Air Traffic Management Simulation Data Visualization and Processing Tool

Todd J. Callantine*
San Jose State University/NASA Ames Research Center, Moffett Field, CA, 94035-1000

Large-scale distributed simulations in which pilot participants fly simulators through multiple airspace sectors managed by controller participants create a rich operational environment for investigating new air traffic management concepts. This paper describes a Java™-based tool that aids in integrating, visualizing, and transforming data collected from large-scale human-in-the-loop air traffic management simulations.

I. Introduction

FUTURE Air Traffic Management (ATM) concepts commonly leverage flight deck and ground-based automation and new interface tools to improve efficiency. To evaluate the feasibility and benefits of proposed concepts, researchers need ways to assess changes in practitioner roles and responsibilities, in addition to traditional system performance, human performance, and acceptability metrics. One way to investigate air-ground integration issues is to use large-scale simulations in which pilot participants fly simulators through multiple airspace sectors managed by controller participants, creating a rich ATM environment. This paper describes a Java™-based tool called DProc that aids in visualizing, integrating and transforming data collected from large-scale ATM simulations.

DProc was implemented to process data from simulations in the Airspace Operations Laboratory (AOL) at NASA Ames Research Center. Controller stations, piloted and pseudo-piloted aircraft simulations—as well as simulation manager and 'host computer' components—all produce data describing particular aspects of an overall simulation. A given controller station, for example, logs actions performed by that air traffic controller subject, while the host computer provides a repository for aircraft state and trajectory information and logs descriptive data. Researchers may be interested in tracing event sequences, information flows, and operational contexts associated with certain outcomes of interest. This may entail identifying what other human subjects or automation agents are doing when one performs an action, and measuring relationships such as the time between various events or actions.

Analysts can, of course, use traditional tools like spreadsheets to examine ATM simulation data and analyze interactions. However, the different types, formats, and sheer quantity of data can make this difficult. Moreover, data collected in a research environment with prototype automation tools may require integrity checking and other preprocessing steps prior to analysis. DProc addresses these issues by creating a database of merged simulation data from various sources. The DProc interface enables researchers to replay simulated traffic and visualize recorded events together with aircraft states. Visualization data may be color-coded according to traffic characteristics (e.g., aircraft weight class, equipage, engaged autoflight modes), or filtered to highlight events associated with a particular controller, aircraft, or class of aircraft. In addition to replaying data, DProc is also capable of producing plots of aircraft tracks or event locations.

In addition to visualizations, DProc also produces batch output suitable for input to a spreadsheet or other analysis tool. As an example, the paper notes how DProc produces input files for data mining applications. Finally, the paper describes potential enhancements to DProc. For example, audio and video, as well as other subjective data collected during or after a simulation run might also be integrated and used to support analysis.

II. Background

The NASA Ames AOL supports large-scale, distributed human-in-the-loop ATM simulations. The AOL has control rooms for Center and TRACON air traffic controllers, as well as pseudo-aircraft operators and simulation coordinators. Piloted flight simulators in other laboratories can connect to and participate in AOL simulations. A voice communication network enables pilot-controller radio communications.

---
* Senior Research Engineer, Human Factors Research and Technology, Mail Stop 262-4, AIAA Member.

The AOL simulation infrastructure used in recent en route Distributed Air Ground Traffic Management (DAG-TM) simulations[1] is shown in Figure 1. Principle components are the Multi Aircraft Control System (MACS) and the Aeronautical Data link and Radar Simulation (ADRS).[2] MACS stations may be configured as piloted flight simulators, pseudo-pilot multi-aircraft control stations, Display System Replacement (DSR) or Standard Terminal Automation Replacement System (STARS) air traffic controller stations, or as a simulation manager console. ADRS processors perform host computer functions and link the various MACS stations to flight simulators in other laboratories and other automation tools.

This paper focuses on an application of DProc to analyze the results of a DAG-TM simulation of terminal-area Flight Management System (FMS) arrivals with airborne spacing and merging[3] airborne spacing, traffic scenarios began with traffic that was well coordinated to merge, and ended with poorly coordinated traffic. Some aircraft from one meter fix were assigned to an alternate runway; aircraft from the other meter fix filled primary runway landing slots vacated by these aircraft. After obtaining ownership of each aircraft from a Center controller confederate, a Feeder controller issued a clearance to continue the descent on the terminal area FMS routing. The Feeder controller could adjust the aircraft's spacing before handing the aircraft off to the Final controller, who was responsible for merging the flows, establishing the proper interarrival spacing, and handing the aircraft off to a Tower controller confederate. Outcomes of interest included methods used to accomplish these functions, as well as controller strategies that dictate how methods are applied and coordinated. A variety quantitative performance metrics, such as spacing accuracy, also required computation. Simulation conditions examined operations with and without air and ground-side tools designed to aid the pilots and controllers.

The next two subsections describe the contents of the MACS and ADRS output data. A third subsection then provides background on the objectives of ATM data analyses and discusses the importance of data visualization for meeting these objectives. It also presents examples of related research on ATM data visualization.

A. MACS Output Data

Figure 2 shows notional classes encapsulating simulation data recorded by MACS. First, `MACSPilotData` contains information MACS records whenever pilots or pseudo-pilots perform control actions (e.g., engaging an autoflight mode). `MACSPilotData` includes information about the event itself (`event` and `event_data`) together with an assortment of information about the state of the aircraft. `MACSATCEventData` similarly encapsulates the information MACS records information about controller activities, while `MACSATCStateData` holds information about the traffic and ATC automation behavior, including detected conflicts. Each pilot or pseudo-pilot station also records aircraft state data (not shown in Figure 2).

MACSPilotData			
aircraft_id	x	meter_fix	cpdlc_equipped
time	y	next_waypoint	self_spacing_equipped
raw_time	ground_speed	runway	spacing_status
who	airspeed	meter_fix_eta	lead_aircraft
event	altitude	meter_fix_eta_reported	spacing_interval
event_data	magnetic_heading	meter_fix_sta	
owner	cruise_altitude	eta_sta_diff	
geo_sector	speed_target	datalink_equipped	
ac_type	altitude_target	rta_equipped	
flight_type	heading_target	free_flight_equipped	

MACSATCEventData			
aircraft_id	ac1_free_flight_status	meter_fix_eta_reported	trajectory_errors
controller	ac2_free_flight_status	meter_fix_sta	lnav
time	x_pos	faf_eta_displayed	vnav
raw_time	y_pos	faf_eta_reported	self_spacing_status
ac_type	ground_speed	faf_sta	lead_aircraft
ac_owner	airspeed	full_datablock_speed	spacing_interval
geo_sector	altitude	conflict_aircraft	who_triggered
flight_type	heading	conflict_probe	event
datalink_equipped	next_waypoint	active_trial_plan	event_data
rta_equipped	meter_fix	time_to_conflict	additional_data
free_flight_status	final_approach_fix	min_predicted_lat_sep	
cpdlc	runway	min_predicted_vert_sep	
self_spacing_capable	meter_fix_eta_displayed	4d_trajectory_status	

MACSATCStateData			
aircraft_id	mins_to_initial_los	rta_equipped	meter_fix_ata
time	vertical_sep	free_flight_equipped	meter_fix_eta_displayed
raw_time	lateral_sep	cpdlc_equipped	meter_fix_eta_reported
event	ac1_ac_owner	self_spacing_capable	meter_fix_sta
event_data	ac1_geo_sector	x_pos	delta_arrival_time
ac_owned_by_controller	ac2_ac_owner	y_pos	self_spacing_status
ac_controlled_by_pilot	ac2_geo_sector	ground_speed	lead_aircraft
ac_in_geo_area	ac1_free_flight_status	airspeed	spacing_interval
afr_ac_owned_by_controller	ac2_free_flight_status	altitude	
ifr_ac_owned_by_controller	ac_type	heading	
afr_ac_controlled_by_pilot	ac_owner	next_waypoint	
ifr_ac_controlled_by_pilot	geo_sector	meter_fix	
afr_ac_in_geo_area	flight_type	final_approach_fix	
ifr_ac_in_geo_area	datalink_equipped	runway	

Figure 2. MACS pilot data, ATC event data, and ATC state data.

B. ADRS Output Data

An ADRS outputs two important types of data in its role as ATC host computer (Figure 3). The first is the state of all flight simulators participating in the simulation on each 1 Hz update. ADRSStateData is useful for checking the integrity of state information recorded in other output. The second type of data is Flight Management System (FMS) trajectories for all participating flight simulators, logged whenever the trajectory changes (including when an aircraft sequences a waypoint). The trajectory data include a list of all the current trajectory points represented as an array of ADRSTrajPoints contained within an instance of ADRSTrajData. The next subsection provides background on the importance of tools for ATM data analysis.

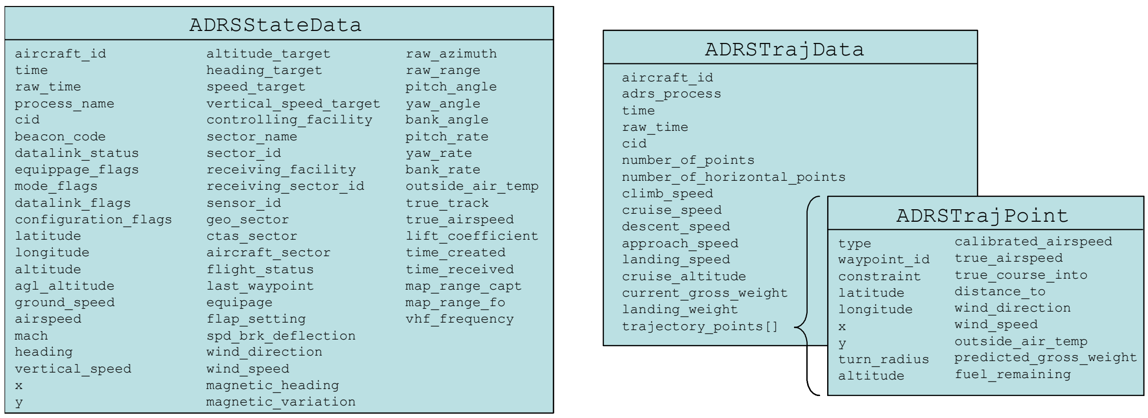

Figure 3. ADRS state and trajectory data.

C. ATM Data Analysis Support Tools

The large volume of data of various types in different formats is one reason for developing tools to support ATM data analysis. Merging data from multiple sources enables a tool to check data for integrity before computing performance metrics. A tool with data parsing and batch processing capabilities can produce performance metrics more rapidly than otherwise be possible. This helps to make the latest, reliable simulation results available in a timely fashion to support iterative ATM concept development. A tool may also be tailored for transforming data to produce metrics that are not directly recorded. For example, DProc infers clearances from pilot actions, under the assumption that actions are performed to comply with clearances (a slight time lag is understood; pilots generally take, say, three to ten seconds to execute inputs to comply with clearance). A tool can also provide insights researchers might otherwise miss by enabling them to visualize and thereby rapidly identify interesting classes of traffic and air-ground interactions.

Data visualization is by now well accepted as an important first step in data mining—examining data to find hidden structure.[4] Indeed, data visualization is a central component of other initiatives aimed at improving the ATM system. A prime example is the Performance Data Analysis and Reporting System (PDARS) used by the Federal Aviation Administration.[5] PDARS computes performance metrics from actual flight data recordings and uses them to create visualizations analysts can use to rapidly identify and examine characteristics of particular traffic classes (e.g., aircraft that flew a particular Standard Terminal Arrival Routing).

Visualization is also useful for understanding interactions among agents in the ATM system. For example, previous research illustrated how visualizing interactions between the flight crew and cockpit automation is useful for examining the effects of using the FMS for terminal-area arrivals on flight crew activities. In that research, visualizations graphically illustrated increased flight crew task loads when attempting to resume FMS arrivals interrupted by tactical clearances.[6]

The present work applies both traffic and interaction visualization to the analysis of human-in-the-loop ATM simulation data. It attempts to support tracing event sequences and capturing operational contexts associated with interesting events. This entails identifying what other humans or automation agents are doing when one performs an action, and measuring relationships (e.g, the time between various events or actions). Identifying and understanding key interactions is essential for evaluating safety and potential benefits of new concepts. The next section describes the DProc tool in detail.

III. DProc Data Visualization and Processing Tool

DProc is a Java™-based tool designed to support integration, visualization, and transformation of human-in-the-loop ATM simulation data from the NASA Ames AOL. The data files for a particular simulation trial are specified in a configuration file. An analyst begins a DProc session by selecting the desired simulation trial or batch-processing mode. Figure 4 shows the DProc configuration window. Selecting a particular simulation trial leads to a second selection window for choosing one of the two parallel simulations from which to view data.

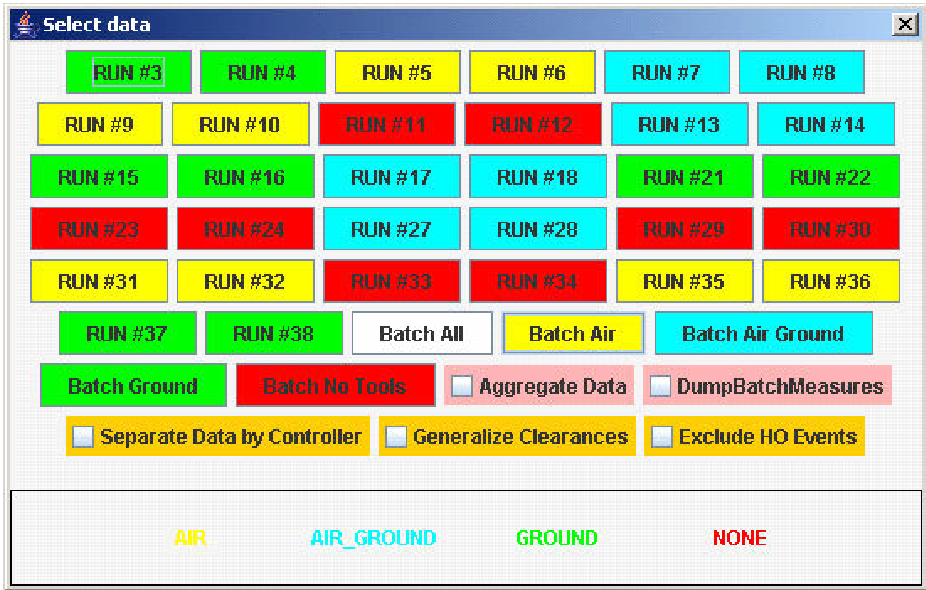

Figure 4. DProc configuration window.

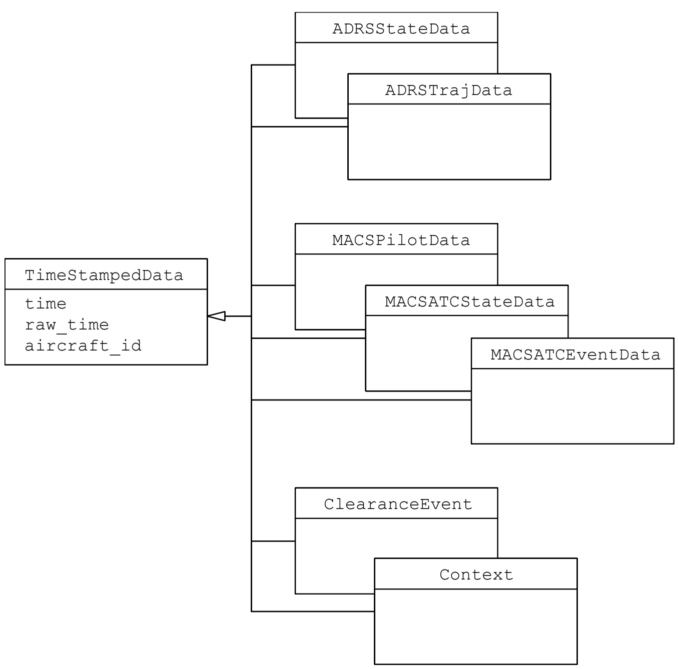

Figure 5. DProc class hierarchy.

When the analyst selects a specific trial to examine, DProc parses and merges the required data into a database that enables rapid retrieval and display of data at a particular time. DProc represents the AOL simulation data described in sections II.A and II.B above as subclasses of `TimeStampedData`, as shown in Figure 5.

Figure 5 also shows two additional subclasses of `TimeStampedData`: `ClearanceEvent` and `Context`. Using a set of filters and a simple model of actions pilots nominally perform to comply with certain clearances, DProc infers clearance instructions from `MACSPilotData` and represents them as `ClearanceEvent` objects, which also contain the ADRS data for the aircraft at the associated time. To support detailed analysis of controller activities, `Context` objects not only encapsulate a particular `ClearanceEvent`, but also the previous `ClearanceEvent`, the states of lead and trail aircraft and the most recent `ClearanceEvents` for lead and trail aircraft. Thus, a `Context` object contains a variety of information useful for understanding the context surrounding a controller's decision to issue a clearance to a particular aircraft. It is not required for basic analyses, but is useful for explorations of controller strategies.

D. DProc Visualization Functionality

Figure 6 shows the main DProc window for replaying and plotting data. Simple mouse actions control replay; a few keyboard shortcuts are available for toggling between airspeed and groundspeed display in the aircraft datablocks and selecting standard views. The slider control enables forward or backward replay.

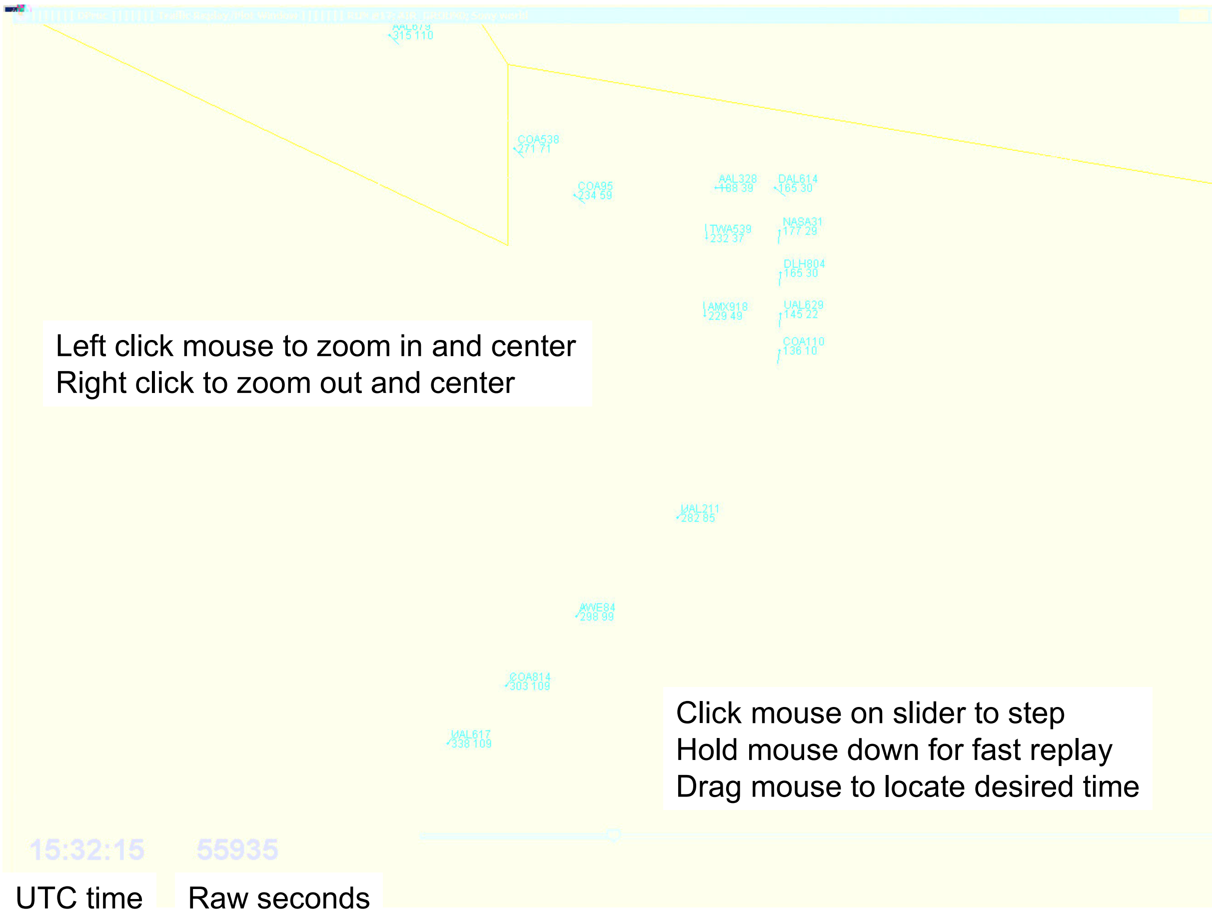

Figure 6. DProc traffic replay/plot window.

In addition to the main display, DProc also creates an Options window for specifying visualization options (Figure 7). The analyst uses the topmost controls to activate either interactive replay or plotting functionality. The second block specifies whether to visualize raw data or derived clearance information. The remaining controls apply various filters. In this implementation, the analyst can specify whether DProc should replay or plot clearance data for one or both controllers. The analyst can also select which aircraft to replay or plot, and which types of inferred clearances are of interest. The aircraft identifiers are color-coded to indicate which are equipped for airborne spacing (pink), and which are MACS flight simulators flown by subject pilots (white). Finally, when the plot mode is selected, the 'Dump Excel Data' button is enabled, so that analysts can export the data currently plotted to a comma-separated data file readable by a spreadsheet program.

Figure 8 is a screen snapshot of DProc replaying raw simulation data. As the slider is moved, DProc displays the traffic state using `ADRSStateData`, along with any `ADRSTrajData` for the currently selected time. The ADRS records trajectory data anytime aircraft FMS trajectories change; thus, Figure 8 shows `ATCEventData` indicating that a waypoint is sequenced at the selected time, along with the updated `ADRSTrajData`.

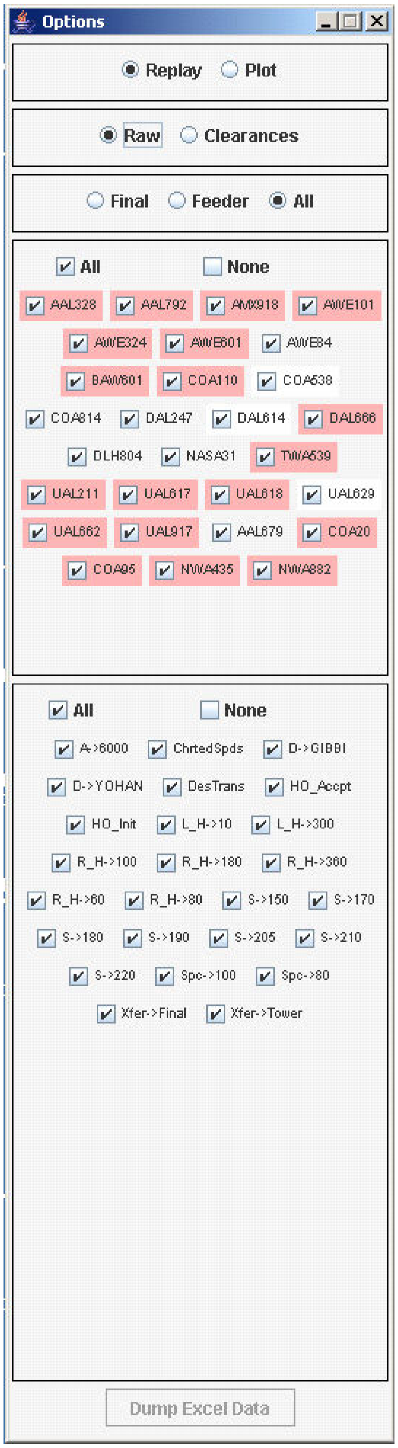

Figure 7. DProc visualization options window.

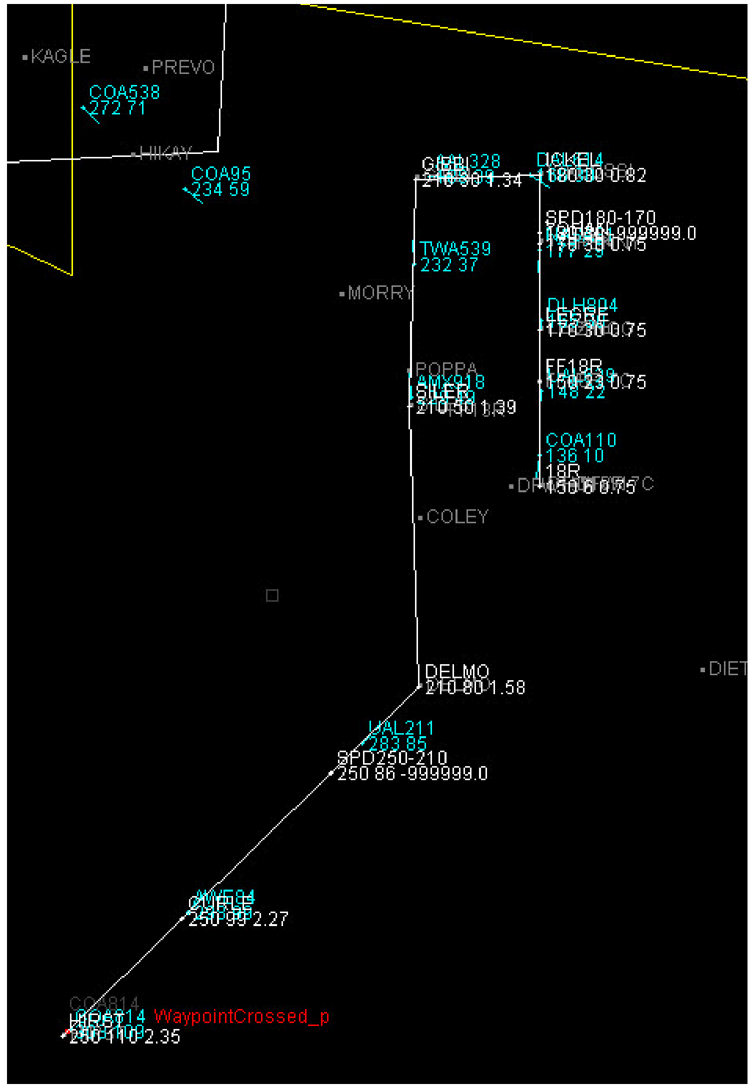

Figure 8. Raw data replay.

Figure 9 shows a screen snapshot from the clearance replay mode. Clearance replay mode is useful for visualizing the order in which air traffic controllers address the needs of aircraft in various situations. A text description of the clearance is constructed within each `ClearanceEvent` object for use in this mode. Where possible the text description also includes contextual information (available in the raw `MACSPilotData`). For example, Figure 9 shows a spacing clearance issued by the Final controller. The text description includes the current temporal spacing to the lead aircraft at the time the clearance is inferred to have been issued. (For consistency, Figure 9 and all subsequent screen snapshots are from the same simulation trial as Figures 6 and 8).

DProc also provides data plotting functionality. Figure 10 shows the DProc plot created by selecting plot mode for clearance data for both controllers, and checking only the 'Spcg->80' and 'Spcg->100' boxes (for airborne spacing clearances with 80 and 100 s temporal intervals, respectively). The plot indicates that both the Feeder controller (clearances with orange labels) and the Final controller (yellow labels) employed airborne spacing clearances, but the Feeder issued more than the Final controller in this particular

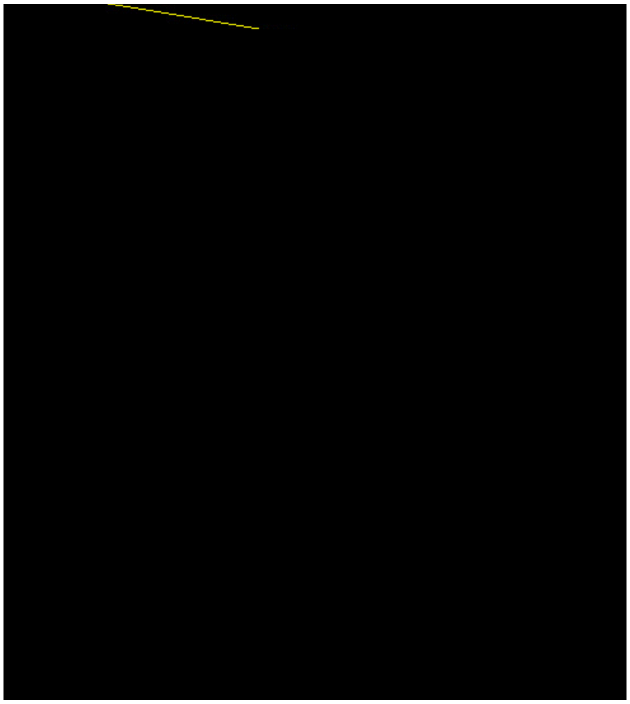

Figure 9. Clearance replay.

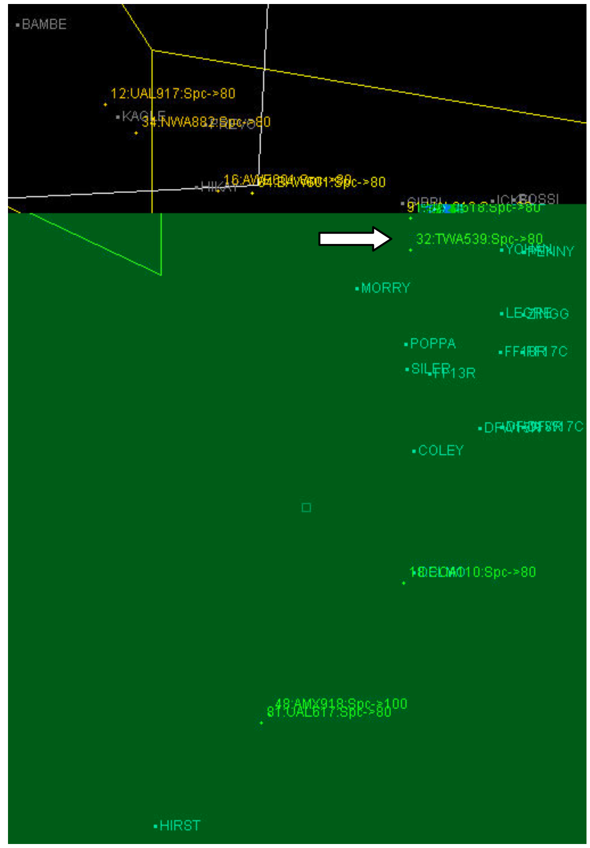

Figure 10. Plot of all airborne spacing clearances.

trial. (The numbers that precede the clearance labels are the clearance number, e.g., the spacing clearance issued to TWA539 was the thirty-second clearance DProc inferred the Final controller to have issued.)

Figure 10 provides an example of the value of visualizations for guiding further analysis. A hypothesis about controller strategy for the terminal area self-spacing concept held that the Feeder controller would use an arrival timeline tool to assess the spacing of aircraft as they entered the terminal area. The Feeder would use speed clearances or, in more extreme cases, heading vectors, to adjust aircraft to be ‘close’ to their assigned scheduled time of arrival (STA). Then, the Final controller would use airborne spacing to ‘lock in’ desired temporal spacing intervals. Figure 10 indicates that, in this trial, the Feeder controller also sought to use airborne spacing to achieve and maintain desired spacing values for sequential aircraft. An examination of the context surrounding the Feeder controller spacing instructions in this trial shows that in every case, the aircraft were currently spaced too closely, and in four of the six cases, the aircraft were approximately ten seconds too close. A plot of all heading vector clearances (Figure 11) indicates that while the Feeder controller did make significant adjustments to the traffic flow, issuing all but one of the heading vectors observed in the given trial, the Feeder may have also expected the airborne spacing guidance to achieve the required spacing when the temporal spacing was within approximately ten seconds. This observation may warrant possible refinements of requirements for spacing guidance performance, timeline tool design, and/or controller training.

Another set of plots focuses on handoff acceptance and transfer of control of aircraft. Ownership-related activities such as these are interesting as indications of workload and attention to the state of aircraft exiting a sector. Figure 12 shows the handoff acceptance locations for both controllers (Feeder in orange, Final in yellow). The locations were fairly consistent in all but one case, in which the Feeder accepted a handoff of a southwest arrival fairly late. Figure 13 shows the locations at which control was transferred to the Final controller; again, all but two transfers were consistent; additional analysis is required to determine to reason(s) behind the late transfers. Figure 14 shows a pattern observed across trials in which the Final controller had tools available to assess spacing: the Final controller tended to maintain control of aircraft longer, which suggests the Final controller attended to spacing more closely and went to greater lengths to more precisely achieve required spacing when support tools were available.[3] Lastly, Figure 15 illustrates a plot, color-coded by altitude, of all the arrival tracks in the selected trial. DProc can also be configured to color-code track plots and data blocks according to aircraft type or level of FMS usage (e.g., Lateral Navigation (LNAV) mode engaged, Vertical Navigation (VNAV) mode engaged, or both engaged).

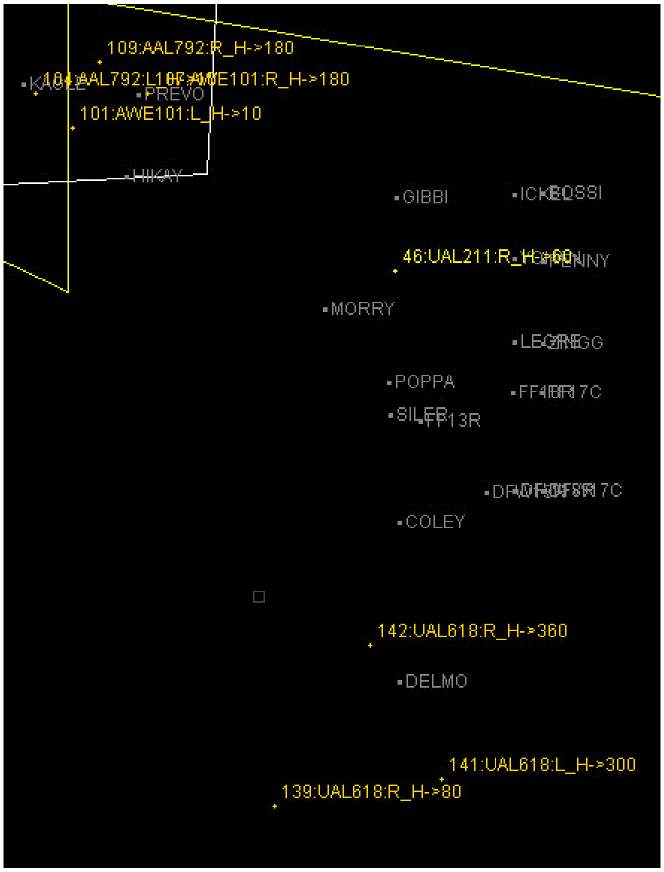

Figure 11. Heading vector clearance plot.

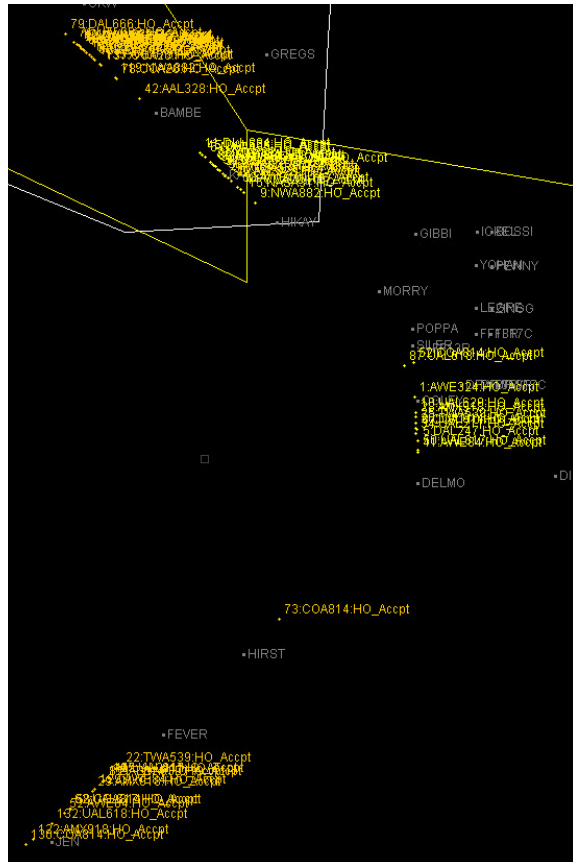

Figure 12. Handoff acceptance plot.

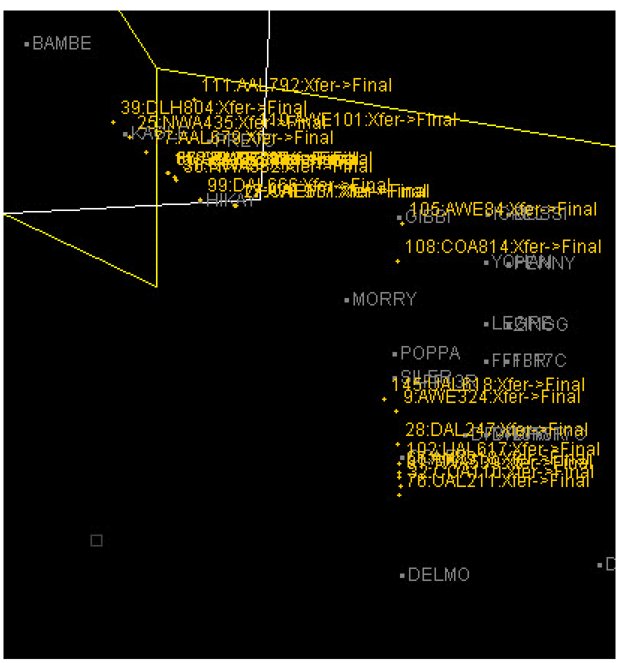

Figure 13. Transfer from Feeder to Final controller.

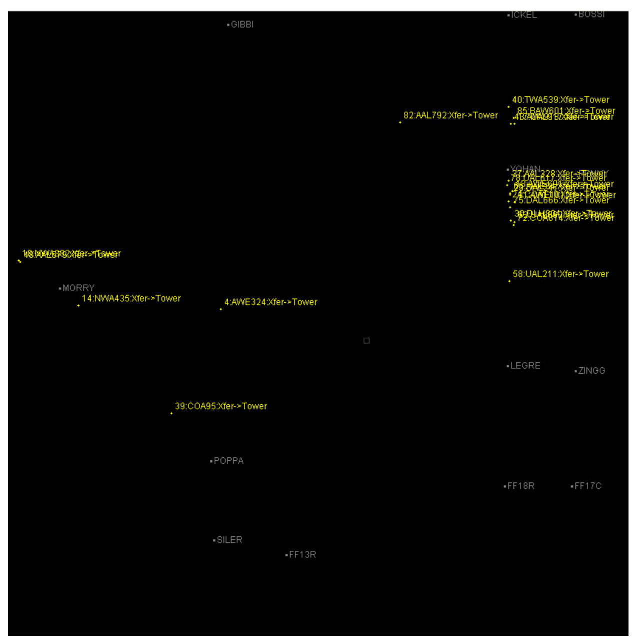

Figure 14. Transfer from Final controller to Tower (zoomed out to show transfers for runway 13R).

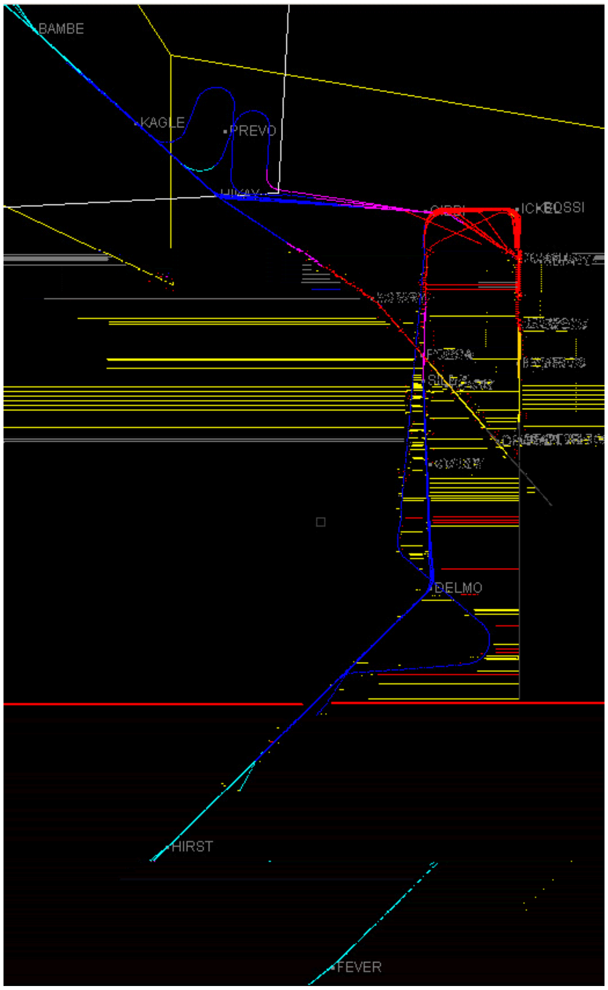

Figure 15. Track plot with colors indicating aircraft altitude.

E. DProc Batch Processing Functionality

As shown in the configuration window in Figure 4, DProc also provides several batch processing options geared toward integrating and outputting data for use in spreadsheets or other analysis tools. First, an analyst may choose to batch-process a single experimental condition or all the data at once. Checkboxes control the content of the batch output. The ‘Aggregate Data’ option produces one output file for the batch. The ‘Dump Batch Measures’ option produces performance metrics (e.g., throughput, spacing accuracy data, etc.) in addition to visualization and plot data. ‘Separate Data by Controller’ produces separate output files for Feeder controller and Final controller data. ‘Generalize Clearances’ specifies the level of granularity at which to analyze inferred clearances. Selecting it enables analyses of clearance *types*, without regard to precise clearance values (e.g., all speed clearances, as opposed to speed clearances to specific speeds). This option may also be used in interactive visualization and plot modes. Finally ‘Exclude HO Events’ enables an analyst to selectively remove aircraft ownership-related events from consideration. This is useful for analyses focused solely on clearance instructions that require aircraft to maneuver.

With ‘Aggregate Data’ selected, DProc also produces Attribute-Relation File Format (.arff) output files used by the Weka collection of machine learning algorithms for data mining.[4] This functionality supports preliminary research on mining data to identify patterns of controller activity that support particular control strategies. Efforts in this area have identified interesting differences among models of controller behavior learned using various schemes, but the research is still in its infancy.

IV. Extensions

Several extensions to enhance the usability and usefulness of DProc as a data analysis tool are possible. Simple visualization and plot extensions might include configuration windows for data block information and color-code selection. Other extensions contribute more to core DProc analysis capabilities. Foremost among these is the inclusion of recorded digital audio and video recordings of controller and pilot activities. In NASA Ames AOL simulations, digital audio data are collected from the voice communications system and movies of controller interfaces that include mouse cursor activities are recorded during simulation trials. The capability to rapidly access a slice of audio or video for a particular time from within DProc would enable analysts to better assess air-ground interactions and the use of decision support tools. Commercial products have been developed for integrating and viewing audio-visual data, but they have typically been relatively expensive and limited to audio-visual data.

A second extension is to include subjective data collected from participants. For example, in the DAG-TM terminal-area airborne spacing simulation, data were collected from Workload Assessment Keypads (WAKs) at each controller position in the AOL at five minute intervals during each simulation trial. Data such as these could be added to the configuration file and integrated with the other digital data. Analysts could focus on time intervals in which controllers reported workload to be high and use visualizations to determine the reasons for it.

Another obvious, useful extension is the capability to print visualizations and plots as graphics or hardcopies. DProc presently requires the analyst to obtain and further manipulate screen snapshots, which needlessly increases the difficulty of presenting interesting findings. Finally, DProc could benefit from a mechanism to easily configure the contents of data representation classes and control parsing. At present, the DProc code must be modified anytime researchers change the content or format of simulation output files.

V. Conclusion

DProc helps analyze data from large-scale ATM simulations. DProc automatically integrates and transforms data, and provides useful visualization and plotting functionality. While in some situations it may identify more questions about observed interactions than answers, it generally makes data more accessible and inspectable, and thereby improves analysis quality. This paper has presented some examples of DProc capabilities, and indicated how DProc could be extended to improve its usefulness. DProc is slated for continued use for analyzing data collected in upcoming AOL simulations.

Acknowledgments

This work was supported by the Distributed Air Ground Traffic Management Element of the NASA Airspace Systems Program Advanced Air Transportation Technologies Project. Thanks to AOL researchers Dr. Everett Palmer, Dr. Thomas Prevôt, Vernol Battiste, Nancy Smith, Dr. Paul Lee, and Joey Mercer.

References

[1] Prevôt, T., Sheldon, S., Palmer, E., Johnson, W., Battiste, V., Smith, N., Callantine, T., Lee, P., Mercer, J., "Distributed Air/Ground Traffic Management Simulation: Results, Progress and Plans," AIAA-2003-5602, American Institute of Aeronautics and Astronautics, Reston, VA, 2003.

[2] Prevôt, T., Palmer, E., Smith, N., and Callantine, T., "A Multi-Fidelity Simulation Environment for Human-In-The-Loop Studies of Distributed Air Ground Traffic Management," AIAA-2002-4679, American Institute of Aeronautics and Astronautics, Reston, VA, 2002.

[3] Callantine, T., Lee, P., Mercer, J., Prevôt, T., and Palmer, E., "Air and Ground Simulation of Terminal-Area FMS Arrivals with Airborne Spacing and Merging," *Proceedings of the 6th USA/Europe Air Traffic Management Research and Development Seminar*, Baltimore, MD, June, 2005.

[4] Witten, I., and Eibe, F., *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., Morgan Kaufmann, San Francisco, 2005.

[5] den Braven, W., and Schade, J., "Concept and Operation of the Performance Data Analysis and Reporting System," SAE Technical Paper 2003-01-2976, SAE International, Warrendale, PA, 2003.

[6] Callantine, T., and Crane, B., "Visualizing Pilot-Automation Interaction," In Abbot, K., Speyer, J., and Boy, G. (eds.), *Proceedings of the HCI-Aero 2000 International Conference on Human-Computer Interaction in Aeronautics*, EURISCO, Toulouse, France, 2000, pp. 87-92.