Simplify ISS Flight Control Communications and Log Keeping via Social Tools and Techniques

Hugh S. Cowart[1] and David W. Scott.[2]
NASA - Marshall Space Flight Center, Huntsville, AL, 35812, USA

and

Daniel J. Stevens[3]
Barrios Technology, NASA - Johnson Space Center, Houston, TX, 77058, USA

The heart of flight operations control involves a) communicating effectively in real time with other controllers in the room and/or in remote locations and b) tracking significant events, decisions, and rationale to support the next set of decisions, provide a thorough shift handover, and troubleshoot/improve operations. International Space Station (ISS) flight controllers speak with each other via multiple voice circuits or "loops," each with a particular purpose and constituency. Controllers monitor and/or respond to several loops concurrently. The primary tracking tools are console logs, typically kept by a single operator and not visible to others in real-time. Information from telemetry, commanding, and planning systems also plays into decision-making. Email is very secondary/tertiary due to timing and archival considerations. Voice communications and log entries supporting ISS operations have increased by orders of magnitude because the number of control centers, flight crew, and payload operations have grown. This paper explores three developmental ground system concepts under development at Johnson Space Center's (JSC) Mission Control Center Houston (MCC-H) and Marshall Space Flight Center's (MSFC) Payload Operations Integration Center (POIC). These concepts could reduce ISS control center voice traffic and console logging yet increase the efficiency and effectiveness of both. The goal of this paper is to kindle further discussion, exploration, and tool development.

I. Introduction

SPACE flight control requires a river of decisions, actions, and observations, often with parallel streams. Voice communication is a primary means of navigation and adaptation. That said, flight controller performance can be saturated by an excessive amount of voice traffic and/or demands from other systems.

Console logs have been used since the beginning of the Space Program to document significant observations, conversations and actions of flight controller console positions. From the 1960's through the 1990's, pen and paper were used. Voluminous physical storage was required and document search was mentally exhausting. During that era, terminals for mainframe/microcomputer based flight control and planning systems took up the real estate that was not needed for communications gear. Application switching on the terminals was awkward and slow. Due to reliability requirements on the control and planning systems, the evolution of workstation and Personal Computer (PC) servers/clients took quite a while. Figure 1 shows a typical Spacelab-era console log.

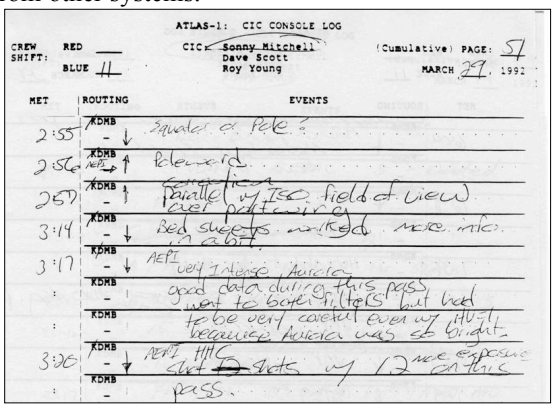

Figure 1 Spacelab Console Log, 1992

[1] Data Management Coordinator, Mission Operations Laboratory, MSFC/EO30, hugh.s.cowart@nasa.gov
[2] Computer Engineer, Ground Systems Development & Integration, MSFC/EO50, scotty@nasa.gov
[3] ISS Flight Controller, Mission Operations Directorate, JSC/DI2, daniel.j.stevens@nasa.gov

When desktop computers with Graphical User Interfaces (GUI) became commonplace in control rooms at the beginning of ISS operations, log keeping based on office automation applications evolved rather quickly. Word processor or spreadsheet programs were typically used and had these advantages over handwritten logs:

- More legible.
- Easier to search.
- Cut and paste from other sources saved typing time.

That said, automation brought new challenges:

- Search was serial - could not easily gather all entries related to a particular topic.
- Hard to search across files from different console positions or time intervals.
- Cut and paste could produce a glut of information.

While Shuttle-based missions could last for a few weeks, ISS is permanently manned. From 2000 to present, log keeping approaches evolved to handle years of data. MCC-H uses a Microsoft® Word template and macros so that log entries can be harvested into a centralized database archive that can then be searched. POIC has been using Filemaker Pro to maintain position-specific logs with flexible capabilities for search, selection, one-click setting of status flags and the like. Figure 2 shows a recent excerpt from a Filemaker Pro® based log at POIC. Examples from MCC-H's current approach appear in Section IV.

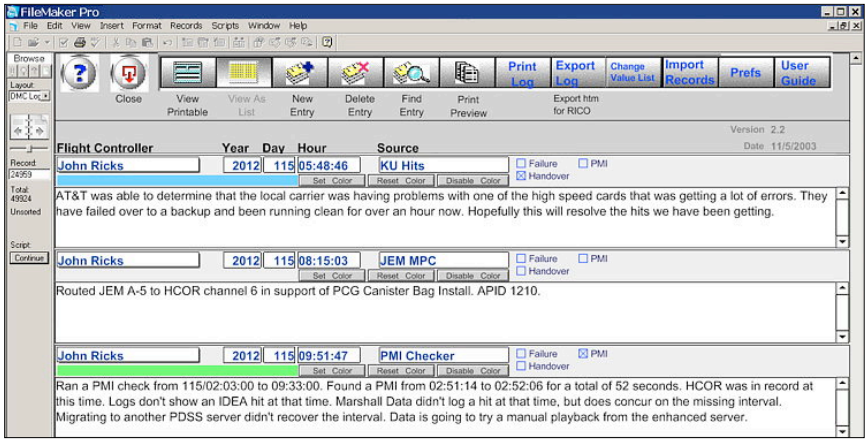

Figure 2 POIC Filemaker Pro® Based Log, 2012

A time-honored principle in operations is "Better And Best Are The Enemy Of Good Enough." While this has merit, the last few years have seen massive growth in communications and planning activity due to the increase in crew size and the transition to post-assembly utilization. Yesterday's Good Enough is today's Hanging In There and could well be tomorrow's Coming Up Short. Since a gram of prevention is worth a kilogram of cure, the authors offer three log keeping systems (one recently deployed, two seed concepts) that ought to be Good Enough To Grow With Operations For Quite Some Time:

- A web-based Console Log Tool (CoLT) that a) with a log owner's permission, supports real time viewing and/or commenting across operator/discipline logs by both on-duty and off-duty personnel, and b) links to other processes;
- A scheme that uses commonly accepted microblog syntax (e.g., @targetid, #hashtag) to perform text conversation via the logs themselves (only portions of the log that the console operator wishes to share would be viewed);
- A communications dashboard that integrates a console log, focused text chat channels for certain types of conversations, status updates of general and discipline-specific interest, and some aids for handling spot discussions and vectoring longer ones.

The following sections of this paper assume that the reader is familiar with ISS Operations Protocol, i.e., standard phrasing conventions for verbal and written communications in the real time flight control environment. Translations of acronyms referring to flight controller positions, systems, payloads, and statuses are generally not given, as they are not central to describing how the concepts work.

II. How Social Media Methods Could Empower Log Keeping Systems

When people think of social media, recreational friendship typically comes to mind, but the term "social" just as readily refers to "interactions among organisms and their systems." In that context, social interactions merit just as much attention as we pay to interactions among our electronic systems. Earlier papers explored functional characteristics of social media that can benefit operations.[1,2,3] Figure 3, which was adapted from presentation charts, reviews some *principles* and *behaviors* that relate particularly well to this paper.

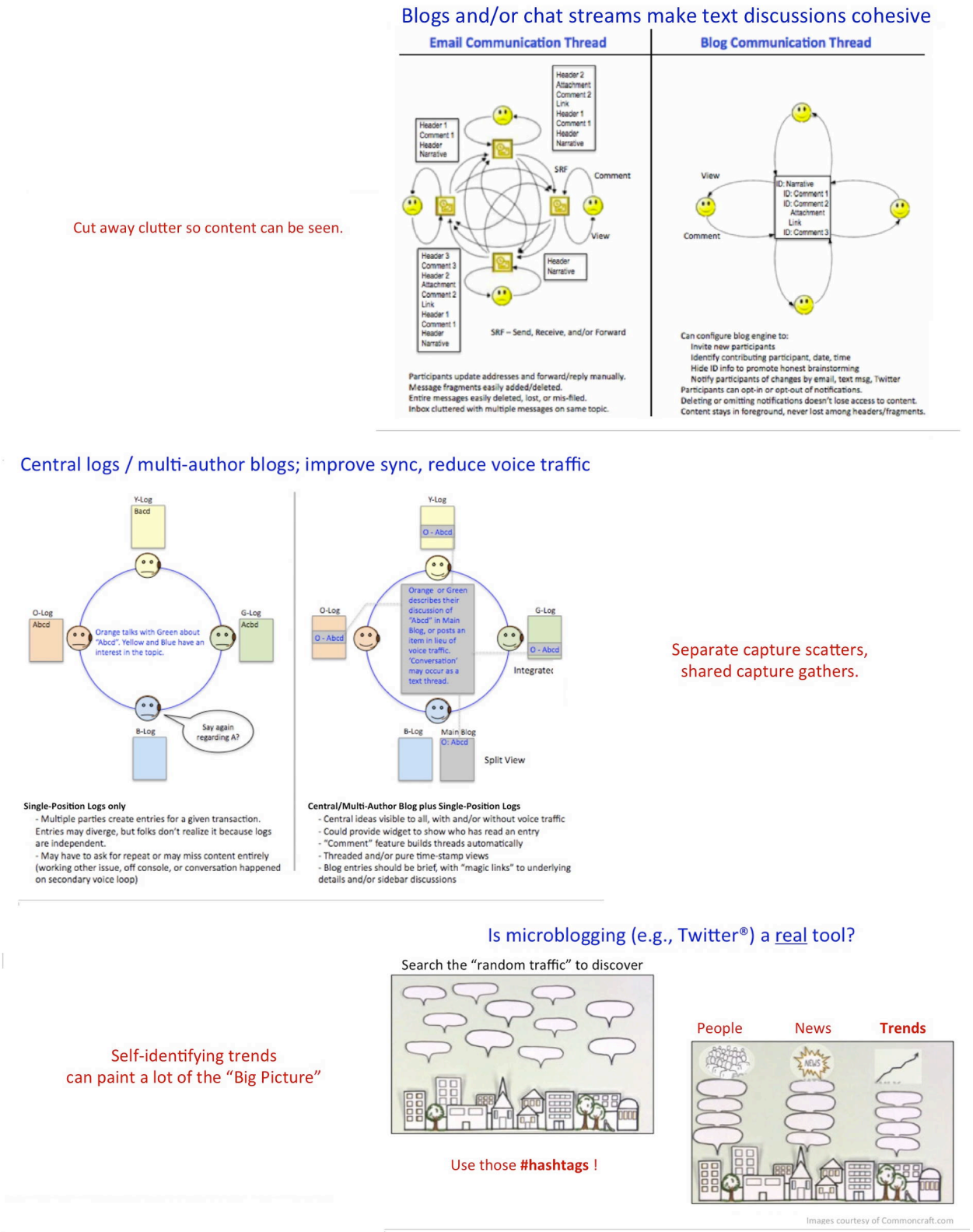

Figure 3 **Some Social Media Functional Characteristics – Implications for Log Keeping**

III. Console Log Tool (CoLT) (by Hugh S. Cowart)

A. CoLT Introduction

The POIC software developers and operations cadre recently deployed a new web-based console log utility to replace the legacy Filemaker Pro® based system. First and foremost, CoLT provides a streamlined interface for producing flight operations log entries. In addition to this fundamental capability, it provides functions that directly support the bidirectional flow of operational information between a real time operator and their "back room" support. As a result, CoLT has the potential to morph 'The Log' into a central communications hub that produces a higher fidelity log of real time support activity, yet requires less traditional logging effort on the part of the real time operator. The following is a review of some prominent CoLT features and how they combine to support new communications processes.

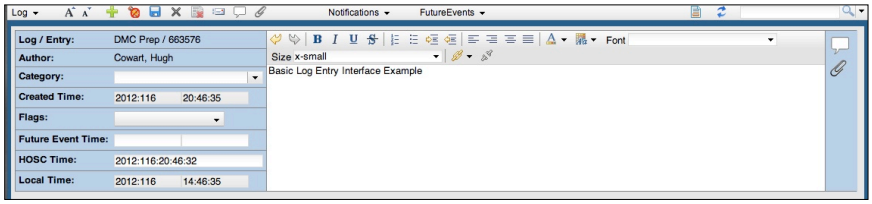

Figure 4 CoLT Log Entry in "Edit" Mode

B. Accessibility

A primary issue with legacy POIC logs was limited access. Due to firewall issues and the use of local clients, direct viewing of logs was restricted to locations inside the POIC. Information logged during the day often has a very short shelf life. With daily export going out after normal business hours, back office support could experience up to 24-hour delays in information receipt unless the console operator makes additional/duplicate efforts to send out information via email or a phone call. CoLT is far more accessible. In basic terms, anyone with the proper credentials and a VPN connection can view console logs in real time. In addition to providing real time access to logs on a team-by-team basis, CoLT permits viewing multiple logs concurrently in separate windows/tabs or in an interleaved presentation. This is very advantageous for grasping a "big picture" view of operational progress or post-ops analysis.

What about remote editing of the log? While the idea is somewhat controversial in terms of traditional operations concepts, it could, if well managed, significantly streamline front room communications processes. The tool has these capabilities:

- Remote and local creation, editing, and/or commenting to logs.
- Control is on a per log basis, administered by assigned log owner(s).

In the future, console applications could be enabled to auto-generate log entries based on telemetry or user-defined triggers, e.g., commanding, with or without a confirmation dialog. Auto-generated entries could be embedded in the operator's log so that "it's all one story" or placed in a separate log for interleaving or hiding with respect to the operator's log as needed.

C. Flags

In CoLT, each log has a "Flags" field with a customizable pull-down value list. Typical flags are Failure, Anomaly, Handover Item, etc. When editing a log entry, the user can set multiple flags as appropriate. A user with Group administrative privilege can, in addition to modifying flag names, associate automatic actions with a given flag's status, such as sending a notification email of the entry to a flag-specific distribution list when the flag is set. Flag status also serves as filtering criteria for CoLT's search engine.

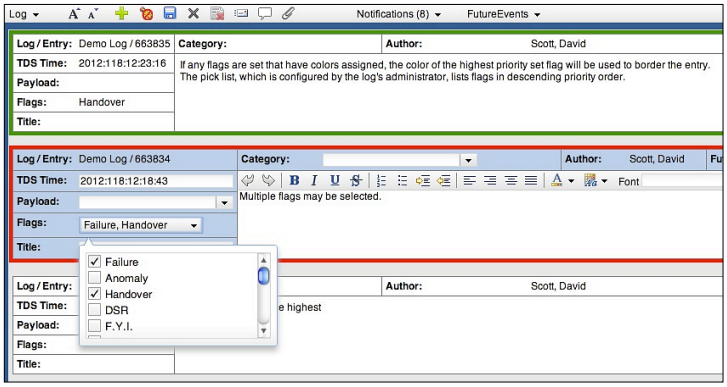

Figure 5 Flags - Behavior During Log Keeping

Future versions of CoLT might allow other actions, e.g., generating a linked action item to be dispositioned by back office processes. Consider how this might simplify anomaly processing. In the legacy log keeping system, an operator determines an event is an anomaly, notes it as such in the log and hopes the back room sees it when reviewing the daily log export. If back room processing needs to begin prior to the daily log export, the operator has to open an email, copy and paste the log entry, determine who it goes to (which may require several phone calls), then address and send it. In CoLT, when the Anomaly flag is checked, the operator would only have to review the auto-generated email/text message that will notify the back room and press OK. The idea is to reduce the amount of coordination and actions needed to pass the ball to someone off-console who can run with it.

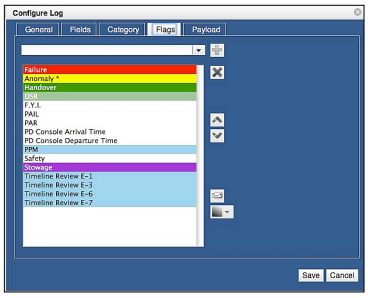

Figure 6 Defining Flags

D. Future Events

Many log entries pertain to a future event. This information would be useful to whoever is on duty when the event actually occurs. In traditional logs, this information often becomes buried as time passes and new log entries are created. One could carry the information in a shift handover log, but this is unwieldy if the future event is a long way out. CoLT allows an author to include a future event date/time in a given entry. When that date/time arrives, CoLT notifies anyone viewing that log (which at a minimum consists of the on-console operator) and provides a link to retrieve the given entry for reference.

Of course, discussions about a significant upcoming real time activity happen in back rooms (and halls) as well. Useful information from these discussions can now be recorded as a log entry, *authored by the back room,* with an assigned future event date/time. Additional conversations related to the same activity will logically lead to new entries and/or updates to existing ones, so that the log now carries the activity as a 'pre-loaded' event. While entries pertaining to the activity may originate at different times, they can be tied to a common future event time that is early enough for the pre-loaded information to be reviewed and acted on prior to the actual activity.

This scheme is a little bit like having an alarm clock that wakes you up *and* gives you a copy of written "to do" lists. This improves communications between real time operations and back room support in both directions. The log becomes more than a serial set of text entries regarding operations events and evolves towards an integration tool. After all, if we are not integrating, we are probably disintegrating!

E. Quality Records

POIC is required to maintain an *uneditable* archive of console logs for record and/or investigation purposes. These fulfill the role of "Quality Records" (QR) in the International Organization for Standardization's (ISO 9000) quality management system. Traditional and current practice is for each console position to email a Hypertext Markup Language (HTML) file of each day's console log entries to an archivist. CoLT supports manual and automatic QR generation, with options to minimize the human effort needed.

F. Rich Text Formatting (RTF)

A common challenge in the legacy tool was unfriendly formatting. The most popular layouts only showed 5 lines of text; using scroll bars became old after a while. Alternative layouts could show the entire content of large entries, but then displayed vast amounts of blank space for small entries. Text formatting options were limited and did not include common features such as bullet lists, numbered lists and hyperlinks. CoLT provides a robust Rich Text Formatting (RTF) suite and auto-adjusts the sizes of individual entries as needed to display all of the content. This makes it much easier to view and edit detailed entries.

G. Attachments

Log entries are often associated with documents or other forms of communication. CoLT allows information to be attached to an entry just as attachments are added to email. The tool should evolve to include seamless integration with document repositories and/or other mission-related systems.

H. Comments

During CoLT requirements development, the concept of external participation in the log spawned a requirement for a non-intrusive means of interacting with an existing entry. The result was an integrated commenting system that works very similarly to blog sites that let users comment to an article.

Commenting allows someone to provide information or request clarification regarding a specific log entry without changing its main content. During pre-deployment CoLT testing, *comment discussions often brought out more useful information than the main entry*. One team plans to update main entries occsaionally based on the essence of the comments, retaining or deleting comments as appropriate. This could be very beneficial when a back office process stores actionable information in a log entry with a future event time, then updates that information as needed until the future event time is reached, e.g.:

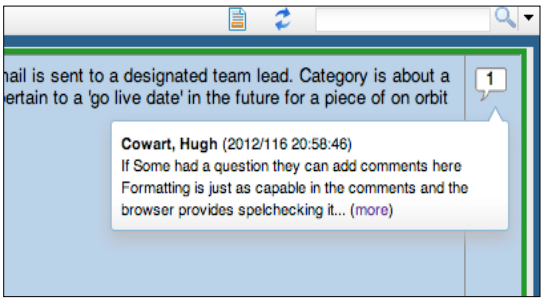

Figure 7 Viewing a Comment

1. Back office team discussion results in a log entry containing actionable information and a future event time.
2. Comments to entry and/or changes in the plan result in updates to the entry.
3. At the future event time, CoLT notifies the operator that this entry pertains to the related activity.
4. Operator edits the entry and/or adds/edits comments to best capture performance of the event.
5. This "cradle to grave" log entry could be "reincarnated" by duplicating it into a new log entry and editing for the next instance of the same event.

I. Notifications

In a traditional log, information is provided by one operator at a time. Any new information, e.g., new entries or updates to existing entries, is introduced by their own action. In CoLT, multiple sources can edit entries and/or add comments (depending on assigned privileges). It is important to advise a given user (particularly the operator on console) of changes made by other parties. CoLT's Notifications feature functions similarly to that seen in Facebook®:

- A user may subscribe to an entry of interest and is automatically subscribed to entries they create, edit, or comment on.
- A user may unsubscribe to any entry.
- Change notifications appear in a pull down menu and include a link to the entry or the ability to pull up a filtered search list returning all entries with related notifications.
- A user may opt-in to receiving emails for each notification. (A future build may offer daily or weekly digests.)

J. Emailing of Individual Entries, Standard Reports, and Search Results

If a user selects an entry and wants to send it to someone, a single click on an envelope icon converts that entry, with attachments and comments, into an email and presents it for addressing and review. A compact dialog makes it very easy to generate and email standard reports (per POIC operational requirements) and customized reports (via CoLT's search engine). Access to global address lists and predefined mailing lists is being improved. Future development will include other ways of streamlining communications processes and eliminating the tedium of manual cut-and-paste.

K. How It Fits Together

While CoLT supports traditional narrative log keeping as practiced for decades of space operations support, its true strength lies in power-boosting the flight team's efforts.

- "Anywhere" access allows back room support, off-duty personnel and management to maintain situational awareness without waiting for daily log export. Search capability across the entire log is no longer limited to on-console personnel.
- Console operators can push information as needed far more easily (Email options, report generation) and sometimes CoLT takes care of it automatically (email alerts based on Flag status).
- Comments capability gathers additional knowledge about a log entry at the same "location" without interfering with the entry.
- Notifications feature keeps users abreast of changes in items of interest.
- Future Events scheme puts accumulated support information for certain activities where it is needed (right in front of the operator), when it is needed (shortly before the activity), and without multiple transfers.

- Future builds could include automatic linkages between CoLT and other communication, coordination, and/or planning systems, reducing the manual keying load on console operators.

IV. Communicating Among Logs Via Social Techniques (by Daniel J. Stevens)

With the advent of Facebook®, Twitter®, and other social media networks, the average person is becoming increasingly comfortable with data sharing and now expects such capabilities. Data storage is shifting from local storage to distributed servers and/or clouds. Applications are moving from local client based software to server side applications with web-based or mobile client interfaces. These paradigm shifts in computing are being embraced by all. NASA could benefit greatly by adopting and adapting these techniques throughout the Agency.

For the first 35 years of the space program, console logs were kept on paper, which required large amounts of physical storage and could be searched only by the human eye. The rise of the PC spawned a gradual evolution to electronic logging. A major challenge with early efforts was that logs might be spread over hundreds of individual Word® files, so searches were quite cumbersome. Thanks to advancing technologies (and some enterprising flight controllers!), most current JSC ISS Flight Controllers use a Microsoft® Word based template with specialized macros. One major function exports log entries to an SQL Database, which allows for a quick and thorough search across all of a specific discipline's logs via a convenient web-based client. The Word® format and SQL database viewer are depicted in Figures 8 and 9.

While this log keeping system has been stable for around 8 years and has worked well, traffic is increasing and it is time to think and act forward.

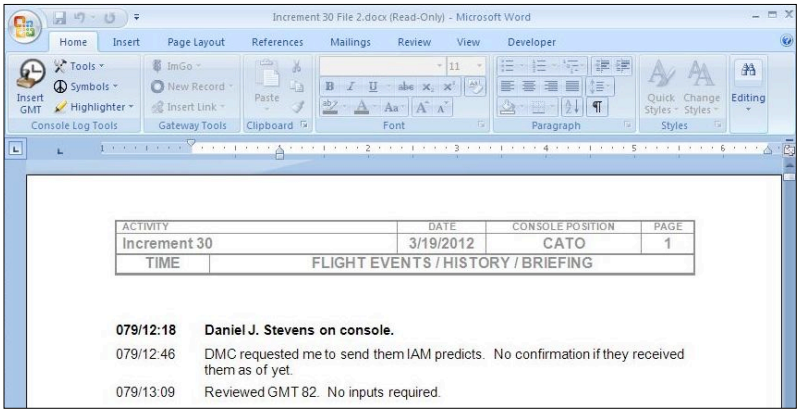

Figure 8 MCC-H Console Log Template

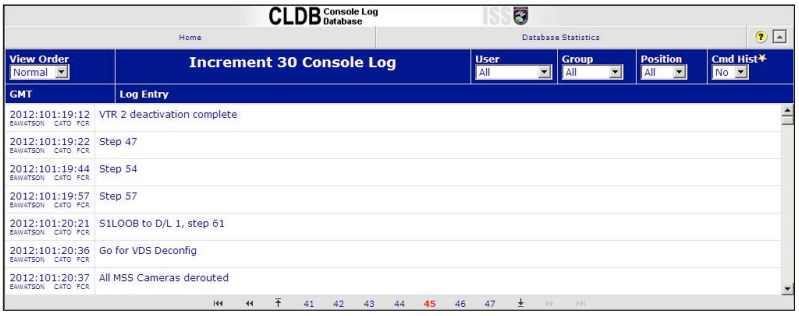

Figure 9 MCC-H Console Log Database

A. Social Media Meets Console Logging

The techniques and mindsets made popular by social media could be harnessed to dramatically improve Log accessibility, cross-discipline communication and the dissemination of information. For example, microblog sites such as Twitter® use the following "tag" protocols:

- To refer to a person, place the "at" symbol (@) in front of their username to reference that person. The post "MyUserID - @JohnDoe see you tonight." will appear in @JohnDoe's stream of posts and will be visible to others who are following @JohnDoe.
- To refer to an object or topic, place a "hash" (#) in front of the item name. The post "MyUserID - is attending the #AIAA #SpaceOps2012 conference this week" will appear in MyUserID's stream. Anyone searching for #SpaceOps2012 will see a stream containing MyUserID's post and anyone else's post referencing #SpaceOps2012. These streams are sometimes called trends when they are receiving a lot of traffic. There are several ways to generate and display lists of top trends.

How can these microblogging techniques be translated for use in the control center? The @ tag would be used to refer to other disciplines' call signs, entire control centers, other organizational units or even a particular person. Examples: @FlightDirector would reference the Flight Director on console at the time; @MCC-H would refer to all disciplines within the MCC-H control center; @Org would be used to reference all users who are members of a particular organizational unit; @myAUID would be used to reference a particular person by their Agency User

IDentifier (AUID). The #tag would be used to reference objects, such as, a piece of hardware or software, procedures, a timelined activity, Flight Notes, Anomoly Reports, and many other frequently used console products.

Using @tags and #tags within a log would cause entries from one log to appear in other logs. For example, Flight Controller X has created a new Flight Note to document XYZ.

- Currently this exchange of information would require:
  - Flight Controller X logs "Created new Flight Note that is ready for Flight Director approval."
  - Flight Controller X calls Flight Director - "FLIGHT, CONTROLER X, new Flight Note created and ready for approval." "CONTROLLER X, FLIGHT, copy."
  - **Flight Director has to search in Flight Note application**, then opens Flight Note, reviews, and approves.
  - Flight Director calls Flight Controller X - "CONTROLLER X, FLIGHT, Flight Note approved." "FLIGHT, CONTROLLER X, Roger, thanks."
  - Both positions create log entries discussing Flight Note's approval.
- Using microblogging techniques, voice loop traffic could be significantly reduced:
  - In Flight Controller X's log - "GMT 300/10:30 @FlightControllerX has created #FNxxxx to document XYZ and it is ready for @FlightDirector approval." Entry would automatically appear in Flight Director's log and would include a direct link to the Flight Note.
  - **Flight Director takes direct link** to Flight Note, reviews, and approves.
  - In a manner similar to a Facebook® comment, Flight Director responds in his log - "GMT 300/10:35 @FlightDirector has approved #FNxxxx." Entry would appear in both logs as a child to Flight Controller X's original entry.
  - Taking this concept a step further, the Flight Note system might be enhanced to generate appropriate @tag and #tag traffic, sometimes automatically, sometimes with dialogs for Proceed/Cancel decisions. This would free up both Flight Director and Flight Controller from logging the event by hand.

This communication would not have to be restricted to just the Flight Control team. A Flight Controller could easily make a similar log entry to contact a Subject Matter Expert (SME) or other personnel in the office, e.g., "@jdoe equipment #XYZ is showing an error X. How should console proceed?" The SME would then receive an email or a client web application popup with the question. Then @jdoe could respond to that question. Likewise, if a SME or other person from the office would like to inform the console of some update or new product, they could then create an entry, "@FlightControllerX the following Anomaly Report is ready for closure: #ARxxxx (Title of the AR)".

With the stream and trends concept, each user would have access to a list of "popular" references. For example, when several controllers make multiple log entries about an observed failure, the associated #tag would quickly become visible to all users as a top trend. (See Figure 10.) This would allow office members to keep up with important issues as they evolve, without requiring console operators to send out additional emails and with less third party reporting.

Streams could dramatically simplify shift handovers. Currently, the Flight Director calls each control center and each discipline individually for reports on significant events of the prior shift and upcoming activities on the timeline. If @tags are adopted by all team members, the Flight Director could initiate a log entry to the entire team and each position would respond with their status.

- "@ISS ready for #Handover items"
- "For #Handover, @FlightControllerY reports all systems functioning nominal and nothing on the timeline."
- "For #Handover @FlightControllerY needs @ISS to review #SomeProduct."

All logs would show the original post from the Flight Director and all replies as children of the original post. In all likelihood, there would still be a need and desire for some handover-related voice traffic, but the existence of the stream would mean: a) less additional log keeping to document the content of the handover, b) shorter voice handover briefing and c) talking points are visible throughout and after the briefing (very handy if someone is working an urgent issue and has to miss the briefing).

B. What Would the Tool Look Like?

The first step in this shift would be to move away from file based Microsoft® Word Templates to a web client which accesses a more robust database. This move alone would reduce the support burdens and redundant storage associated with use of local clients. The end product would resemble a chat window with all the added functionalities of the current Visual Basic® Word Macros (e.g., Timestamps, Hyperlinks to Flight Products). The user would define the number of old entries to view and the client web interface would only maintain those entries in the viewable screen. Figure 10 is a conceptual image of this type of Console Log.

In this example, previous shift entries have a gray background. Colored highlighting identifies external entries, which may be hidden or revealed via the "X" box. The orange "active box" shows which entry the user has selected for editing, commenting or other actions. Note the "Trends" frame showing which items are receiving the most attention.

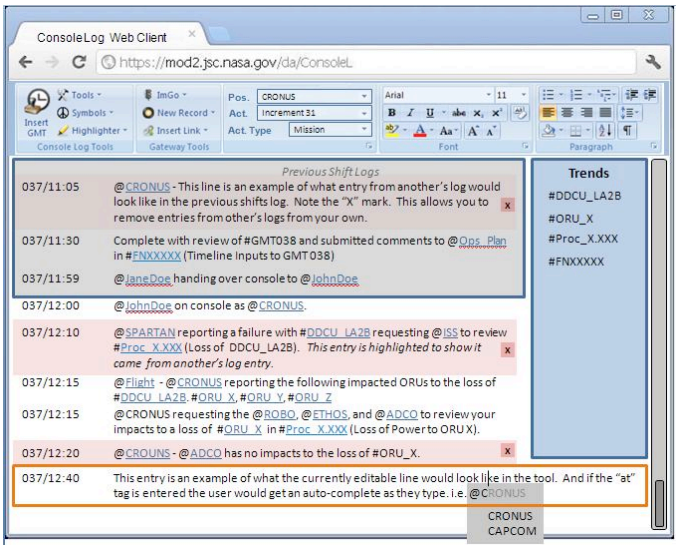

Figure 10 Console Log with @tag and #tag Features

C. Autocomplete

Since this system would be working with an established list of Discipline names and userIDs, it is possible to have autocomplete functionality on @ and # tags. As shown at the bottom of Figure 10, if a user began typing "@C", a window of autocomplete options would appear with the closest completion selected. The user could press Enter to autocomplete @CRONUS or press the down arrow followed by Enter to autocomplete @CAPCOM. This could also be extended to AUIDs, Procedure numbers or various other Flight Product references.

D. Unique Considerations

The Flight Control environment has some more stringent performance needs than better known forms of social media.

- The client side interface must be able to work off-line in the case of network outages and then synchronize with the Database when connectivity is restored.
- Access privileges must be managed with extreme care. Sensitive material may exist within a log entry that must not be shared outside of a specific group. For example, the Biomedical Engineer (BME) reports to the Flight Director via an @tag post in the Biomed log that a Crew member has fallen ill. Since medical information is sensitive, BME would need to be able to turn protected posting on so that his post would only be available to his group and to anyone he references in that post. Protected posts would not be visible to anyone else, e.g., via #tag or @tag searches or following. For posts that are not sensitive, BME could turn protected posting off. Administration of protected posting merits much additional discussion.

E. Where This Might Lead

This concept could port very easily to mobile devices and even other control centers around the world. The flexibility of server side processing would allow for thin-client applications and ease of distribution. "Anywhere" access to detailed log information would make it easier for a manager or On-Call flight controller to maintain situation awareness or answer calls for assistance. (ISS core system flight control operations have been streamlined with multiple console positions often consolidated into a single Operator who is trained to safe malfunctioning systems until a certified Specialist can resolve the issue. With log details literally "in their hand," a remotely located Specialist would be in a much better position to talk the Operator through a quick resolution or to assess the need to go on-site.)

By harnessing microblogging's simple yet novel tools to log keeping, NASA would "grease the machines" for the next generation of flight controllers who have grown up Tweeting and Facebooking. Given that our forebears in space operations did amazing things wielding slide rules, just imagine what tomorrow's controllers could accomplish with tools built around a technique that is as natural to them as breathing. Those of us currently in the business might also find some fresh air!

V. Communications Dashboard (by David W. Scott)

Over the years, electronic messaging and documentation have lightened the paper burden and made searching easier but have spawned the unintended consequence of a text explosion! (Just because we *can* generate more words more quickly does not mean we should; nonetheless, we do.) Much effort goes into transferring text from one communications or planning system to another. With the increased pace of operations and addition of more console operators, all kinds of voice discussions wind up being repeated to bring folks "up to speed" and/or get transcribed or summarized into text and injected into the text explosion, making the blast even bigger. At times, the barrage of background can cause the foreground to go underground.

If we could integrate presentation and exchange of flight control communications in intra-center and/or inter-center contexts, we might drastically reduce repetition of both text and voice information, thus empowering flight controllers to manage more operations with fewer words and with greatly reduced stress. These techniques could apply to a wide variety of real time control environments, e.g., aircraft flight control, factory operations, military operations, 911 call centers and inter-agency responses to large-scale disasters.

A. Dashboard Components

A communications dashboard could: a) use Social Networking Service (SNS), micro-blog, and/or fortified text chat methods to supplement voice loop communications and b) display various information streams in an arrangement that requires less human processing to grasp the "big picture" of operational events and context. Typical SNS applications, such as Friends, games and free-form chatter, would naturally give way to more operations-oriented services, e.g.:

- Console log contents.
- Text chat channels analogous to traditional voice loops and/or text chat streams created "on the fly" and dedicated to work a particular issue. Text chat *is* a bit slower than voice, but when something is "said", it remains visible. New or existing participants can catch up or keep up with the conversation just by reading, and the conversation is self-documenting.
- Mapping capability with a quick, intuitive interface so that console log entries can be kept brief by providing links to access underlying details stored in text channels or streams. This also reduces duplication of cut and pasted text.
- Consolidated indexing of mapped text conversations and streams for easy browsing or searching.
- Individual/Group "hailing" capability that would advise operators of someone else's need to communicate. Unlike a voice loop call which may not be heard or have to be repeated, a dashboard hail would persist visually until acknowledged.
- Announcement capability. As for hailing, the advantage over voice loop transmission is visual persistence.
- High-level status indicators, e.g., Comm-link status with ISS, time to major events, and other items of general and/or position-specific interest.
- Scrollable status updates from manual (console operator) and/or automated (telemetry monitoring) sources.

These services could be displayed integrally to slash redundancy among multiple threads. Also, if people look at the same instance of a conversation instead of creating separate accounts of it, there will be fewer disconnects. Those that do appear will be more obvious and can be quickly resolved.

B. How Social Media Techniques Could Improve Information Display

Social media methods present intriguing potential for powering a communications dashboard. Well-known schemes with a personal, professional, or institutional focus (e.g., Facebook®, LinkedIn®, and Yammer®) are clearly not suited for direct application to real time mission operations. If we examine <u>how</u> these services engage users, we find some visually simple yet extremely powerful system behavior. For example:

- A few very small icons, representing events of keen interest, change to a brilliant color when those events happen, indicating how many instances have happened and easily attracting the operator's attention. A simple click on the icon summarizes the new events and a simple highlight-and-mouse-release opens a specific instance's details.
- Multiple streams of dynamic information appear in frames proportionate to their "importance" (based on assumptions about a typical user and some user-configurable preferences). When moving about the workspace, some frames change while others remain the same, just as scenery outside our car changes as we drive, yet certain items in the car stay in place, though their values may fluctuate.

C. Conceptual Mockup

Figure 11 illustrates the concept of integrating several text streams, a console log, hailing, and status indicators to present a "big picture" to the console operator. A description of related voice loop traffic appears above the mocked-up display window.

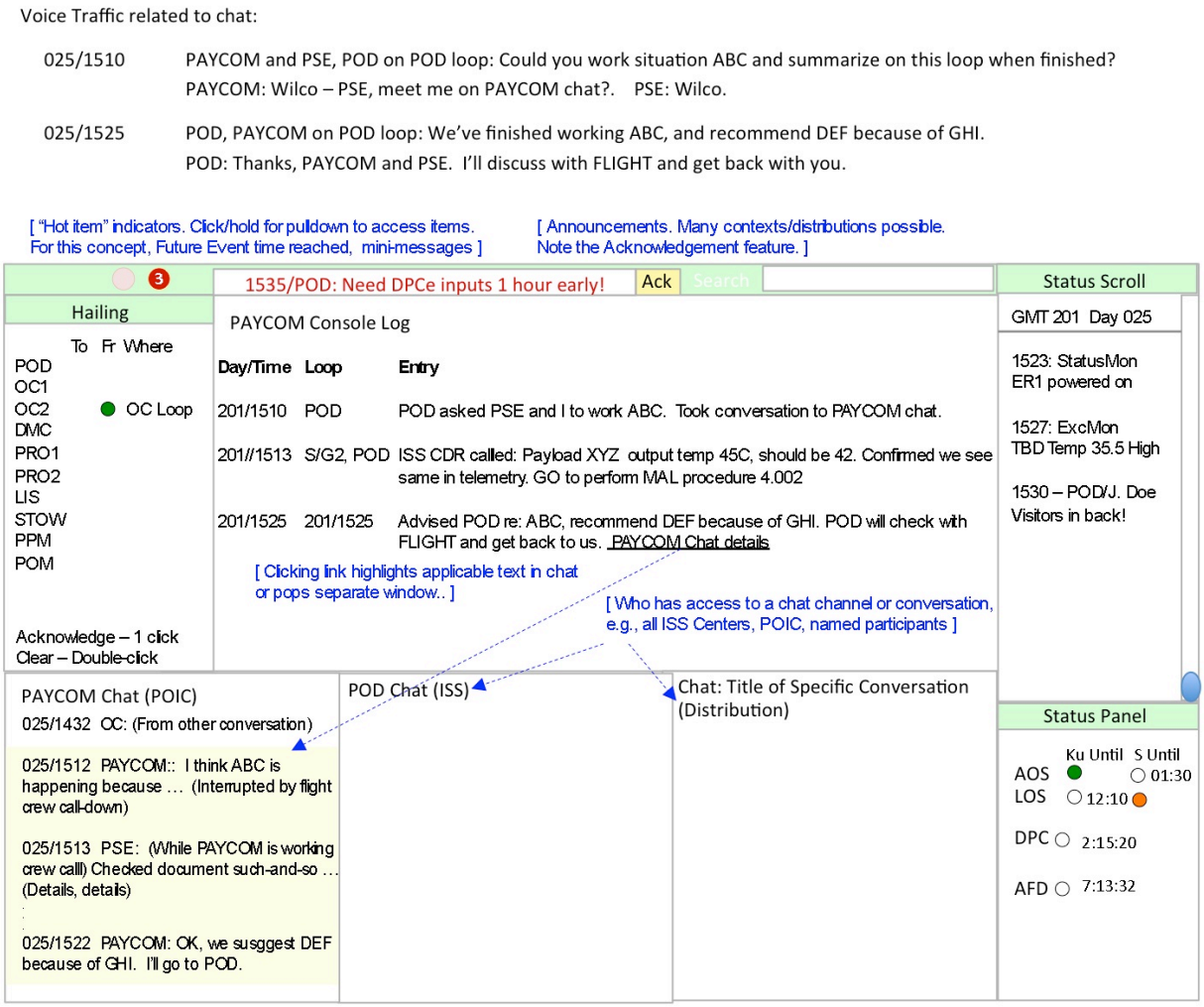

Figure 11 Communications Dashboard Concept

An added benefit of building a communications dashboard around a SNS model is that, since SNSs are common in our personal and office cultures, training to use one in an operations setting would be relatively easy.

D. What Fain Would a Communications Dashboard Provide for Voice Communications?

With more robust text communications and display of appropriate status items, some voice traffic from announcements and detailed discussions could be moved to dashboard media. Voice is vital but is serial in nature. Text is semi-parallel and graphics is parallel, and both have the advantage of visual persistence. If the most essential voice traffic is retained due to time and/or criticality and underlying details are worked elsewhere with less repetition, aural saturation and competition between audio and visual stimuli are reduced. This *will* lower stress on the console operator.

A communications dashboard lab could be established to:

- Develop and demonstrate dashboard capabilities and their effects.
- Allow flight controllers to a) experiment with new representations and communication methods to see how they might like to evolve their operations concepts and b) generate ideas for additional capabilities and features.
- Develop ways to access appropriate existing mission support software, e.g., CoLT, via either their traditional interface or a dashboard. Existing SNS schemes are essentially entities unto themselves. Applications running inside the SNS are not available outside of it. The author knows of no SNSs that integrate with real time operations systems.
- Explore the effects of various layouts in a mission operations context and draft guidelines for console operators to optimize their setups.
- Qualify and quantify what voice traffic might be moved to dashboard media.

VI. Conclusion

Innovation is the art of the "adjacent possible."[4,5] This paper offers three innovative, web-based technology approaches to console log keeping (one in initial deployment, two in concept stage) that could reduce stress on space flight controllers while increasing their productivity. The three approaches are:

- Console Log Tool (CoLT) - A recently deployed log keeping application at MSFC's POIC that provides "anywhere" access, comment and notifications features similar to those found in SNSs, report generation and email transmission at several levels of automation, and a future events tracking capability that could foster appropriate bidirectional communication via the log. Future versions might include connections to other ISS communications and/or planning systems.
- Cross-Log Communication via Social Techniques - A concept being explored at JSC's MCC-H that would use microblogging's @tag and #tag protocols to make information/requests visible and/or discoverable in logs owned by @Destination addressees. Some #Object designations would also serve as links to products/locations in other systems, e.g., Flight Notes.
- Communications Dashboard - A MSFC concept that proposes a SNS approach to integrate console logs, text chat streams analogous to voice loops, text chat streams dedicated to particular conversations, generic and position-specific status displays/streams, and a graphic-based hailing display.

These approaches can reduce flight controller stress because of these goals and principles:

- Information is represented in fewer places, so there is less opportunity for disconnects.
- Multiple parties look at the same instance of information, reducing disconnect opportunities even further.
- Manual cut/copy-and-paste operations are minimized.
- Information is discoverable when needed and disappears when it is not.
- Less voice traffic is needed because the log keeping system can now carry many conversations with excellent retention, organization, and visual persistence.

References

[1]Scott, David W., "Using Social Media in Engineering Support and Space Flight Operations Control", 2011 IEEE Aerospace Conference, http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5747662

[2]Scott, David W., "Using Web 2.0 (and Beyond?) in Space Flight Operations Control Centers", SpaceOps 2010, 210896, http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100021135_2010020435.pdf

[3]Scott, David W., "How to Boost Engineering Support Via Web 2.0 (Seeds for the Ares Project ... and/or Yours?)", Unpublished (available on request from scotty@nasa.gov)

[4]Johnson, S., "The Genius of the Tinkerer," *The Wall Street Journal*, September 25, 2010, http://tinyurl.com/76aeqqd

[5]Johnson, S., "Where Good Ideas Come From," http://www.youtube.com/watch?v=Mb0ssmoXG1I