

19<sup>th</sup> International Command and Control Research and Technology Symposium  
“C2 Agility: lessons Learned from Research and Operations”

**Workstation Analytics in Distributed Warfighting Experimentation:**

Results from Coalition Attack Guidance Experiment 3A

Topics

Topic 3: Data, Information, and Knowledge

Topic 4: Experimentation, Metrics, and Analysis

Name of Authors

Richard McCourt, Natalie Nakhla, Irene Collin, Alan Hill  
Centre for Operational Research and Analysis  
Defence Research and Development Canada  
Ottawa, Ontario, Canada

Point of Contact

Richard McCourt  
Warfare Centre Science Team  
Canadian Forces Warfare Centre  
3701 Carling Avenue, Ottawa, Ontario, Canada  
richard.mccourt@drdc-rddc.gc.ca

# Workstation Analytics in Distributed Warfighting Experimentation:

Results from Coalition Attack Guidance Experiment 3A

By

Richard McCourt, Natalie Nakhla, Irene Collin, Alan Hill  
Centre for Operational Research and Analysis  
Defence Research and Development Canada  
Ottawa, Ontario, Canada

## Abstract:

The Canadian Forces Warfare Centre (CFWC) uses state-of-the-art joint battle labs (JBLs) to test the integration of command and control (C2) systems from all of the Canadian Armed Forces services, as well as the integration of international and experimental C2 systems. Typically, this is done through large-scale distributed warfighting experimentation involving several geographically dispersed and networked JBLs. One such series of experiments is the Coalition Attack Guidance Experiment (CAGE). The third event in this series (CAGE 3A) was a multi-national human-in-the-loop experiment (including Australian, British, and Canadian operators) to investigate the use of specific tool suites in the planning and coordination of cross-boundary, joint and combined fires. Participants in the experiment played their corresponding operational roles in the interconnected command posts, and the performance of the people, processes, and tools was observed.

The CFWC developed a customized automated data collection tool to capture operator workstation activity. The tool was developed to improve the fidelity and detail of the collected data, to augment the data with information which could not be collected by human observers, and to collect data unobtrusively. The tool used a combination of hooking, packet sniffing, and virtual network computing to record screenshots, screencasts, and details of all typed text, mouse clicks, and network activity, e.g. voice over internet protocol. CAGE 3A was the first time this tool was successfully distributed across participating nations in a multi-national experiment. At the end of the experiment the data was pulled back from other nations for post-event analysis. Workstation analytics from this raw data set helped to assess the usage of applications in the provided C2 tool suite, to monitor adherence to pre-defined fires processes, and to mine for emergent fires processes. Additionally, workstation metrics from this experiment will be used as a benchmark for future experiments.

## 1.0 Introduction

The Canadian Forces Warfare Centre (CFWC) in Ottawa uses state-of-the-art joint battle labs (JBLs) to test the integration of command and control (C2) systems from all of the Canadian Armed Forces services, as well as the integration of international and experimental C2 systems. Typically, this is done through large-scale distributed warfighting experimentation involving several geographically dispersed and networked JBLs. One such series of experiments is the Coalition Attack Guidance Experiment (CAGE)[1]. The third event in this series (CAGE 3A) was a multi-national human-in-the-loop experiment (including Australian, British, and Canadian operators) to investigate the use of specific tool suites in the planning and coordination of cross-boundary, joint and combined fires. Participants in the experiment played their corresponding operational roles in the interconnected command posts, and the performance of the people, processes, and tools was observed.

Typically, in past experiments, data were collected from application log files, Personal Storage Tables, the chat server database, and surveys. Any information that was missed by these disparate data sets was left to human observations. These observations were extremely useful; however, they did not provide a “ground truth” recording of what the participants did on each workstation. Detailed recordings of workstation activity can help to recreate and understand specific events and processes. These detailed recordings act like a microscope or telescope, capable of exposing greater detail in each event or process, e.g., monitoring adherence to existing tactics, techniques, and procedures (TTPs) and potentially revealing emerging ones. It would have been difficult for human observers to collect detailed workstation activity on a single computer; collecting on all computers would have been unrealistic.

The need for detailed collection and analysis of workstation activity, i.e. workstation analytics, is not uncommon for large and complex organizations. Computer monitoring software is used to monitor and police employee workstation usage [2,3,4,5], and full service workstation activity tracking and reporting solutions exist [6,7], but several requirements drove the decision to develop custom data collection software at the CFWC.

The JBLs at the CFWC operate in a classified state and it is important to only install trusted software on these systems. Currently, many of the commercially available computer monitoring software solutions are proprietary, licensed, and closed-source. Not surprisingly, computer monitoring is also a field riddled with software designed with malicious intent. However, even if a company were proven to be trustworthy, the procurement process could be lengthy and require training. Additionally, asking partner nations to procure the same software would be extremely difficult.

One notable open-source project existed prior to development [8], however, it was a fairly simple program. The publisher’s description of the software states that it is ideally suited “to keep watch on childrens [sic] (two to seven years) activity on the computer” [9]. Furthermore, it had several infeasible restrictions, including a required connection to a Moldovan web service to enable remote control of the software.

Finally, in large events there can be hundreds of workstations to monitor. In a recent exercise, there were nearly 200 workstations to monitor over a three week period. The large amount of classified data collected from events like this cannot easily be moved off of the experimentation network where it was recorded. In past events, it has proven easier to leave the data on the network and process it in place. Also, it is easier to perform the data analysis in-house with the resident operational research team embedded in the CFWC.

For these reasons, the CFWC developed Thea, a computer monitoring tool named after the Titan goddess of sight. Thea is required to:

- Be unobtrusive;
- Have a small/non-existent network footprint;
- Record details of every mouse click;
- Record details of all keystrokes;
- Have the ability to record video screen captures;
- Have the ability to record screen captures on each mouse click;
- Have the ability to record Voice over Internet Protocol (VoIP) communications;
- Be controlled from a central server on the same network;
- Store data locally; and,
- Have no restriction on the number of computers that can be monitored or the size of data to be stored (other than physical restriction imposed by existing hardware).

Thea has gone through several name changes and development iterations over the past two years. It has completely transformed from the computer monitoring tool that was originally built at the CFWC[10]. The different versions of the tool have been used in four major events at the CFWC. CAGE 3A was the first time this tool was successfully distributed across participating nations in a multi-national experiment. At the end of the experiment the data was pulled back from other nations for post-event analysis. Workstation analytics from this raw data set helped to assess the usage of applications in the provided C2 tool suite, to monitor adherence to pre-defined fires processes, and to mine for emergent fires processes. Additionally, workstation metrics from this experiment will be used as a benchmark for future experiments.

## 1.1 Aim of Paper

The computer monitoring techniques employed by Thea are not new; however, the application of these techniques to large-scale distributed warfighting experimentation is new. The use of Thea in CAGE 3A was one example of this, but during the experiment, Thea was still under development and monitoring computers outside of Canada had not been tested. For this reason, deployment of Thea into partner nations was limited to a small, close-watched set of computers only in the Australian JBL. Thea was fully deployed on Canadian workstations, but due to scheduling and planning issues, Canadian participation was limited to support and experiment control.

For these reasons, the data that Thea collected during CAGE 3A does not fully represent actual military warfighting TTPs. Therefore, the aim of this paper is to discuss the usefulness of Thea data in C2 experimentation, rather than reporting actual experimental results found.

## 2.0 Collection Method

Thea continues to undergo further development: adding new features and fixing existing bugs. In its current state, Thea collects:

- Details of all mouse clicks.  
What application and window was clicked in, which user did the click, when the user clicked, what button or menu was clicked (if applicable), and which mouse button was clicked
- Details of all keystrokes.  
What application and window the text was typed/cut/copied/pasted into, which user typed the text, and when the user finished typing (marked by a mouse click or change in active window)
- Tracking of mouse movement.
- Screenshots are captured after every mouse click.
- Details of Windows shortcut keys used.  
What application and window was active when the shortcut was used, what keys were pressed, what action was taken, which user pressed the keys, and what time they were pressed.
- Video screen capture for full playback capability.
- Network packet capture to and from the recorded workstation.

Like many other computer monitoring applications, Thea uses a computer programming technique known as hooking to intercept function calls, messages, or events passed between software components, and run mouse click and keystroke logging software during each interception. Video screen captures are recorded via remote framebuffer protocol through a virtual network computing application. Lastly, network traffic is captured in a local packet capture (pcap) file using the WinPcap application programming interface (API).

All of the collected data, except for the packet data, is stored locally in a SQLite database on the workstation where Thea is collecting. A separate SQLite database and pcap file are created each time Thea recording is started and stopped (typically once per day per workstation). Packet data could have been collected at a centralized location (i.e. on servers and routers, where most network monitoring software run). However, by monitoring and storing network traffic locally, Thea (an in-development tool) will not affect key network infrastructure if a problem with the tool arises. Moreover, packet data is automatically meta-tagged with workstation hostname and the corresponding user, in addition to the network addresses attached to each packet. Finally, capturing packets through Thea at the workstation network interface gave CFWC scientists and data collectors a very simple way of creating custom data collection plans for each workstation.

Post exercise, when the network can afford the bandwidth, each of the database and pcap files are pulled to a central location. Each of the SQLite databases are converted and stored in one MySQL database and the pcap files are parsed to extract call information from the specific communications applications used during the experiment. This extracted data is then stored in the same MySQL database.

With all the collected data in one place, the database is queried for the desired information, as demonstrated in the following examples.

## 3.0 Description of Data and Sample Results

As previously mentioned, mouse clicks, keystrokes, and VoIP calls are all captured for each monitored workstation. From the collected data come many analysis possibilities, some of which are:

- Tracking application usage to determine the distribution of operator time in a given tool suite;
- Process mining to determine if operators are adhering to trained procedures, or if any insightful behaviour emerges when they go off-procedure;
- Social network analysis considering all forms of communication (i.e. e-mail, chat, VoIP) all from a single database;
- Behavioural biometrics (e.g., keystroke and mouse dynamics) to estimate participant workload, fatigue, boredom, etc.;
- Detecting and tracking experimental trends in activity or language.

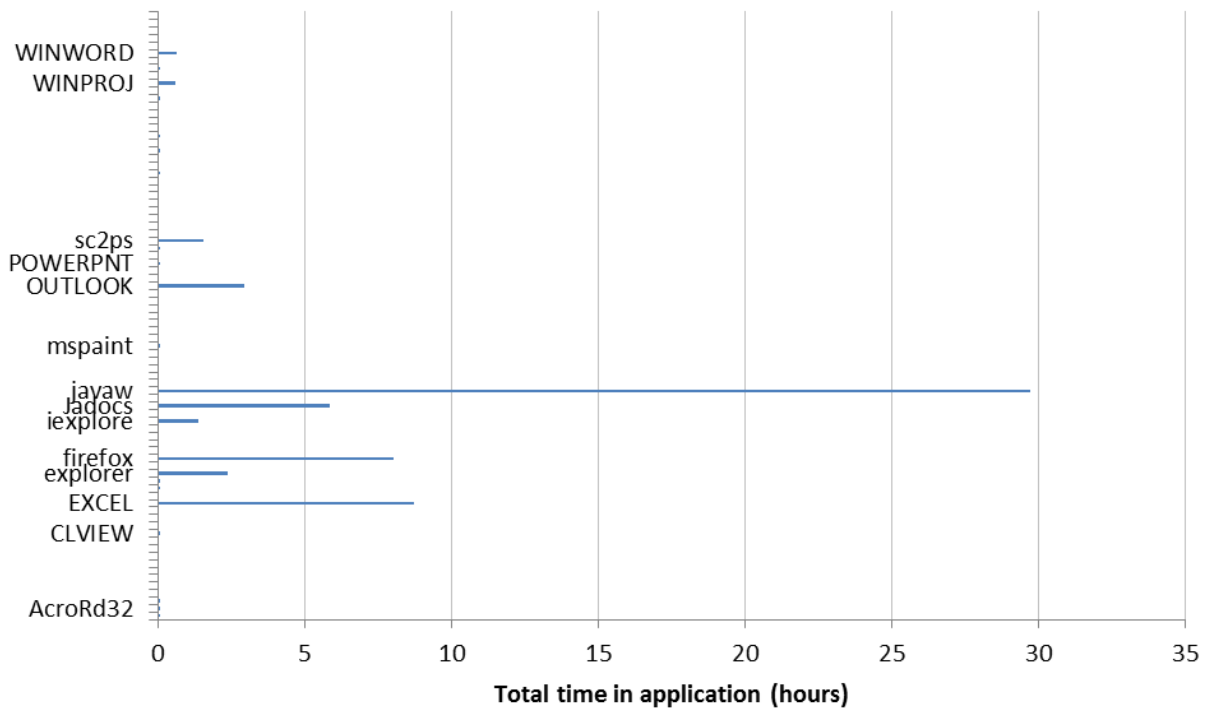
To demonstrate some of these capabilities, a sampling of the results is presented in this section. These results are derived from data collected during CAGE 3A. During this experiment, Thea monitored 29 users in Australia and 20 users in Canada over a two-week period. In comparison to other events at the CFWC, CAGE 3A was fairly small and, therefore, a perfect opportunity to test out this data collection capability.

### 3.1 Application Usage

#### 3.1.1 Total Application Usage

In this example, we consider the application usage by the participants. Note that the results provided are only for the workstations which were monitored with Thea, which was all of the participants in Canada, approximately 50% of the participants in Australia, and none of the participants in the United Kingdom.

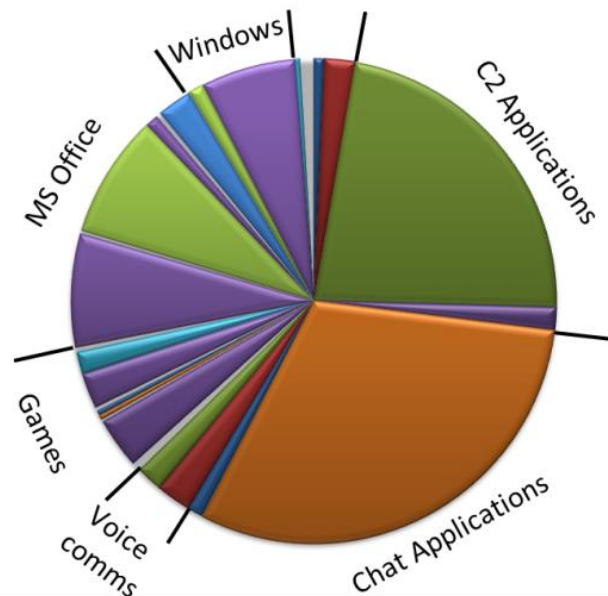
Figure 1 shows the total application usage for the Canadian experiment director (EXDIR). This result is of significant importance since one of the Canadian objectives of the experiment was to capture all aspects



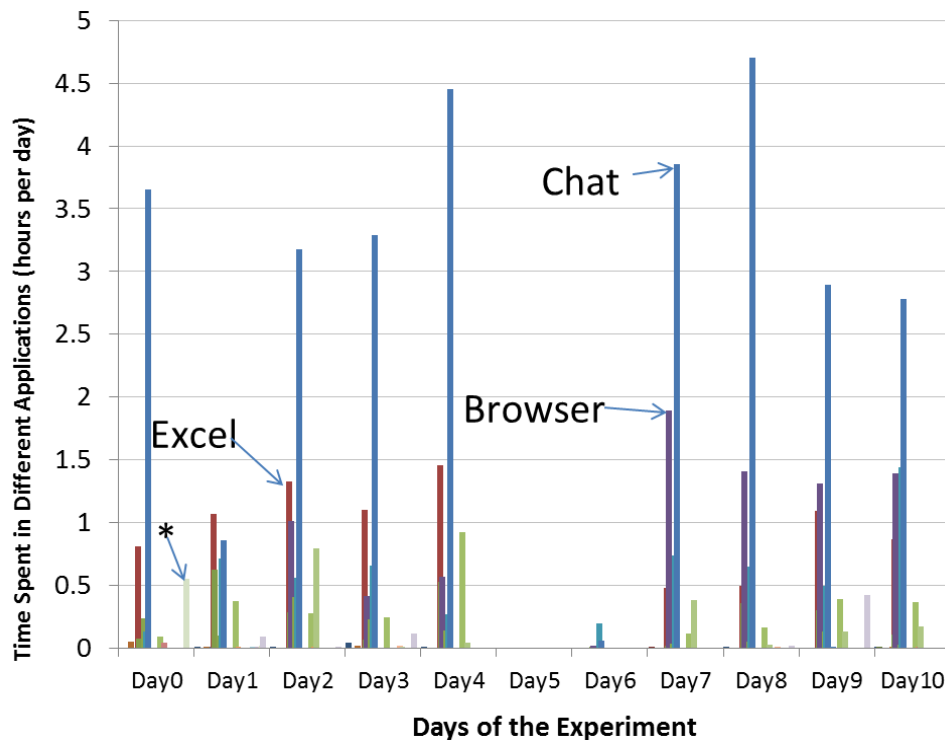
**Figure 1 – Total time that the experiment director (EXDIR) spent in each application**

of Experiment Control (EXCON), including which tools they used and what their processes were. As seen, the results demonstrate that the EXDIR relied heavily on the chat program (labeled “javaw”) to facilitate experimental coordination. He spent a total of 29.7 hours, or 47.9% of his time with the chat application active. This result confirms what the observer-analysts suspected while monitoring EXCON during the event.

Next, Figure 2 provides the percentage of total time spent in all of the applications by all monitored participants. The different colors indicate the various applications used by the participants. Note that the legend identifying the actual applications is omitted due to the sensitivity of the information. As seen from the figure, most of the time was spent in chat applications (32%), which was the main tool used for coordination during the event. This graph provides insight into what tools are essential for the next instalment of the CAGE series (CAGE 3B) and can be used in the planning of this next event. It is important to realize that just because an application was not widely used by all of the participants, it does not mean



**Figure 2 – Percentage of total time spent in different applications for all monitored participants**



**Figure 3 – Application usage (in hours) per day for EXDIR**

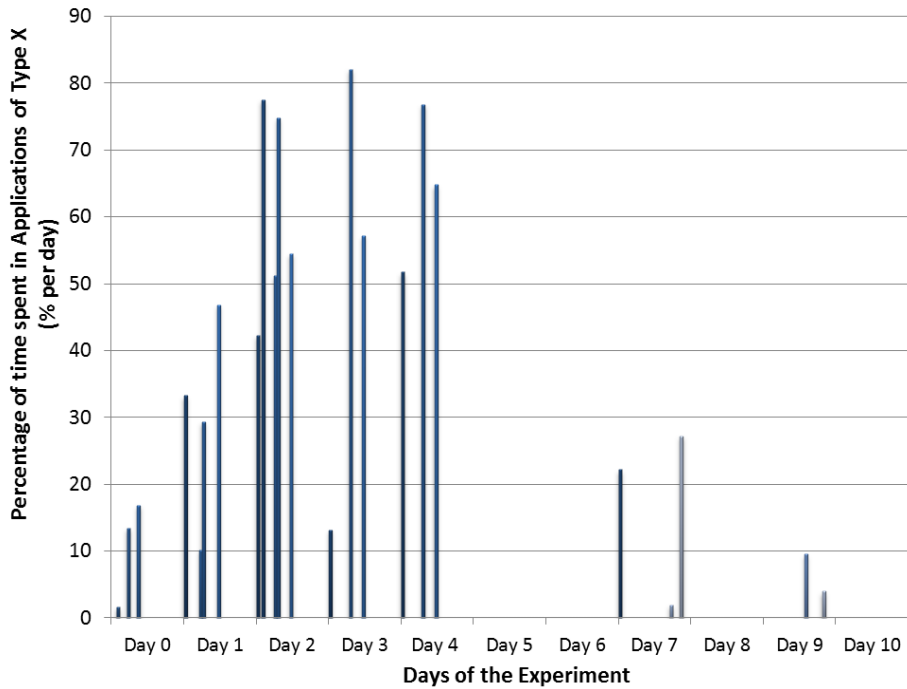
that it was not vital to the event. For example, one of the smallest slices in the plot (0.17% usage) was the Thea manager, which was essential in capturing the data. Thus, plots like the one in Figure 2 must be interpreted carefully.

### 3.1.2 Application Usage Over Time

In this example, we consider application usage over time. This helps detect experiment trends, participant behaviours and the battle rhythm of the event. Figure 3 demonstrates the application usage over the course of the experiment for the EXDIR. As seen, the EXDIR spent the majority of his time in the chat program in each day, and in the first week, Excel was the second most used program. In the second week, Excel usage dropped and a web browser took its place in level of usage. This change corresponded with a change in coordination tools that were used during the experiment.

Anomalies in this graph can cue inquiries about their activity. For example, in Day 0, the application marked by '\*' was Microsoft Project, and took up approximately 10% of the EXDIR's time, but had no usage throughout the rest of the experiment. This corresponded to a brief, last minute effort at the beginning of the experiment to try and use Project instead of Excel to coordinate the event.

Next, application usage over time for a certain group of participants is shown in Figure 4. The results show the usage of a specific proscribed type of application (Type X). Each bar in the graph represents the usage of Type X applications by a specific participant on a particular day. Usage of Type X application increased throughout the first week. At the end of the first week, it was observed that the participants became aware that the analysts were monitoring and recording their application usage. During the second week, the usage of application of Type X decreased dramatically.

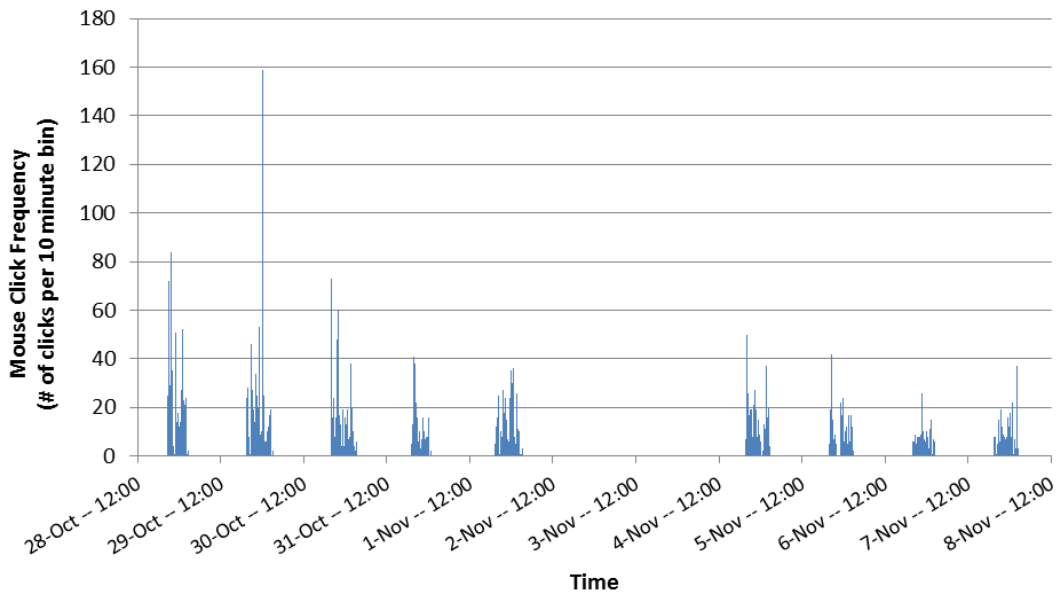


**Figure 4 – Decreased usage of application Type X after participants informed**

### 3.1.3 Mouse Click Frequency

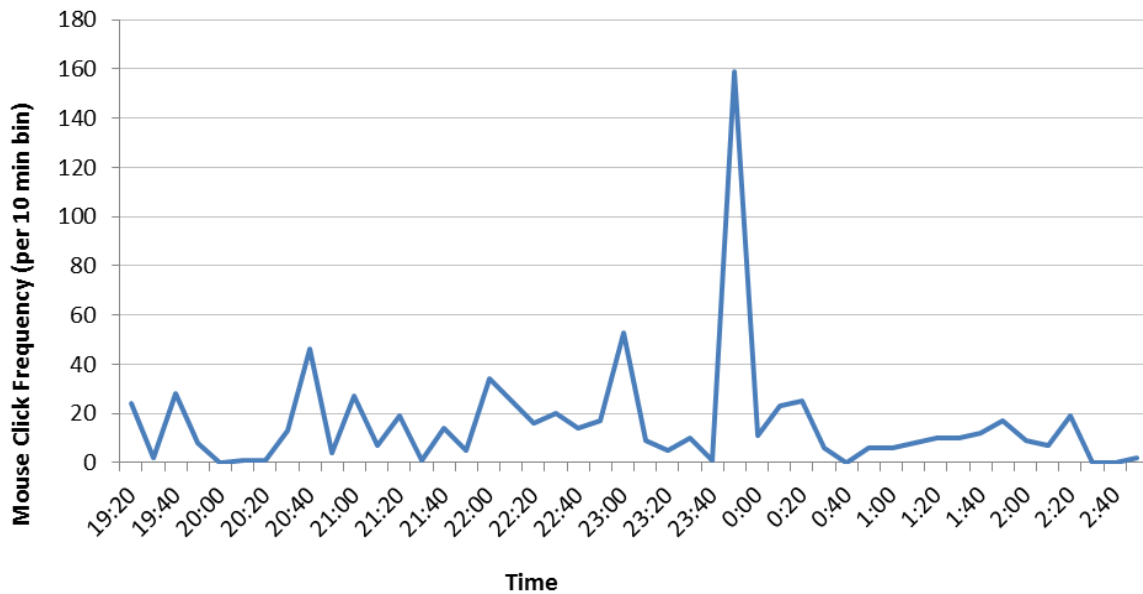
In this next example, the frequency of mouse clicks is considered. Mouse click frequency is one of many behavioural metrics that can be used to infer the participants’ state of being or their activity level. In this case, the data could be used to indicate a level of effort within a specific application, or simply which tool the participant was actively clicking in.

Figure 5 shows the frequency of mouse clicks over time for the Canadian G3 Ops while in the chat



**Figure 5 – Mouse click frequency for the Canadian G3 Ops in the chat program for the entire event**





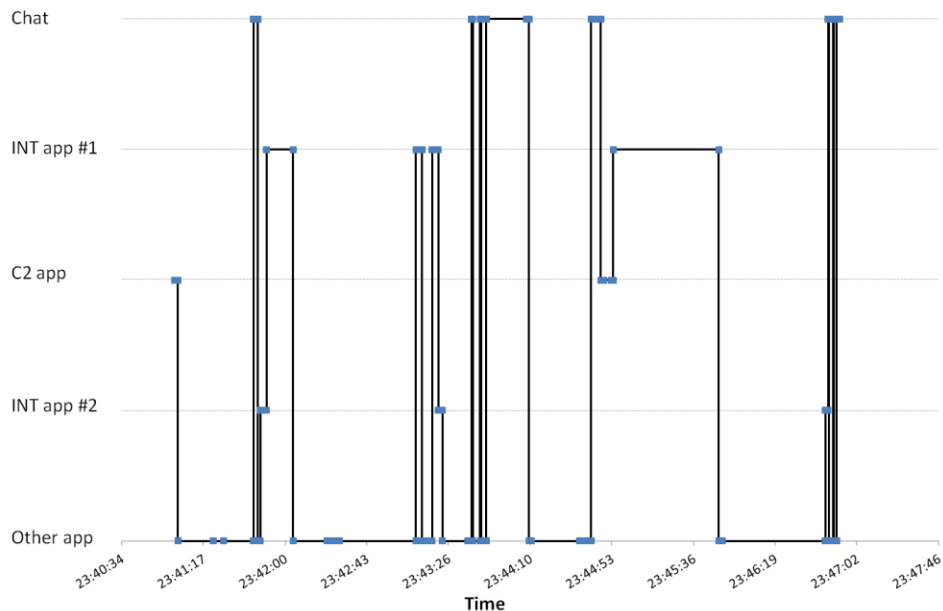
**Figure 6 – Mouse click frequency for the Canadian G3 Ops in the chat program on Day 1**

application. As seen, the G3 Ops clicked the most in the chat program on Day 1 of play. Day 0 was a training day, and Day 1 was the first real day of play. During this initial phase of the experiment, there would be a large amount of coordination between different chat rooms and configuration of the chat application itself. It is important to note that a high volume of mouse clicks does not imply that the G3 Ops necessarily typed a high volume of messages; it just means that the chat program was open and the mouse was clicked within that window. This could imply that the G3 Ops had chat open and was reading a large number of messages which were being passed. Further analysis can be performed by combining mouse click activity with other Thea data, for example, combining with typing speed and mouse movement (among other metrics) to infer operator fatigue [11].

Figure 6 presents a zoomed in view of the G3 Ops mouse clicking activity for Day 1. As seen there is a large spike in the plot. This corresponds to the 10 minute interval between 2350Z and 0000Z, during which the chat server went down numerous times. Every time the chat servers went down, all the chat rooms had to be re-opened and the windows re-arranged. This explains the high volume of mouse clicks during that time period.

### 3.1.4 Process Detection and Use of Tools

In this example, we consider the application use and process flow for the Canadian Intelligence officer (CA.INT). Four particular applications, which are referred to as Chat, INT app #1, INT app #2, and C2 app (real names are omitted due to the sensitivity of the information), are very important for the CA.INT to accomplish his tasks. The Thea results from monitoring the CA.INT’s workstation show that during week 1 (Condition 1), only Chat and the C2 application (out of the four applications of interest) were used. During week 2 (Condition 2), all four applications were employed; however, with quick back and forth movements between the four tools. Cross-referencing with the analysts’ observations indicated that during week 1, the coalition participants were not using the two INT applications, and during week 2, they were using all four tools. The quick back and forth between tools for the CA.INT occurred because there were many inconsistencies between some of the applications regarding some relevant operational information; as a result, the CA.INT spent a lot of time searching through the two INT applications for



**Figure 7 – Process detection for an individual participant through tracking of the active application**

the correct information, and when it was not found, the CA.INT went back to Chat to cross-check and then into the C2 application to enter new information. Figure 7 presents a sample of the results (for an illustrative time window) which demonstrates this fluctuating behaviour. Note that points at the zero mark represent mouse clicks in application of type 'Other'.

## 3.2 Participant Text Analysis

### 3.2.1 Description of Editable Text

From each user, Thea was able to collect editable text data. This information contains every word, number and character typed, copied, or pasted by the user, and thus, forms a picture of his actions and communication patterns throughout the experiment. The information can be sorted by time period, so that it is possible to see all of the issues being discussed or worked on during any time frame, whether months, weeks, days, hours or minutes. Moreover, it is relatively simple to determine which issue is being given the most attention at any time, as well as which users are in frequent contact. Even the nature of the user's role can be observed through these data, with some user data being more numerical (as in target positions), some more verbal, some quite brief (as in orders) and some more conversational (as in more detailed discussions regarding options for approaching a problem). Further, the use of certain terms or phases may be indicative of the user's level of satisfaction or frustration.

### 3.2.2 Participant Language Use

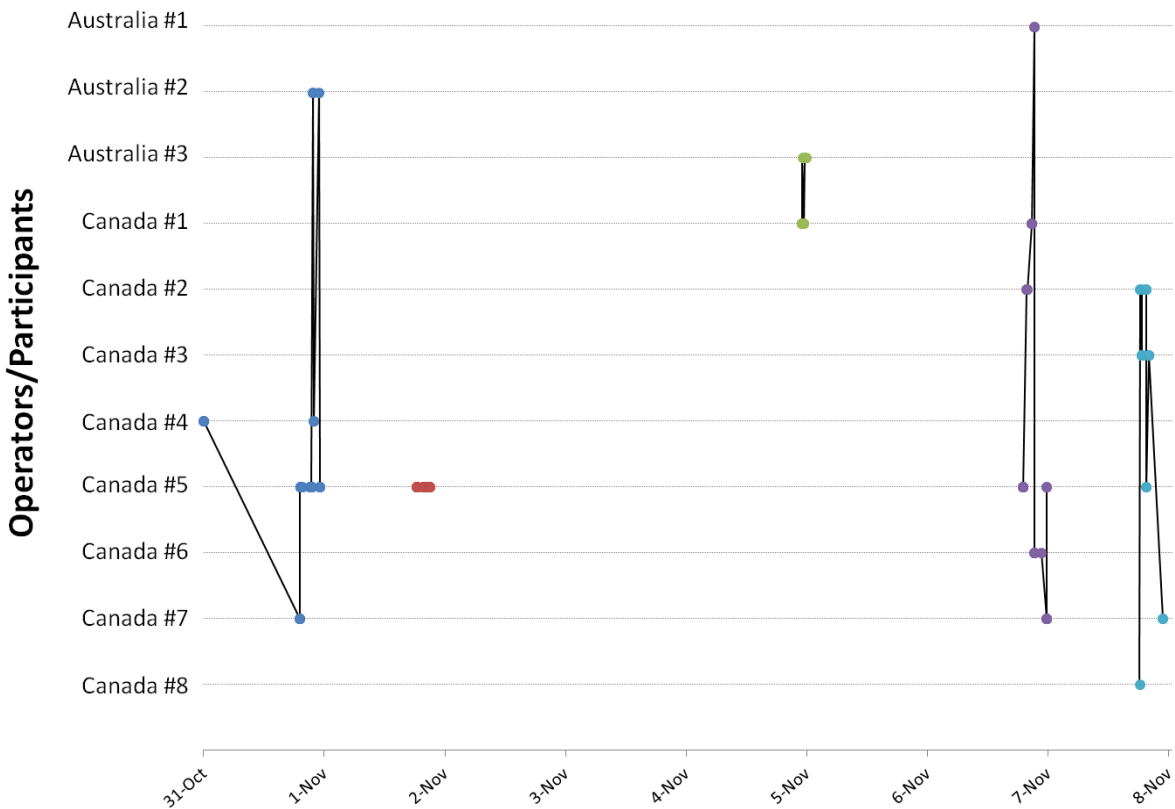
The text data can be displayed in numerous forms, and one such method is through the creation of word clouds. These clusters of words and numerical data blocks show the most frequently used words or terms in the largest font, tapering off to the rarely used terms in the smallest fonts. This form is appropriate since it allows all of the terms to be grouped into one small picture, unlike graphs and frequency plots, which would produce an inordinately long graph (especially considering terms used





4. two days later, coordination across the spectrum of involved operators, probably confirming the target status and approving a course of action;
5. finally, messages between the operators involved in executing the fires mission and reporting.

These data extractions showed that it is possible to trace processes using Thea, from start to finish. The sequence and level of personnel involvement can be determined, as well as the communication workload. If an important step in the process were to be absent or if certain sequences were out of order, this would be apparent. Unexpected sequences would at least provide an indication regarding problems in the process, thus alerting the analyst to areas for further investigation.



**Figure 10 – Visualization of an extracted process: the prosecution of the target “Two Fingers”. A point on the figure indicates that a user typed/copied/pasted text something related to this target.**

### 3.3 Packet Capture and Analysis of VoIP Calls

#### 3.3.1 Description of Packet Data

As previously mentioned, Thea used the WinPcap API to filter and store packet data at the workstation where it was installed. During the collection, the filtered packets were stored locally in several pcap files. Post experiment, the packet data was parsed by a custom Python script to extract phone conversation details, e.g. call start time, call end time, audio encoding, caller name/number, receiver name/number. The voice data from the phone conversation was also extracted, but only the call details were used in the following example.

CAGE 3A was the first experiment where this type of data was collected by the CFWC. The tool is still under development, and there is much work to be done in refining the collection of VoIP and simulated radio calls. The data presented here is incomplete, but demonstrates the current capability of the tool, its limitations, and where future improvements can be made.

### 3.3.2 Example Results

Thea collected data on 377 calls made during CAGE 3A. Unfortunately, a large number of the call logs were missing either who the call was from, who the call was to, had incorrect start dates, or were captured from before the experiment began. In total, only 128 of the 377 call logs (~34%) were complete and relevant for analysis.

Table 1 shows the 128 complete and relevant calls (hereafter referred to as the filtered data) broken down by type. It is important to note that no voice-to-text conversion was performed, so the call type is inferred only from the associated participants in the call, as opposed to the discussion made during the call. Of note, there were 0 player-to-player calls recorded in the filtered data set. Given that no calls between Australian and Canadian participants were successfully recorded, this is not surprising as all the Canadian participants were in the same room.

**Table 1 : VoIP calls by type**

Call Type	Filtered Count	Percent
Conference Call	91	71%
Technical Support	24	19%
Player-to-EXCON	10	8%
Analyst	3	2%
Player-to-Player	0	0%
<b>Total</b>	128	100%

The filtered calls are broken down by associated experiment day in Table 2. Additionally, a more complete list of calls is also shown. As can be seen from the percentage columns, there is little difference between the filtered call log and the more complete call logs. The high number of calls on Day 1 of Condition 1 and Condition 2 is most likely associated with training and initial coordination, as well as a higher than usual number of technical support calls associated with the beginning of a new experiment condition.

Tables 3 and 4 deal with call duration. Note that in Table 3, not all categories are of the same duration (e.g. some categories are 5 minutes, some are 20 minutes, etc.).

**Table 2 : VoIP calls by experiment day**

Call Count by Day	Filtered Count	Percent	Complete <sup>2</sup> Count	Percent
<b>Condition 1 Day 1 (C1D1)</b>	47	37%	103	30%
<b>C1D2</b>	4	3%	28	8%
<b>C1D3</b>	10	8%	35	10%
<b>C1D4</b>	5	4%	29	8%
<b>C2D1</b>	29	23%	79	23%
<b>C2D2</b>	10	8%	20	6%
<b>C2D3</b>	14	11%	38	11%
<b>C2D4</b>	9	7%	11	3%
<b>Total</b>	128	100%	343	100%

(e.g. some categories are 5 minutes, some are 20 minutes, etc.). The vast majority of the zero duration calls and of the longer (greater than 60 minute) calls were calls to the conference call phone number made by EXCON.

<sup>2</sup> The term "Complete" is technically not accurate, here and in later tables. First of all, not all calls made were captured. Secondly, of those captured, 10 calls had a corrupt start date, and 24 calls were recorded outside of the experiment dates (during technical testing).

From Table 4 we can see that it is a highly skewed distribution (since the mean and median are so far apart). In particular, based on the two tables, it appears that the longer calls are skewing the average call length. Looking more closely at Table 4, the means and standard deviations of the two data sets are likely significantly different due to two incredibly long calls in the more complete data set.

**Table 3 - VoIP call durations**

Call Duration (minutes)	Filtered Count	Percent	Complete Count	Percent
<b>0</b>	41	31%	43	12%
<b>Between 0 and 1</b>	18	14%	48	13%
<b>Between 1 and 5</b>	15	12%	81	22%
<b>Between 5 and 10</b>	13	10%	52	14%
<b>Between 10 and 15</b>	11	9%	30	8%
<b>Between 15 and 20</b>	2	2%	18	5%
<b>Between 20 and 40</b>	7	5%	34	9%
<b>Between 40 and 60</b>	2	2%	16	4%
<b>Greater than 60</b>	19	15%	45	12%
<b>Total</b>	128	100%	367	100%

Looking beyond the general descriptive statistics above, call logs could be utilized in numerous useful ways, mostly as part of a larger communications analysis. Generally, communication occurs either directly (e.g. VoIP call, face-to-face, etc.) or indirectly (e.g. JADOCs coordination cells, changes in a SharePoint document, etc.). The long term goal for Thea would be to capture and catalogue all non-in-person communication (with in-person communication being captured by observers, survey, or other recording equipment). This would include VoIP phone, simulated radio, chat, email and direct messaging, as well as the system based indirect communication.

Once communications were comprehensively captured, several data analyses could be performed. Social Network Analysis techniques could be applied to determine which participants are the most central/important, what cliques/clusters existed, whether the networks were highly transitive, etc. Additionally, part of automatically mapping existing and emergent processes could be undertaken, specifically the communication flows within those processes (as opposed to a task analysis). Finally, information spread and diffusion could be analysed to better understand how significant information is distributed through an organization.

The primary reason most of these analyses could not be performed on the given data set was due to the fact that most of the VoIP calls would be expected to be between participants from different nations. However, there were no call logs of that type (with sufficient data). For future development, it is important that all calls are captured.

**Table 4 – VoIP call duration high level statistics**

Statistic	Filtered Data	Complete Data
<b>Mean</b>	23m 40s	30m 17s
<b>Median</b>	1m 13s	6m 01s
<b>Standard Deviation</b>	46m 55s	113m 17s
<b>Longest Call</b>	225m 15s	1490m 26s

## 4.0 Conclusion

Thea, the computer monitoring tool, has provided a significant and import data collection capability to the CFWC. This tool is capable of providing experiment analysis and metrics that were not possible before.

The examples presented in this paper outline the extent of analysis that has been done with the collected data to date. Future analysis of the data is being considered in the areas of speech recognition, process mining, and behavioural biometrics. Speech recognition is being attempted to



convert VoIP and simulated radio conversations to text, which will then be added to the text analysis methods in Section 3.2. Process mining will help the CFWC better understand C2 processes; in particular, planning and coordination of joint fires. Behavioural biometrics will help to infer human factors like fatigue and workload, which previously could only be extracted through participant surveys.

Another future application of these detailed workstation recordings would be to compare analytics from one experiment to the next, and in general, between all events at the CFWC where this monitoring tool is used.

Thea remains under development at the CFWC. The capability of the tool that has been demonstrated in CAGE 3A will help in the development of the data collection and analysis plan for CAGE 3B and future experiments and exercises at the CFWC.

## References

1. Coalition Attack Guidance Experiment (CAGE II) Final Report, TTCP TR-AER/JSA-1-2013, The Technical Cooperation Program, 2013
2. System Surveillance Pro by Golden Plains Software, <http://www.gpssoftdev.com/> (accessed 19 February 2014)
3. Argos Monitoring by Argosafe, <http://argosafe.com> (accessed 19 February 2014)
4. SpyAgent by SpyTech, <http://www.spytech-web.com> (accessed 19 February 2014)
5. Spector 360 by SpectorSoft, <http://www.spector360.com> (accessed 19 February 2014)
6. Desktop Analytics by OpenSpan, [http://www.openspan.com/products/desktop\\_analytics/](http://www.openspan.com/products/desktop_analytics/) (accessed 19 February 2014)
7. NICE Interaction Analytics by NICE Systems Inc, <http://www.nice.com/cross-channel-analytics/desktop-analytics>, (accessed 19 February 2014)
8. KidLogger, <http://kidlogger.net/> (accessed 19 February 2014)
9. Tesline-Service publisher's description. (Feb 27, 2009). KidLogger reviews. Retrieved February 20, 2014, from [http://download.cnet.com/KidLogger/3000-27064\\_4-10418527.html](http://download.cnet.com/KidLogger/3000-27064_4-10418527.html)
10. Allen, D., Autonomous Workflow Reconstruction for Command and Control Experimentation, Proc. 17<sup>th</sup> ICCRTS, Fairfax, VA, 19-21 June 2012
11. Pimenta, A., Carneiro, D., Novais, P., Neves, J., "Monitoring Mental Fatigue through the Analysis of Keyboard and Mouse Interaction Patterns", Proceedings of 8<sup>th</sup> International Conference on Hybrid Artificial Intelligence Systems, 11-13 Sept 2013, Salamanca, Spain, pp 222-231.