17th ICCRTS

"Operationalizing C2 Agility"

Autonomous Workflow Reconstruction for Command and Control Experimentation

Topic(s)

Experimentation, Metrics, and Analysis

Name of Author(s)

Dave Allen

ISR Operational Research Team Leader

Point Of Contact

Dave Allen

Centre for Operational Research and Analysis

NDHQ, 101 Colonel By Drive, Ottawa, Ontario, Canada

Dave.Allen@forces.gc.ca

Autonomous Workflow Reconstruction for Command and Control Experimentation

Dave Allen
Centre for Operational Research and Analysis
NDHQ, 101 Colonel By Drive, Ottawa,
Ontario, Canada
Dave.Allen@forces.gc.ca

Abstract

This paper describes a semi-autonomous system that has been developed to support the mining, visualization, and analysis of the process as performed by a team of operators when fulfilling specific military command and control tasks. The system focuses on operational centre activities. It is composed of several independent modules. Two primary modules provides the basic input to the system: a monitoring module that logs all keystrokes, mouse clicks and screen snapshots on each workstation used by the operators; and, a reference module providing a repository of activity models describing expected sequence of activities. In addition, the system uses an optical character recognition (OCR) application as well as pattern recognition algorithms. It provides information on the structure of the workflow as performed by the operators and support the comparison between the expected and observed activity flow. The system has been used to support Joint Fires Support experiments and although improvements to the OCR are required, the developed methodology appears very promising.

Keywords: Model-driven experimentation; process modeling; process mining, process recognition; command and control.

1. INTRODUCTION

1.1. Background

Teams are essential to the proper functioning of military units. Large military operations depend on the coordinated effort of tens of thousands soldiers dispersed over a broad area of operations (in the battlefield, deployed operating bases, as well as, at national operational bases back in the homeland). It depends on intelligence cells gathering and processing sensor data and other information to provide a global picture of the situation to the commander who, supported by a team of planners, will provide directives and guidance to teams of air, naval and/or ground officers who will execute the required actions. All these teams are further supported by units that provide core-infrastructure and cyber protection and a complex logistical chain that ensure the provision of fuel, ammunitions, parts, as well as other required resources. The assessment of such operations is therefore inherently an assessment of the performance of interconnected and coordinated teams.

The North Atlantic Treaty Organization (NATO) Research and Technology Organization (RTO) for Human Factor Modeling (HFM) has developed a framework for assessing team, called the Command Team Effectiveness (CTEF) instrument. [1] The premise for this instrument is that the assessment of a team cannot be reduced to the capability of the team to meet its goal, but the complexity of the goal, the context under which the tasks are being performed as well as the team structure need to be considered within the assessment.

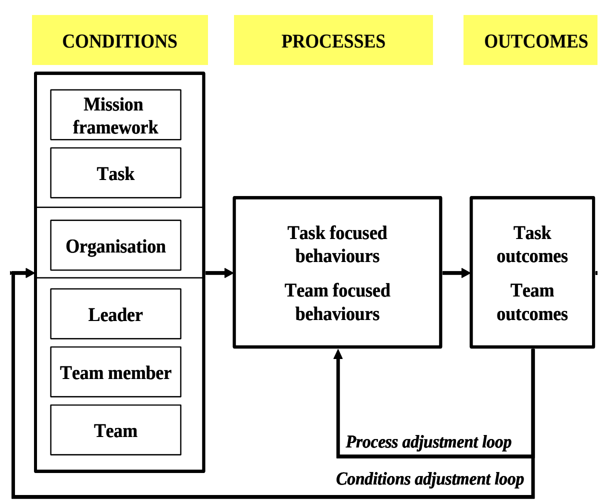

Figure 1: ***High level overview of the Command Team Effectiveness instrument***

Figure 1 summarizes the various factors considered within the CTEF instrument. Essentially, there are three families of measures within the instrument: working conditions; process; and, outcomes. Each of these elements is further described in Table 1.

The list of factors considered within the CTEF model is provided in Table 1. Multitude measures have been developed over the years focusing on factors similar to those considered under conditions and outcomes. Such instrument includes the Personal Value Questionnaire, [2] various methodologies for assessing situation awareness, [3] as well as confidence evaluation instruments. [4] [5] With regards to the process evaluation, a lot of effort has been devoted to assessing the information exchange within a team of operators. These assessments have traditionally used clustering methodologies (to identify possible patterns for effective and ineffective information flow) and social network analysis. In particular, a series of experiments was run collaboratively by the University of Pittsburgh and the Mitre Corporation between June 2000 and April 2002 to investigate factors hindering the collaboration of a distributed team interacting using information technology. Using clustering methods, these experiments indicated four types of interactions that could lead to inefficient team collaboration (see [6] for a summary). These four types of interaction correspond essentially to Webb's factors for ineffective collaboration [7]:

1. Not requesting collaboration;
2. Lack of timely and relevant support for collaboration;
3. Lack of clarity of provided information; and,
4. Lack of follow-up, i.e., not implementing or using the provided information.

Table 1: ***List of factors within the CTEF model***

	Conditions		
Mission Framework	Situation uncertainty Stress potential Constraints Stakes	Leader	Skills Knowledge Match personal to organisational goals
Task	Complexity Workload Unclear goals Instable goals	Team Member	Skills Knowledge Match personal to organisational goals
Organisation	Goal congruity Clarity of Command structure Autonomy Organisational support	Team	Composition Size Structure Maturity Match team to organisational goals
	Processes		
Task Focused Behaviours	Managing information Making decisions Planning Directing and controlling Interactions with other teams	Team Focused Behaviours	Giving and maintaining vision Maintaining shared strategy Collaborating within the team Motivating Adapting to changes Providing team maintenance
	Outcomes (*for intermediate or end goals)*		
Task Outcomes	Goal achievement Stakeholders' expectations Staying within limits/intentions	Team Outcomes	Mutual trust Morale Cohesion Collective confidence in success Shared vision Mutual respect

Another important factor to team performance is the frequency and type of feedback received by the operators (the feedback loops are illustrated at the bottom of Figure 1 but are not explicitly included in Table 1). Traditionally, the feedback is largely limited to individual task performance. However, as indicated by Kozlowski's group, [8] an appropriate balance between individual and team feedback should be received as efforts are made towards the desired goal. This balance ensures that an appropriate focus and competitive atmosphere is reached within the working environment. This feedback is in fact essential to ensure an effective team motivation and adaptation to changes shown within the process category of factors (see the rightmost column under "processes" in Table 1).

Due to its dynamic nature, the assessment of the processes requires a continuous data collection that captures the communication among the various operators as well as the operators' activities. For small team of operators this can be achieved by a team of observers. However, for large team of operators, a full and complete analysis requires time and effort and an automated tool to support this assessment is desirable.

1.2. Aim of the Paper

The aim of the paper is to describe the process monitoring, mining, and evaluation system that was recently developed to support command and control experiments at the Canadian Forces Warfare Centre (CFWC). The paper focuses particularly on the overall approach to command and control experimentation and the analysis system to automate the analysis of the workflow as performed by a team of operators.

Following this introduction, the paper introduces a classification scheme for the various types of experiments and identifies the context in which the proposed system would be used. This discussion is build using a comparison with approaches used for software application testing. This comparison shows how the developed system will support a model-driven approach to experimentation. The rest of the paper provides details on the system itself.

2. METHODOLOGICAL APPROACHES TO EXPERIMENTATION

2.1. Approach Overview

The various approaches to experimentation can be categorized similarly to the software testing approaches. Figure 2 shows various approaches that have been developed for software testing (the figures are based on reference [9] with the exception that here no differences are made between the "Manual Testing" and "Capture&Replay" approaches; the only difference being whether a Capture-Replay tool is used).

The graphs within Figure 2 are build using traditional data flow representations where a parallelogram is used to represent data, a rectangle with a wavy based is used for documents, rectangles are used for automated processes, and trapezoids display manual processes.

The comparison between the software testing and the considered experimental approaches are summarized by the following relationships:

- Manual testing ~ Table-top experiment;
- Script-based testing ~ Simulation-driven experiment;
- Keyword-driven testing ~ Lesson-based simulation-driven experiment;
- Model-based testing ~ Model-based experiment.

Although the type of entities tested are very different – software applications in one case and a socio-technical system composed of hardware, software and operators in the other case – many of the same pros and cons to both type of activities. In particular, table-top experiment, like manual testing, are rather inexpensive, but the feedback collected is largely limited to subject matter expert judgments. Within an inter-organizational setting, it provides a useful to compare various organizations concept of operation, but it is less reliable for testing physical limitations. An example of the limitations is provided by the Hurricane Pam table-top exercise that fails to adequately test both the transportation and communication issues that might be experienced following a Hurricane incident in the New Orleans area. [10]

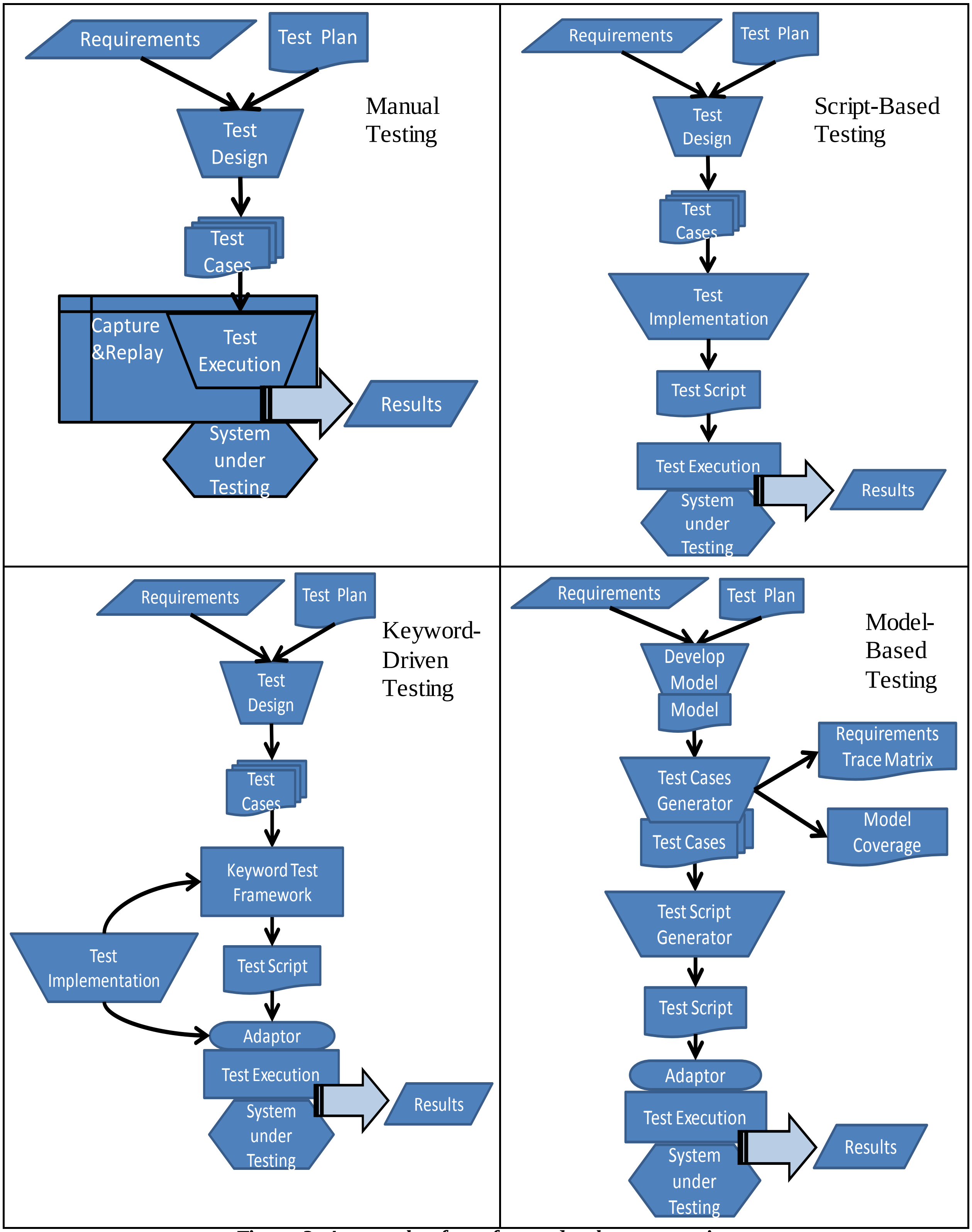

Figure 2: ***Approaches for software development testing***

The simulation-driven approach to experiment is similar to the script-based testing in the sense that a more complete consecutive set of tasks are being followed. This type of experiments is more costly since they require more initial work to set-up the simulation and also personal to control and adapt the simulation when required. It provides a more rigorous testing than the table-top experiment, but does not provide measurement for the coverage of the test. Simulation-driven approaches can be very prescriptive, as used in a test and evaluation approach, or more adaptive to the decisions made by the participants to the experiment.

The lesson-based simulation-driven experiment build on the simulation-driven approach by providing a set of previously tested script build from previous experiments and/or lessons obtained from military operations. This approach is fairly similar to the simulation-driven approach, but simplifies the development of scripts through the use of keywords and repository of scripts linked to specific test cases. Within an adaptive type of experiment, this approach can support a variation of the test script depending on decisions made by the participants to the experiment (the adaptor allowing a feedback to the test case via the test implementation).

Finally, the model-based approach for planning and executing the military experiments follows the model-driven testing methodology used for testing software applications [11]. It requires more initial work for the development of models. These models would represent the expected work flow of the tested socio-technical system. However, it leads to a more complete set of testing and also to better appreciation of the limitations to the performed tests (measurement of the coverage of the test).

At this time, the Canadian Forces Warfare Centre (CFWC) has been performing mostly the first two-types of experiments. The third approach has been used, but with limited appropriate support mostly due to the inadequate database tools to categorize and store simulation initialization data. Also, some initial steps have been made to support model-based experiments. This latter type of experiments will leverage on-going work within the CF to develop various business processes. The process mining and evaluation tool described here will play an important role in supporting such experiments.

Figure 3 indicates the central role to be expected from the process mining and evaluation tool. Models and test cases are used as starting points from which a prototype system (technical, procedural, and/or organizational prototype) and test script are developed. Using the developed test script, the prototype is tested within an experiment or training exercise within a controlled environment. An audit trail is captured to then verify whether the initial requirements were met. The system discussed within this paper is used to provide a semi-automated comparison between the expected activity models and the captured audit trail.

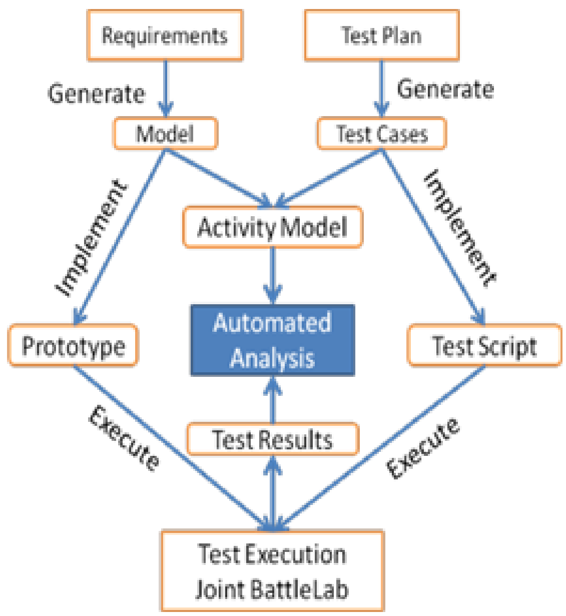

Figure 3: ***Overview of the planned used of the Semi-Automated Analysis system***

3. PROCESS MINING: AN OVERVIEW

3.1. Limitations and Issues with Process Assessment

Various process monitoring, mining, and evaluation methodologies and tools (ProM, PISA, Process Miner, CPAS) have been proposed over the past years (see [12] for a recent review). Although these methodologies can worked well in a structured and restricted environment, limitations have been identified in open unrestricted work environment were operators can adapt to the situation and respond in unanticipated manner. [13] The review of case studies has indicated five major problem areas with currently available process mining systems: [14]

1. Process-oriented Information Systems (IS) require continuous adaptation due to the frequent process changes.
2. Process-oriented IS exhibit hard-coded process logic complicating the adaptation of such systems.
3. The existing software components lack possibilities to customize process logic at a sufficiently flexible and detailed level.
4. In many cases, the provided business functions do not effectively support the business processes.
5. Structures of log data vary from system to system and keeping track of the process and mining the logs require large efforts.

Within the context of monitoring and mining the processes performed by the military staff within an operation centre, these difficulties are exacerbated for several reasons. Military operators use several specialized IS, such as, ADSI, AMPS, GCCS, JADOCS, LogFAS, MAAS, SC2PS, to name but a few. Although these systems possess their own logs, often the content of these logs is limited and does not provide enough information for an adequate process audit trail. Furthermore, some of these systems have been obtained through Foreign Military Sales and no access to the internal code is possible.

In addition, some of these systems limits the access to their own databases to ensure information assurance (i.e., it ensures that no external agents have the opportunity to change, delete, or otherwise interfere with the database content). All these factors complicate the recording of valuable audit trail to monitor and assess the military operational processes.

Beyond these issues, the staff members within an operation centre are often dealing with several different processes at any one time. Some processes, such as the development of summary reports and of briefs are routine and performed on a regular basis. Other processes are performed on an ad-hoc basis in response to on-going crises. These include medical evacuations, search-and-rescue missions, fire support, surveillance and tracking missions, etc. These crises often lead to the interruption of other on-going processes and the re-direction of resources. In addition, the Chief of Staff performs several tasks to ensure the cohesion of the staff work; these are important tasks that might at first sight appear as noise within the analysis of the staff workload.

For these reasons, the Canadian Forces Warfare Centre initiated the development of new tools and methodologies for the monitoring, mining and evaluation of the processes performed by the military staff participating in experiments and training exercises performed within its Joint BattleLab.

3.2. Requirements for Process Assessment

The principal assumption for the design of the system to support the processes assessment was that all processes would be prescribed ahead of the collection of the audit trail. The proposed approach consisted in assuming an initial set of process models describing the expected activities by the operators. The processes would be described indicating the subsequence set of tasks expected. Each of the indicated tasks would also be associated to a list of specific actions indicating the procedure to be followed for accomplishing it.

As a matter of clarification, the following terminology was agreed to differentiate respectively the terms action, task, approach, method, procedure, and process:

- **Action**. An action was considered a complete observable movement performed by an operator (e.g., striking a key on his or her keyboard, or a set of continuous eyes saccade by the operators to read given information).
- **Task**. A task is defined as an activity that is accomplished by a single operator or performed simultaneously by a group of operators and which leads to a single output. A task is typically composed of a sequence of actions (e.g., the production of a brief is a task that requires several actions).

- **Approach**. An approach is defined as an attitude or manner (modus operandi) to perform some task. In order words, an approach is a philosophy for accomplishing a task.
- **Method**. A method is defined as a way of accomplishing specific tasks. It can be vague or detailed by specifying the tools to be used.
- **Procedure**. A procedure is a series of actions specifying a precise way of accomplishing a task.
- **Process**. A process is defined as a collection of causally related tasks, which solve a particular issue. It specifies the goal (or expected outputs) to be achieved. The complete definition of a process requires four sets of items: the set of interrelated tasks; the set of resources associated with the tasks; the set of expected outputs or goals; and, the set of possible triggers of the process. It corresponds to a WorkFlow Net where the source corresponds to the trigger and the sink to the output. [13]

The requirement for the system was to capture the processes performed by a team of operators with the aim of assessing whether each process was performed as and when expected. If differences were observed, the tool shall also support the classifications of these differences and the identification of the context in which these differences occurred. The system must also support the analysis of the communication logs to identify its efficiencies, to identify the type of feedback and motivation received by the operators, their ability to maintain a shared strategy and vision, and their adaptability to changes. The processes of interest were centered on tasks such as information management, decision making, planning, directing and controlling.

4. AUTOMATED PROCESS RECOGNITION SYSTEM

This section describes the system developed to support the desired process analysis. The first subsection discusses the audit trail used by the system and the second focuses on the data analysis. Figure 4 provides an overview of the system ("UI" designates the User Interface).

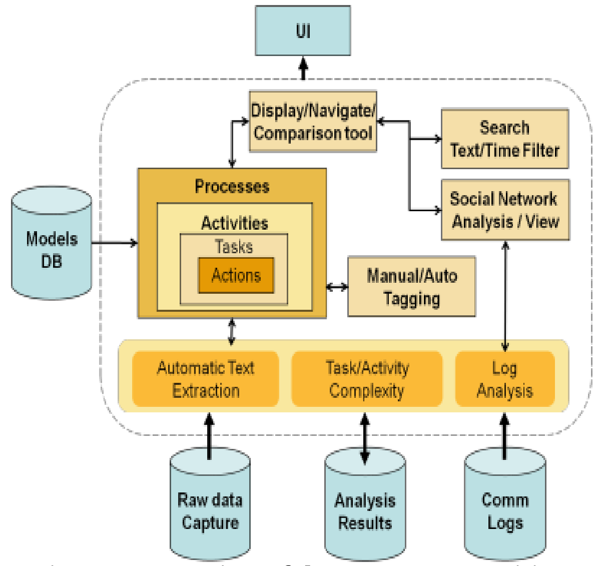

Figure 4: ***Overview of the process recognition system***

Three databases are used by the system: the audit trail database which includes the data recorded through the monitoring of all operators workstations; the expected database which includes a description of the expected processes; and the analysis database which includes the results from the analysis of the two previous databases. The user can navigate, filter and search any of these three databases.

4.1. Audit Trail Content

Due to the limitations with the information systems used (see discussion in section 3.1), the selected audit trail consisted of the logs from all communication systems as well as the recording of all mouse clicks, keystrokes, and frequent recording of all monitors screen snapshots.

The audit trail system was designed to have a component installed on all operators' workstations and a component installed on the analyst workstation to remotely control the data capture over the network. Open source software was selected as the main components on which to build the audit trail system. More precisely, Cygwin, TightVNC, rfbproxy, and SQLite were selected. To capture as much operators' actions as possible, the list of applications were brought together to capture the screen snapshots, the mouse clicks and the keystrokes performed by all operators. For flexibility, the screen capture was designed to be performed at a user selected frequency.

The screen capture tool, rfbproxy, is a Windows program that runs as a service on each target machine. The program is controlled remotely over the network, allowing the data analyst to control the beginning and end of the capture as well as the transfer of the required screen/keyboard/mouse data. The captured data is stored on the local disk of each machine by default to avoid interfering with normal network traffic. The files are transferred back to the data acquisition control computer when the network is not being used for the actual experiments.

All the collected data (screen snapshots, mouse clicks and key strokes) are time tagged. In addition, the location of the mouse clicks within the operators' display is associated with the mouse click. This audit trail system has been used over the past experiments. Typically, 10 to 60 gigabytes of data were captured per operator and per day of experiment and screen capture frequencies of 1 to 5 Hz have been used. To speed up the analysis, the collection database has been divided into various sessions. A session is typically associated with the data captured on a single operator during a given day of experimentation.

This audit trail is very rich and provides the benefits of clearly identifying the context during which each action was performed. Upon completion of the analysis, the audit trail can also be used to review particular identified issues with the participants to the performed experiment. Since, the ultimate goal of all experiments is to support some learning objectives, this benefit is very significant. However, the diverse content of the audit trail complicates the process mining and evaluation. A particular issue is that the content is largely made of events and attributes defining the context in which these events occurred rather than a simple list of tasks (note that typical process mining tools are assuming an audit trail containing task lists). This is further complicated by the fact that a given action might be performed to accomplish two or more different tasks, and similarly a given task might be performed within several different processes. For this reason new methodologies and tools were required to perform the process mining and evaluation.

4.2. Process Analysis System

The process analysis component of the system has been developed since the summer of 2010 and is still under development. This component is designed to enable specific transformations of the audit trail as well as to perform comparisons with expected process models. The expected process database is designed to capture operational architecture diagrams (in particular the process models provided within operational views OV-5b and the event-trace diagrams provided by the operational views OV-6c). This was done to leverage on-going work for the development of the Department of National Defence Architecture Framework (DNDAF). [11]

The analysis system is composed of several components:

- An audit trail browsing component to review and vet the captured data;
- A text extraction component to identify the information content within the operators displays (from the screen snapshots);
- A search functionality to mine all extracted data;
- A tagging functionality to cluster and label particular actions;
- An association functionality to associate a set of actions with a given task;
- A results visualization module.

Each of these components is described below.

4.2.1. Audit Trail Browsing and Review

Figure 5 provides an overview of the process analysis component of the system. Using, the "File" menu (top left), the user can load various databases. The loaded databases and their associated sessions appear on the upper-left window. Once a session is selected, its content in terms of screen snapshots, keystrokes and mouse clicks can be visualized on the right. In particular, the consecutive list of screen snapshots associated with a given workstation can be played as a movie. The user also has access to the usual functionalities, such as, rewind, pause, forward, zoom-in and zoom-out.

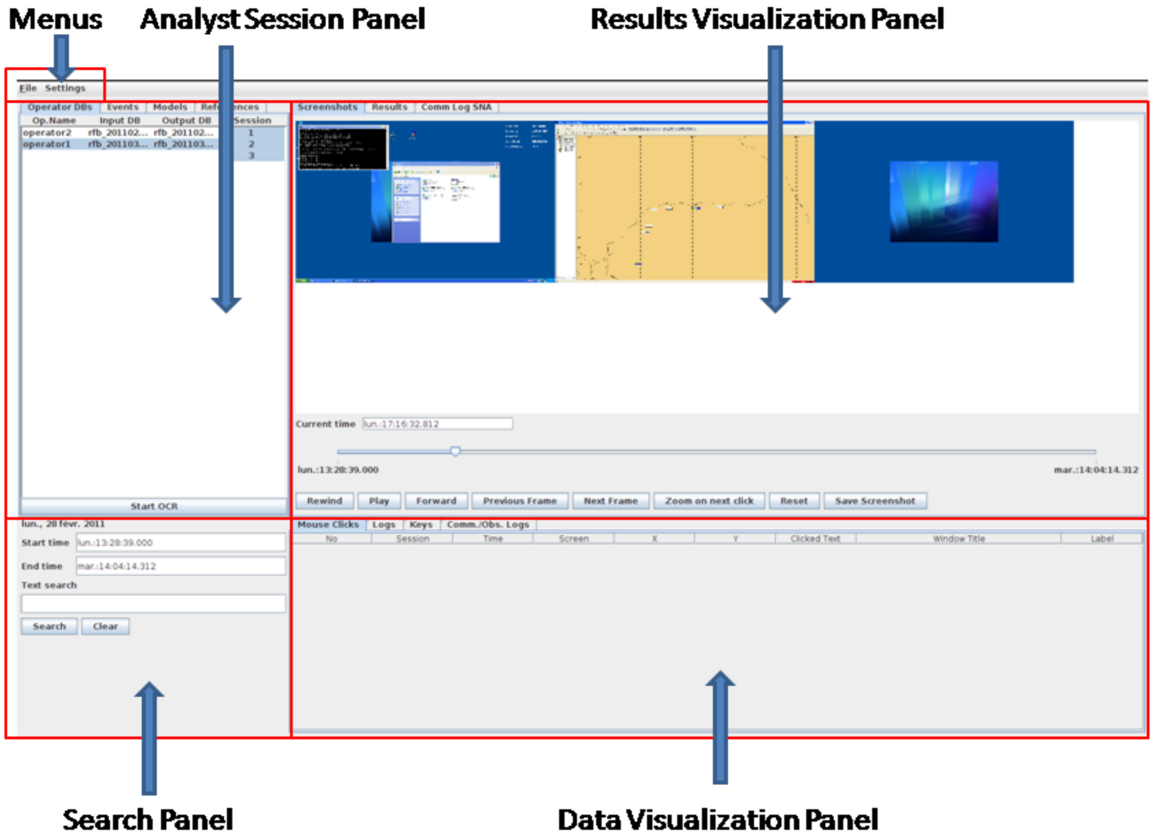

Figure 5: ***Overview of the process analysis component of the system***

Communication logs can also be loaded and appear in a different tab on the upper right (see tab called "Comms Log SNA"). When visualized, the communication log shows in a table format the list of chat entries made the various operators. The name of the chat room and time of entries are also displayed. In addition to chat entries, it is also possible to visualize email exchange and phone logs if these logs are provided in a suitable format.

When viewing the different screen snapshots, communication logs, and actions (records of mouse clicks and keystrokes) performed by the operators, the user can associate a label to any items of interest. The labelling capability permits the identification of items of interest, focusing on a subset of the dataset. An important feature of the labelling is the automated detection of all identical actions (if similar labels have been associated with them). The resulting detected actions are automatically bookmarked and shown in a special folder of the Bookmark (or Tag) section and can also be shown, in the Result Window, ordered in time. Labels can be removed, removing as well the related bookmarks produced.

4.2.2. Text Extraction

Upon selection of a session, the user can extract the information within the screen snapshots (by pressing the "Start OCR" button on the left). This procedure will identify the titles of the active window in which the key strokes and mouse clicks were performed. The label of command buttons where mouse clicks were performed is also extracted and added to the "Mouse clicks" table. Finally, all additional text identified is compiled within a "Logs" table. The three tables "Mouse Clicks", "Logs" and "Keys" appear at the bottom right of the Data

Analysis window and can be selected subsequently.

The extraction of the text from the screen snapshots is performed through consecutive transformation to reduce the risk of errors with the identification of characters. As an example, Figure 6 shows the list of transformation performed to extract the text from clicked buttons. The single image is divided into different Hue-Saturated-Value (HSV) channels and then processed before recombining the various images and applying further transformations.

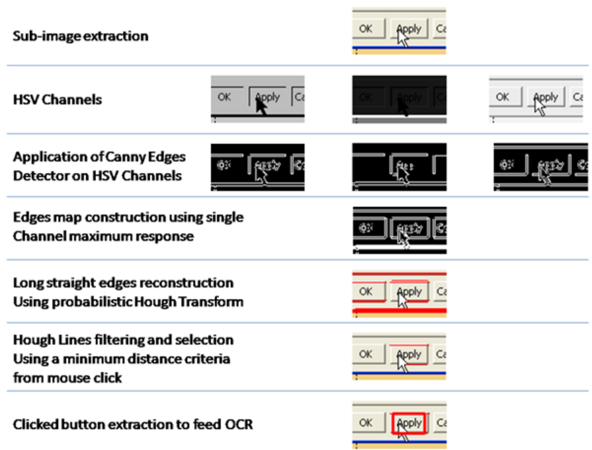

Figure 6: ***Consecutive transformations for extracting text from clicked buttons***

4.2.3. Data Mining

Once data has been extracted from the screen snapshots, this data (as well as other data including key strokes) can be mined using the search panel on the lower left portion of the window. The result of the search is used to filter the data displayed within the "Data Visualization Panel". To bring back all data within the panel, the user can perform a search without specifying any characters for the search (an empty string is interpreted by the engine as an unrestricted search) and by extending the start time and end time for the search to the beginning and end of the session.

The search engine is designed to be flexible enough considering that operators will typically make typographical errors when typing. Thus, this engine uses a similarity evaluation between two strings which is based on the Levenshtein distance. [16] The Levenshtein method makes the assumption that the words are not necessarily misspelled, but rather mistyped. When computing the proximity of two expressions with the Levenshtein metric, operations such as changing a letter, deleting or adding a character, inserting a blank space, or interchanging two adjacent letters are performed on the first word. If these steps result in the second word, then the number of steps is used to estimate how far the first word was from the second. To obtain the score in percentage, the following transformation is applied:

score = 100 x (1 - Nb of operations / Length of the longest string).

The similarity threshold was set to 0.75 (except for short words with less than five characters where a perfect match is required). While there is no widely adopted value for such a threshold, it usually varies from 0.70 to 0.90.

It was decided to fix the threshold at a relatively low value (0.75) to avoid the discrimination of potentially relevant information. In the present case, it is less damaging to present information not perfectly matching the query than missing relevant information that includes typos.

The search engine for the communication log provides a more advanced search (through the use of logical operators and wildcards) to allow searching on all aspects that would relate to a given work object. For example, if the considered work object is a target, then the user can include in the search the name and aliases associated with this target, its basic encyclopedia number, its associated targeting number, the location at which the target was engaged (using latitude and longitude or military grid coordinates), any coordinating measures associated with the target, and also the name of resources that were designated to track, identify, designate, engage, or perform battle damage assessment on the target.

4.2.4. Tagging

The action of tagging consists in categorizing items (audit trail or processed data) within groups of interest in order to be able to retrieve a group of items quickly when needed.

In that sense, the intent goes beyond the simple identification of a screenshot or mouse click of interest. To perform the desired analysis, one needs to categorize the observed actions; or in other words, associate a particular tag to the action.

The tags are stored in a tree-like user-defined structure. The user can create new nodes in the tree structure and rename it to his or her convenience. The retrieval functionality uses the user selected timestamp string to fetch and display the corresponding item.

There are two main categories of tags: manual and automated. Manual tags include a special category called "expected" which is used to capture the expected/predicted sequence of actions (or tasks) to perform a task (or process). The automated tags identify sequence of actions from within the raw data that reproduce the "expected" list of tags. Several automated tags are generated from a single "expected" tag if a given process was performed several times by the operators.

The user has complete control over the content of all tags. In particular, the content of the automated tags can be reviewed and modified if the user considers that specific actions were incorrectly tagged.

4.2.5. Association

The generation of automated tags based on the expected tags is performed through an association engine. The input of this engine is the list of actions identified through the text extraction. The activity association is initially comprised of two sub-tasks: event mapping and sequence matching. Figure 7 provides an overview of the procedure used by the association engine.

As shown in Figure 7, the output of the text extraction (through the use of an Optical Character Recognition engine), is used to build a list of event. An event is defined as a given action occurring at a given time. In other words, an event corresponds to an ordered pair (action, time). Since a given action might be expected in more than one modeled task, the association of events to expected tasks is performed by finding the expected tasks to which the events might correspond and by using a consensus ranking methodology to determine the best association. More precisely, an ordered list of actions is built based on the sequence of events associated to a given operator. This ordered list of actions is then compared to the expected sequence in each task where the considered actions show up. A distance between these sequences is computed using the Kendall tau-b measure. [17] The task with the shortest distance (as defined by the Kendall tau-b method) is selected as the best fit for the association of the actions.

All actions not appearing in any expected sequence will be classified in a "Miscellaneous" tab. The actions within this tab will be ordered by frequency of their occurrence, the actions occurring very often being on top of the list.

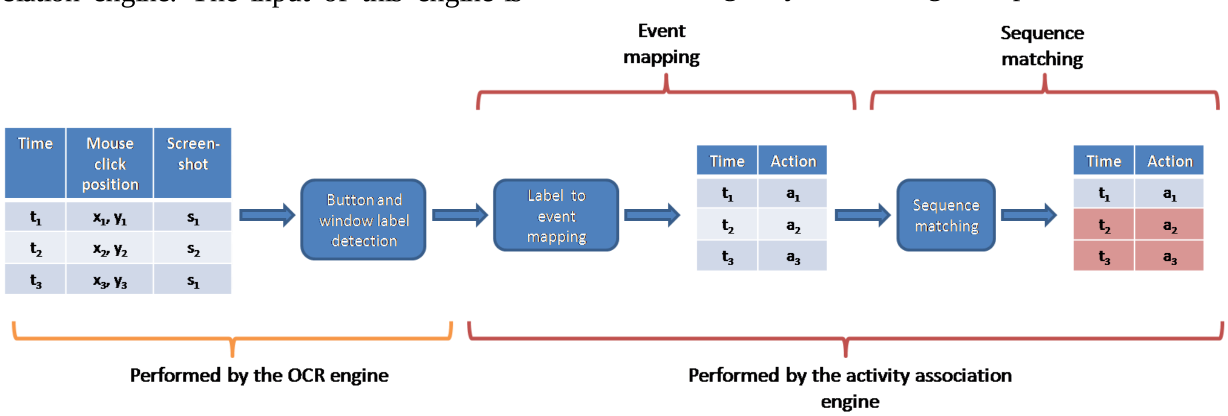

Figure 7: ***Performed operations for the association of performed actions to the list of expected actions***

Once actions have been associated with tasks, a similar procedure is used to link tasks to process. However, in this case the content of the windows is used to identify the most likely work object used within the task. Using the developed test script for the experiment, the association with a given process is performed.

The results of the various associations appear in tags. All actions associated with a given task are grouped in a tag labeled with the name of the associated task and the time at which the task was initiated. Similarly, all tasks associated with a given process are grouped in a tag labeled with the name of the associated process and the time at which the process was initiated.

4.2.6. Results Visualization

The upper right panels of the main window (see Figure 5) can be used to visualize various results of the investigation performed. The consecutive time when the operators performed specific actions (mouse clicks, key strokes) can be visualized in Gantt chart format. The gaps between observed and expected processes can be observed using stop lights. Finally, the communication flow relating to specific work objects as well as communication statistics associated with each operator can also be obtained.

Figure 8 to Figure 12 provides example of results that can be displayed within the system. The design of the visualization engine is based on the JFreeChart package and was developed to provide as much flexibility as possible for the type of chart displayed. The figures displayed below only provide a limited list of results that can be visualized within the system. Some of the check lists on the provided graphs indicate additional display available.

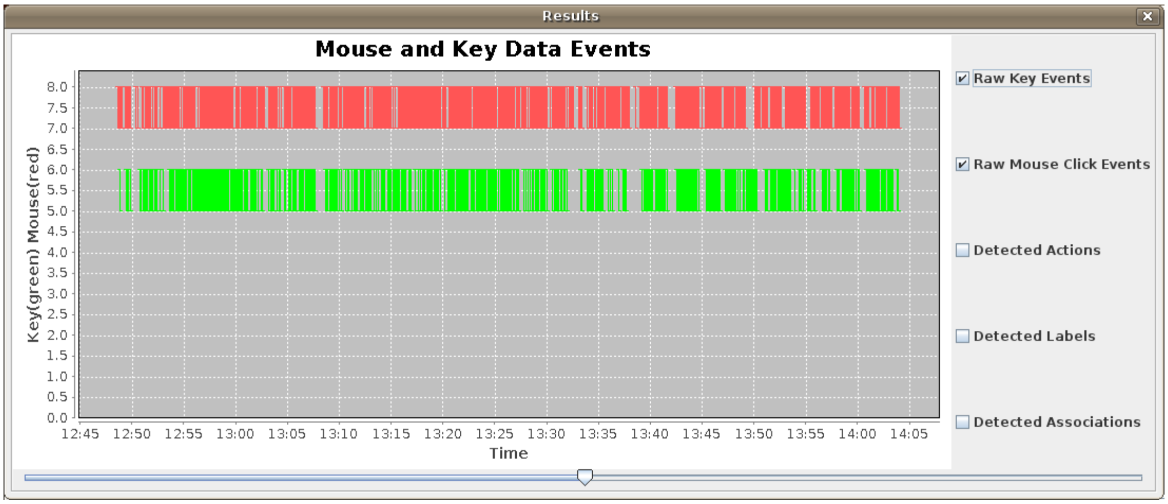

Figure 8: ***Visualization of operators' activities for a given time window***

Figure 8 shows the time of occurrence of mouse click and key strokes performed by all operators. As indicated in this figure, the user can also select additional items for display (list on the right end). The observation of these events provides an easy identification of period of higher intensity with those of lower intensity. As can be seen in this figure, there are short period of time when no mouse click or keystrokes are observed (for example near 13:08 and just after 14:00).

Figure 9 provides an example of the current visualization of the model vs. observed process comparison. For each process (task) instance, a stop light is assigned that informs the user whether all tasks (actions) within the associated expected model also appear in the identified instance. The stop light is green if all tasks (actions) appear; otherwise, the stop light is red. The assigned color is done regardless of the order in which the tasks (actions) are being performed within the process (task).

On-going work is underway to provide a more complete visualization of the differences between process (task) instances and the expected models that would include the task (action) order. In addition, algorithms will be used to identify patterns that repeat across all instances of a given process (task) as well as identify portion of processes (tasks) that vary across their associated instances.

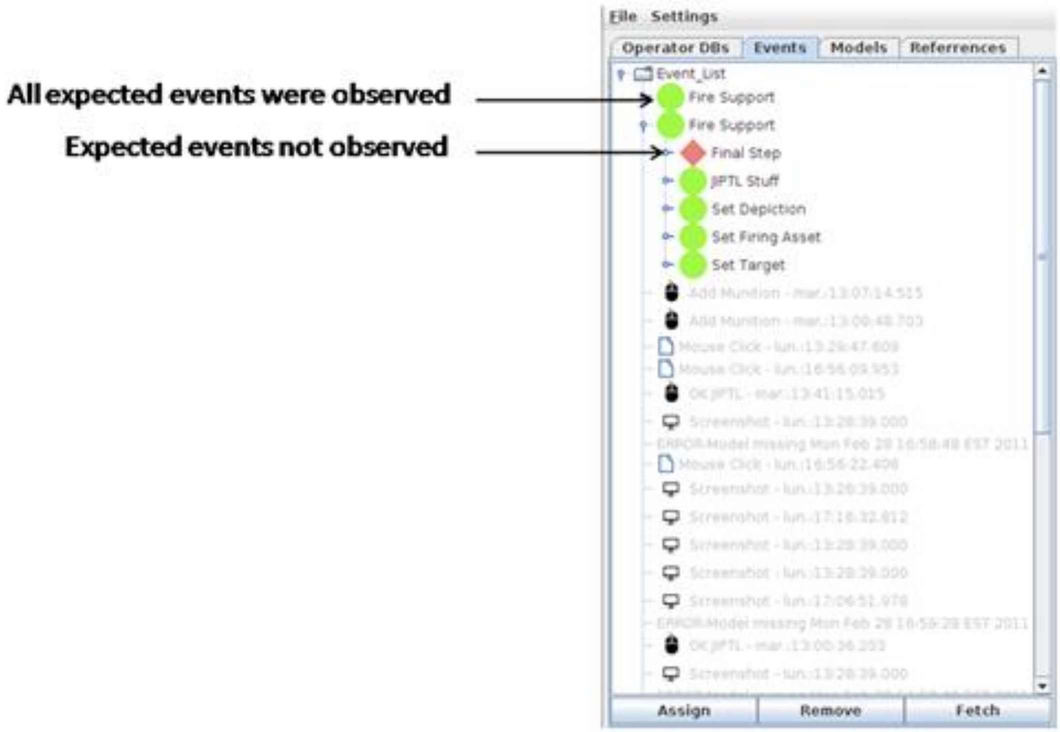

Figure 9: ***Visualization of gaps between each process and task instance and their associated model***

Figure 10, Figure 11, and Figure 12 provide examples of the visualization of data obtained from the communication logs. At the top of these figures, five tabs are used to display 5 different work objects (“TIC at ruins”, “COI#127”, “down helo”, “mortar attack”, and “INS”). Each of these tabs is displaying the subset of communications that relate to a given work object (these subsets were obtained through data mining as discussed in section 4.2.3). The last tab (“Statistics”) provides overall communication statistics (as shown in Figure 12).

Figure 10 provides a display of the communication flow that relates to each work object. More precisely is displayed all identified communication related to the work object in terms of who initiated the communication (red dot) and who was on the receiving end (green dot). The vertical axis represents the list of operators while the horizontal axis shows a specific time window. Additional details with regards to the content of the information shared are displayed when the user brings the cursor above a given node. The panel to the right provides global information with regards to the overall time window associated with the selected work object as well as the list of all operators involved in all identified communications.

Figure 11 provides a more traditional Social Network Analysis (SNA) display. Both a directed or undirected network display can be selected. The display shows the list of all operators that were involved in communication related to the selected work object regardless of the time of their involvement. The right panel provides to traditional SNA data: the betweeness and closeness of operator within the considered network. These data provide useful information to identify the relative importance of each operator in coordinating the communication required to dealing with the selected work object.

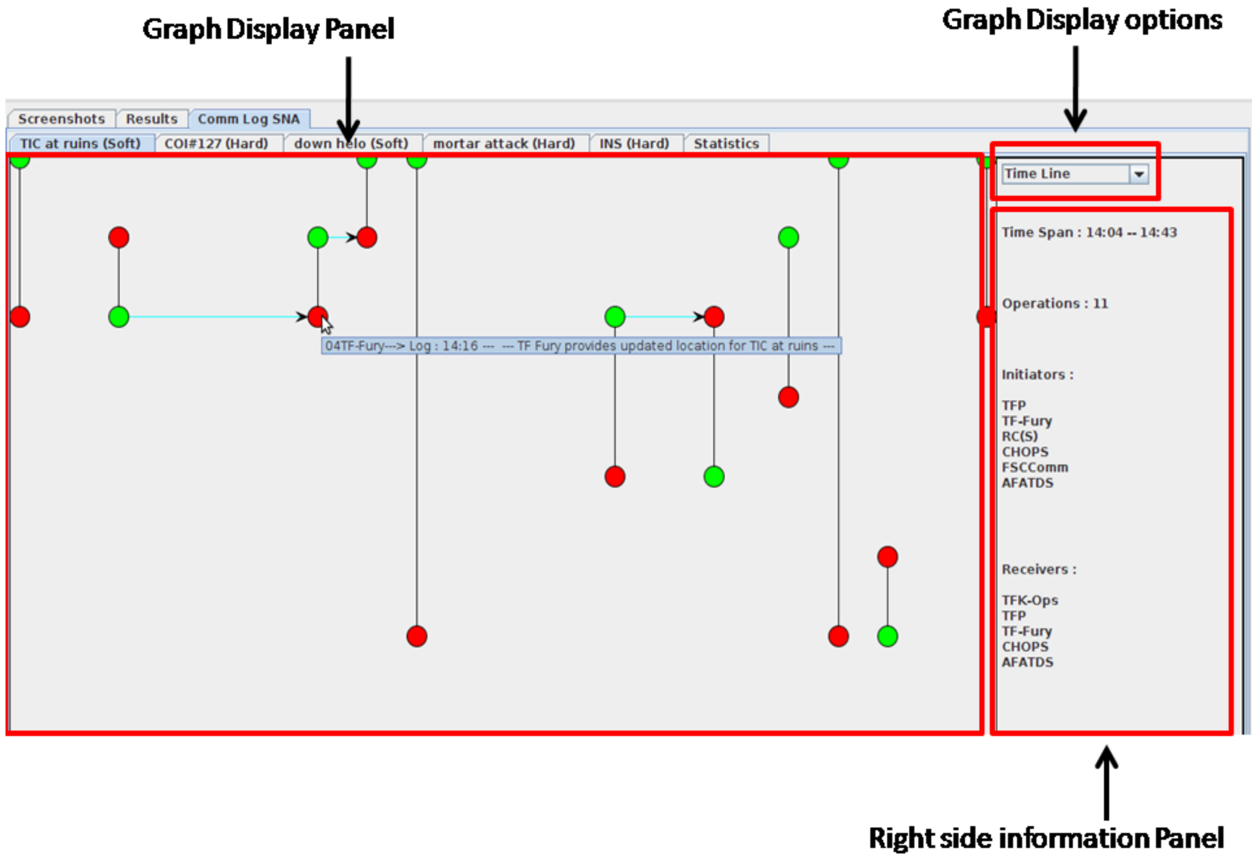

Figure 10: *Visualization of the sequence of communications associated with a given work object*

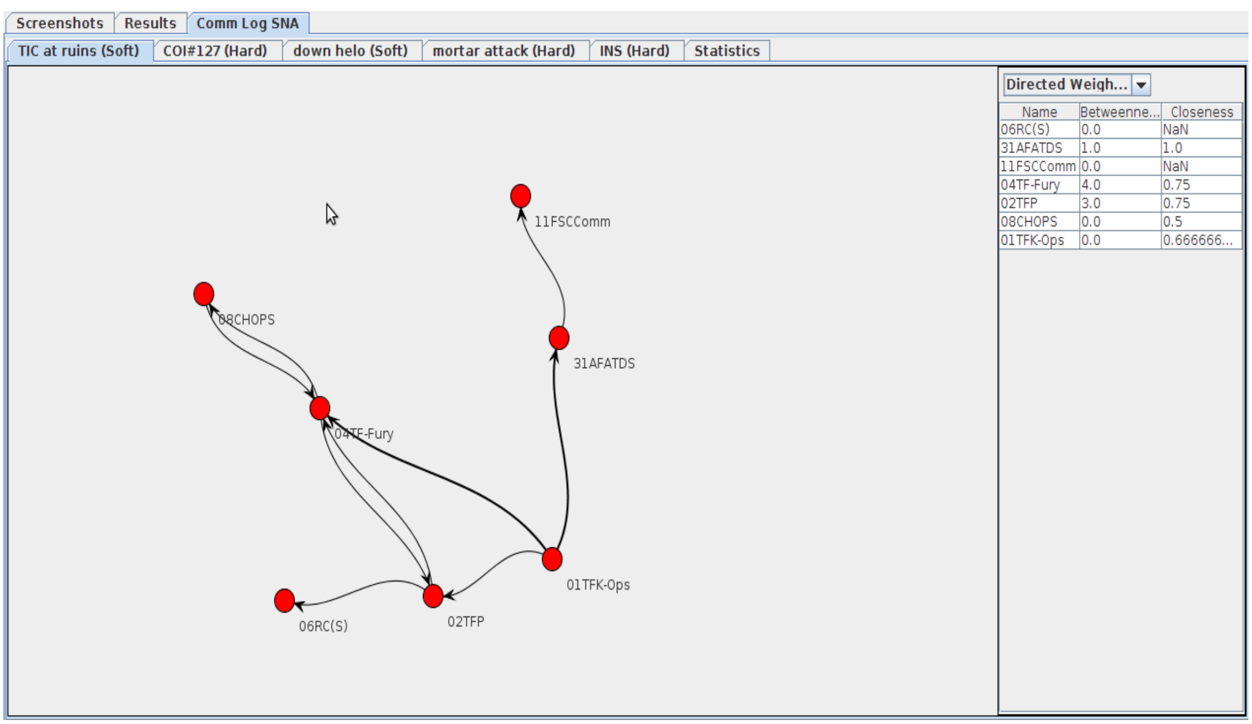

Figure 11: *Visualization of overall social network associated with a given work object*

Figure 12 shows the list of global statistics that are computed by the system considering all work objects. For each operator (column 1), the system compute the ratio of work object in which the operators was involved in at least one communication link (column 2). Then considering only the work object in which the operators were involved, the system computes the average number of communication that the operator initiated and the average number of communication for which they were a receiver. The example shown in Figure 12 indicates that the ASCC (first row) was involved in dealing with 40% of all work objects. For those work objects in which the ASCC was involved, he initiated on average 0 communications but received on average 1.92% of all communications associated with this work object.

Screenshots | Results | Comm Log SNA

TIC at ruins (Soft) | COI#127 (Hard) | down helo (Soft) | mortar attack (Hard) | INS (Hard) | Statistics

Operator Name	Ratio of call involvement	Average number of time "Initiator"	Average number of time "Receiver"
16 - ASCC	0.4	0.0	0.01923077
09 - SigOps	0.2	0.009615385	0.0
31 - AFATDS	0.2	0.01923077	0.009615385
10 - JFSC	0.6	0.14423077	0.115384616
15 - Avn	0.6	0.028846154	0.03846154
12 - JAG	0.2	0.01923077	0.0
13 - STACC Hi-Locon	0.2	0.0	0.009615385
11 - FSCComm	0.8	0.10576923	0.01923077
08 - CHOPS	0.8	0.09615385	0.01923077
29 - Log	0.6	0.0	0.03846154
01 - TFK-Ops	1.0	0.0	0.4903846
13 - STACC	0.6	0.17307693	0.009615385
07 - TF1-12	0.2	0.0	0.009615385
06 - RC(S)	0.6	0.048076924	0.0
27 - AirLocon	0.4	0.009615385	0.03846154
04 - TF-Fury	0.4	0.03846154	0.028846154
02 - TFP	0.4	0.01923077	0.028846154
03 - TF3-09	0.8	0.20192307	0.03846154
14 - TACP	0.2	0.0	0.009615385
17 - FSCC	0.6	0.03846154	0.028846154
26 - FSCC-BG	0.4	0.057692308	0.009615385
21 - J2Ops	0.4	0.115384616	0.0
22 - J2Plans	0.2	0.009615385	0.0

Figure 12: ***Statistics for each operator indicating the ratio of mission thread involvement and the average number of initiated and received communication for the mission thread in which they were involved***

5. CONCLUSION

We reported on a semi-automated system developed to identify the process followed by a team of operators when accomplishing a series of tasks. The system was recently developed and was tested in recent experiments. This system supports the development of a model-based approach to experimentation.

The current usage of the system has been to focus on specific processes such as the fire support process, and the airspace management process. However, as mentioned in Section 1.1 (see the CTEF model discussion), the assessment of the overall process aspects need to include managerial tasks such as the motivation of team members, the maintaining of a shared strategy, and the adaptation to changes. These tasks are currently stored in the Miscellaneous tag, but work is on-going for developing models for these activities and adding assessment tools to analyze the adequacy of those tasks.

Other on-going work includes the development of better visualization tools for the output of the analysis as well as tools to enable the development of the expected tags from DNDAF diagrams.

References

[1] Essens, P. J. M. D., A.L.W Vogelaar, J.J.C. Mylle, C. Blendell, C. Paris, S.M. Halpin, and J.V. Baranski, J. V. 2007. “Team effectiveness in complex settings: A framework for commanders.” In: E. Salas, G. F. Goodwin, & C. S. Burke (Eds) Team Effectiveness in Complex Organizations: Cross-disciplinary Perspectives and Approaches. SIOP Frontier Series. Mahwah, N.J.: Lawrence Erlbaum Associates.

[2] England, G.W. 1969. “Managerial value systems.” In Emerging Concepts in Management. Max S. Wortman and Fred Luthans Eds. Collier Macmillan Ltd.

[3] Endsley, M. R. 1995. “Toward a theory of situation awareness in dynamic systems.” Human Factors, 37, pp.32-64.

[4] Griffin, D. and A. Tversky. 1992. “The weighing of evidence and the determinants of confidence.” Cognitive Psychology, 24, 411-435.

[5] Bingi, P., D. K., Turnispeed, and G. M. Kasper. 2001. “The best laid plans of mice and men: The role of decision confidence in outcome success.” North American Journal of Psychology, 3, pp.91-108.

[6] Soller, A., 2002. “Computational analysis of knowledge sharing in collaborative distance learning.” Doctoral Dissertation, University of Pittsburgh, PA.

[7] Webb, N, 1992. “Interaction in cooperative groups: The theoretical anatomy of group learning.” R. Hertz-Lazarowitz and N. Miller (Eds.), Cambridge University Press, p. 105.

[8] DeShon, R.P., S.W.J., Kozlowski, A.M. Schmidt, K.A. Milner and D. Wiechmann. 2004. “A multiple-goal, multilevel model of feedback effects on the regulation of individual and team performance.” in Journal of Applied Psychology, Vol. 89, No. 6, pp. 1035-1056; December 2004.

[9] Utting, M. and B. Legeard. *Practical Model-Based Testing: A Tools Approach*. Morgan-Kaufmann, 2007.

[10] Special Report of the Senate Committee on Homeland Security and Governmental Affairs. *Hurricane Katrina: A Nation still unprepared*. U.S.A. S. Report 109-322. 2006

[11] Allen, D., F. Lichacz, and K. Wheaton, 2010. “Observational studies of a joint fire support coordination cell.” In proceedings of the 15th International Command and Control Research and Technology Symposium, June 2010. Santa Monica, CA.

[12] Tiwari, A., C.J. Turner, and B. Majeed. A Review of Business Process Mining: State-of-the-Art and Future Trends. Business Process Management Journal, Vol 14, No. 1, pp. 5-22, 2008.

[13] Günther, C.W., and W.M.P. van der Aalst. Fuzzy Mining – Adaptive Process Simplification Based on Multi-Perspective Metrics. In G. Alonso, P. Dadam, and M. Rosemann (Eds.) LNCS Vol 4714, pp.328-343. Springer-Verlag, Heidelberg, 2007.

[14] Mutschler, B., M. Reichert, and J. Bumiller. Unleashing the Effectiveness of Process-Oriented Information Systems: Problem Analysis, Critical Success Factors, and Implications. IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 38, No 3, pp. 280-290. May 2008..

[15] Defence Enterprise Architecture. 2008. “DND/CF Architecture Framework Version 1.7” Department of National Defence Information Management Group, April 2010.

[16] Levenshtein VI. 1966. "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady 10: 707–10.

[17] Emond, E.J. and D.W. Mason. 2002. “A new rank correlation coefficient with application to the consensus ranking problem.” Journal of Multi-Criteria Decision Analysis, Jan/Feb 2002, Vol. 11 Issue 1, p. 17-28.