TwitApp: In-product Micro-Blogging for Design Sharing

Wei Li, Tovi Grossman, Justin Matejka, George Fitzmaurice
Autodesk Research
210 King St. E., Toronto, Ontario, Canada
{firstname.lastname@autodesk.com}

ABSTRACT

We describe TwitApp, an enhanced micro-blogging system integrated within AutoCAD for design sharing. TwitApp integrates rich content and still keeps the sharing transaction cost low. In TwitApp, tweets are organized by their project, and users can follow or unfollow each individual project. We introduce the concept of automatic tweet drafting and other novel features such as enhanced real-time search and integrated live video streaming. The TwitApp system leverages the existing Twitter micro-blogging system. We also contribute a study which provides insights on these concepts and associated designs, and demonstrates potential user excitement of such tools.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Micro-blogging, Design, Twitter, Sharing

INTRODUCTION

Social media and networking systems have been rapidly evolving over the last 10 years. The information streams of web 2.0 feeds provide a popular means for information awareness [6]. In particular, micro-blogging systems, such as Twitter, have become extremely popular for keeping community members aware of each other's activities. For example, Twitter has approximately 200 million users, who send 140 million tweets daily – about one billion a week[12].

Given its prevalence, there has been a recent surge of research in the HCI community studying numerous aspects and potential uses of micro-blogging sites, and in particular, Twitter, ranging from friend network analysis [14], workplace applications [28], to content visualizations [19]. However, one unexplored area which we see as having a rich potential, is the use of micro-blogging systems for sharing design activities within software applications.

The emergence of the World Wide Web has indeed made sharing design activities and workflows quite popular. It is common to see discussion forums with specific threads for users to share their work, with step by step instructions of how they proceeded. Many designers also maintain personal blogs, where they can post information about their current projects on a day to day basis.

While technologies such as blogs and discussion forums are effective ways to share designs and software application activities, they have several limitations. First, these sites are typically external to the actual software application. As such, the content must be manually created which can be laborious. Second, because of the high authoring cost and low feedback ratio, blog users do not update their blogs as frequently as micro-blogging [20]. On discussion forums and tutorial sites, posts are typically only made once a project is completed, preventing the community of users from following along with the author's progress.

Micro-blogging provides a different sharing experience for designers. In one potential usage scenario, Adam, a young architect, shares his personal design portfolios to his followers using micro-blogging. His current design may get comments or critiques while he is still working on it. His colleagues discuss and collaborate with him at work. And he learns new skills by following senior architects' designs. Sometimes, he also receives notification of live software tutorials in real-time.

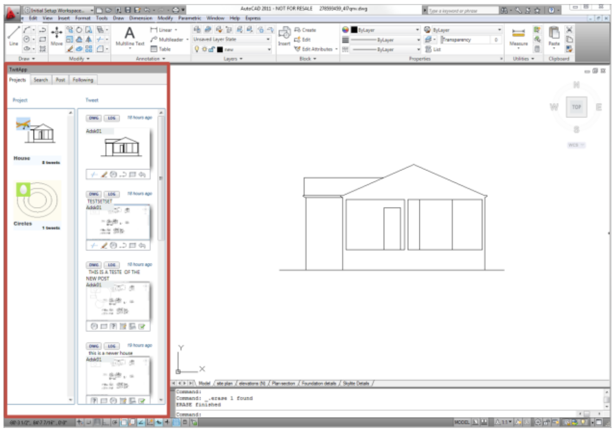


Figure 1. The TwitApp plug-in is displayed as a palette within AutoCAD.

In this paper we introduce TwitApp, an in-product micro-blogging system that allows users to both author and follow information about software design activities (Figure 1). We developed TwitApp within AutoCAD, a popular computer-

[1] http://www.bbc.co.uk/news/technology-12805216
[2] http://en.wikipedia.org/wiki/Twitter

*UIST '11,*October 16-19, 2011, Santa Barbara, CA, USA.

aided design application used by over 10 million users worldwide. Using Twitter's existing back-end architecture

TwitApp allows an author to easily and frequently post messages about their current design work: authors can add useful metadata to their posts such as screenshots, application drawing files, and command logs; and draft posts can be automatically authored based on time or command triggers. TwitApp also provides a mechanism for users to search for and follow specific designers or projects of interest, communicate in real time with the content authors, and even view a live video stream of an author's work.

The benefits of TwitApp are two-fold. First, by placing the micro-blogging client within product, the transaction cost for authoring and sharing content becomes extremely low, and as mentioned, can even be done automatically. Second, by leveraging existing micro-blogging technologies, authors can relay information about their design work on a frequent basis, while followers can keep a close eye on a project of interest, with an opportunity to engage the author if specific workflows of interest are viewed.

After reviewing related work, we provide a thorough description of the TwitApp system and its implementation details. We then present an evaluation, where we invited users to try out the features of the system, to gather usage observations and user feedback. The results of the study showed that TwitApp possesses a number of features that excite potential users, and indicated the general approach of supporting in-product micro-blogging is a promising one. We conclude by discussing potential usage scenarios, limitations of our work, and possible future research topics.

RELATED RESEARCH

Micro-blogging

Micro-blogging is proving extremely useful for the fast exchanges of thoughts, ideas and information sharing [10]. It has been widely used in sharing information and exchanging thoughts [13, 16], tracking conferences and events [9, 24, 25], facilitating live broadcasting [26], training and learning [2, 10], and updating one's status to family and friends [20].

Some specialized social network sites are targeted towards specific user groups, such as professionals in medicine, finance and advertising industries [7, 27]. This indicates that micro-blogging can provide value to a wide spectrum of users and professional domains.

The research thus far on micro-blogging has been focused on four aspects: network analysis [13, 14], content [6, 21, 25], motivations [11, 28, 29], and beneficial applications. Very little research has been published on improving the existing designs of micro-blogging systems.

Design Sharing

The focus of this paper is on design sharing, which can span many different domains, including architecture, landscape and urban design, media and entertainment, digital photography, animation, and many more. Amongst all of these domains, there are some unique aspects of sharing. For example, most designers share very rich visual content with colleagues, customers, friends, or the public, and they use design applications to create their work.

Previous research in design sharing focuses on collaborative design, which supports design teams operating on a project in the same location [4], or in physically independent locations, simultaneously or asynchronously [15, 18]. Some research has been done for sharing informal ideas in the workplace [22, 29]. However, the idea of sharing design using micro-blogging has not been explored.

In-Product Micro-Blogging

Low sharing cost is one major reason why people like micro-blogging. The 140-character limitation on text messages helps reduce the cost of sharing [29]. However, current micro-blogging platforms are not ideal for designers to share their work with others. As an example, Twitter users need to go though several steps to locate and attach an image as a shortened URL link in a tweet. Furthermore, the brief text updates plus URLs are not optimized to provide a visual presentation that a designer may desire.

To minimize the switching cost, we propose an in-product micro-blogging system. The only related example we are aware of, is a command line based Twitter client within Eclipse [23]. While this client is provided in-product, it was not developed to support rich content sharing, and only supports short text messages.

With TwitApp we also explore the use of live video broadcasting to share design workflows in real-time. In order to share videos with Twitter users, a previous approach is to post the video description and a link to Twitter for all followers to see [3]. Our approach is similar, except it updates the author's live streaming status in real time.

In summary, previous research has been focused on the application and analysis of general micro-blogging systems. The problem of redesigning a micro-blogging system for a specific user group has not been explored. In our work, we introduce TwitApp, a new micro-blogging system that has been specifically developed to support in-product design sharing with low transaction cost, and places a particular emphasis on creating a rich visual experience.

DESIGN GOALS AND METHODOLOGY

Design Goals

Based on the requirement of design sharing, we indentified three design goals when developing our new system.

Sharing Rich Design Content. In addition to a short message, the system should support sharing rich design content.

Maintaining a Low Transaction Cost. We should minimize switching cost, so that designers can generate tweets using their design data automatically without interrupting their workflow.

Supporting Real-time Awareness. When a new tweet matches the follower's interest, the system should provide unobtrusive real-time notifications.

Design Methodology and Considerations

To achieve an appropriate design that achieved the above outlined goals, we explored multiple system designs. We explored both stand-alone and in-product clients, and both command-based and graphical interfaces. We also considered tweets containing pure text and tweets embedded with multimedia content. The text-based messages prevented us from easily embedding multimedia data, such as images and videos. Although we could put URLs in messages, it would require a browser to view the content.

We also explored different presentation layouts, from a simple tweet list to an automatically generated magazine layout for mobile devices. Our final design, which will be presented in the next section, evolved through a process of several iterations. For example, one of our original prototypes used a command line interface.

TWITAPP SYSTEM

TwitApp was implemented within AutoCAD, a 2D and 3D design and drafting software program used by over 10 million users. We choose AutoCAD for developing TwitApp because of the large user base, the large number of design domains supported, and the ease of extending the program with customized plug-ins via the ObjectArx programming API. However, we believe it is straightforward to generalize this concept and the prototype to design software in other domains.

Enhancements to Twitter

We chose Twitter because it is probably the most well-known micro-blogging platform currently available on the web. Although TwitApp leverages the existing Twitter platform and back-end API, the functionality of TwitApp goes beyond traditional Twitter clients in the following ways:

Rich Tweet Content: In addition to a short text message, TwitApp also integrates high resolution images, design data, user command history and video streaming within each tweet.

Project Organization: TwitApp extends the traditional way of displaying every tweet you received from friends in the temporal order as a timeline and introduces the concept of a *project*. In TwitApp, tweets are organized by both users *and* projects. Thus, each user can have multiple projects and each one can be individually followed or unfollowed.

Tweet Drafting: TwitApp also introduces the concept of automatically drafting tweets, triggered by user configured application events. This further helps reduce the sharing cost, allowing designers to create new tweets while maintaining focus on their work and workflow.

Enhanced Search: TwitApp also extends the notion of the "real time search" that is largely synonymous with searching tweets [5]. In TwitApp, the real time search is enhanced to not just search across the text messages within the tweet, but also the meta data attached with tweets, such as the users' command steam. This allows users to find relevant projects to follow.

System Overview

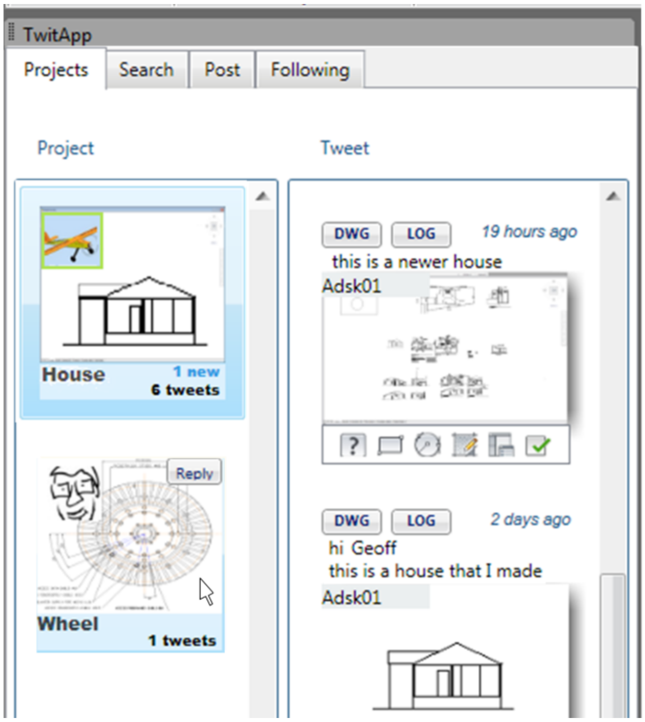
The TwitApp client manifests as a palette in the user interface of AutoCAD (Figure 1). The credentials for the Twitter account are saved in the AutoCAD system preferences for each designer. The user does not need to leave the working environment to send or receive messages, as the functionality is all presented within-product.

The functions of TwitApp are organized by four tabs in the palette: *Projects*, *Search*, *Post*, and *Following* (Figure 2). In the *Projects* tab, design projects which the user is following are displayed on the left side of the window and individual tweet messages for the selected project are shown in the right side of the panel. The *Search* tab allows users to find designers and projects to follow. The *Post* tab is used to send tweets, and the *Following* tab is used to select projects and users to follow or un-follow.

Viewing Followed Projects

Following a designer on TwitApp means that a user is subscribing to some or all of his/her Project Tweets as a *follower* and those tweets will appear in the user's stream of updates when he/she logs in. When the user initially follows the designer, he/she may select from a list of known projects to follow. When one unfollows someone, all of that designer's projects are automatically unfollowed.

One contribution of TwitApp is the ability to select a specific project to follow. We define *a project* as a sequence of tweets which provide updates on the same design file. This concept allows us to organize tweets into a set of projects.



this is a newer house
Adsk01

hi Geoff
this is a house that I made
Adsk01

Figure 2. The TwitApp palette is organized into four tabs: Projects, Search, Post, and Following.

Projects

The *Project* tab in TwitApp shows a list of all projects the user is currently following on the left (Figure 2). Users can scroll through the list and select a project to view its details. The projects are displayed as thumbnail images that are associated with that project (Figure 3 Left). A thumbnail image in the upper left corner represents the author of that project. Below the project image, we display the name of the project, as well as the number of new tweets and the total number of tweets. If the author is currently broadcasting their work, a "live" icon will be displayed. When a user hovers over a project thumbnail, a Reply button is made visible for the user to quickly send a tweet to the project.

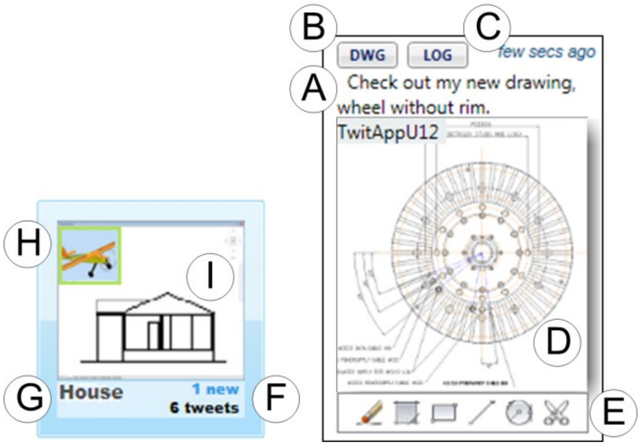
Tweets

When a project is selected, each of its associated tweets are displayed in a scrollable list, sorted by time, on the right side of the panel. On the right side, Figure 3 illustrates how individual tweets are displayed in TwitApp.

The top portion of the tweet contains buttons to access attached data and a message showing when the tweet was sent (see Figure 3 B and C). If the current design file is attached to the tweet a *DWG* button is displayed. If a log of the user's commands is attached a *LOG* button is displayed. Clicking the *DWG* button loads that file into the user's application while clicking the *LOG* button displays a text file, showing a log of commands used since the previous tweet.

Below this, the tweet contains the content of the user's textual message. Next, a large thumbnail is displayed in the center of the tweet. Clicking on it opens a high resolution screen capture image of the drawing in the user's default image viewer. The user name is overlaid on the top-left of the thumbnail image.

A row of command icons, which represent commands used by the author, but which have not been used by the current user, are displayed under the thumbnail (Figure 3E). This serves as a potential learning aid.



Check out my new drawing, wheel without rim.

Figure 3. A project (left) and a rich content tweet in TwitApp A) Text message, B) Drawing data file, C) User command history, D) Thumbnail screen image, E) Command icons, F) New tweet and total tweet count, G) Project name, H) User screen image, and I) Screen image

Managing Followed Projects

Within the *Following* tab, users can see the collection of users (i.e. designers) they are currently following in a scroll list ordered from most-to-least recently added (Figure 4 A). Each designer has their personal thumbnail, name, and two control buttons. For each designer being followed, the user can click a check mark to temporarily un-follow that user. By clicking the *Follow Projects* button, a user can view all the projects created by that designer and add or remove projects to follow (Figure 4 B). Modifications to the projects being followed will update which projects are displayed in the *Projects* tab.

Searching

Users can find new users and projects to follow by using the *Search* tab. There are three options in the search window: *View All Latest Tweets*, *Keyword Search*, and *User Search*. Regardless of which search options are used, the search results are updated in near real-time. Whenever a new hit that satisfies the current search criteria is received, the search results are updated. The number of new hits since the search was performed is shown in brackets in the *Search* tab, for example, "Search (1)". See Figure 5.

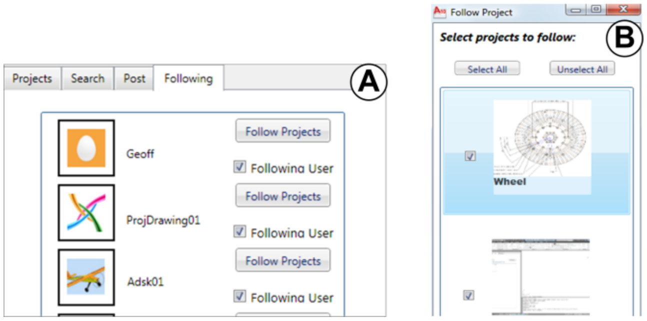
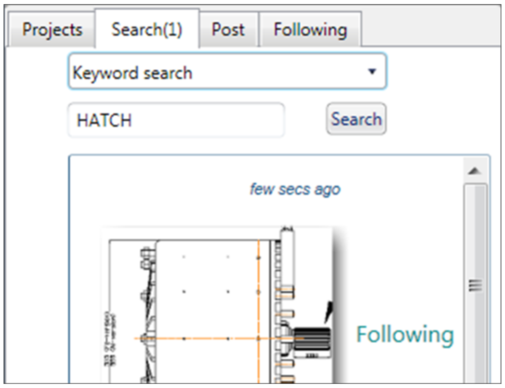


Figure 4. A) Following Tab, B) Project dialog window for a given designer.

The default *View All Latest Tweets* option, shows all of the latest tweets produced by all TwitApp users (in the results list, sorted by time). This allows users to browse the TwitApp community for new projects or users. The *Keyword search* option allows users to specify a search phrase and presents matches to those tweets whose text message or command log contains the search phrase, ordered by date. This feature, in combination with the real time search, is useful for when a user is interested in a specific command. For example, if a user searches for the *HATCH* command in TwitApp, when a designer later uses the *HATCH* command and subsequently posts a tweet, the *Search* tab title will change and notify the user that there is a new tweet satisfying the search query.

The *User search* option returns user names that match a given search phrase. The result list shows matched user profile images and all the projects created by that user. From the search results, users can follow a user, or a specific project of that user. Thus a user can explore and find new tweets, projects, and users.



Keyword search

HATCH

Figure 5.Real time search find a new tweet, which used command HATCH.

Tweet Authoring

Like regular Twitter clients, TwitApp users can create text messages and post new tweets. However, as a benefit from the in-product design, TwitApp also allows users to attach multiple data formats with their tweets. To create a new tweet, the user clicks on the *Post* tab and opens the Post New Tweet dialog (Figure 7) by clicking the *New* button shown in Figure 6.

The Post New Tweet dialog allows users to enter their message using a text field at the top of the dialog. A screen capture image is automatically displayed below. Clicking on the drop-down menu presents four options to alter the attached image. They can also specify if they wish to attach the current *DWG* file and command LOG file.

Users can also quickly author tweets by replying to existing projects in the *Project* tab. When the cursor hovers over a project, a reply button is displayed. Clicking that button will also display the Post New Tweet dialogue.

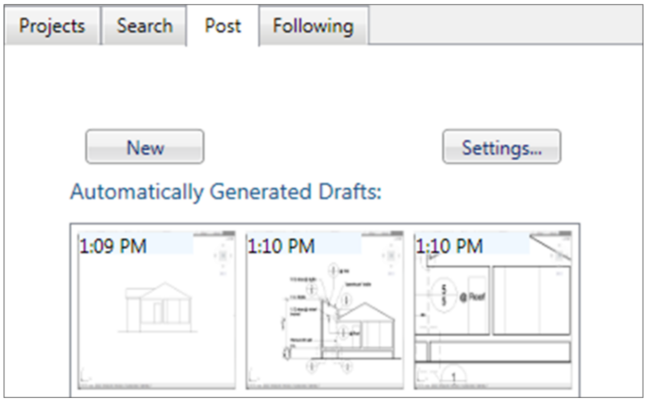
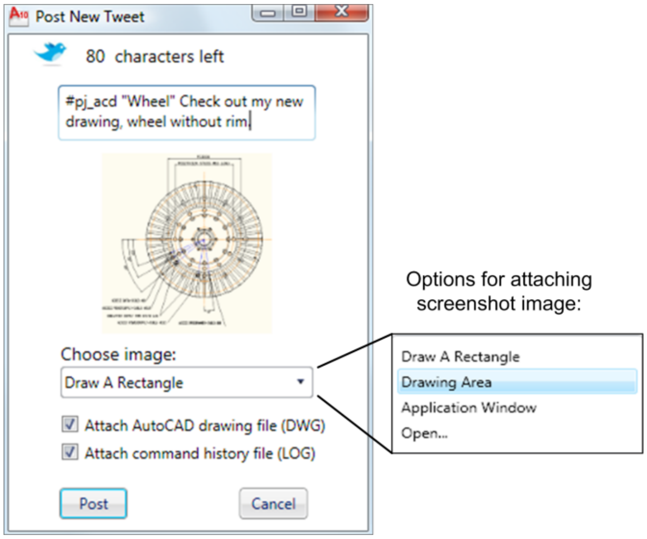


Figure 6. The Post Tab.

Tweet Drafts

A main contribution of our system is the ability to automatically create tweet drafts. This allows the system maintain a low transaction cost. Designers are able to maintain updates without explicitly breaking from their primary task. A tweet draft is a tweet which has editable attachments and has not been posted yet. Tweet drafts that are created are added to the "Automatically Generated Drafts" section of the *Post* tab (Figure 6). Hovering over a tweet draft, the user has the option to click on the *Delete* or *Preview* button. Clicking the *Delete* button will remove the draft from the collection. Clicking on the *Preview* button will bring up and populate the Post New Tweet dialog (Figure 7) with the project information, draft image and a timestamp message of when the draft was originally produced. The system also defaults to attaching a snapshot of the DWG file and command log reflecting the moment in time when the draft was originally captured. The user can adjust any of the information before posting the tweet.



#pj_acd "Wheel" Check out my new drawing, wheel without rim

Choose image: Draw A Rectangle

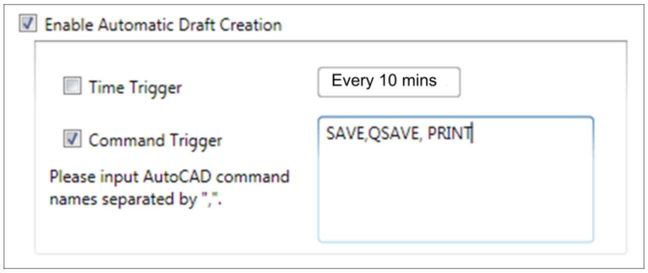
☑ Attach AutoCAD drawing file (DWG)

☑ Attach command history file (LOG)

Options for attaching screenshot image:

Figure 7. Post new tweet window

TwitApp provides two mechanisms for triggering the automatic creation of tweet drafts: time triggers and command triggers. These triggers are configured in a settings dialog (Figure 8).



☑ Enable Automatic Draft Creation

□ Time Trigger Every 10 mins

☑ Command Trigger SAVE,QSAVE, PRINT

Please input AutoCAD command names separated by ",".

Figure 8.The setting dialog provides UI controls to setup the time trigger and the command triggers.

Time Triggers

The time trigger is used to create tweets in a periodic time interval. Users can select from 1 minute, 5 minutes, and up to an hour. Figure 9 shows three tweet drafts created by the time trigger with the time interval set to one minute.

Command Triggers

In the situation when the user does not need to change the current drawing or has simply left his desk, the time trigger will repeatedly create the exact same tweet drafts. To avoid a large number of duplicate during this idle time, and to allow for more flexibility, TwitApp provides a command trigger. This allows users to specify a set of AutoCAD commands to control the automatic creation of drafts. To specify a command trigger, the user types the command name in a text box within the settings dialog (Figure 8). Multiple commands can be separated by comas. Drafts will be triggered if *any* of the commands are used.

We noticed that several commands are good indicators of when the project has been significantly updated. For example, AutoCAD users often use the SAVE command after some meaningful design or drafting work has been done. However, some users may frequently save their projects, even nothing been changed. In this case, the combination of a time trigger and command trigger can be useful. In these configurations, drafts will only be created when both the time *and* command triggers have been satisfied.

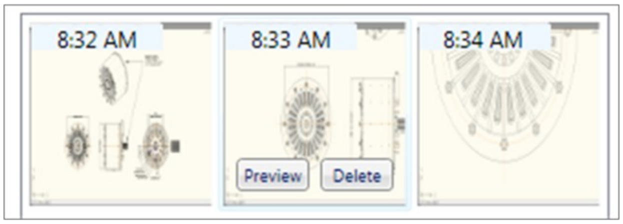


Figure 9.Automatically generated tweet drafts.

Live Video Streaming and Following

While the micro-blogging that TwitApp provides give users the chance to follow snapshots of a designer's work in near real-time, they cannot see firsthand how authors carry out their work. To support true design workflow sharing, we implemented a live broadcast feature within TwitApp.

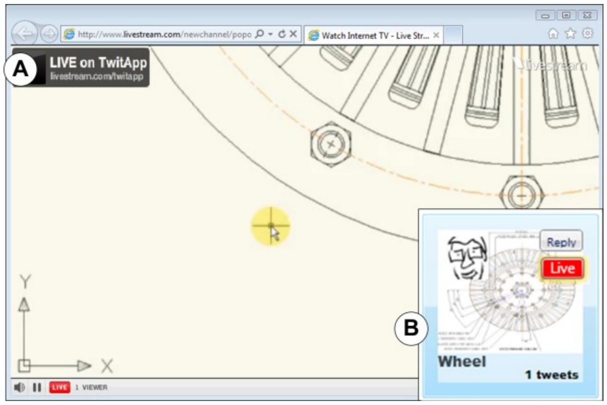


Figure 10.A) Live video player window lauched by TwitApp, B) A live broadcasting button (red) displayed over a project control.

As a proof of concept, we used the LiveStream[3] web service to enable video streaming (Figure 10 A). Additionally, TwitApp recognizes when the LiveStream service has started and displays a *Live* broadcasting button on the associated project in the *Project* tab (Figure 10 B). Followers of that project can click on it to open a live video window. While watching the live stream, the user can post comments and questions using the reply button.

IMPLEMENTATION

Our system consists of the TwitApp client and several backend services (Figure 11). The client communicates with the servers using web-based APIs.

TwitApp leverages the existing micro-blogging system and the user community. The content delivered with each tweet is stored and managed using multiple web services. Tweets sent from TwitApp are compatible with the existing system, and, as such, clients can view the message and images on the Twitter website or mobile devices. The cost of building and maintaining this system is extremely low. As such, this architecture serves as a good approach for experimenting with micro-blogging research systems and techniques.

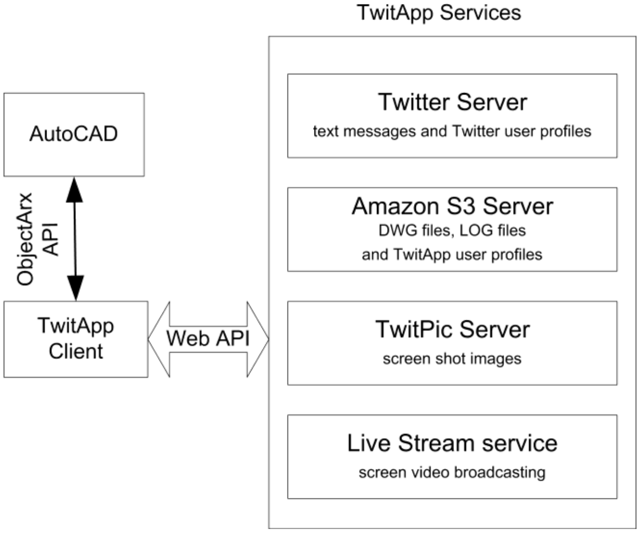


Figure 11. TwitApp architecture.

Project and Verification

Twitter hashtags are a simple way of grouping messages with a "#" sign followed by a name or a special code which will form a unique tag for a specific purpose. TwitApp uses the hashtag "#pj_acd" to label every tweet sent from a TwitApp client. This label is followed by the name of the specific project, such as "House". Once the TwitApp client receives a tweet with the "#pj_acd" in its text message, TwitApp starts to verify this tweet with data distributed in other websites. After confirmation that the tweet is in the valid TwitApp format, the data will be downloaded from multiple services and displayed in the client.

It may be desirable to allow users to check regular tweets in the design software. In our current design, TwitApp does not display tweets without the "#pj_acd" tag.

TwitApp Web Service

We implemented the TwitApp web services using: the web API provided by Twitter, the Amazon Simple Storage Service, TwitPic and LiveStream. The integrated data is decomposed for each tweet and distributed on the appropriate servers in the required format.

The Twitter web service provides user account and credential management, text message storage, and user profile

[3] http://www.livestream.com/

storage (including user's ID, screen name, following and follower lists). Once the user successfully logs into Twitter, a unique key is generated for each tweet. The Amazon S3 server was created to organize and store the TwitApp customized data, such as drawing files and command log files. Also, TwitApp maintains a list of TwitApp users who have communicated with this server. Finally, TwitPic and Live Stream are used to store and access images and videos.

TwitApp generates a unique key number for each tweet and uses this key to name files stored in S3, TwitPic and Live Stream servers. Using the key and TwitPic API, the client can easily map the user data with the user account stored on the Twitter server.

TwitApp Client

The TwitApp client integrates within AutoCAD as a plug-in. The user interface of the TwitApp client was developed with Visual Studio C#, Windows Presentation Foundation and ObjectArx (an API to access AutoCAD's functions). For users who want to broadcast live screen video, a video broadcast client is also installed.

Twitter Rate Limiting

To protect its service from overloading, Twitter only allows clients to make a limited number of calls in a given hour, in our case 350 per hour[4]. Although it is important to reduce the frequency of calls made, TwitApp clients must remain responsive and useful despite the rate limitation.

We implemented a Twitter request planning component which checks the current rate limit of a user account and determines when to send requests directly to Twitter, and when to send requests to a queue maintained by our client. If the queue is not empty and the rate limit has not been reached, a request will be made, and the results will be passed to other components of the client asynchronously.

A caching system was also implemented for lowering the operating cost of the system. Before downloading an image or a data file from a server, the client checks its cache first. This way, we improve the download speed and overall system performance. During our user study, the user's experience was not affected by the rate limiting.

EVALUATION

To evaluate the concepts and features of TwitApp, we conducted a qualitative user study. The main goal of the study was to generate initial observations and to assess which features would be most welcomed, and identify any features that may need to be revised or rethought.

Participants

We recruited eight external users, aged 26-40, from an online posting. Because we wanted to get observations from potential end users, participants had to be familiar with both AutoCAD and Twitter. Three participants use AutoCAD daily, three of them used AutoCAD at least once a month, and the rest used AutoCAD occasionally. All participants were familiar with Twitter, and five of them had active Twitter accounts. Participants were paired to bring a more personal feeling to the evaluation, and so users could experience the real-time sharing features.

Apparatus

The study was run on two Xeon 3.0Ghz Windows 7 workstations with a broadband internet connection. Participants were sitting in the same room but on opposite sides of a table so were not able to watch each other's screen.

Procedure

The whole evaluation lasted approximate 1 hour, and was divided into 4 parts. Participants were first given a 5 minute introduction to the system. This was followed with a 35 minute full walkthrough of the system. During this walkthrough, the participants were shown every feature of TwitApp, and asked to accomplish 49 "atomic" tasks, such as "find the project called 'wheel design'" and "create a command trigger using the HATCH command".

After the walkthrough, participants completed two tasks together. In the first task, one participant acted as the author, and was asked to modify an AutoCAD drawing of a house, and post tweets using the tools that they were introduced to. The other user "followed" the author, and viewed and replied to the tweets which were received. After 5 minutes, the participants switched roles.

In the second task, one user again acted as an author and worked on the design of a house. This time, the author enabled the live broadcast. The other user watched the broadcast and posted comments to the author using TwitApp. The participants switched roles after 5 minutes.

After completing all 4 parts, users filled out a questionnaire about the individual features in the system. For each feature the questionnaire asked them to rate, on a 7-point Likert scale, the statements "I found it easy to use" and "I think it would be useful", on a scale from 1 (poor) to 7 (excellent).

Results

The observations from the study were very encouraging. Users were impressed with the features of TwitApp, and were enthusiastic about the idea of sending out tweets from within the application.

The system seemed easy to use for all users. Users completed every "atomic task" independently after the brief introduction, and were able to complete the final 2 tasks in both the "author role" and "follower role".

The results from the questionnaire were generally positive. Average response for the overall system were 6.4 for *easy to use* (all scores out of 7) and 6 for *would be useful*.

The features users found most useful were:

- Ability to quickly send out tweet inside AutoCAD (6.8)
- Video broadcasting (6.6)
- Including screen shot images in a tweet (6.6)
- Organizing and viewing the tweets by projects (6.3)
- Send/Receive/View AutoCAD drawings in tweets (6.3)

[4] http://dev.twitter.com/pages/rate-limiting

The features users found easiest to use were:

- Real time search of commands and keywords (6.9)
- Including screen shot images in a tweet (6.9)
- Following and unfollowing projects (6.8)
- Browsing and following the latest tweets (6.8)

Only 3 features got less than 5 for *would be useful*:

- Attaching a command log in a tweet (4.1)
- Viewing the command log attached to a tweet (4.1)
- Viewing icons of commands used by others (4.3)

The exact same 3 features were the only features that got less than 5 for *easy to use.* Their scores are 4.5, 4.8 and 4.9. This data makes it clear that features related to logging and displaying command histories were not as well-accepted as the other features of the system.

Many positive comments were made by the participants in the questionnaire. Five of the participants provided comments indicating that they would recommend TwitApp as a learning tool. One user, who is an architect and uses AutoCAD every day, wrote "I think this is a terrific idea, especially for AutoCad training, [other] technical training or project sharing, also for video conferencing." Another user wrote "In a learning environment, TwitApp could be [a] useful tool to accelerate learning the software." These types of comments indicate that TwitApp could provide a new medium to support community-based learning.

Four of the participants indicated they would like to use the system for the purpose of sharing. An AutoCAD instructor commented "it is very easy to share my project in real time." A senior AutoCAD user wrote "it is excellent for teaching and sharing modifications of DWG", and another user commented "it is a really useful application for users to share their design".

Participants mentioned several potential limitations about TwitApp. Almost every user had concerns about the privacy and confidentiality. For example, a user commented that it could be useful in many situations, but sharing a drawing file with outside people is against his company policy. Another user wrote "this application should be used in professional social networks, not in general social networks." These comments are important to consider, and a commercial deployment would need to be careful about allowing users to send files and commands logs to the public.

Three users also wrote comments about using TwitApp inside the office environment. Another user, who works in an architecture firm, wrote "TwitApp could be an alternative tool for the (internal) email system, I can see it will reduce emails". He also commented "TwitApp is also a good tool for project managers for revising, coordinating, [and] integrating documents. I see this tool as a collaboration tool between members of the same team and even companies."

Overall, we are very encouraged by the observations from our study, as well as the feedback questionnaire results.

MOTIVATING SCENARIOS FOR TWITAPP

We see a number of potential scenarios of usage where micro-blogging could be a welcomed feature within design software applications. Some of these scenarios are directly related to comments received from our evaluation, while others are based on our own intuitions and experiences.

Design Sharing

Most notably, TwitApp reduces the transaction cost to support design sharing within user communities. Users who want to share their work, or show off their skills, either to the broad user community, or to specific followers, can use TwitApp to easily share their designs.

Micro-Critique

By supporting micro blogging, TwitApp also supports micro-critiquing. Designers who use TwitApp can receive close to real-time feedback on their designs. Again, this feedback can be received from the public user community, or from work colleagues or managers. Micro-critiquing will allow designers to consider feedback as they work, rather than waiting for feedback after a draft is completed.

Online Tutoring

TwitApp gives instructors a new medium to provide live, interactive, tutorials. For example, instead of posting a video tutorial to a website, instructors can advertise a tutorial session that they will be carrying out at a specific time, so members of the user community can follow the tutorial in real-time, with an opportunity to interact with the instructor.

Learning

TwitApp can also support learning, by allowing users to follow projects that demonstrate relevant workflows of interest. Users can look at any project of interest, and view its history of posts, to get a sense of the workflows that were used. If further information is needed, users could send messages to the originating author.

Workplace Co-operation

TwipApp can be used as a mechanism for members of a workplace to maintain an awareness of the status of each other's work. It allows users to receive notifications when projects are updated, or to look at a project of interest that a colleague is working on, to see its current status. This may also help colleagues share design ideas among one another.

Personal Journal

TwitApp can also be used for completely personal reasons, allowing a user to keep a journal of their own workflow activates. This may be useful for users to review or remember what they have done, or to simply have a record of it – with an indifference to community members viewing it. This type of behavior is common among bloggers, who wish to "document their life". TwitApp allows users to document their designs.

DISCUSSIONS

Application Dependence

Most features in TwitApp leverage contextual information in the application. For example, tweets are organized using the current design file, tweet drafts are generated based on application commands, and command streams are used to search for relevant tweets. Although TwitApp was developed for AutoCAD, our vision is that its main features could be integrated into any application used for design. From an implementation and architecture standpoint, there should be no real barriers in doing so in other applications.

Privacy

One of the main concerns expressed by our participants was the issue of privacy. Privacy is indeed a concern for social network sites in general [1, 7, 8]. Although Twitter offers basic privacy settings, which allow users to control who can read their updates, users still need to monitor their followers' list to block suspicious following. Of course, when messages involve confidential architecture drawing files, privacy is a much more serious concern.

A typical practice adopted by many social network sites is to design an access control policy. For example, to control who sees what content, users can limit the visibility of their sensitive data files to their immediate connections or they can choose to allow any followers to see their content. Because TwitApp was implemented to store and manage different types of data distributed over separate web servers, TwitApp's architecture can support numerous policies for access control. In this way, TwitApp could support users who want to use the system to exchange design data with their close colleagues, as well as support users reaching out to the entire community.

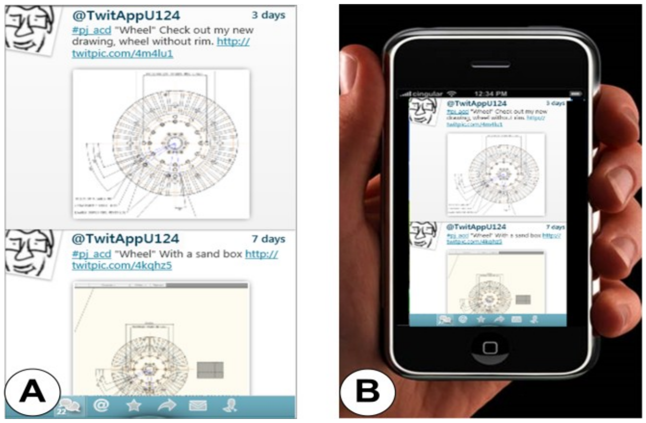
TwitApp in The Workplace

We can see many useful applications if TwitApp can be served as a coordination and communication tool inside design companies. Combined with version control applications, TwitApp could provide a very cost efficient cooperative solution.

For privacy issues at work, TwitApp could be setup behind the enterprise firewall. Previous research shows that privacy is less of a concern within the enterprise [7]. Because of this, we see no need for more extensive privacy controls for intranet environments compared with the internet environment. Furthermore, open sharing across the enterprise specifically could enable connecting with colleagues, showcasing one's designs, building up one's profile and personality, developing one's career and facilitate campaigning.

From Twitter to TwitApp

Our implementation, which uses the existing twitter framework, allowed us to quickly and inexpensively develop a new and robust micro blogging system. We believe this implementation, in itself, is a contribution and hope other researchers will be able to learn from our experiences. However, another reason we used the existing Twitter backend, is that it allows users to follow designers without running TwitApp, but instead from a traditional Twitter client. Tweets sent from TwitApp will have additional hash tags, but otherwise, view quite well on standard Twitter clients, and could also be viewed on mobile devices, potentially by a design manger during a commute (Figure 12).



@TwitAppU124 3 days
#pj_acd "Wheel" Check out my new drawing, wheel without rim. http://twitpic.com/4m4lu1

@TwitAppU124 7 days
#pj_acd "Wheel" With a sand box http://twitpic.com/4kqhz5

Figure 12. a) Layout of actual TwipApp tweets displayed by an existing Twitter client. b) Mockup of the same client within a mobile device.

Visualized Command History

Attaching and viewing command data was the least popular feature in our evaluation. A common concern was one of privacy. For instance, an employee may not want to update their managers with time stamped command streams. One option would be to investigate alternative ways to visualize command usage, or alternatively, this feature could be removed. However, even if command logs were not visible to users, they could still be used to index and search projects.

FUTURE WORK

We have implemented TwitApp and conducted a qualitative user study. We believe this research opens up new future topics of research, related to in-application micro-blogging, and using micro-blogging for sharing designs.

Our evaluation did show that TwitApp, and in-app micro-blogging in general, is a worthwhile concept to pursue. However this study did not formally validate our design. One area of future work we will be hoping to do is to carry out a more comprehensive field study to evaluate our current design and implementation in a larger deployment. This would allow us to analyze adoption and usage metrics in real scenarios, and potentially compare TwitApp with both discussion forums and personal blogs.

In addition to sharing design data, future work could explore real-time sharing of a designer's workflow. This would be similar to Li [17], who explored the idea of sharing users' browsing history using social networks. This could allow designers to repeat someone else's workflow.

In the mobile environment, micro-blogging is a mainstream communication approach. The integration of a rich content micro-blogging client and mobile data also creates many open questions for both HCI research and design.

In our user study, participants gave positive feedback to the automatic tweet drafting feature. We would like to advance upon our command and time triggers, and further consider more advanced methods for triggering automatic drafts. For example, Grabler [12] proposed to use the application state to help aggregate the user's input events into higher level event clusters. By exploring the application's contextual information, the modification history, and the user's action history, a more optimized and intelligent approach of automatically generating tweets could be possible. Furthermore, we believe it may be possible to define content-based triggers for users to register portions of the application data that they want to be notified by a tweet if it is changed.

CONCLUSION

In this paper, we introduced a system design and architecture of TwitApp, an enhanced Twitter system integrated within AutoCAD for design sharing. The system introduces numerous novel features that could be utilized by practitioners. We also contribute a study which provides insights on these concepts and associated designs, and demonstrates potential user-excitement of such tools. Initial user reaction is very positive and supports our belief in providing within-application, data rich, micro-blogging capabilities for flexible and enhanced design sharing.

REFERENCES

1. Acquisti, A. and R. Gross (2006). Imagined communities: awareness, information sharing, and privacy on the Facebook. *6th Workshop on Privacy Enhancing Technologies*. 36-58.
2. Borau, K., C. Ullrich, et al. (2009). Microblogging for language learning: using Twitter to train communicative and cultural competence. *8th International Conference on Advances in Web Based Learning*. 78-87.
3. Bornoe, N. and L. Barkhuus (2010). Video microblogging: your 12 seconds of fame. *ACM CHI-10*. 3325-3330.
4. Buxton, W., G. Fitzmaurice, et al. (2000). "Large displays in automotive design." *IEEE Computer Graphics and Applications,* 20(4): 68-75.
5. Chen, C., F. Li, et al. (2011). TI: An efficient indexing mechanism for real-time search on tweets. *SIGMOD'11*. 648-660.
6. Chen, J., R. Nairn, et al. (2010). Short and tweet: experiments on recommending content from information streams. *ACM CHI-10*. 1185-1194.
7. DiMicco, J., D. R. Millen, et al. (2008). Motivations for social networking at work. *ACM CSCW-08*. 711-720.
8. Dwyer, C. and S. R. Hiltz (2007). Trust and privacy concern within social networking sites : a comparison of Facebook and MySpace. *AMCIS 2007.*
9. Ebner, M., H. Mühlburger, et al. (2010). Getting cranular on Twitter : tweets from a conference and their limited usefulness for non-participants. *WCC 2010*, 102-113.
10. Ebner, M. and M. Schiefner (2008). Microblogging-more than fun. *IADIS Mobile Learning Conference 2008.* 155-159.
11. Ehrlich, K. and N. S. Shami (2010). Microblogging inside and outside the workplace. *ICWSM-10.*
12. Grabler, F., M. Agrawala, et al. (2009). Generating photo manipulation tutorials by demonstration. *ACM SIGGRAPH 2009.* 1-9.
13. Java, A., X. Song, et al. (2007). Why we twitter: understanding microblogging usage and communities. *9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. 56-65.
14. Kivran-Swaine, F., P. Govindan, et al. (2011). The impact of network structure on breaking ties in online social networks: unfollowing on Twitter. *ACM CHI-11*. 1101-1104.
15. Kvan, T. (2000). "Collaborative design: what is it?" *Automation in Construction* 9: 409-415.
16. Kwak, H., C. Lee, et al. (2010). What is Twitter, a social network or a news media? *WWW-10*. 591-600.
17. Li, I., J. Nichols, et al. (2010). Here's what I did: Sharing and reusing web activity with ActionShot. *ACM CHI-10*. 723-732.
18. Li, W. D., W. F. Lu, et al. (2005). "Collaborative computer-aided design—research and development status." *Computer-Aided Design* 37(9): 931-940.
19. Marcus, A., M. S. Bernstein, et al. (2011). TwitInfo: aggregating and visualizing microblogs for event exploration. *ACM CHI-11*. 227-236.
20. McFedries, P. (2007) All a-Twitter. *IEEE Spectrum*
21. Naaman, M., J. Boase, et al. (2010). *Is it really* about me?: message content in social awareness streams. *ACM CSCW-10*. 189-192.
22. Pederson, E., K. McCall, et al. *(1993). T*ivoli: An electronic whiteboard for informal workgroup meetings. *ACM CHI-93.* 391-398.
23. Reinhardt, W. (2009). Communication is the key - support durable knowledge sharing in software engineering by microblogging. *Software Engineering Workshops 2009*
24. Reinhardt, W., M. Ebner, et al. (2009). How people are using Twitter during conferences. *5th EduMedia Conference*.
25. Shamma, D. A. and E. F. Churchill (2010). Tweetgeist: can the Twitter timeline reveal the structure of broadcast events? *ACM CSCW 2010*.
26. Shamma, D. A., L. Kennedy, et al. (2009). Tweet the debates: understanding community annotation of uncollected sources. 1st *SIGMM Workshop on Social Media*. 3-10.
27. VASCELLARO, J. E. (2007) Social networking goes professional *Wall Street Journal*
28. Zhang, J., Y. Qu, et al. (2010). A case study of micro-blogging in the enterprise: use, value, and related issues. *ACM CHI-10*. 123-132.
29. Zhao, D. and M. B. Rosson (2009). How and why people Twitter: the role that micro-blogging plays in informal communication at work. *ACM Group-09*. 243-252.