thesis, we designed a user scenario based on the needs of our users. The scenario describes a general use case that can be relevant to any researcher who collaborates on data sets. For example, Winnie is a cancer researcher at an MIT lab, and she works with colleagues at universities all over the world. Having just finished a biopsy on a cancerous liver tumor, she has some interesting findings that she would like to share with her colleagues. These findings are stored in a data set with numeric parameters, along with images of the tumor and the cells. Winnie can easily create a table for her projects and decide which tables to keep make public and which to keep private and share only with her colleagues.
Winnie decides that she wants to give some of her colleagues permission to edit and collaborate on the data sets. She will allow them to add similar data measurements to her latest findings, make interesting observations, and even raise questions about her data set. Some of her collaborators are in different time zones, and they appreciate Fusion Tables' asynchronic collaboration tools. For example: they can add special comments to Winnie's data or create interesting snaps to point out intriguing facts. Since Winnie likes to respond quickly to any type of collaboration, she signed onto a notification service that will inform her about any changes or comments to her data sets.

1.2 Thesis Outline

The thesis consists of the following five chapters:

- Chapter 2 - Related Work - This chapter discusses and compares applications that provide functionality similar to that of Fusion Tables, while comparing

them on two levels: projects that create and combine data sets and projects that focus on data sets and visualization collaboration.

- Chapter 3 - Discussion Tools - This chapter discusses the functionality and the implementation of the discussion features that are available in Fusion Tables: features such as comments, the discussion listing panel, and the comments notifications.

- Chapter 4 - Saving Fusion Tables States - This chapter discusses the design and implementation of a feature that allows users to save customized queries and visualization settings for future reference or to share this state as a link with other collaborators.

- Chapter 5 - Collaboration Outside of The DMS - This chapter starts the discussion about new ways that collaborators can begin their discussion outside of the DMS, where they can use their own tools for discussion. It also presents our first attempt to accomplish this by providing a Fusion Tables - Google Wave gadget.

- Chapter 6 - Evaluating Fusion Tables' Collaboration Tools - This chapter discusses the analysis of our user evaluations. The goal of the evaluations was to answer the following questions:

  - How do users actually use Fusion Tables to collaborate on structured data?
  - Do users find the collaboration tools that were discussed in this thesis useful?
  - How can the collaboration tools of Fusion Tables be improved?

Chapter 2

Related Work

This chapter discusses and compares two categories of applications that provide functionality similar to that of Fusion Tables: projects that create and combine data sets and projects that focus on data sets and visualization collaboration.

2.1 Projects That Create and Combine Data Sets

During the last several years, researchers have implemented applications that provide an easy and intuitive way to publish and visualize structured data sets. One of these applications is Exhibit, which was developed at MIT [2]. Exhibit allows individuals without any advanced computer skills to easily build interactive websites on top of different types of data sets. Exhibit also supports the use of semantic web metadata. This metadata allows future semantic web tools to utilize the data for different uses and interpolations. Since Exhibit is relatively easy to use, it surely contributes to the number of visualized sites that provide structured data with advanced semantic web tools in the public domain (see figure 2-1). Exhibit, unlike Fusion Tables, allows only a passive view of the data and does not provide discussion tools on top of the platform. It should be noted that it is possible to combine Exhibit with Fusion Tables. Exhibit can generate a web page based on Fusion Tables' data sets by retrieving the data as a JSON file.

Another framework that aims to combine different types of data sources under

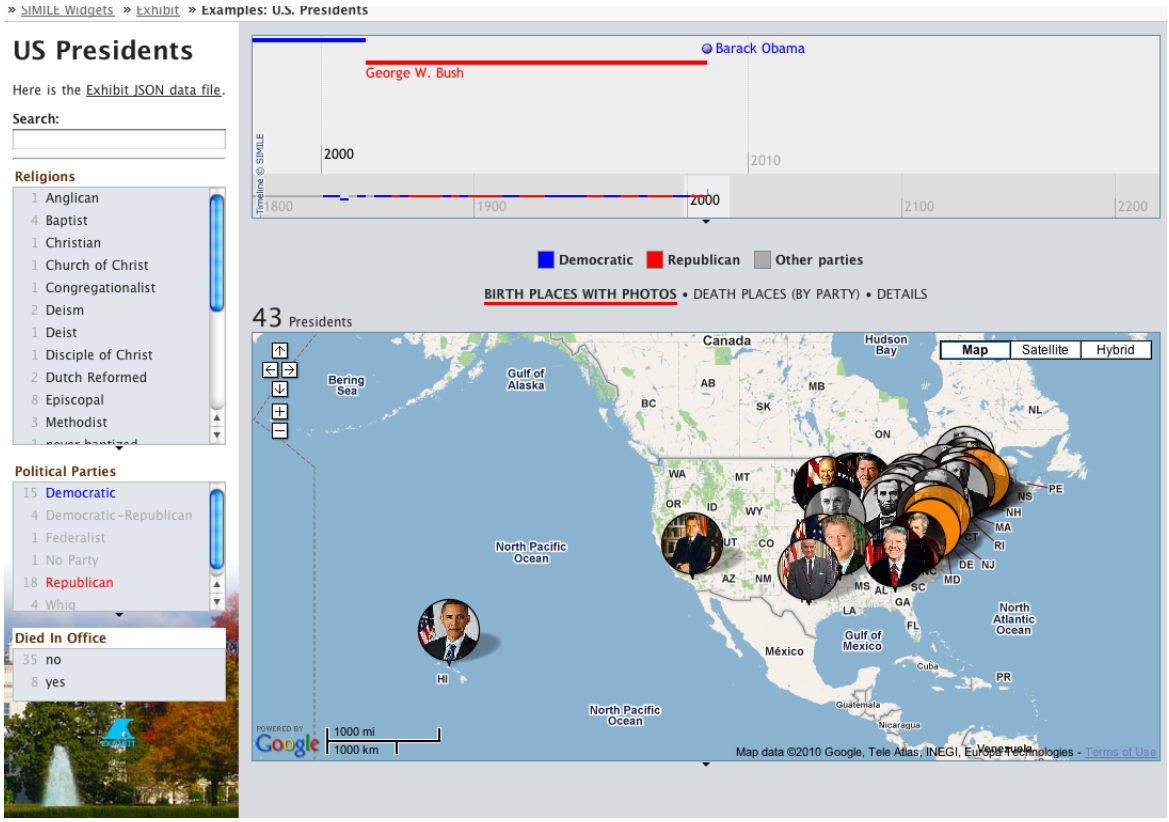

Figure 2-1: Example of Exhibit visualization - List of US presidents

one umbrella is DataSpaces. Franklin, Halevy et al. [3] proposed a new approach to the Database Management Systems (DMS). In the new approach, the DMS is not required to have complete control over the data sets. Data Spaces allows the data to be managed by the participant system, but it provides a set of services that offer better querying, searching and cataloging of the shared data. DataSpaces or any other framework that will allow collaborators to easily manage different types of data sets, will act as the primary framework for any large scale multiuser collaborations system for structured data. Fusion Tables provides this functionality by allowing users to easily upload different types of structured data sets and manage them all under the centralized "DataSpace," which is Fusion Tables.

2.2 Data Set Visualization and Discussion

Several applications provide tools for discussion and visualization of shared structured data, each with its own advantages and drawbacks. In this section I discuss the different applications and compare them to Fusion Tables.

The basic feature for every type of online discussion, especially blogs, is leaving some sort of comment on the provided data or article. The ability to leave relevant comments becomes harder when the comments need to be left on specific points on the visualization or on a particular entry in the data set. The first application that tried to solve this problem was "Sense.Us," which was built by Heer et al. [5]. It provides a novel way to create an asynchronous social interaction regarding visual data set representations. Sense.Us allows collaborators to bookmark the screen visualization (similar to "snap") for future reference, add text and graphical annotations, and conduct discussions of specific data. Sense.Us, unlike Fusion Tables, is only concerned with the mechanisms for asynchronous collaboration and not with other data management features; this is mostly because it provides only a few limited data sets that are available for discussion. Shortly after the Sense.Us paper was published, IBM's website "ManyEyes" [6] was launched in collaboration with Sense.Us. ManyEyes provides the missing data management tools to supplement the ideas that were discussed in the Sense.Us paper. Since the launch of ManyEyes in June 2007 and that of Fusion Tables in June 2009, three more sites with similar goals related to improving collaboration on structured data and providing better visualization have joined this space. These sites are Swivel [9], Factual [10] and Socrata [11]. Below is a list of all the different collaboration-related features that each site provides (see figure 2-2]:

- Visualizations - All the sites understand the importance of generating visualizations to better analyze data, and therefore all the sites provide advanced tools to generate visualizations. But there are still some differences. Socrata does not allow users who are not collaborators or owners to generate their own views of the dataset, unlike Fusion Tables, which allows every user with permission to view the data to generate new visualizations. The benefit of this approach is

that it gives only the owner of the table control over the how the data is being represented, which ensures only valuable and meaningful visualizations. The negative side of this approach is that users are not able to intimately interact with the data or try to find new findings. Also, Factual is the only site that requires users to modify HTML code in order to generate visualizations. This modification provides better customization to the visualizations, but it prevents users who are not computer experts from using this functionality. Fusion Tables not only provides visualizations that are easy to generate but also has auto schema engine that can recognize types of columns type (e.g. columns that represent countries can be used to place the data on a map).

- Data Collaboration - All the sites support presenting data in a table view, and allow editing them. All the sites but ManyEyes allow the data set owner to provide collaboration permissions to other users and allow them to modify the data as well. By doing so, ManyEyes identifies itself as a site that provides collaboration only on the analysis side and not on the data management aspects. By providing both, Fusion Tables may attract users who want both types of collaboration.

- Discussion Tools - All the sites but Factual allow users to leave comments on the table level. Factual allows comments on the cell level, only if the users have modified them due to an error or in order to add new information. Fusion Tables is the only application that supports comments on the different levels of the data (table/row/column/cell) and that has a discussion listing panel that allows easy navigation between the cell discussions. It seems that Factual focuses only on the most common scenario of cell discussion, which is error fixing. Fusion Tables, on the other hand, has a more lenient approach that allows users to leave comments on a cell even if they did not perform any changes to the data. This provides more flexibility to the user and therefore could be useful if users just want to point an interesting fact in the data.
  ManyEyes considers each visualization as a comment; therefore, each comment

on the data sets itself will not appear on the data set visualizations and every new comment on an already created visualization will be considered, as well as a new visualization. The advantage of this approach is that it is easy to link the visualization and its specific comment since they are tied together, but at the same time it can generate an overload on the server because of the overuse of the same visualizations. To solve this problem, Fusion Table and Socrata provide threaded discussions, such that users can view the history of the discussion and reply to previous comments on the same visualization.

- Saving Query and Visualization Settings - All the sites allow users to save and share their customized queried visualizations. Fusion Tables is the only site that allows users to continue modifying the queries and changing the visualizations over the saved query in order to generate a modified saved query.
  This ability provides the users with more flexibility to continue from the place that the visualization and query were saved and not start from the beginning. Factual allows saving data sets as queries, but only as a new data set table. This can create an overload of tables instead of just allowing users to bookmark specific queries and visualizations inside a data set, as Fusion Tables snap does.

- Cell Value Change - Only Factual keeps track of every cell value change. Fusion Tables keeps track of the changes only when there are discussions on the cell. The other sites do not keep track of any cell changes at all. It seems that keeping track of every cell change is a desirable approach when there are only few changes per table, but if the table cells' values are being modified over and over again, it can become a huge overload for the system.

- Ratings - ManyEyes, Factual and Socrata allow ranking of their data sets. Socrata also allows ranking of comments within each data set. This feature allows the sites to present users with the most popular data sets. Fusion Tables does not allow ratings at the moment, but some sort of ranking is probably going to be added in the near future.

- API - Factual, Socrata, and Fusion Tables allow users to collaborate, query and modify data outside of their framework with an API. This feature can be used to build different applications and customized visualizations based on the stored data sets.

- Merge Data Sets - Only Fusion Tables and Factual allow easy merging of data sets. Users can pick two data sets and, based on a shared column, join both data sets into one. This feature allows a collaboration between users who own different data sets and can save a lot of time for the collaborators by preventing them from collecting the data themselves.

Another two approaches for leaving comments that are similar to Fusion Tables are an annotation management system that was developed at Purdue University [4] and NB [12].

The annotation system from Purdue provides the ability to use different granularity levels of comments (cell/row/column/subset) and the means to define how the annotation should propagate within the database (i.e, which queries are going to present the annotation). Unlike Purdue's implementation, Fusion Tables' discussion tools focus on ways to provide the ability to conduct threaded discussions and monitor the comments from the discussion listing panel and sort and filter the discussions there. Fusion Tables, like Purdue's system, shows the comments that are relevant only to the specific data query.

NB offers a framework for collaborating using asynchronic annotations. It provides the ability to leave public or private text annotations on PDF documents. This type of annotation is very similar to the cell comments of Fusion Tables, but they allow distinctions between public and private comments, which Fusion Tables does not.

	ManyEyes	Swivel	Factual	Socrata	Fusion Tables	Purdue
Allow collaboration on the data	Only owner can modify data	Only owner can modify data	Yes	Yes	Yes	Yes
Discussions	Visualizations are discussions	Table level	Only when cell value change	On tables and visualization level +threaded	Table/row/ column/cell + threaded	Table/row/ column/cell/ subsets
Allow modify query/ visualization bookmarks	No - Saves as a new visualization	No	Each query bookmark is saved as different table	No	Yes	No
Track of cell change in the data set	No	No	Yes	No	Only within the comments	Only within the comments
Ranking	Yes	No	Yes	Yes (Comments and Data-sets)	No	No
Data set merge	No	No	Yes	No	Yes	No
API	No	No	Yes	Yes	Yes	NO

Figure 2-2: Comparison of the different collaboration applications.

Chapter 3

Discussion Tools

This chapter discusses the functionality and implementation of the discussions features available in Fusion Tables. Features such as comment, the discussion listing panel, and comments' notifications are included. Fusion Tables' discussion tools contain the following functionalities:

1. The tools enable discussions on different data levels: rows, columns and individual cells. This granularity allows collaborators to have different types of conversations, whether it is a comment on the whole data set or a specific comment that points to an error in one data cell.

2. We designed a discussion listing panel that helps users view, filter and sort all the available discussions on a data set. This panel helps users find the relevant discussion. It shows only the discussions that are relevant to the filtered and aggregated data, and keeps track of any new discussions (see figure 3-1).

3. A discussion is created from a comment thread where each comment relates to the other comments, such as in the case where the comments are on the same data cell. Each comment contains not only the text, user, and time but also keeps track of the cell value at the time of the comment. This feature is very useful to collaborators who want to perform changes to the data but keep track of the reasons for the specific change, or to better understand a discussion that later resulted in a data cell change (see figure 3-2).

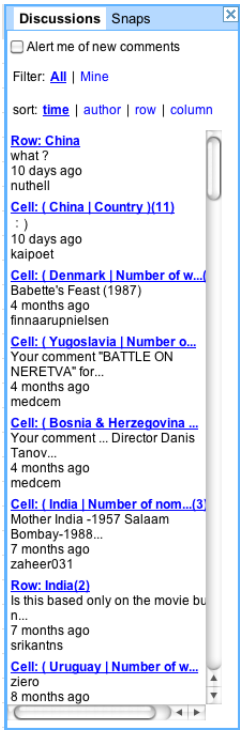

Figure 3-1: The Discussion Listing Panel allows to users easily scroll, sort and filter comments.

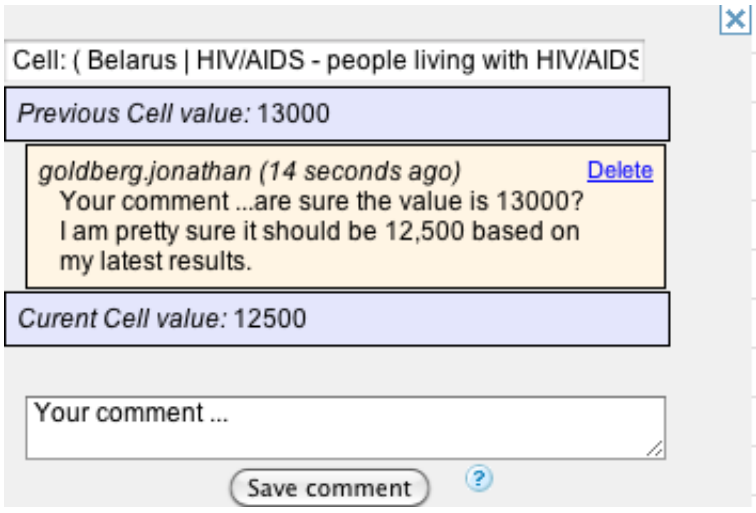

Figure 3-2: Discussion thread example that keeps track of the changed cell values.

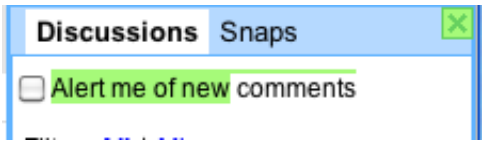

Figure 3-3: Comment alert notification - checking the checkbox is all that is needed to set a notification alert for new comments on the entire data set.

4. The discussions shown are based on the filtered query of the data set. The users can see only the discussions that are relevant to their current pages' view. For example: A data set with 100K values may have 3000 discussions: in the case that a query resulted only 10 values, the user will only see the discussions that are relevant for these values.

5. The discussion / comment notification, similar to an RSS feeder, allows the user to sign up to receive an email with updates on all the new comments on a data set. This feature helps collaborators keep track in real time of every change that occurs in the data set (see figure 3-3).

6. It is possible to view the discussions of all the different kinds of visualizations. For example, users are able to read and add discussions on a map, bar, etc., instead of moving back to the data set (table) view.
   This feature was available only on my internal version and was not launched

7. Fusion Tables encapsulates an advance access control mechanism on public documentation that allows users without permissions only to view the data and leave comments on it. It is up to the owner of the table to decide how to handle the comments.

3.1 Comments - Design and Implementation

This section discusses the main design decisions and the unique implementation aspects of Fusion Tables' comments. The main design decision was to abstract the comments outside of the data sets in such a way that the comments could be stored in a DB separate from the table's cell values.

This design has two benefits: The first is performance. It allows loading the dataset and the discussions with a separate server call. This is especially useful when the table has many comments and we want to show the data before fetching all of the comments. Also, without this separation, we would need to traverse throughout all of the data set entries to find all of the table's comments.

The second benefit is comment propagation. Because the comments are not part of the cells, we can decide whether we want to import the comments with the data cells when two tables are merged. Since the comments are not part of the data sets, we are able to present them as a reference to the original data sets instead of part of the current merged table. For example, if a comment discusses an error that was corrected in a cell, it may not be necessary to have this comment be a part of the new merged table, but it may still be useful to allow the user to know that there was a comment that was left on the original table's cell.

The comments are stored on a Big Table [7] database (similar to the actual data sets). In the database, each comment is stored as part of a discussion, and therefore each discussion may contain more than one comment. At the moment it is possible to have only one discussion per element (table, row, column, cell) but possible to allow separated discussions on each level. Each discussion contains the following ids: table, column, row, unique ID. Each comment contains the unique discussion ID and a comment unique ID. The comment also stores the actual text, the time of creation, the cell value, and the user ID of the creator. For optimized retrieval of the comments for a specific discussion, we use a tuplet as the primary key. The tuplet is: <Table, Column, Row, Discussion ID>. As mentioned above, the comments store the cell value at the time of creation because this helps a user to better understand the discussion thread, especially if the discussion resulted a cell value change. The comments store the user's ID not only for referencing but also for ownership reasons: Only the table's owner and the person who wrote the comments have the permission to erase a comment from the table (See figure 3-4).

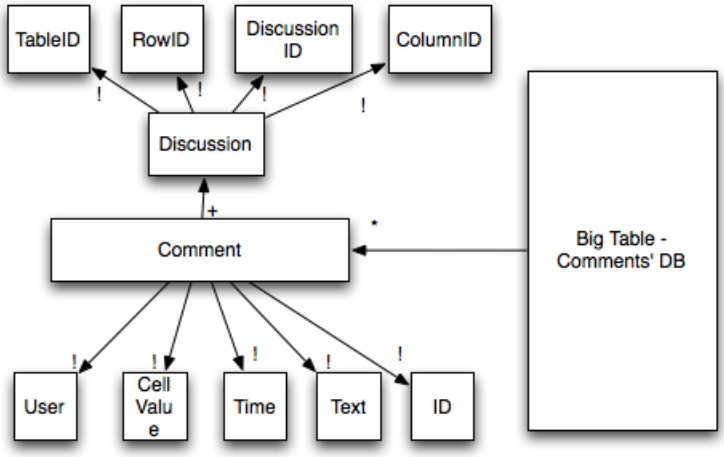

Figure 3-4: The comment's dependency diagram

3.2 Caching Comments on the Client Side

Because performance is a key factor in the user experience, we decided to improve the performance of reading and browsing through the comments by caching all the table's comments on the client's browser once that data set was loaded to the client. Once the user loads the data set, he sends a request to fetch all the comments on the table (with a threshold limit in case there are too many comments). Then the application stores the response on the client's cache. This design allows the user to easily browse the discussions without being restricted to the latency of more RPC calls that would have been added if it was necessary to make an RPC call for each comment view. Nevertheless, in order to prevent the scenario that a comment thread has been updated, because the cache is loaded every time a discussion is opened, the client checks whether there is an updated version on the server side, and if so, updates the discussion thread for the client.

3.3 Comments Retrieval Algorithm

A common scenario that poses a challenge is how to efficiently fetch a table's comments when the table has many rows and only a few comments; for example, more than ten thousand rows but fewer than ten comments. The problem arises because

at the moment Fusion Tables can query and show only up to one hundred rows at a time. Because many users may be interested in viewing all the available comments on a table query even if they can view only a subset of it, we had to come up with an efficient algorithm to filter all the relevant comments for a specific query and not only for a view. Because the comments are also indexed with their row ID, it is possible to know whether a comment is relevant for a specific query by examining whether its row ID is valid for the current query. One solution is to find all the rows that are valid for the current query and then retrieve all the comments that match these rows. The downside of this approach is that if we have many more rows to query than comments, we will have to query them all before filtering the comments, even though we can present only the first hundred rows on the client view. This approach will result in a very high unnecessary overhead on the DB. Therefore, we proposed a better algorithm that checks whether the comment's row is part of the hundred rows that are already presented and, if not, the algorithm will validate whether the comment 's specific row answers the query. Because in this scenario there are many more rows than comments, this process will take less time to run if our proposed approach is used.

3.4 Discussion Listing Panel

As more comments started to appear on the page, we noticed that it was very hard to browse through the comments or to notice whether someone had left a new comment. Originally Fusion Tables worked in the same way that other applications, such as Google Docs, left comments on structured data: The common way was to add an unthreaded discussion to the cell, with a small colorful marker inside the data cell (See figure 3-5). This type of comment marker makes it almost impossible to track all the discussions in the dataset, filter them, or sort them.

To solve this problem, we introduced the "discussion listing panel." This panel aggregates all the discussions relevant to the current data query in a collapsable panel on the right side of the screen (See figure 3-1). The discussion panel allows users to

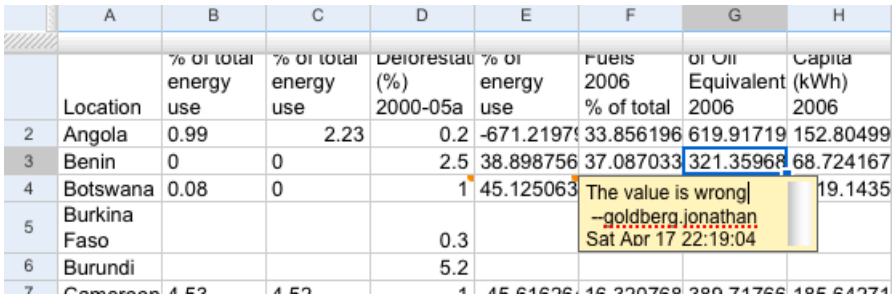

Figure 3-5: Example of structured data comment on Google Docs

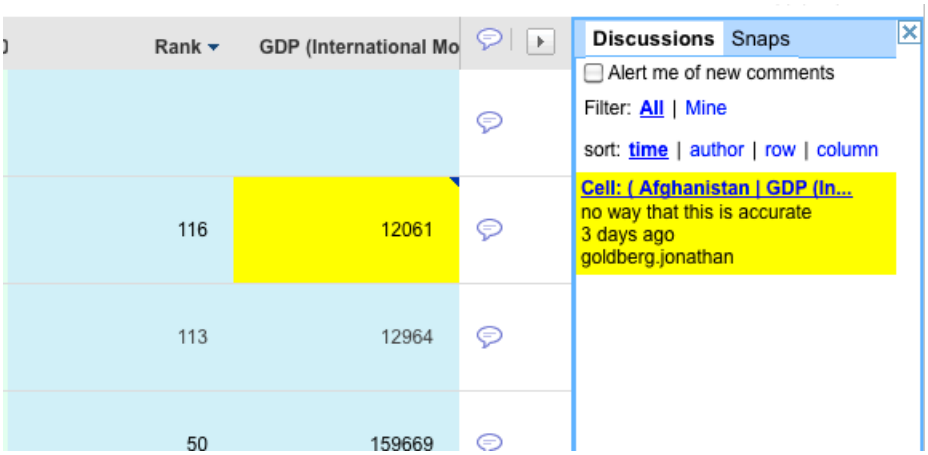

Figure 3-6: The discussion listing panel aggregates, sorts, filters and highlights the discussions in the dataset.

easily view all the discussions using different sorting and filtering combinations (time, author, rows, etc.). The discussion panel also embraces its relation with the data by highlighting in yellow the relevant data cell as the cursor hovers over the discussion listing. This feature helps users to directly relate the data itself and its discussion (See figure 3-6). The implementation of the discussion listing panel also required us to write an algorithm that resized the number of viewable columns based on the size of the window. The algorithm dynamically calculates the number of viewable columns on the screen based on the actual browser's window size subtracted from the width of the discussion listing panel and then divided by the average column width. This algorithm sets the mapping between the comment's location on the screen (HTML location) and its actual column/row location in the dataset. It also supports reversed ID conversion, from the IDs that are stored on the DB to the current HTML IDs.

Another feature, which we added in order to provide the user with a better indi-

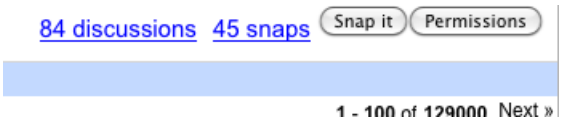

Figure 3-7: The discussion and snap counters provide a quick indication of the number of discussions and snaps available on the data set table.

cator of the number of comments that each table has, was a counter that indicates how many discussions and snaps are relevant to each dataset query (See figure 3-7).

3.5 New Comment Notification

Based on the feedback we received from some users, it was apparent that one of the most demanded features was a notification of when a new comment was posted. The solution that we provided to meet this demand was to implement a modular user event system that can perform different actions. We implemented a new module that can be extended to store different personalization settings, such as, in this example, comment notification. Another option that this module can later be used for is to set different personalization setting for users, such as which panel to keep open.

For the comment notification, we added a table that uses the table-id as the primary key, and next to it we stored the user-ids that are registered to receive notifications. Every time a new comment is saved on the table, the DB is queried and a message is sent to all of the users who asked to receive a comment notification. At the moment it is impossible to set alerts to specific cells or rows, but this additional functionality could be easily added in the future.

We also designed the UI to be very simple and straightforward. All that the user needs to do in order to sign on or to remove himself from the service is to check or uncheck the notification box in the Discussion Listing Panel (See figure 3-3).

Chapter 4

Saving Fusion Tables States

This chapter discusses the design and implementation of a feature that allows users to save customized queries and visualization settings for future reference or to share them as a link with other collaborators. We named this feature "snap" since it reminded us of the action of taking a snapshot of the current state of the client.

We were motivated to create this feature for three reasons. First, we wanted to provide a tool that was more collaborative and shareable than regular the browser's bookmarks. The downside of a regular bookmark is that it is saved only on the user's browser and therefore other viewers of the same data set are not aware of it. Snaps by default are shared with all of the data set viewers.

Second, we wanted to make the snap a dynamic, non strict, bookmark. For example, if user A shares a snap with user B, and user B thinks that some parts of the snap's query need to be changed, user B can modify only the specific query and then replace the old snap. The fact that snaps are dynamic allows users to continue working and modifying the snaps from the point that they were saved and does not require them to start from scratch, but at the same time they can save (unless deleted) the original snap for future reference

Third, Fusion Tables is a web-app and therefore most of the actions that the client will perform do not change the url link of the app. This fact prevents users from saving unique bookmarks of the application state, and to solve this problem we introduced a quick and easy way to save states within the web app.

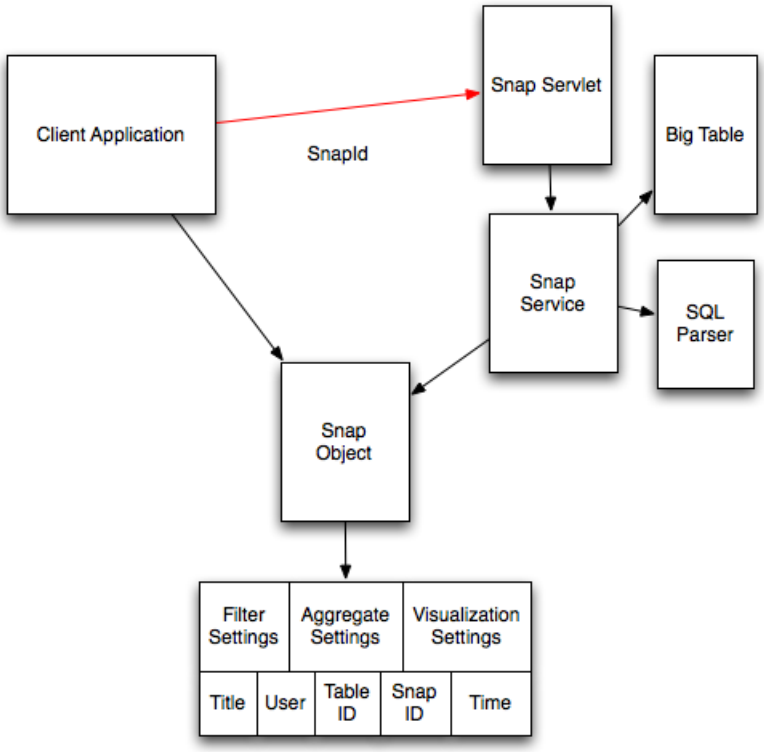

Figure 4-1: The snap's dependency diagram shows the relationship between the client and the server.

4.1 Storing and Loading Snaps

Snap was added as a feature after most of the current infrastructure of Fusion Tables was already in place. Therefore, snap is based on the same DB frameworks as the Fusion Tables discussions and it is stored on Big Table and Megastore, Google's data storage infrastructure, as well.

A snap is a serialized object that is sent between the application and the server. It contains the filter, aggregation and visualization settings that the user defined on the application. The snap also contains the user ID that generated it, time, label, table ID, and unique Snap-ID. The filter and the aggregation settings are implemented as serialized objects as well, and the client uses them to easily modify and control the views and queries on the application. Because it is impossible to use these objects by themselves to query the DB, an SQL serializer was developed, which is responsible for transforming these objects into SQL statements. These SQL statements can be

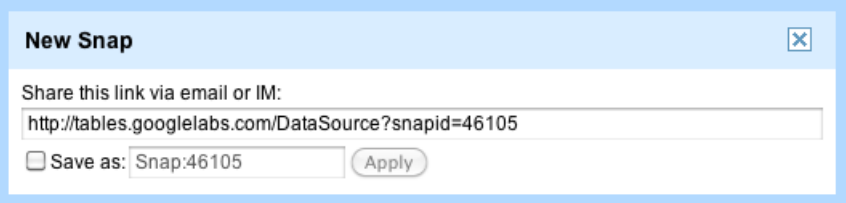

Figure 4-2: This is the snap's save dialog, which is promoted once a snap is saved on the server.

stored as strings and saved on the snaps DB for future retrieval. In a similar way, the visualization settings were serialized in such a way that it will be easy to store them on the DB as strings and integers. The snaps also have a unique ID. This ID is not only used as an identifier for the DB but also as a url servlet parameter to load the snap directly, or when the client asks the server to load a specific snap from the discussion listing panel.

Snaps are generated by clicking on the "Snap It" button (See figure 1-6). Once clicked, the client's app gathers all the unique settings of the snap and sends the snap object to the server. The server generates a unique ID, stores the snap on the DB, and sends it back to the client. Once the client receives the updated snap, a dialog popup prompts the user with the snap's unique link and also allows the user to provide the snap with a unique title.

The Fusion Tables discussion listing panel has also been extended to support the saved snaps. Users are able to view and load the available snaps of the data set. Similar to the discussion listing panel, it is possible to sort and filter all the snaps based on their creation date, author or title (See figure 4-3).

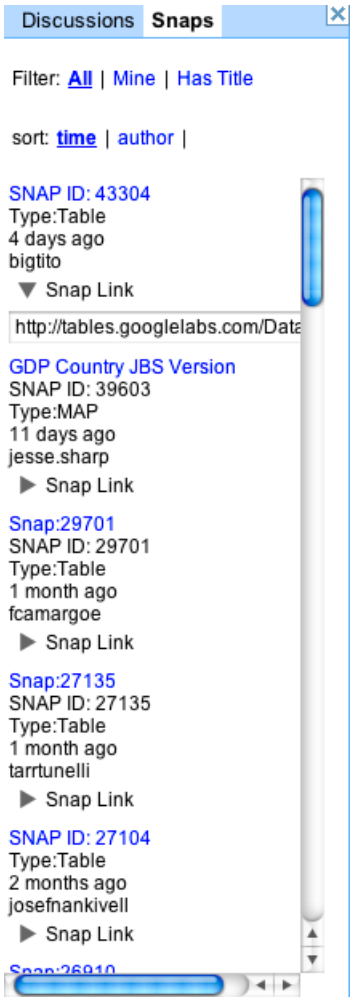

Figure 4-3: The snaps panel shows the list of all the snaps and allows sorting and filtering operations.

Chapter 5

Collaboration Outside of The DMS

As discussed in the previous chapters, Fusion Tables provides several approaches for collaborating on its data sets. Nevertheless, as much as we believe that our implementation answers the needs of our users, it may still not be the ideal solution for some of our user scenarios. Some users would like to use their own tools or preferred platform for discussion, but at the same time enjoy the DMS benefits and visualizations that Fusion Tables provides in the cloud. While users collaborate on their favorite platform, the DMS can still communicate and monitor the sites where the discussions take place and, based on that, make new assumptions and analogies about the data sets. For example, Fusion Tables can keep track of which blogs or Wikipedia articles are using the specific data sets and then provide links to these blogs on the data sets. Another use case could be that if two or more datasets were referred to in the same blog, email or wave discussion, Fusion Tables could most probably assume that the data sets are related and suggest that users view them as well. This chapter starts this discussion by presenting one application that allows to conduct a discussion outside of Fusion Tables.

5.1 Fusion Tables - Wave Discussion Gadget

This section discusses the first attempt to provide a tool that allows discussion outside of the DMS. We designed a Google Wave [13] gadget (see figure 5-1) that enables

users to execute SQL queries on the available datasets that are stored on Fusion Tables and then pick the best visualizations to represent them with. Wave provides a framework that allows the users to present these visualizations and conduct synchronic and asynchronic discussions around the gadget. Once the gadget is added to the discussion, users can easily create new visualizations or modify their peers' visualizations. Similarly to snaps (as described in the previous chapter) the visualizations are not static images and they change if the actual data changes.

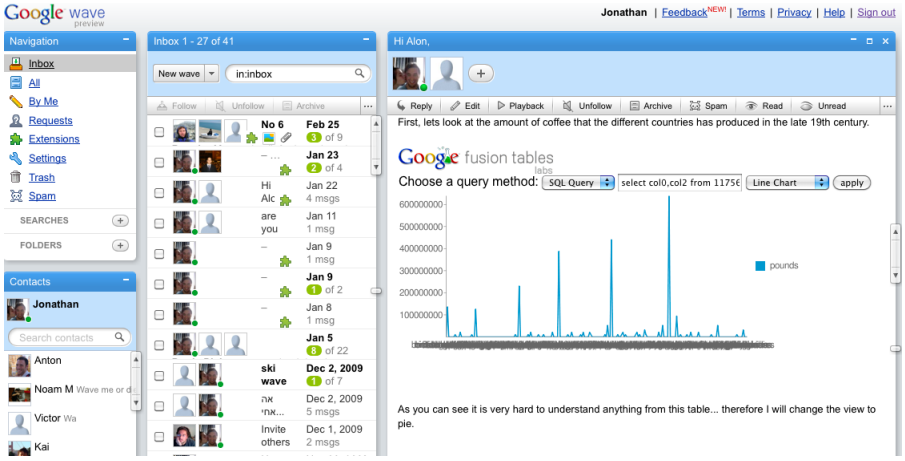

Figure 5-1: Fusion Tables-Wave gadget is our first implementation that tries to separate the data set collaboration outside of the DMS.

Wave can store the different states of a discussion, and therefore it provides the ability to "play back" all of these states, by viewing one state at a time. This feature is useful if users want to see how a discussion evolved over time, or to tell a data story by adding subtitles to each visualization. Our implementation of the gadget supports the playback mechanism and allows us to save different states of the gadget (chosen sql query and visualization type). Then when the users click "playback" they can see how the sql and the visualizations evolved over the gadget (see figure 5-2).

The implementation of the gadget is based on java-script and it is stored and run from a server at MIT. The implementation consists of an HTML part, which is the UI, and a javascript part, which handles the UI controller and communicates anmong three APIs:

1. Wave's API - used to load the gadget and to store the different states of the

gadget.

2. Fusion Tables' API - retrieves the data in a JSON representation based on the query from the gadget.

3. Google Visualization's API - provides the tools to draw the data that was received from the Fusion Tables.

Other scenarios we wanted to introduce into this gadget include a direct method to load snaps and a way to control the table's queries in a way similar to the interface with which they are controlled within the Fusion Tables application interface.

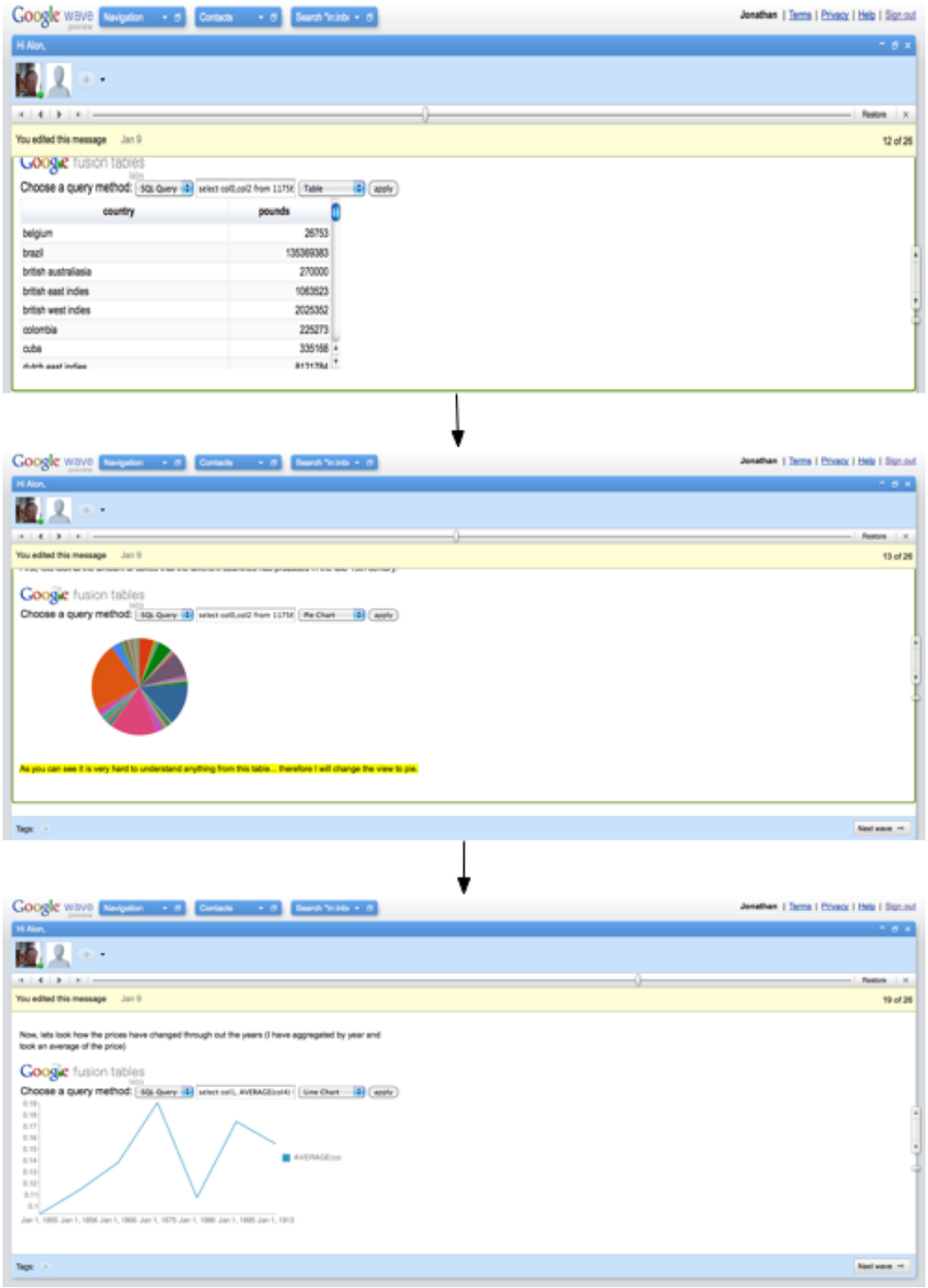

Figure 5-2: Wave gadget's playback example, showing the different states of a gadget as it evolves through three steps in a conversation.

Chapter 6

Evaluating Fusion Tables' Collaboration Tools

This chapter describes the user evaluations that we conducted and their analysis. The theme of these experiments was inspired by the collaboration experiment conducted for SearchTogther by Morris [8]. The goal of the evaluations was to answer the following questions:

- Do users use Fusion Tables for collaboration?
- Do users find the collaboration tools that were discussed in this thesis useful for their collaboration?
- How can the collaboration tools of Fusion Tables be improved?

The evaluation is based on three sources:
The first is results from Google Analytics, which has monitored the usage of the collaboration features since the day of their implementation.
The second is a user study within the MIT community to evaluate the UI and the collaboration tools.
The third is a field study that we conducted. In this study, we contacted actual users of Fusion Tables and provided them with a survey to learn more about their collaboration usage scenarios.

6.1 Google Analytics Results

To assess the general usage of the collaboration tools that were introduced in Fusion Tables, we set up different events that monitored the usage of those tools. These events have been traced by Google Analytics. The events are saving a new comment, viewing a comment, saving a snap, viewing a snap within a data set or from a link. By comparing the event results to the total number of visitors to the site, these results can assist us to better understand whether the users are aware of the features and use them. The results do not provide any information regarding the type of usage or the UI experience.

6.1.1 Results

By analyzing the the ratio between the number of visitors and the use of the collaboration tools that were discussed in this thesis, which is about 40%, we can see that there is almost a constant ratio of feature usage to the number of visitors to the site (see figure: 6-1).

6.2 User Study

Our user study collected data from eight members of the MIT community, who were separated into four pairs. All the participants were experienced with data web collaboration tools and had participated in group projects that required collaboration on structured data. None of the participants had used Fusion Tables prior to the experiment. The user study simulated collaboration between two researchers who are not located in the same area and need to collaborate on the same data set and generate a visualization output. The testing module was separated into two parts. The first part was a UI evaluation survey, to assess the ease of use and the users' experience

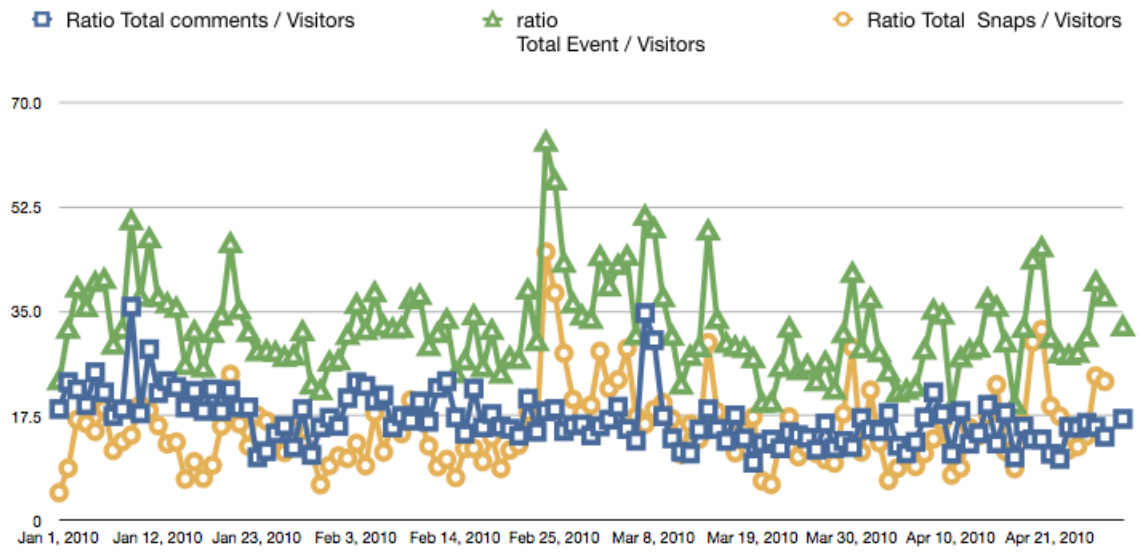

Figure 6-1: Analytics results for the collaboration events, representing the ratio between the total snap and comment events and the total visitors to the site between 01/01/2010 to 04/28/2010

of Fusion Tables. The second part was a collaboration task where the users had to collaborate in their pair on a shared data set and generate a combined outcome.

6.2.1 Preparation

The initial UI evaluation was done separately for each user in the pair. They were given a brief overview of Fusion Tables and then asked to perform basic tasks. During the execution of these tasks, the computer screen and comments were captured.

During the second part of the user study, we asked each pair to sit in different rooms and try to simulate a remote collaboration. Then we allowed each pair to pick a topic of their choice and generate a valuable blog post about any merged table, from the public data sets that are available on Fusion Tables. For example, they could have used data sets that cover GDP, Coffee Production, Oscar Winnings, Homicides in the US, etc.

At the end of the of the task, we asked the participants to fill out a subjective satisfaction survey to evaluate the collaboration tools of FT. (For more information, see Appendix B).

6.2.2 Results

UI Evaluation

All the users were able to perform all of their tasks and some were excited by the simplicity of Fusion Tables. But there were a few UI issues and in some cases without guidance, the users could not successfully accomplish the task. The issues were:

- Terminology: there are some terms that are used in Fusion Tables, which, without a hint, were not clear to most users:
  - "Snap" - Most often the users could not correlate "snap" with saving the current query and visualization. They looked for this type of feature under the file menu.
  - "Merge" - Some users thought that this term refers to column merging and not to merging a data set based on joined column.
  - "Table Gallery" - Table Gallery is used on the home page to refer to the public tables folder, but most of the users could not correlate between "table gallery" and public tables.
- In the merge screen (see figure: 1-3), it is almost impossible to notice that the text box that lists the other available data sets is clickable. It should be replaced with a scroll down listing.
- The Discussion listing panel should be open when the users log in for the first time to a table. It was hard for users to figure out how to open it. Also, users would like to see a preview of the latest comment and not the first on the listing panel.
- Comment notification - Users wanted to have the ability to mark a comment as private / public or to notify specific users or all users about it. It is worth mentioning that all the users were able to easily find the comment notification checkbox once they opened the discussion listing panel at least once.

Post-Collaboration Task Evaluation

At the end of the evaluation the participants had to reflect on their collaboration task in an online survey. Based on the results, we can see that the visualization tools were the most important feature for success and then equally important the merge tools and the snaps. The comments were the least important for the collaboration task (see figure 6-2).

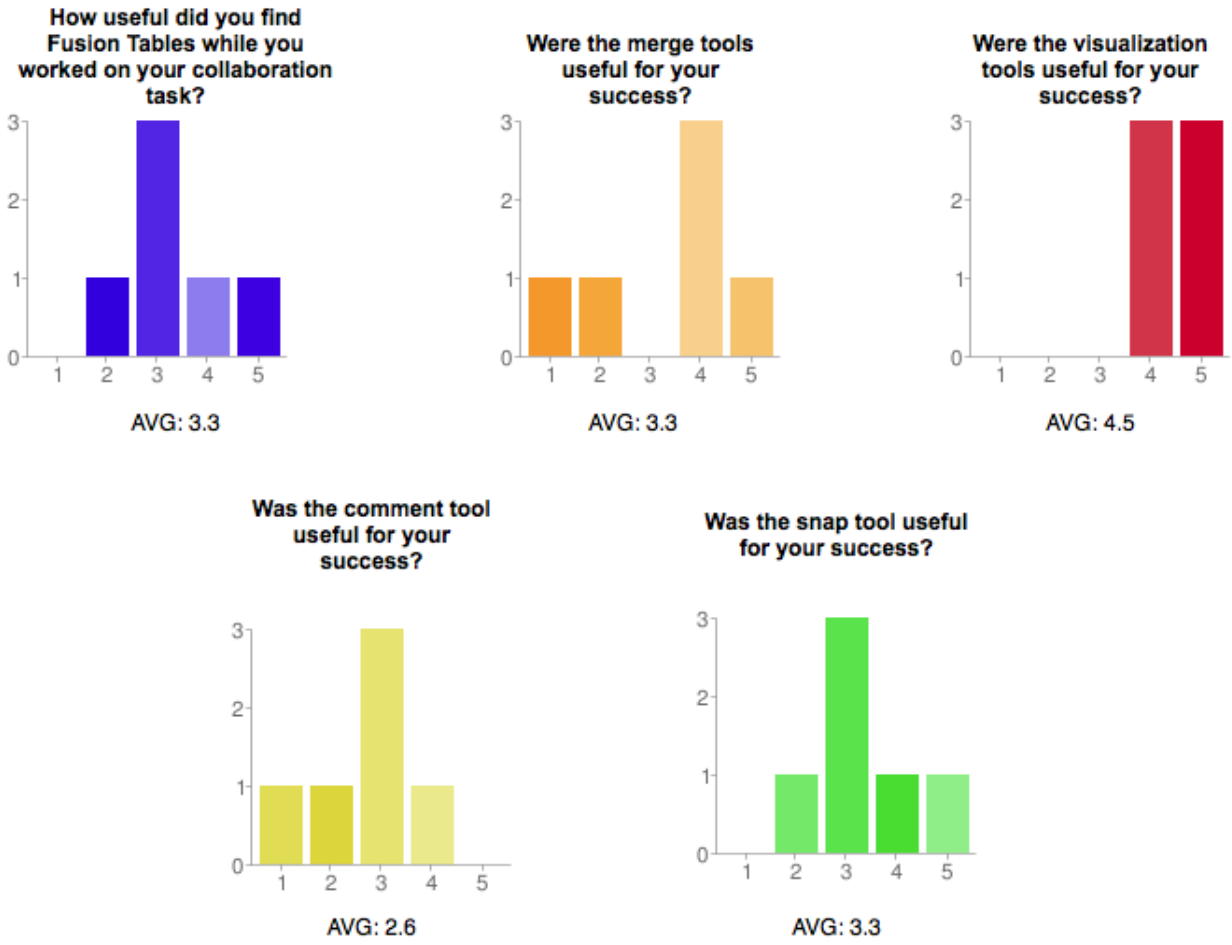

Figure 6-2: Summary of the user study post-collaboration task evaluation.

Most of the users indicated the need for an instant messenger in order to collaborate in real time. The users thought that the comments were not useful for real time synchronic collaboration because they left a trace of unimportant messages that had no real value for the table and they should have been part of an online chat. The

users indicated that the snaps feature was useful, but they wanted to have the ability to have a separate discussions on each snap, similar to the discussions in ManyEyes.

6.3 Field Study

This study aimed to gather the responses from Fusion Tables' most active users. The goal of the study was to better understand whether these users were aware of and used the collaboration features. Based on their inputs, we can better evaluate the current collaboration tools, and provide users with the tools that they need.
The field survey was posted on the Google Groups page of Fusion Tables, and in twenty other related blogs that discussed Fusion Tables during the last six months. In return for participating in the survey, we offered a $40 gift certificate to one of the participants who was chosen at random. The survey questioned the users about the ways in which they use Fusion Tables for collaboration (the list of questions appears in Appendix B).

6.3.1 Results

Thirteen users submitted responses to the surveys. Seven reported that they were using Fusion Tables for collaboration purposes.
Based on the users who indicated that they were collaborating via Fusion Tables, 38% indicated that they use the embedded visualization, snaps and the sharing options (the user's permissions, such as collaborators, viewers and owners). Twenty-six precent indicated that they used the discussions, and none of the participants indicated that they used the comments notification tool.

One of the participants indicated that he used the comment tools "to argue about whether the assumptions we make on the data are correct, incorrect, irrelevant. Basically it's a G-talk discussion that is saved with the data itself, so we can refer back to the assumptions in a better way than we would with comments on an Excel sheet." This usage is very similar to our main user scenario for the comments.

Almost all of the participants indicated that they used email and messenger to

collaborate on their data sets. Based on the user study we know that the need for an email usage is to support asynchronic collaboration because it allows the collaborators to notify each other when then they left a comment for a specific user. This usage can be addressed if the comments will support a personal notification. The usage of a messenger client is necessary during synchronic collaboration sessions when the collaborators want to freely discuss a data set without leaving permeant comments on the data.

The users also mentioned that they would like to have better merging tools, the ability to import more files, and to use spreadsheet formulas.

6.4 Analysis of the results

In this section we will analyze the results while answering the questions that were stated at the beginning of this chapter. Our first question was whether users use Fusion Tables for collaboration. Based on the field study, we learned that there are users who see Fusion Tables as a collaboration tool. Due to the low response rate we are not able to provide a real estimate of the users who use it as a collaboration framework. Nevertheless, based on the analysis, about 40% of the visitors to Fusion Tables used some type of collaboration feature. This usage can happen for different reasons that we cannot state precisely, but based on the user study that we conducted, we can assume that the UI issues that were detailed in the previous section may have prevented from users from discovering these features.

As for the users who do not use Fusion Tables for collaboration, they use it for its merging and visualization tools.

The second question was whether the collaboration tools that were discussed in this thesis are useful for collaboration. Based on the results of the field study and the user evaluation, it appears that half of the users use the visualizations and snaps. Combined with the user evaluations, it appears that the snap feature is not easily recognized and that may be one of the reasons for its lower rate of usage.

As for the comments, they were used by an even lower number of people, but as

one of the users indicated, comments are mainly used by experts who comment on data sets about assumptions and errors.

The third question was how Fusion Tables collaboration can be improved. One of the common answers was that users need better ways to synchronically communicate with each other. The main issue that users found was an inability to communicate their thoughts or messages with each other solely on the Fusion Tables platform. This inability leads to the use of email and instant messengers for synchronic collaboration. It seems that an integrated chat client could solve the synchronic collaboration problem, and, for the asynchronic collaboration, the ability to notify specific users about a comment or a snap may turn out to be useful and eliminate the need to use email.

Chapter 7

Conclusion

Google's Fusion Tables provides a framework for users who want to share, merge and collaborate on their data in the cloud. These principal contributions of this thesis focused on collaboration tools that can help collaborators on structured data sets better conduct discussions, point out outliers and convey ideas on the data compared to the available tools. This thesis makes three main contributions, which were primarily designed with the goal of improving collaboration via Fusion Tables. The first is an integrated panel that provides a new way to view, find and sort the comments of a structured data. These comments allow threaded discussions on the different granularities of the data set. The second is the ability to use "snaps," a dynamic state bookmarking that allows collaborators to save queries and visualizations and share them with other users or save them for later reference. The third is the implementation of the the Fusion Tables - Wave Gadget, which initiated a discussion about extending the collaboration outside of the DMS. These contributions were evaluated through user experiments, the results of which show that approximately 40% of the visitors to the site use the collaboration tools. Based on the user study, it appears that UI improvements can increase exposure to these features, and some additional functionality could be added to improve the collaboration features.

7.1 Future Work

The contributions of this thesis work have provided a new selection of tools for online collaboration on structured data, but there are still many ideas that, due to lack of time, I was not able to address.

Allow Advanced Discussion Outside of The DMS

With the implementation of the Fusion Table-Wave gadget, we showed the possibility of having discussions that are related to a data set on different domains without losing the connection to the data or to the functionality that Fusion Tables provides. By allowing users to interact with the data, whether by using an API or other means, we can allow the users to collaborate, without any restrictions or limitations, in their favorite environment.

Propagate Comments From Underlying Data Sets and Data Correlation

Derived from the above idea, the ability to find related data sets can immensely improve if the DMS is able to track the comments that were left on a data set, and propagate it up to dataset based on the original data set or to correlate between two unrelated data sets if there were discussions that involved the two sets.

Greater Personalization of Fusion Tables

Based on the framework that was implemented for the comment notification, it is possible to personalize the user experience even more. Users may want, similar to an email inbox, to keep track only of new discussions, and mark the discussions that were resolved or the ones that still require their attention. It may also be possible to leverage this idea to implement a method to view all the differences in the data set that occurred since the last time the user logged in.

Ranking

One of the users in the field study commented that he would like to see a different home page, which represents a selection of highlighted tables and data. One way to handle this need is allowing users to rank the tables and the comments.

Nevertheless, this thesis has three main contributions: the discussion listing panel, snaps, and the Fusion Tables - Wave gadget. Taken together, these three features provide a richer collaboration experience for Fusion Tables.

Bibliography

[1] *Google Fusion Tables: Web-Centered Data Management and Collaboration.* **Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen, Jonathan Goldberg-Kidon**, 2010, Sigmod

[2] *Exhibit: Lightweight Structured Data Publishing.* **David F. Huynh, David R. Karger, Robert C. Miller**, 2007, IW3C2

[3] *From Databases to Dataspaces: A New Abstraction for Information Management.* **Michael Franklin, Alon Halevy, David Maier**, 2005, SIGMOD.

[4] *Supporting Annotation on Relations.* **Mohamed Y. Eltabakh, Walid G. Aref, Ahmed K. Elmagarmid, Mourad Ouzzani, Yasin N. Silva**, 2009, EDBT.

[5] *Voyagers and Voyeurs: Supporting Asynchronous Collaborative Information Visualization.* **Jeffrey Heer, Fernanda B. Viegas, Martin Wattenberg**, 2007, CHI.

[6] *Many Eyes: A Site for Visualization at Internet Scale.* **Fernanda B. Vigas, Martin Wattenberg, Frank van Ham, Jesse Kriss, Matt McKeon**, 2007, IEEE.

[7] *Bigtable: A Distributed Storage System for Structured Data.* **Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber**, 2006, OSDI.

[8] *SearchTogether: An Interface for Collaborative Web Search.* **Meredith Rigel Morris, Eric Horvitz**, 2007, UIST.

[9] *www.swivel.com*

[10] *www.factual.com*

[11] *www.socrata.com*

[12] *http://people.csail.mit.edu/sacha/nb/*

[13] *www.wave.com*

Appendix A

Additional Work

This chapter describes the additional features that I have implemented during my 6-A internship, which are not part of the main contributions of this thesis. These features mostly needed to be implemented as a pre-requirement for the collaboration features presented in this thesis or as a high priority need to improve the user experience.

- Auto-complete: Implemented the Auto-Complete feature for the filter query panel. The uniqueness of this implementation is that instead of loading each column word separately, once the auto-complete is initialized, it retrieves all the possible wordings for every column at once. The ability to have an auto-complete feature was a crucial need for an improved user experience and for increasing the usage of the filters (which are part of the Snaps).

- UI enhancements - Cleaned and redesigned major parts of the home and the main table pages.

- MVP pattern and testing - Worked with Ano Langen and re-factored numerous parts of the code based on the MVP pattern to introduce more scalability, modularity and testability.

- .xlsx files support - Added the support of .xlsx file types to Fusion Tables.

Appendix B

Evaluation Questionnaires

B.1 Field Study Questionnaire

The field study questionnaire was sent to active users of Fusion Tables, who used it to collaborate with other users:

- How often do you use Google Fusion Tables?
- Approximately when did you start using Google Fusion Tables?
- Approximately how many people do you collaborate with on data sets using Google Fusion Tables?
- Are your collaborators located in the same office as you, or in a different location?
- Which features are most important for this collaboration (from the following list)?
  - Embeded Visualization
  - Snaps
  - Commenting / Discussion on Table/Cell/Column/Row
  - Comment Notifications

  - Sharing Options (setting users as collaborators/viewers/etc.)

- Can you please describe how you use the selected featuers?
- What other applications, websites, or communication tools do you use for this collaboration? Please describe how you use them.
- Which other collaboration tools or improvements would you like to see available on Google Fusion Tables?
- What other applications, websites, or communication tools do you use for this collaboration? Please describe how you use them.

B.2 MIT User Study

The study will be conducted in two parts:

- UI evaluation - Evaluates the features and the user interaction within Fusion Tables.
- Collaboration task- Simulates a collaboration between two remote users followed by a short survey.

B.2.1 UI Evaluation

During the UI evaluation, the computer video output will be captured and the test coordinator will note down the user comments and their feedback. The UI evaluation consists of the following tasks:

- Load a data set into fusion table.
- Merge a data set with another data.
- Leave a comment on the data.
- Run a query on the data set.

- Generate a visualization on the data set.
- Save the visualization and the query for future reference and share its link.

B.2.2 Collaboration Tasks

This questionnaire followed the successful completion of the pairs blog post. The following questions were answered using a scale of 1-5:

- How useful did you find Fusion Tables while working on your collaboration task?
- Was the merge tool useful for your success?
- Was the visualization tool useful for your success?
- Was the comment tool useful for your success?
- Was the snap tool useful for your success?

Open questions:

- Did you use other tools than Fusion Table to accomplish your task? If yes, please describe which tools and how you used them.
- Are there any tools or features missing from Fusion Tables that could have helped you in your task?