IP-QAT: In-Product Questions, Answers & Tips

Justin Matejka, Tovi Grossman, and *George Fitzmaurice*
Autodesk Research
firstname.lastname@Autodesk.com

ABSTRACT

We present IP-QAT, a new community-based question and answer system for software users. Unlike most community forums, IP-QAT is integrated into the actual software application, allowing users to easily post questions, answers and tips without having to leave the application. Our in-product implementation is context-aware and shows relevant posts based on a user's recent activity. It is also designed with minimal transaction costs to encourage users to easily post, include annotated images and file attachments, as well as tag their posts with relevant UI components. We describe a robust cloud-based system implementation, which allowed us to release IP-QAT to 37 users for a 2 week field study. Our study showed that IP-QAT increased user contributions, and subjectively, users found our system more useful and easier to use, in comparison to the existing commercial discussion board.

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

General terms: Design, Human Factors

Keywords: Q&A, Community, Help, In-Product Learning

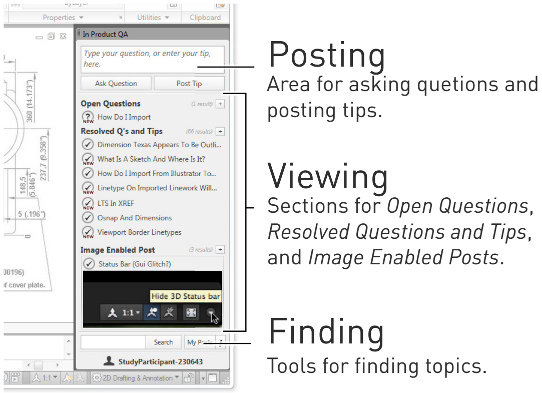
INTRODUCTION

Despite years of HCI research, learnability continues to be a challenge for software applications [9]. This is especially true for complex and feature rich programs which have been shown to continuously grow in their number of features with each release [15]. However, new trends and technologies are available for providing assistance to learning, which were not prevalent in the earlier days of HCI research.

In particular, the world-wide-web has the power to create online communities, allowing users to share their knowledge with one another. For popular software applications, one can find dozens of discussion boards, where community members can discuss topics about the software, including learning challenges. However, we would argue that such forums are not optimized for allowing users to get help with specific learnability challenges, and we highlight two particular problems and associated opportunities.

First, discussion forums are accessed and presented outside of the context of the software application, which can impair a user's awareness and access of relevant material [18]. Furthermore, once out of the application, it can be difficult and cumbersome for question askers to express their questions, using proper terminology and necessary images or attachments. Some online systems, such as Intuit's TurboTax, have community Q&A services integrated into their websites [20]. We believe there is an opportunity to build upon such work to also improve community help for desktop software applications.

Second, threads in traditional discussion forums often turn into true "discussions" making it difficult for a user to find a concise answer for a specific problem or question. Outside of the software domain, many *question and answer* (*Q&A*) sites, such as Yahoo! Answers [2] and Stack Overflow [16], have become popular, and have shown to be a fast and effective way to learn from the community. Users are encouraged to ask short, concise questions, answerers are motivate to provide high quality responses, and the presentation and layout of the content makes it easy for subsequent users to see questions and their associated answers. We believe that there is an opportunity for developing integrated Q&A systems to target the learning of complicated software applications.



Posting
Area for asking quetions and posting tips.

Viewing
Sections for *Open Questions*, *Resolved Questions and Tips*, and *Image Enabled Posts*.

Finding
Tools for finding topics.

Figure 1. IP-QAT system interface overview.

In this paper, we investigate both of these opportunities, by presenting IP-QAT: "In-Product Questions, Answers & Tips", a community-based question and answer system that is provided as a palette within the target application (Figure 1). IP-QAT was developed for AutoCAD, a popular but difficult to learn application for 2D and 3D architectural design. IP-QAT allows users to: post questions, answers, and tips; include annotated images and attachments with their posts; and tag their posts with relevant UI components. All interactions are accomplished with a low transaction cost compared to traditional out-of-product

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

community forums. Particular effort was made to provide an image-rich experience, to facilitate shared understanding between users referring to spatial data within complex software [5]. IP-QAT also supports opportunistic learning, by displaying answered questions and tips relevant to a user's current context of usage.

Our main contribution is a fully-functional implementation of a community Q&A system integrated into a graphical desktop application. We also contribute an investigation and discussion of the challenges and opportunities related to the development of such a system. Finally, we contribute a 2-week user study, comparing our system to an online software discussion board. Our study shows that IP-QAT increased user contributions, and subjectively, users found our system more useful, easier to use, in comparison to the existing commercial discussion board, and provided numerous enthusiastic and encouraging comments which we summarize and discuss.

RELATED WORK

Discussion Forums and Q&A Sites

Usenet newsgroups [7], pre-dating even the internet, and more recently web-based discussion forums, have been a popular method for people to communicate online. Singh et al. explored discussion forums of 8 open source software applications [21,22] and by qualitatively analyzing the threads was able to identify a number of problematic interactions encountered during the help-seeking process.

As opposed to general purpose discussion forums, which as the name implies are generally designed to promote discussion, Q&A sites have distinctive characteristics designed to promote the specific tasks of asking and answering questions. An early example of a Q&A based knowledge system is Ackerman's Answer Garden [1] which maintained an organically growing database of questions and expert answers for a single domain.

A sizable body of work focuses on the analysis of general purpose Q&A sites such as Yahoo Answers [2,3,6,11], Windows Live QnA [24], and a popular Korean Q&A site, Naver [12,19]. Mamykina et al's detailed analysis of Stack Overflow [16] looks to understand the factors behind the site's success and attributes it, in part, to the daily involvement of the design team with the community of users it serves. The Q&A site Aardvark [14] differs from most others in that it routes questions directly to individual users who are known to be online to elicit faster responses.

These explorations into Q&A sites provide useful guidance towards designing our system, however, none of the existing systems examine the implications of integrating the Q&A system directly into the target application.

TurboTax Live Community

Intuit's TurboTax[1] has an integrated Q&A service alongside the main window [20]. As a highly structured, form-filling application, the context of the user's activity can be determined by which page or field they are working on, which is used to select contextually relevant questions. Only textual questions and answers are supported, which is natural given the text-only nature of the system.

The IP-QAT system builds upon the Live Community concept, to create a system that works with non-linear, graphically intensive applications. Since these programs are not form-based, the user's context cannot be trivially determined, so we infer context based on recent command usage. Additionally, the increased complexity and visual nature of content-authoring applications creates more difficult problems for users to express simply using text [5], which motivates the IP-QAT system to promote and make extensive use of imagery.

Software Learning

To address the prevalent problem of software learnability [9], several research projects have focused on in-product learning techniques.

With Blueprint [4], Brandt et al. embedded a task specific search engine into a programming integrated development environment (IDE) to help users locate appropriate sample code. Hartman et al.'s HelpMeOut [13] is also integrated into an IDE and uses a recommender system to match build and debugging errors the user is having to solutions which have worked for other users in similar situations. While integrated into the target application, these systems are designed for a single discipline (coding), where our system could be used in multiple application domains.

CommunityCommands [15,17] leverages the power of the user community by using collaborative filtering algorithms to generate personal, contextual command recommendations. The AmbientHelp system [18] uses the context of recently used commands to find and automatically display relevant existing help content. Several systems present tool-specific animated help within the application including ToolClips and Google SketchUp's "Instructor" panel [10]. These systems use application context to present useful commands and existing help content, while our system selects relevant discussion topics. Additionally, none of these system support direct communication between users.

Researchers have also looked at ways outside of online forums that users can help each other. Twidale [23] studied "over the shoulder learning": informal help shared between colleagues in a workplace. This work serves as an important reference; however, it focuses on collocated assistance, as opposed to remote, online assistance.

DESIGN PRINCIPLES

We created a set of five design principles to guide the development of IP-QAT. The names in parenthesis after each heading indicate how each principle will be referred to in the system description.

Promote Rich Visuals (*visuals*)

Discussion boards and Q&A sites are typically dominated by text-based posts. Most sites provide support for

[1] http://turbotax.intuit.com/best-tax-software/why-choose-turbotax/help-live-community.jsp

attaching files or images; however, the process is often less than straightforward. Users are generally required to take and edit a screenshot using 3rd party software, and then manually find and upload it as an attachment to their post.

In Chilana et al's research into product support [5], support specialists described screenshots as being the most useful attachment type for diagnosing software problems, with over 90% agreeing they were useful. Despite their usefulness, less than 8% of support issues initially included screenshots, and often the specialists would need to explicitly request them.

Besides promoting the inclusion of visual attachments, increasing the prominence of such images while browsing the system is also desired. In current systems you need to click on a post, and often an "attachment" link before you are able to see an image. In other cases only registered users are able to see images associated with a post. Additionally, images can be a useful way to quickly "scan" and find interesting topics [25], so we would like to promote visual information to the top level of the system.

Make Use of Contextual Information (*context*)

There are many posts in modern discussion and Q&A sites where a potential question answerer begins by requesting additional information from the asker such as *"What version of the program are you using?"*, or *"What plug-ins do you have installed?"* Answers to these and other similar questions could have easily been included in the original question, but the asker either didn't think it was relevant, or perhaps thought it might be, but didn't want to bog the question down with potentially irrelevant details.

Some help systems are able to make use of the user's context to recommend commands [15,17] or pull up potentially relevant pre-existing help content [18] by looking at aspects of a user's context such as which commands have recently been used. Similarly, it would be useful for a community help system to be able to call up relevant community discussions based on the user's history.

Present Content Within the Application (*in-app*)

Current discussion and Q&A sites are accessible through a web browser, separate from the primary application the user is working with. In order to access these communities, the user must switch out of their primary application and search for help or interact with the community in this secondary program. If the content is presented in the primary application we gain several benefits:

Increased Awareness. Users are more likely to be aware of the community help system if it is an integrated feature of their program as opposed to an independent website they must leave to find and access.

Supports Opportunistic Learning. With content hosted in an external application, users will only switch to that program when they explicitly want to learn something. While working, users often have many short periods of down-time [17], and if the community help content is being displayed within the application the user will have a chance to glance over during this time. Additionally, since new material is being "pushed" to the user as opposed to the traditional "pull" model of help, the user has the chance to learn things they never would have considered looking for.

Minimize Transaction Costs (*min-cost*)

A primary consideration of our system is to minimize the transactions costs associated with both authoring content, and consuming content. For authors, we don't want the feeling of *"it's too much work"* to prevent anyone from contributing content. It should also be easy to include images and document source files. For those viewing the content, it should be easy to quickly browse through many topics to see if any are of interest. Topics should load immediately without any additional delay caused by waiting for the content to be downloaded from a server.

Small reductions in the transaction cost may individually not seem to make a huge difference, but previous work has shown that small improvements in interactions can cause categorical behavior changes that far exceed the benefits of the resulting decrease in task times [4,8,18].

Create a Positive Initial Experience (*initial*)

The first experience for a user at most traditional discussion and Q&A sites can be quite overwhelming. There are many different groups, boards, sub-boards, and forums to look at and it is not obvious which ones are appropriate to the user's interests. If the user decides they want to ask a quick question, or even propose an answer, they may first be inundated by login screens, registration procedures, confirmation emails, etc. Creating a good first experience with the system makes it more likely that user's will continue to work with it and both find help for themselves, and eventually contribute to the community.

SYSTEM DESCRIPTION

Based on the above mentioned design principles, we created the In-Product Questions, Answers & Tips (IP-QAT) system, a fully functional implementation of a community help system integrated into a graphical desktop application (Figure 1). In this section, as features relate back to the guiding design principles they will be noted in parenthesis (e.g., *initial, in-app*)

Overview

The IP-QAT system resides in a dockable panel within the host application. The system automatically creates a unique user-id when upon first being launched. There is no requirement to create a new account, which contributes to creating a positive initial experience (*initial, in-app*).

Main Interface

The view of the system consists of three main sections, one each for: *posting*, *viewing*, and *finding*.

The interface is designed to accommodate screens of varying sizes and resolutions and will gracefully reduce the exposed functionality as the window height is decreased (Figure 2). The panel can also be "rolled-up" into a narrow 15px wide bar which is restored when the cursor moves over it (Figure 2E). Even at very small sizes the system remains usable, and allows the user to customize the display to fit their preferred workspace (*initial*).

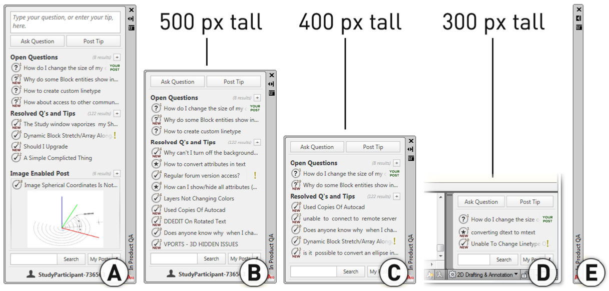


Figure 2. Progression of the interface from the full experience (A), to a very small compressed version (D). The rolled-up version is shown in (E).

Posting

The top part of the interface contains a text field for the user to enter their question or tip. The text field is relatively small to encourage conciseness, and is visible at the top level of the interface to encourage frequent use. On many discussion forums you will see minimally helpful "titles" such as *"I need help"* or *"What am I doing wrong?"* Non-descriptive titles like these may make it difficult for users to browse the topics; they will need to open up the thread to see what it is about. We omitted an explicit "title" field to encourage users to maximize the fraction of useful content in their posts (*min-cost*).

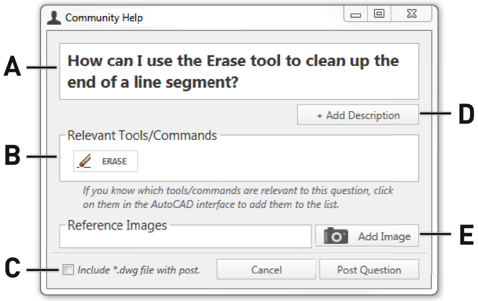
Questions and Tips

In addition to providing support for traditional question and answer type threads, the IP-QAT system creates a top-level mechanism for users to generate "tips". The main interface includes two buttons: "Ask Question" and "Post Tip". Tips give users a way to contribute to the system even if they are unable to answer any currently open questions. Also, tips provide a way for expert users to share their knowledge about smaller features, or "tricks" that they use which might not be something that a user would ever think to explicitly ask about. An example of a tip in a desktop setting might be *"Press F2 to rename a file"*; most people already know how to rename a file in some way, but this tip might let them do it more efficiently.

Supporting Information and Materials

Once the user has clicked on either of the "Ask Question" or "Post Tip" buttons, they are brought to the second-level posting dialog (Figure 3). If no additional information is needed, pressing *Enter* will post the question or tip with only the text entered in the main interface (Figure 3A). However, if additional supporting information is required, it can be added with this top-level modeless dialog.

Clicking on the "Add Description" button (Figure 3D) expands the window to include a second, larger text area to include a more detailed account of the issue. Figure 3B contains a list of tools and commands which the user feels are relevant to this question or tip. Rather than including command names as free-form tags or choosing commands from a long, exhaustive list, the user adds relevant commands to the post by clicking on them in the main interface of the application (*min-cost, in-app*). Since the dialog is modeless, a convenient workflow for asking a question involves launching the post dialog, and then attempting the task you have a question with. In doing this, the commands you use in your attempt will be automatically captured in the relevant tools list (*context*).



How can I use the Erase tool to clean up the end of a line segment?

If you know which tools/commands are relevant to this question, click on them in the AutoCAD interface to add them to the list.

Figure 3. Dialog for posting a question or tip.

The "Add Image" button (Figure 3E) launches the built-in screen-shot utility (*visual*). Similar to the "rectangular snip" mode of the Windows 7 Snipping Tool or TechSmith's SnagIt, the function puts a semi-transparent white overlay over the entire desktop and allows the user to drag a box over the area of interest. The image is inserted directly into the dialog and the user is able to create red "ink" markups on the image to highlight important areas (Figure 4).

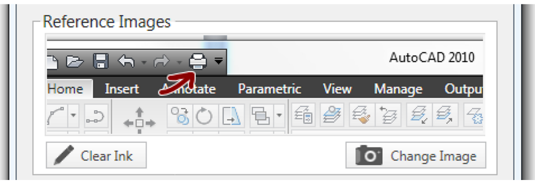


Figure 4. Image portion of the posting dialog box.

Including the active document as an attachment to the post in typical systems involves saving a copy of the document and then finding it through a file system dialog. Since the IP-QAT system has access to the application data, simply selecting a check box (Figure 3C) saves a copy of your document and uploads it with the post (*min-cost, visuals*).

Viewing

The majority of the IP-QAT interface is dedicated to displaying questions and tips, and is divided into three subgroups: *Open Questions*, *Answered Q's and Tips*, and *Image Enabled Posts* (Figure 1). Each section contains a specific selection of topics: *Open Questions* are questions which are still looking for an answer; *Answered Q's and Tips* contains a mixture of questions which have already been answered and contributed tips; and *Image Enabled Posts* selects one or more posts with an image component and displays the topic and image at the top level of the interface (*visual*). Promoting visual posts to an additional dedicated portion of the interface serves as an advertisement to other users that image-based posts are encouraged, and promote them to include images in their posts.

Selecting Topics to Display

Typical discussion boards order topics in chronological order based on the most recent post; all of the threads which were most recently updated are at the top. This helps in finding the newest posts, but potentially interesting threads without any recent activity quickly slip past the first page of results and become more difficult to find.

The IP-QAT system selects which topics to display by running two separate algorithms and then combining the results into one list. The first algorithm sorts all topics in the category by date of last activity; this ensures that the threads with recent activity (including newly created topics) are shown. The second algorithm looks at the context of the application (*context*) and gives each topic a relevance score, $\{S\}$. The algorithm takes the topic $\{t\}$, the user, $\{u\}$, and the ten most recently used commands $\{C\}$. and weights topics using those commands more highly. The particular algorithm used also considers if the user has seen all of the posts in the topic already or is the author of the topic, if they and other users have marked the thread as "interesting", and how many views the thread has:

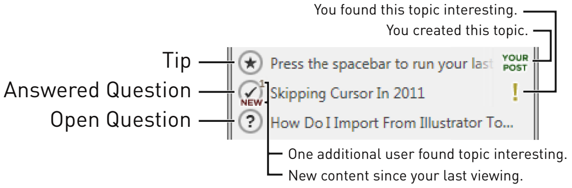
$$\begin{aligned} S(t,u,C) = {} & u.hasNotSeen(\forall t) * 11 \\ & + u.author(t) * 25 \\ & + u.foundInteresting(t) * 17 \\ & + \textstyle\sum_C (t.uses(c_i) * 3) \\ & + \textstyle\sum_{\forall u} (u_i.hasSeen(t) * 1.1) \\ & + \textstyle\sum_{\forall u} (u_i.foundInteresting(t) * 4.1) \end{aligned}$$

This is just one possible scheme, which we found worked well in our pilot studies, but has not been formally tuned. The results from the two algorithms are interleaved to generate the list of topics presented to the user.

The system checks for new posts on the server, and to minimize potential distractions, the list of topics will update at most once per minute if new posts are created or the user's context has changed. To guard against the list re-ordering while the user is interacting with the system, changes are queued up while the cursor is over the IP-QAT window and will take affect once the cursor has been away from the window for three seconds.

Displaying Individual Topics

The individual topic rows in the main view are marked up with various informative icons (Figure 5). The main icon on the left indicates the type of post (either a tip or a question), and, if a question, whether it has been answered or not. Additional graphical badges on the left indicate if there are posts in this thread that you have not read, and a counter of how many users have found the topic interesting.

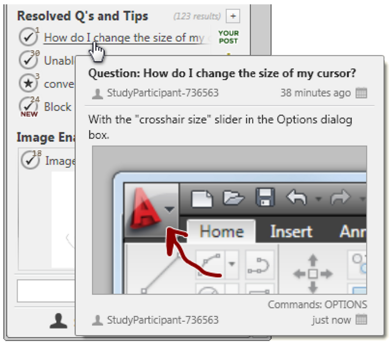

You found this topic interesting.
You created this topic.
Tip
Answered Question
Open Question
YOUR POST
NEW
!
One additional user found topic interesting.
New content since your last viewing.

Press the spacebar to run your last
Skipping Cursor In 2011
How Do I Import From Illustrator To...

Figure 5. Interface elements for individual topics.

To mark a post as interesting, users click on the '!' icon on the right side of the display bar. These topics receive a higher weighting in the selection algorithms; that is, if a topic you are interested has a new response that you have not seen, it will be promoted to the top of your display list. Additionally, there is a "Your Post" graphic to indicate which topics were created by this user.

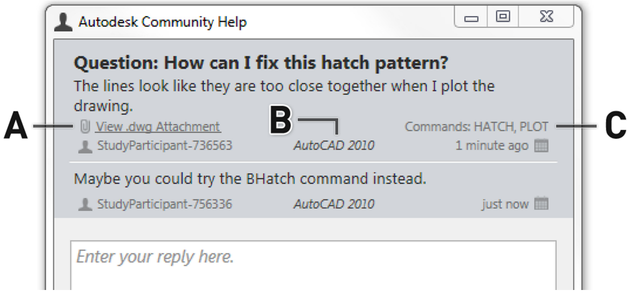
When placing the cursor over an individual topic display, the user is presented with a detailed tooltip showing the thread of posts (Figure 6). If any of the posts have images associated with them, they are also displayed (*visuals*). All posts and images are downloaded in a background process and stored locally so there is no loading delay when activating the topic tooltips (*min-cost*) making it possible and convenient for a user to quickly run over many topics and essentially "flip through" them to see if any are of interest. The detailed tooltip also contains the poster's name and when the post was submitted for each post.



Question: How do I change the size of my cursor?
StudyParticipant-736563 38 minutes ago
With the "crosshair size" slider in the Options dialog box.
Commands: OPTIONS
StudyParticipant-736563 just now

Figure 6. Tooltip interface for quick browsing of topics. When available, images are also displayed.

Clicking on a topic brings up the modeless view/reply dialog (Figure 7). The bottom portion of this window is similar to the posting dialog (Figure 3), while the top portion displays the content of the posts which make up the thread. This view not only shows the commands and tools relevant to each post, but also additional contextual information about which version of the software the author was using (e.x., *AutoCAD Electrical 2011*).



Question: How can I fix this hatch pattern?
The lines look like they are too close together when I plot the drawing.
View .dwg Attachment
StudyParticipant-736563 AutoCAD 2010 Commands: HATCH, PLOT 1 minute ago
Maybe you could try the BHatch command instead.
StudyParticipant-756336 AutoCAD 2010 just now
Enter your reply here.

Figure 7. The view/reply window showing the textual post content as well as a link to open the source file (A), the version of the program the post originated from (B) and the relevant commands (C).

Combining the thread viewing and responding functions into a single window streamlines the interface for the user, reducing the number of clicks required to contribute to the community, reducing overall transaction cost (*min-cost*). Once an open question has been replied to, it becomes an "answered" question. If the original asker does not feel the question has been adequately answered, when they reply to the topic they are given the option to select whether or not this question should be marked as answered. If they choose "no", then the question becomes open again.

Finding

While the collection of posts in the interface dynamically updates to display the newest and most relevant topics, there will still be times where the user wants to search for specific content. The search box at the bottom of the interface allows searching for keywords, commands and user names over the entire collection of postings. As characters are entered in the text field, the main display updates in real time to show all matches (*min-cost*).

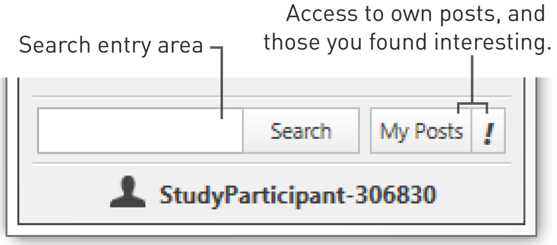


Figure 8. Interface elements for finding posts; by keyword search, those you have authored, or those you marked as interesting.

The "My Posts" button gives users the ability to find all of the content they have contributed, and it is possible for a user to see all topics which they have marked as interesting by clicking on the '!' button.

Context Menu Access

To permit access to the system even if the main interface panel is closed, we implemented a context-menu augmentation which adds IP-QAT specific items into the application's exiting context menu (Figure 9).

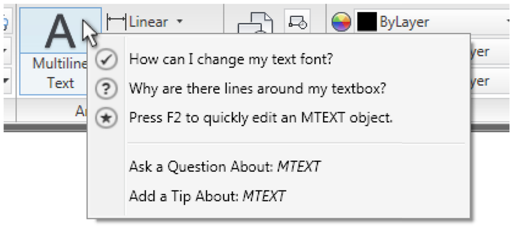

A
Multiline Text
Linear
ByLayer
How can I change my text font?
Why are there lines around my textbox?
Press F2 to quickly edit an MTEXT object.
Ask a Question About: MTEXT
Add a Tip About: MTEXT

Figure 9. Context menu access of IP-QAT system.

These options can be accessed from either the right click menu of the application, or by right-clicking directly on a tool icon (*in-app*). When accessed from an individual tool, the most highly rated topics related to that particular tool are presented. The tool-button context menu also includes items to ask a question or post a tip about the selected tool. When selected, these options launch the posting dialog (Figure 3) with the current tool preloaded into the relevant tools/commands list (*context*).

SYSTEM IMPLEMENTATION

To perform our intended field study, the implementation of the IP-QAT system needed to be fully functional and robust enough to distribute to users and have them use it for an extended time on their primary workstations while performing their regular jobs.

A significant technical challenge was determining how to reasonably communicate and synchronize the posts, images, and attachment files across all the users of the system. A conventional approach is to run a web-server and database on the back-end for communicating with the clients and storing the data. However, configuring and running this type of system requires specialized expertise, dedicated hardware, and involves a significant overhead cost for setup and maintenance.

As an alternative, we chose to use the Amazon Simple Storage Service (Amazon S3) as our back-end. Locally, each post is stored as a folder, with simple text and image files containing the post data. When the user creates a new post, the folder is created locally, and then uploaded to the S3 server. At regular intervals, each client queries the S3 server and downloads any new posts on a background thread, saving them into folders in the file system. Once downloaded, the posts are loaded into the system. As a benefit of having the data downloaded and stored locally, the user is still able to view existing topics while off-line, and they are also able to post new content which gets updated next time they connect to the internet.

The connection with the S3 server also allows the easy and automatic collection of usage data and statistics, which is a significant convenience when performing distributed, remote, user studies.

AutoCAD Plug-in

The system was implemented as a plug-in to AutoCAD using the ObjectArx framework. The logic of the system was coded in C# and the interface was developed using Microsoft's WPF. The entire system was bundled into a single installer which added the necessary registry keys so the plug-in would start automatically when any of the 21 supported versions of AutoCAD 2010 or 2011 were run. It is possible to have more than one version of AutoCAD running on a machine at one time, and the system implementation includes redundancies ensuring that the system continues to work in these cases.

EVALUATION

To understand how the IP-QAT system would work with a community of users, and compare it to traditional software discussion boards, we performed a two week user study.

Participants

We recruited 48 professional AutoCAD users (ages 22-47: μ=36, 12 female) for the two week study. To qualify, the paid volunteers were required to use AutoCAD for at least 10 hours/week. An initial survey asked how often they typically looked at, and/or contributed to, online AutoCAD discussion forums (Figure 10) and to rate their AutoCAD expertise level. In general, most users looked at discussions somewhat regularly, but were far less likely to contribute.

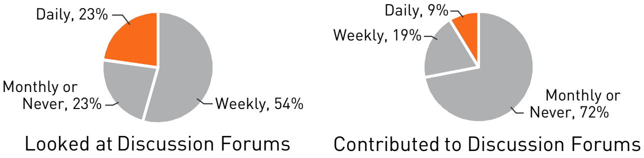


Figure 10. Distribution of user's pre-existing habits of looking at, and contributing to, online forums.

A number of participants dropped out part way though the study, or did not meet the minimum time requirements, leaving us with 37 valid participants.

Design

The participants were split into two groups with the most prolific forum users being evenly divided between the groups, and the rest of the users divided to equally distribute the level of AutoCAD expertise.

Community Help Systems

As an alternative to the IP-QAT help system described in this paper, the *forum* condition consisted of a private discussion forum accessible only to study participants. This private forum was hosted on the existing Autodesk Community site which draws over 200,000 visitors daily. The Autodesk forums typically require users to create an account, but we provided pre-activated accounts for the study to simplify the process. The presentation order of the community help systems were counterbalanced between the two user groups and after one week the groups switched to the alternate system.

Procedure

The participants installed an AutoCAD plug-in which contained both the IP-QAT help system, and the ability to launch a web-browser for directly accessing the private discussion forum of the *forum* condition. When AutoCAD launched, the assigned help system for that week would be automatically loaded (i.e., the IP-QAT panel would open, or a browser would open with the *forum* page loaded). Users were not required to keep the help system open, however, after 6 hours the active system would re-open as a gentle reminder that the study was going on. Participants were told that the automatic opening of both community help systems was a property of the study, and not the standard behavior of either system.

A collection of 140 questions and associated answers were collected from the existing public discussion forum and used to seed each of the communities with content. This allowed users to experience the viewing/consuming aspects of the systems immediately without needing to wait for their fellow participants to contribute topics. 70 topics were used for the first week (the same ones for each community help system), and the remaining 70 were used as the seed questions for the second week. Between conditions the content contributed from the previous group was removed.

Users were encouraged to use the provided community help systems during the course of the study, but were not prevented from looking for help from other sources. At a minimum, participants were asked to contribute three posts in each condition. To better simulate a larger and more active user community, bonus prizes ranging from $100 to $25 were awarded to the top five contributors in each condition for each group, on top of the $50 base amount given to each user for participating. At the end of the study user's filled out a web-based questionnaire.

Results

Volume and Rate of Posting

Over the duration of the study the IP-QAT system generated 316 new posts compared to 252 in the *forum*. Repeated measure analysis of variance showed a main effect for posts per user ($F_{1,36} = 5.17$, $p < .05$) with means of 8.54 and 6.81 for *IP-QAT* and *forum* respectively (Figure 11). This amounts to a 25% increase in the volume of posts for the *IP-QAT* system over the *forum*. We can consider the first three posts to be "mandatory" since that was the minimum requirement to participate in the study and look at posts beyond that to be "voluntary". That breakdown results in means of 5.81 for *IP-QAT* and 3.94 for *forum,* for a 47% increase in the amount of voluntary posting activity ($F_{1,36} = 6.81$, $p < .05$).

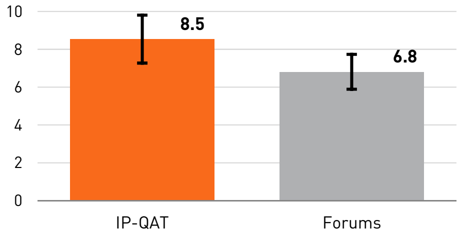


Figure 11. Mean number of posts contributed per user. (Note: error bars report standard error)

The IP-QAT system also led to the creation of more new topics overall (103 for *IP-QAT*, 82 for *forum*), however the effects per user did not reach significance ($F_{1,36} = 1.7$, ns).

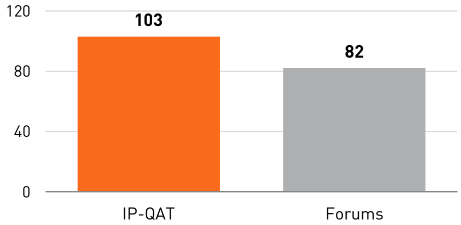


Figure 12. Number of new topics created.

The additional monetary motivation used for the study likely contributed to higher post volumes than a natural scenario. However, since the rewards were paid out for both conditions, we feel the overall trends would translate to a real deployment. Anecdotally, the monetary rewards did not seem to be the main driving factor for any of the participants; for example, there were no instances of users "gaming" the system by creating many posts of limited value. It was clear that the participants were using the system to ask detailed and technical questions that they genuinely had about AutoCAD, and were making an effort to provide accurate answers when responding. Questions that were asked and resolved during the study included topics such as: creating custom hatch patterns, scaling multiple MTEXT objects at once, and temporarily hiding objects to see what is being obscured.

We manually tagged newly created topics from the *forum* as questions or tips. Of the new 185 total new topics created over both conditions, 40 were tips, and 35 of those came during the *IP-QAT* condition. In the questionnaire, users were asked to rate a series of questions on a 5-point Likert scale, including if they found tips useful, to which 28 of the 37 users responded with agreement compared to only 2 who viewed tips negatively (Figure 13).

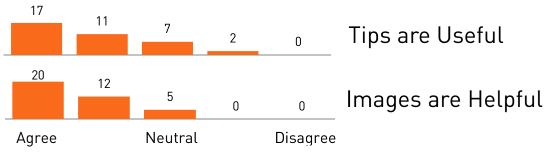


Figure 13. Subjective results for "*It is useful when people post tips, as opposed to only questions and answers.*" and *"I find images helpful for explaining and answering questions."*

Another metric to look at is how long it takes for a question to receive an answer, which can be correlated with the size of the community [16]. On StackOverflow the 300,000 users are able to deliver a median time-to-first-answer (TTFA) of ~11 minutes [16]. In our study with 18 and 19 users in each group the median TTFAs were 61 minutes for *forum* and 38 minutes for *IP-QAT* (Figure 14). This result did not reach significance (Mann-Whitney, *p* > .05).

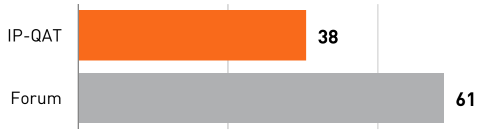


Figure 14. Median Time-to-First-Answer (Minutes).

Use of Images

Overall, 38 images were associated with posts in the *IP-QAT* condition compared to 9 in the *forum* (Figure 15). This amounts to 12.2% of *IP-QAT* posts, and 4.2% of *forum* posts containing images. In contrast the image attachment rate on the public Autodesk discussion boards is approximately 2%.

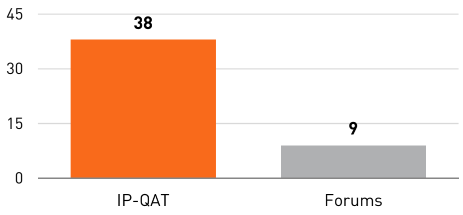


Figure 15. Total number of images used.

Users were also asked to rate the usefulness of images. As found in the product support domain [5], the majority of users consider images useful (Figure 13). Besides simply making images easier to include in a post, the IP-QAT system also simplifies the process of adding pen-type markups to the images. Of the 38 images from the IP-QAT system 8 contained overlaid markups, compared to only 1 from the *forum* condition.

Subjective Comparison

At the end of the exit survey participants were asked to score the statements *"I found this system easier to use"*, *"I found this system more useful"*, and *"I would prefer to use this system"* on a 5-point scale ranging from *IP-QAT* on one end to *Forum* on the other (Figure 16). Valuing the answers on a scale from 2 to -2, all questions reached significance on a one sample t-test: *Easier to Use* (t(36) = 0.76, *p* < .005), *More Useful* (t(36) = 0.78, *p* < .0005), *Would Prefer to Use* (t(36) = 0.89, *p* < .0001).

While the IP-QAT system was the dominant response for these questions, the forum was preferred by some users and several users suggested that it would be useful to have the content from each system available on the other. Likewise, we do not see these systems as alternatives to one another, but rather as systems that could work together to create a more complete community experience.

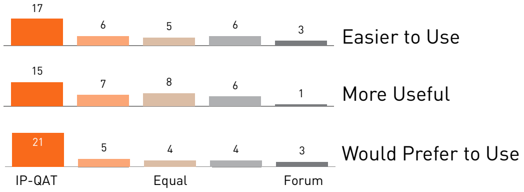


Figure 16. Subjective results comparing the *IP-QAT* and *forum* community help conditions.

Visualization of Activity

With so many topics and posts being created, it is difficult to get an overall feel for how individual threads are evolving. To get a sense of the activities occurring in the community, we created visualizations of each thread and grouped them by week and condition (Figure 17).

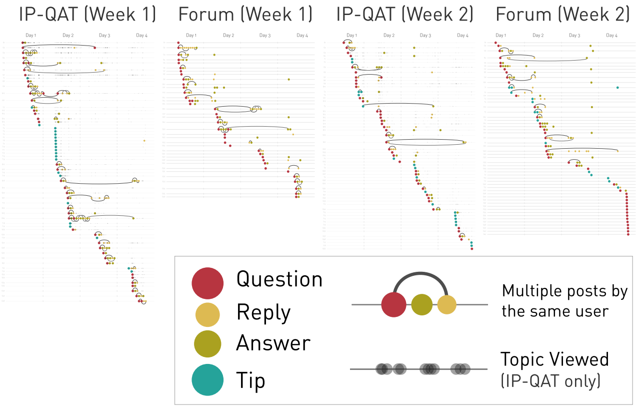


Figure 17. Visualization of all topics created during the study. (Note: this is a vector graphic, zoom in to see in more detail)

To generate the visualization, all posts entered during the study were manually classified into one of four categories:

Questions: Posts of a problem looking for an answer.
Answers: Attempts at answering a question.
Tips: Helpful statements not answering a question.
Replies: Responses not providing answer.

Posting activity was then organized by thread and displayed in chronological order as color coded dots from left to right on the timeline. When a single user made multiple posts in one thread, arcs were drawn connecting all of the user's posts. For example, in Figure 18A we can see that there were 7 total posts, but by looking at the arcs connecting the nodes, we see that the first user made posts #1, 3, 5, and 7, while a second user made posts #2, 4, and 6. This thread is an example of a common pattern where there is one question asker, and one answerer going back-and-forth trying to figure out the solution to the problem.

Another pattern we see and recognized by Singh [10] is where one user asks a question, and many others jump in with replies; more or less saying "me too". Figure 18B shows an example of this type of thread, with 10 total posts, and 7 unique authors.

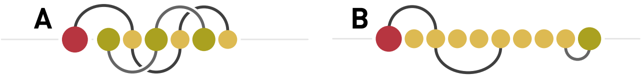


Figure 18. Two example conversations. Between two users (A), and many unique users (B).

Besides posting information, the IP-QAT system is also implemented to know when users viewed the topic. These events are represented on the timeline as small grey dots, and in Figure 19 we see an example where the views had trailed off for a particular thread, and then once a new post was made, the views picked up again.

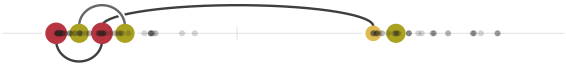


Figure 19. "Topic Viewed" dots showing a resurgence of interest in a thread after a new post is made.

It is interesting to note how the group who used the forums in the first week posted zero tips, while the group in the forums for the second week (who had used IP-QAT the first week) contributed 15 tips; once they became used to tips in the IP-QAT system they decided to try them in the *forums* condition even though the forums contained no explicit support for publishing tips. We are hopeful that going forward, as the community grows, visualizations like this may expose longer term trends.

User Feedback

Numerous positive comments were obtained in the questionnaire. In particular, several users commented favorably about the quick access IP-QAT provides:

I preferred the in product version much better than having to access a separate forum. It seemed much easier to access and in general just more convenient!

Users also mentioned the ease of attaching images:

I liked that it was always on, and I could take a peek at any time, but that it could be rolled up out of the way to save space. I liked that it was easy to attach a snapshot of the portion of the drawing.

In contrast, images were found to be difficult to attach in the *forum* condition:

I found that I needed another program to be able to add an image to my post, and that slowed me down versus the in product system which had the imaging software included.

Users also had a particular appreciation for the instant feel that the IP-QAT system provided:

It gave me a feel that the questions would be more immediately answered. It almost gave me the feel of a blackberry messenger app. or texting in general. I really think if this system is modeled after a platform like that it will be a success. Who doesn't text anymore? Overall, it kinds of creates an environment that allows users to more readily "text" a question. I think for the newer users or younger generation of users, this system will be quite attractive.

Another user provided a similar comment:

A nice idea of bringing experienced users together with those experiencing issues. Easy to throw a question out there at the moment you experience it, while in the program, and easy to check to see if there are any answers.

We were encouraged to see such enthusiastic comments from real-world end-users, giving us confidence that IP-QAT would be welcomed by user communities at large.

DISCUSSION AND FUTURE WORK

A main contribution of our work is a study demonstrating behavioral differences between using an in-product, and out of product community help system. Our study has shown some clear values of integrating the community help system within the application. The study also provided us with future topics of research.

Several of the study participants found it difficult when looking at a question with more than once answer to determine which one the asker found useful. Creating a way for both the author and others to mark which responses were helpful, as well as configurable ways to order the list of available posts, are among our future directions.

While we used the user's command history to determine context, it would be interesting to use other cues such as which GUI elements are active (e.g., ribbon tab, tool pallets, dialog boxes) or gathering semantic information from the active document. Additionally, many questions and tips centered around dialog box elements, so the ability to mark posts with such elements (in addition to commands) will be useful.

Fostering a community of active users is an important aspect of any community helps system, which we did not address in this paper. Established Q&A sites such as Stack Overflow have effective mechanisms for motivating users [16] such as badges, awards and leaderboards which would be useful to integrate into our system in the future.

We would like to deploy the IP-QAT system in the wild for a wider scale deployment, and to do this we will need to address privacy and confidentiality issues related to employees sharing company data. Some possibilities include automatically or manually blurring sensitive areas of screenshots, site controls for managers to block the uploading of files, or the ability to transform a confidential source file into a sharable one which exhibits the same problem as the original.

A wide-scale deployment may also require us to reconsider the system architecture. Besides being easy to setup and use, the Amazon S3 service was extremely cost effective; even with over 1.2 million PUT, GET, and LIST requests and more than 10 GB of data transferred over the course of the study, the web services cost came to only $3.78. As such, we see this is a scalable solution.

CONCLUSION

To conclude, we have investigated a community-based question-answer system within the context of a desktop software application. We discussed a number of relevant design principles, and presented our new system, IP-QAT. Particular attention was given to providing an image-rich experience, and reducing the transaction cost of posting and browsing content on the system. Furthermore, we contribute a robust and cheap architecture that allowed us to deploy the system to 37 users for a 2 week filed study. The evaluation identified a number of quantitative and qualitative benefits on our system, which we hope will serve as inspiration for future research.

ACKNOWLEDGMENTS

The authors thank Eric Wright for his assistance with the Autodesk discussion forums.

REFERENCES

1. Ackerman, M.S., and Malone, T.W. (1990). Answer Garden: a tool for growing organizational memory. *ACM SIGOIS and IEEE CS TC-OA,* 31-39.
2. Adamic L.A., Zhang, J., Bakshy, E., Ackerman, M.S. (2008). Knowledge sharing and yahoo answers: everyone knows something. *ACM WWW*, 665-674.
3. Bouguessa, M., Dumoulin, B., and Wang, S. (2008). Identifying authoritative actors in question-answering forums: the case of Yahoo! Answers. *SIGKDD,* 866-874.
4. Brandt, J., Dontcheva, M., Weskamp, M., Klemmer, S.R. (2010). Example-centric programming: integrating web search into the development environment. *ACM CHI,* 513-522.
5. Chilana, P., Grossman, T., and Fitzmaurice, G. (2011) Modern Software Product Support Processes and the Usage of Multimedia Formats. *ACM CHI*, 3093-3102.
6. Dearman D and Truong KN. (2010). Why users of yahoo!: answers do not answer questions. *CHI,* 329-332.
7. Fiore, A.T., Tiernan, S.L., Smith, M.A. (2002). Observed behavior and perceived value of authors in usenet newsgroups: bridging the gap. *CHI*, 323-330.
8. Gray, W. D. and Boehm-Davis, D. A. (2000). Milliseconds Matter: An Introduction to Microstrategies and to Their Use in Describing and Predicting Interactive Behavior. *Journal of Experimental Psychology: Applied*, 6 (4), 322-335.
9. Grossman, T., Fitzmaurice, G., and Attar, R. (2009). A survey of software learnability: metrics, methodologies and guidelines. *ACM CHI*, 649-658.
10. Grossman, T. and Fitzmaurice, G. (2010). ToolClips: an investigation of contextual video assistance for functionality understanding. *ACM CHI*, 1515-1524.
11. Harper, F.M., Moy, D., and Konstan, J.A. (2009). Facts or friends?: distinguishing informational and conversational questions in social Q&A sites. *CHI,* 759-768.
12. Harper, F.M., Raban, D., Rafaeli, S., and Konstan, J.A. (2008). Predictors of answer quality in online Q&A sites. *ACM CHI* , 865-874.
13. Hartmann, B., MacDougall, D., Brandt, J., Klemmer, S.R. (2010). What would other programmers do: suggesting solutions to error messages. *CHI,* 1019-1028.
14. Horowitz, D., and Kamvar, S.D. (2010). The anatomy of a large-scale social search engine. *WWW,* 431-440.
15. Li, W., Matejka, J., Gossman, T., Konstan, J.A., and Fitzmaurice, G. (2011 – in press). Design and Evaluation of a Command Recommendation System for Software Applications. *ACM TOCHI.*
16. Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. (2011). Design Lessons from the Fastest Q&A Site in the West. *ACM CHI,* 2857-2866.
17. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. (2009). CommunityCommands: command recommendations for software applications. *ACM UIST,* 193-202.
18. Matejka, J., Li, W., Grossman, T., and Fitzmaurice, G. (2011). Ambient Help. *ACM CHI,* 2751-2760.
19. Nam, K.K., Ackerman, M.S., and Adamic, L.A. (2009). Questions in, knowledge in?: a study of naver's question answering community. *ACM CHI,* 779-788.
20. Pal, A., and Konstan, J.A. (2010). Expert Identification in Community Question Answering: Exploring Question Selection Bias. *ACM CIKM*, 1505-1508.
21. Singh V and Twidale MB. (2008). The confusion of crowds: non-dyadic help interactions. *CSCW*, 699-702.
22. Singh V, Twidale MB, and Rathi D. (2006). Open Source Technical Support: A Look at Peer Help-Giving. *IEEE HICSS,* 6, 118c-118c.
23. Twidale, M. B. (2005). Over the Shoulder Learning: Supporting Brief Informal Learning. *CSCW,* 505-547.
24. Welser, H.T., Gleave, E., Barash, V., Smith, M., and Meckes, J. (2009). Whither the Experts? Social Affordances and the Cultivation of Experts in Community Q&A Systems. *IEEE ICCSIT*, 4, 450-455.
25. Yeh, T., Chang, T., and Miller, R.C. (2009). Sikuli: using GUI screenshots for search and automation. *UIST*, 183-192.