

**DEVELOPMENT OF A GIS-BASED ONLINE DISCUSSION SYSTEM FOR SCIENTISTS WITH GOOGLE EARTH API AND TWITTER.** S. Ohtake, H. Demura, N. Hirata, J. Terazono. Department of Computer Science and Engineering, University of Aizu, Ikki-machi, Aizu-Wakamatsu City, Fukushima 965-8580, Japan. (email: s1150041@u-aizu.ac.jp)

**Introduction:** Scientific discussions among researchers are very important. Face-to-face discussion is one of the traditional and a common style, but several online media tools recently including e-mail, bulletin board systems, and chat systems become familiar. These online media have several merits. Participants need not to attend to an online discussion in real time. Minutes of the discussions are usually recorded automatically. In contrast, the participants of the online discussion can't refer common data to be shared in all participants. If they can share and refer the common data sets throughout an online discussion, the online scientific discussion would be more productive and fruitful. Geographical information system (GIS) aggregate and display various type of geographical and geological data. It will be a valuable basis for online discussions providing common data.

From this point of view, we develop an online discussion system cooperating with GIS for researchers in the planetary sciences. With this system, researchers at world institutes can make a discussion on a specific location on a planetary body with shared observation data and their analysis results through the worldwide-web. The shared data on GIS help the participants to understand backgrounds of the discussion. All comments of the discussion are saved, and people can refer them later for further discussion.

We develop this system as a web application by mashup approach with the existing web services; Google Earth API [1] and Twitter [2]. This approach saves the development cost. We also prepare a web server with a database system constructed with MySQL [3] that connects the mashup components and provides backend functions of the system. Google Earth is a virtual globe developed by Google Inc. that renders map images and various geographic information on a 3D globe. Although most of mapping data are provided through streaming, additional user data can be displayed by preparing KML files. It is originally developed for displaying the Earth globe, but it also has capability to display other bodies such as Moon and Mars. Google Inc. additionally provides Google Earth API that is enabled to embed the Google Earth display frame in a web browser window. Twitter is a web system providing social networking and microblogging services. Users of Twitter can send and read text messages up to 140 characters called tweets. Twitter also provide API that enable developers to

make standalone client software or new web services. Personal authentication of users is verified by the OAuth technology [4].

Our group previously proposed an online discussion system with a remote sensing image viewer [5]. It was designed to provide a discussion environment with an image viewer to planetary scientists. That system, however, can't hold multiple threads of discussions. Also, it can't show contexts of discussions on the map. We intend to solve these problems. There is another system constructed by a combination with GIS and twitter that visualize information on a map where tweets are sent from [6]. This system only keeps a location of each tweet, and can't hold threads of discussions. Therefore it doesn't fulfill our requirements.

**System Design:** Figure 1 shows an overview of this system. As mentioned previously, we use two mashup components; Google Earth server provides map information, and Twitter server keeps comments of discussion. They are controlled by JavaScript program and Ruby CGI program, which we developed, respectively. The Ruby CGI program also controls A MySQL DB system.

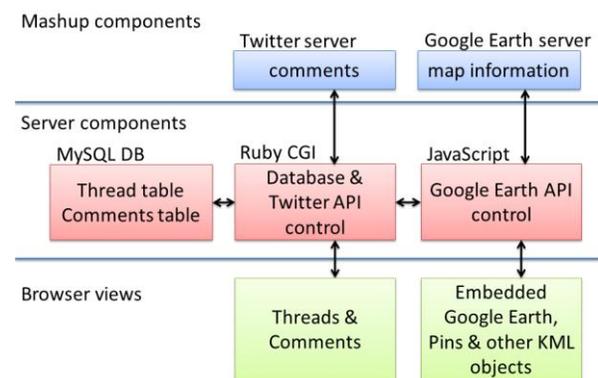


Fig.1 system overview

In this system, a thread of comments is composed of a certain number of Twitter comments. The thread also has a location property that indicates a location of interest of users and at which the discussion is proceeding. As all comments are kept in the Twitter server, our system only manages only IDs of Twitter comments that are given by the Twitter server, location properties of the threads and list of comment IDs. The MySQL database system has two main tables;

Thread table and Comments table. The former keeps a list of all threads with location information, and the latter holds lists of comment IDs for each threads.

A location information of a thread is delivered to the Google Earth API that controlled by JavaScript to display a position of the thread, or delivered from the Google Earth API when a new thread is created. A Ruby CGI program invoked database and twitter API[7]. The Twitter server records comments inputted at a web form. We use a twitter library distributed through the rubygem [8]. This library is a ruby wrapper for the twitter REST and Search APIs. Twitter REST is style of designing loosely coupled applications that rely on named resources rather than messages.

**Implemented Functions:** Users can put a new pin on any points. They can see the thread which ties the location with the pin (Figure 2). Therefore they can find pins displayed visually on interesting regions.

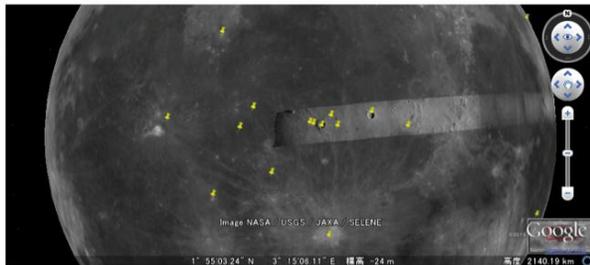


Fig.2 Displayed location of threads as pins on body' surface

Users can write and read comments in the specified thread. The comments are displayed in order of time. The comments have three information; “user name”, “date and hour”, and “tweets”.

This system can also search pins which are corresponding for threads on the map.

**Summary:** In a scientific discussion among researchers, they expect to get a more productive and fruitful discussions. Therefore we propose to share scientific idea on particular location of body on web. We have developed this system with the existing web services, Google Earth and Twitter. By this system, users can find threads at particular location represented as pins easily by viewing body, then they can discuss on the selected location. Therefore this system is simple and convenient.

#### References:

- [1]Google Earth API  
<http://code.google.com/intl/en/apis/earth/>
- [2] Twitter  
<http://dev.twitter.com/start>
- [3] MySQL

<http://www.mysql.com/>

[4] OAuth

<http://oauth.net/>

[5] Kanzawa, M.(2009) Online Geographic Information System Supporting Collaborative Work for Multiple Users, Master thesis in Computer Science and Engineering Graduate School of the University of Aizu.

[6] Geotwitter

<http://ebiquity.umbc.edu/blogger/2007/03/08/twitter-google-maps-mashup/>

[7] Ruby

<http://www.ruby-lang.org/ja/>

[8]Rubygem twitter

<http://twitter.rubyforge.org/>