

Modeling and Simulation of an Integrated Gate Turnaround Management Concept

William Chung, Doug Ahlquist, Carla Ingram, Girish Chachad, Spencer Monheim

SAIC

NASA Ames Research Center

william.w.chung@nasa.gov, douglas.k.ahlquist@nasa.gov, carla.d.ingram@nasa.gov,

girish.h.chachad@nasa.gov, spencer.c.monheim@nasa.gov

ABSTRACT

An Integrated Gate Turnaround Management (IGTM) prototype was developed at NASA Ames Simulation Laboratories (SimLabs) using Dallas Ft. Worth International Airport (DFW) to demonstrate the IGTM concept's feasibility and benefits. The simulation architecture includes: the IGTM controller, an Airline Operations Control (AOC) application, Big Data/Analytics Input (BAI) application, a terminal traffic simulation or known as NASA-developed Surface Operation Simulator and Scheduler (SOSS), and a Database Server. ActiveMQ, a Java messaging service, was used to emulate the System Wide Information Management (SWIM) data network messaging. This paper describes the modeling and simulation of the IGTM concept, and illustrates selected use cases to demonstrate the feasibility and benefits of the IGTM concept for optimizing gate turnaround operations.

ABOUT THE AUTHORS

Mr. William Chung is the Chief Aerospace Engineer with SAIC at NASA Ames Research Center's SimLabs. He has been working on a wide range of flight simulations, including fixed-wing and rotorcraft, and NextGen projects as well as unmanned aircraft systems. He has a Master Degree in Aeronautics and Astronautics from Stanford University, a Master Degree in Mechanical Engineering from Oregon State University, and a Bachelor Degree of Science in Industrial Engineering from Chung Yuan University.

Mr. Doug Ahlquist is a software designer and developer with SAIC at NASA Ames Research Center's SimLabs. He has been in software development for more than 25 years. He had started in software before working on his BS at San Jose State University and continued at the University California Santa Cruz. He joined NASA in 2010 for Lunar satellite LADEE mission to work on a small team of 2 to design and implement the ground data systems, upon completion leaving to work in the Treasury Service division of JP Morgan Chase and came back to NASA's Flight Simulations team in 2014.

Ms. Carla Ingram is the technical lead for NASA's FutureFlight Central facility and the software manager of the Airspace Traffic Generator system. Carla currently works in the Aerospace Simulation Research and Development Branch at Ames Research Center and has been supporting air traffic management simulation research for over 18 years, covering the en route, TRACON, and airport domains. Her prior research experiences include high-fidelity rotorcraft research simulations. She has a B.S. degree in Mechanical Engineering from California State University at Chico.

Mr. Girish Chachad is a Marketing Manager at SAIC and serves as SimLabs Outreach Manager for NASA Ames Research Center's Outreach Team and Simulation Planning Office. Mr. Chachad has 36 years of engineering experience with real-time flight simulations, and in structural analysis and design. Mr. Chachad has an MBA from San Jose State University, an MS in Civil Engineering from the University of Rhode Island, and a BS in Technology for Civil Engineering from the Indian Institute of Technology.

Mr. Spencer Monheim is a software engineer with SAIC at NASA Ames Research Center's SimLabs. Currently working in Air Traffic Management simulations as a developer on the Airspace Traffic Generator, he has also provided data analysis for other projects. Prior to joining SAIC at NASA in early 2015 he worked with the Fermi Large Area Telescope Collaboration at University of California, Santa Cruz, where he earned a B.S. in Astrophysics with a research focus in data analysis of dark matter source candidates.

Modeling and Simulation of an Integrated Gate Turnaround Management Concept

William Chung, Doug Ahlquist, Carla Ingram, Girish Chachad, Spencer Monheim

SAIC

NASA Ames Research Center

william.w.chung@nasa.gov, douglas.k.ahlquist@nasa.gov, carla.d.ingram@nasa.gov,

girish.h.chachad@nasa.gov, spencer.c.monheim@nasa.gov

INTRODUCTION

In 2015, NASA Ames Research Center conducted the “Big Data, Data Analytics, and Net-enabled ATM and Airspace Operations Project” to identify air traffic management (ATM) functions that can benefit from networked, net-enabled, and/or cloud-based architecture. The objective of this effort was to investigate methodologies that could reduce duplication; reduce cost of operations and upgrades related to air traffic management, airline operations, and flight operations; and provide data analytics. Additionally, potential “fee for service” mechanisms for funding the development and operation of a solution application were to be considered. A subject matter experts (SME) team with experience in ATM, airline operations, airport operations, and big data architectures was assembled to tackle this challenge.

Five topics were developed by the team related to the National Airspace System (NAS) performance, they are i) Weather Impacts to Aviation, ii) Big Data and Modeling Infrastructure, iii) Integrated Gate Turnaround Management, iv) Flight Operations Control Management System, and v) Departure and Arrival Management. After a team review, the Integrated Gate Turnaround Management (IGTM) was selected to proceed with a prototype demonstration with the following guidelines.

- Use of simulation in the ATM environment
- Focus on airline side and ramp operations
 - Desire to leverage data to manage the uncertainties in nominal and non-nominal, and
 - Provide performance monitoring and collaborative decision tools to promote integrated gate turnaround operations to meet the on-tie and/or predictable performance upon arrival and departure at the terminal

CONCEPT AND BENEFITS OF THE INTEGRATED GATE TURNAROUND MANAGEMENT

The concept of Integrated Gate Turnaround Management (IGTM) is to leverage i) the data analytics technologies with multiple historical database and live data to establish bounds of uncertainties of dependent parameters associated with NAS performance, [1,2] ii) a distributed data network shared by stakeholders, [3] and iii) collaborative decision tools [4,5] to optimize the arrival and departure performance at an airport through en route, terminal, and the gate. The IGTM concept is shown in Figure 1.

By providing predictable traffic performance information under nominal and off-nominal conditions, e.g., adverse weather, high/low traffic density, runway condition due to rain and/or ice, available flight crew, available ground crew, ground stop orders, and number of transient passengers, collaborative decision tools can be developed for individual stakeholders to optimize the performance under an integrated environment. This coordinated optimization will prevent localized optimization that could lead to system-wide delay.

THE IGRM PROTOTYPE

The IGTM prototype was defined for the service space starting from the final metering fix upon arrival at an airport to takeoff or wheel-off. This will allow the prototype to be integrated with automation tools for terminal and en route airspace operations for future applications. The IGTM prototype covers the flight and ground operations from approach to runway, landing and wheels-down, to taxi-in from the active movement area to the in-bound transfer spot for handover by ATC to ramp control, lastly, to taxi in the ramp area to arrive at the assigned gate. The aircraft, once parked at the gate, would then undergo gate turnaround operations after which the aircraft reverses the process with gate pushback, taxi-out to the out-bound transfer spot from ramp control to ATC, taxi-out to the departure queue, and ending at takeoff to meet the departure slot time. This complex process involves the controlling

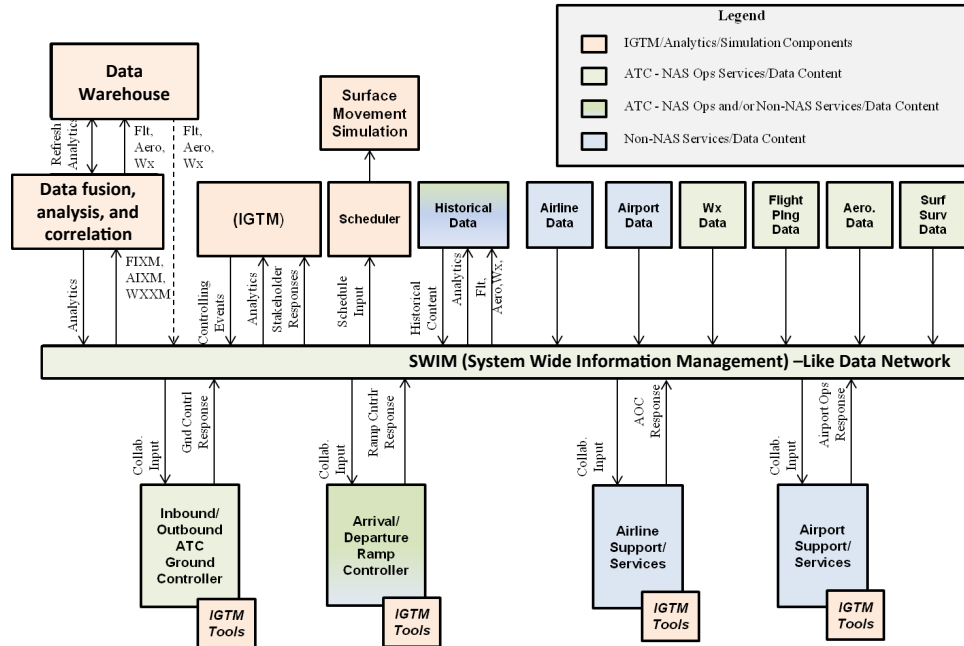


Figure 1. The IGTM concept

authorities and decision stakeholders which includes the ATC ground controllers, that have authority for the airport movement area, and the airport/airline ramp controllers, that have authority for the airport non-movement area. The airline operators, airline dispatchers, pilots and ground handling personnel who determine the readiness of the airplane are also stakeholders in this process.

The system architecture of the IGTM prototype is shown in Figure 2. The prototype has eight major components, listed below. Descriptions of these components are discussed in following paragraphs.

1. IGTM Model/Controller, which controls the IGTM simulation and collaborative decision tools.
2. AOC App, to simulate the Airline Operational Control Center
3. BAI App, to simulate the Big Data/Analytics interface and provide the analytical data.
4. Surface Operations Simulator and Scheduler (SOSS), to simulate live traffic data.

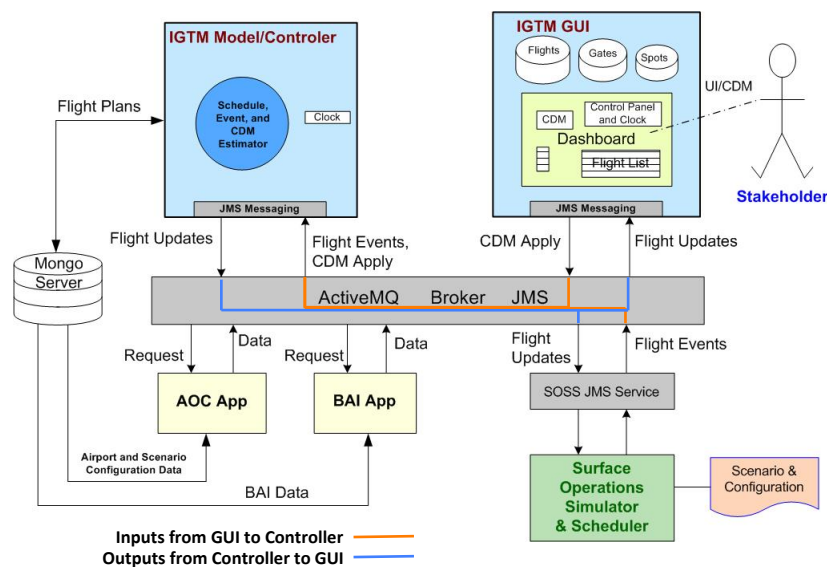


Figure 2. IGTM prototype's system architecture

5. SOSS JMS Service Adaptor, to translate data between the SOSS and the IGTM Controller.
6. IGTM Graphic User Interface (GUI) including the Dashboard, which provides the interface to the stakeholders.
7. ActiveMQ, an open source broker with a Java Messaging Service (JMS), provides the communications network for servicing the data exchanges.
8. Mongo server or MongoDB, an open source database server for the IGTM prototype.

IGTM Host System/Launch Scripts

The IGTM prototype is entirely written in JAVA and works with products, which are OS agnostic and can run on Linux, Windows or a MAC system. The prototype is a collection of 8 processes (See Figure 2, excluding the scenario configuration) that can be configured through XML to run on any number of systems. Scenario configuration is completed pre-simulation, by loading the desired simulated case files into MongoDB. The prototype was designed to allow any number of GUI/Dashboards users from multiple stakeholders and Model/Controllers exchange messages through the ActiveMQ Bus. By modifying the host, port, database and transport prefix in the various XML files one can reconfigure the process home system. Mongo itself can be configured to run on a clustered system solely by making changes in its XML files and the configuration files of the Model/Controller/BAI and the AOC components. These modifications can be made to increase the overall application performance. Modifications to the ActiveMQ configuration can increase the number of throughput channels and the volume of traffic allowed to pass through each channel.

The launch, configuration, and control of the IGTM processes were accomplished with several Linux Bash scripts, which were called from a master script. All major events are captured in log files and are visible in the standard output. Data collection for the events are captured on the MongoDB server.

IGTM Model/Controller

The IGTM Model/Controller, or Controller for short, is an event driven process. The Controller used the data from AOC App, which provided the airline scheduled data, BAI App, which provided the estimated event performance (in this demonstration the mean and standard deviation, σ , of a specified event performance from historical data was used), SOSS for actual flight event data, and commands from the GUI, which provides inputs from the stakeholders for collaborative decision making or gaming on a specific event. The events modeled in the prototype are listed below.

- i. Final metering fix
- ii. Wheels Down (or touchdown)
- iii. Inbound spot
- iv. Gate Arrival/Parked
- v. Gate turnaround (based on completion of all the gate activities)
- vi. Push back
- vii. Outbound spot
- viii. Departure queue
- ix. Wheels Up (Takeoff)

For each flight, three sets of time performance related data, Scheduled Time, Estimated Time, and Actual Time, are required for each event as shown in Figure 3. Scheduled event time, $T_{i_Scheduled}$, for an Event i as listed above for individual flights was drawing from the AOC App. Actual event time, T_{i_Actual} , came from the Surface Operations Simulator and Scheduler (SOSS). Estimated event completion time, $T_{i_Estimated}$ was calculated based on the mean and standard deviation (σ) from the BAI App on a specific event as shown in Equation 1.

$$T_{Estimated_i} = T_{Estimated_i-1} + t_{Estimated_i-1_to_i} \quad (1)$$

where

T is the simulation time, e.g., in zulu in seconds
 i denotes an event, e.g., arrival at the gate
 $i-1$ denotes a previous event, e.g., inbound spot
 t is the duration from Event $i-1$ to Event i

The time to take from Event $i-1$ to Event i is defined in Equation 2.

$$t_{Estimated_i-1_to_i} = \text{Size of the Event/Rate of the Event} \quad (2)$$

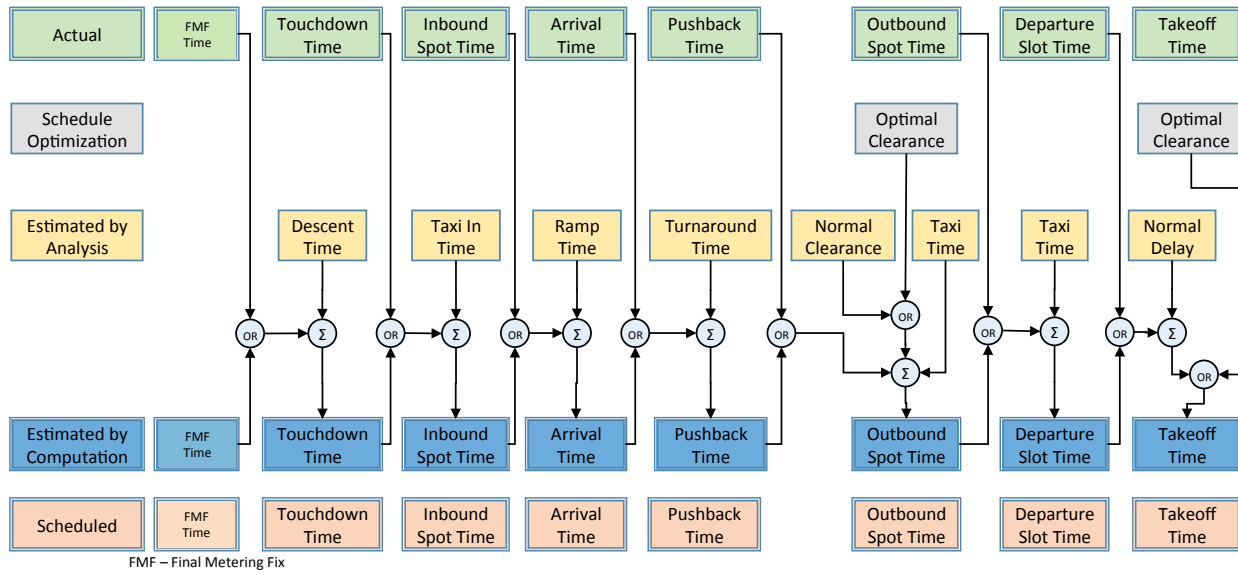


Figure 3. Data flow of the IGTM events

An example for the Inbound Spot Time, “Size of the Event” would be the distance from the touchdown spot to the inbound spot, and the “Rate of the Event” would be the average (or mean) taxi speed. Distance from the touchdown spot to the inbound spot is given by SOSS, and the average taxi speed is provided by the BAI App.

At the gate, the “Turnaround Time,” where applicable, determines the “Pushback Time” in Figure 3, is dictated or triggered by the completion of following gate activities

- i. Deplaning of passengers
- ii. Baggage unloading
- iii. Fueling
- iv. Cabin services
- v. Catering services
- vi. Baggage loading
- vii. Flight crew availability
- viii. Cabin crew availability
- ix. Maintenance
- x. Boarding of passengers

For the gate turnaround events, Equation 1 applies to all these events with the same three sets of time performance, i.e., Scheduled, Actual, and Estimated. In these instances, the Deplaning Time was determined by the number of passengers (Size of the Event), and the deplane rate (Rate of the Event). Number of passengers is given by the AOC App, and the average deplane rate is provided by the BAI App.

For the turnaround time at the gate, assumptions of critical paths were developed as shown in Table 1. Times for critical paths were calculated, and the estimated turnaround time was determined by the critical path that took the longest time. Additional time delay due to door-close to brake-release, late arrival of the flight crew, and/or cabin crew, and time to receive the clearance for a pushback were included in the turnaround time to determine the Estimated pushback time.

Table 1. Critical paths during the gate turnaround events

| Critical Paths | Sequence of Events (from left to right) | | |
|----------------|---|-----------------|------------------------|
| 1 | Deplaning of passengers | Cabin services | Boarding of passengers |
| 2 | Baggage unloading | Baggage loading | |
| 3 | Catering services | | |
| 4 | Deplaning of passengers | Fueling | Boarding of passengers |
| 5 | Maintenance | | |

Collaborative Decision Tools

One of the key concepts of the IGTM is applying collaborative decision-making among stakeholders to gain the integrated NAS performance improvements than the optimization of a local event. Thus, the effects of a single decision-making must be propagated through the NAS beyond the local decision-making domain, and promote the coordinated or collaborative decision-making on scheduling the events. For the IGTM prototype, simple uses cases were developed to demonstrate the tools to mitigate unexpected early and late arrival situations.

Airline Operational Control Center (AOC) App

The AOC App was developed to simulate the data from Airline Operational Control Center (AOC), which consisted of scheduled flight time information at the Final Metering Fix (FMF), wheels-down (touchdown), inbound spot, arrival at the gate, pushback, outbound spot, and wheels-up (takeoff). The AOC data were developed based on the arrivals and departures at Terminal A of Ft. Worth International Airport (DFW). For the prototype, all the AOC data were stored on the Mongo Server to be accessed by the AOC App.

BAI App

The Big Data/Analytics Input (BAI) App simulates the applications of the Big Data and Analytics technologies, which develop predictable results based on the past and current NAS performance according to selected decision trees through a User Interface (UI). The prototype was developed to demonstrate the potential benefits of the BAI data through the UI.

For the IGTM prototype, arrival and departure time of flights in December 2013 at Terminal A of DFW were analyzed based on data from the NASA data warehouse. [6] The mean and standard deviation of the turnaround time were identified by analyzing Actual gate-in and gate-out time at each gate per type of airplanes. Estimated time events based on predicted results, in this case, the mean and standard deviation from historical data, were generated for each events defined in the IGTM Controller. For example, “Rate of Event” such as the flight speed between the FMF and the touchdown was defined by SOSS of a specific type of airplane, e.g. B737-800, with a specified speed variation. Rate of Events for gate turnaround, e.g., passenger deplaning, cabin services, and boarding, were obtained from References [7,8]. BAI data were also stored on the Mongo Server to be accessed by the BAI App.

Surface Operations Simulator and Scheduler (SOSS)

SOSS is a NASA developed Fast-Time simulator, which was used to simulate flights arriving and departing from Terminal A of DFW, with arrivals on runway 17C and departures on 17R. SOSS was connected through a Java Messaging Service (JMS) Interface which provided the Actual events and the simulated clock time.

SOSS Adapter

The SOSS Adapter is the fast-simulation translator, while the SOSS component is the Fast-simulation message emitter/consumer that reads/writes in scenario data from files. The SOSS Adapter translates messages from a proprietary socket connection structured data stream and converts each packet into JSON or Serialize Data, which is then transported over the ActiveMQ message bus.

IGTM GUI

The Control Panel of the IGTM GUI is shown in the top of Figure 4, which allows the user to configure the environment to observe flight data associated with the Airport, Terminal, Spot, and Departure Runway. The Control Panel also provides the user with Collaborative Decision Making (CDM) use case options. The Dashboard, which displays flight status and predicted time events, is shown in the bottom of Figure 4. The color code on the right describes the estimated bounds based on the standard deviation derived by the BAI. A light Green of an event represents a likelihood of an event would be completed within one standard deviation of the scheduled event time or about a 68 percent of successful probability. A light Yellow indicates the event would be completed between 1 and 2 standard deviations or about a 95 percent of successful probability. A Red indicates the event would be completed greater than 2 standard deviations or there is only an about 5 percent of chance to meet the scheduled time. The purpose of the color code is to give users a direct implication of the success rate of completing a given event under the uncertainties the BAI data are generated. Therefore, performance bounds can be established based on the dependencies among performance parameters and specified uncertainties.

For gate turnaround performance at the gate, a Gate Status display can be selected by the users from the Dashboard for a given flight as shown in Figure 5. The display shows the same color code topology and the standard deviation

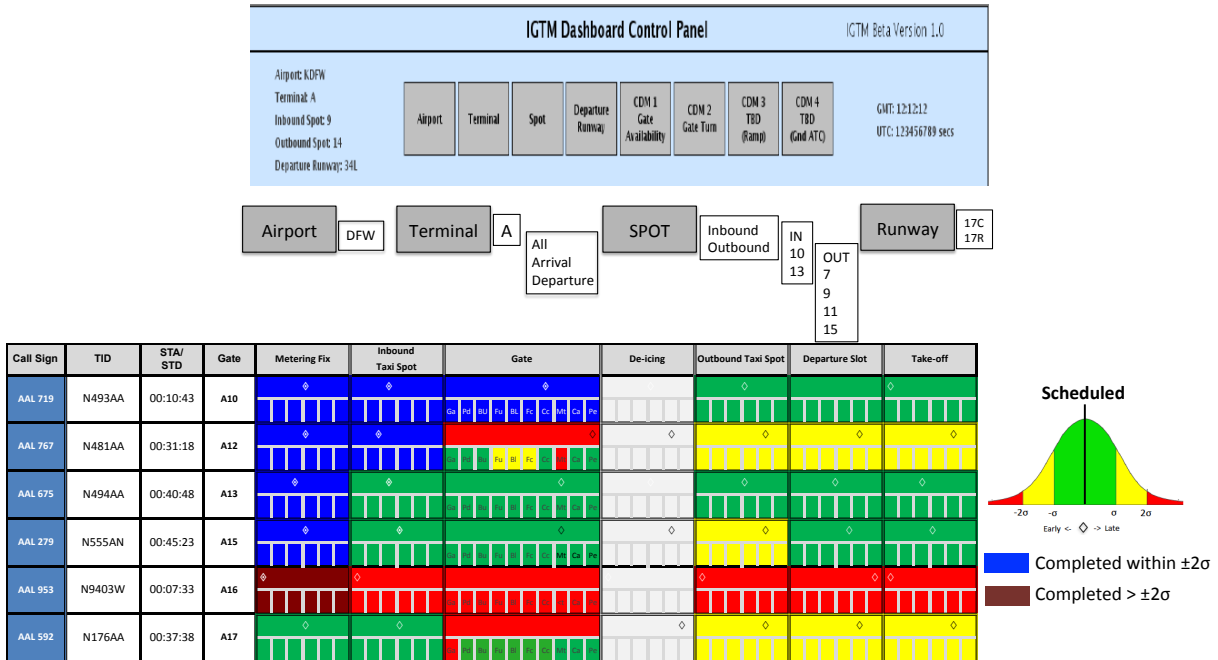


Figure4. The IGTM prototype's Control Panel

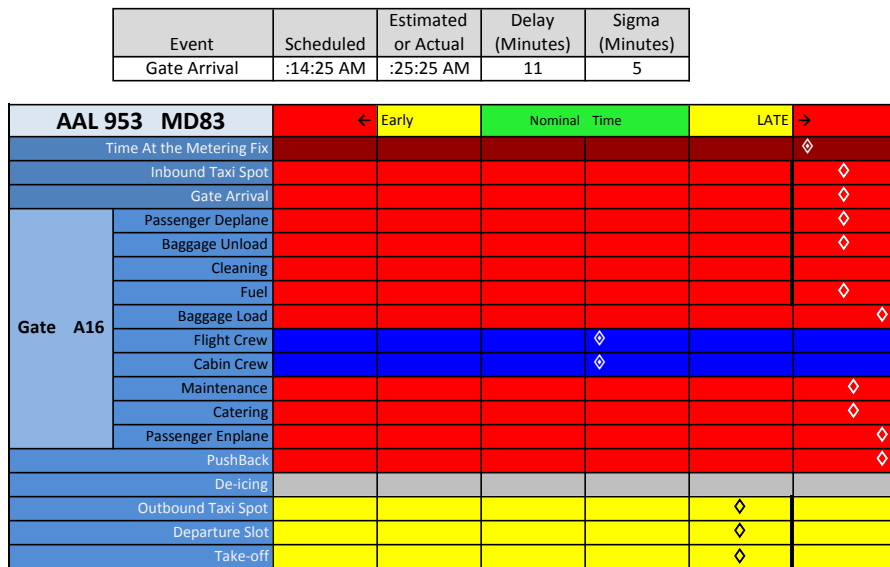


Figure5. The Gate status display

(σ or Sigma) in minutes. Users can then associated the color code to the time objectively.

ActiveMQ

A message 'event' can be defined as "a change in state." In the IGTM application events are as simple as 'an aircraft touches down', 'the aircraft reaches the gate', 'another aircraft has all baggage removed,' or various other changes in an aircraft's state of location or activity as shown in Figure 3 in the Gate-Turn Model. IGTM software components handle numerous events. An ATM's system architecture may treat this state change as an event whose

occurrence can be made known to other processes within the application architecture. From a formal perspective, what is produced, published, propagated, detected, or consumed is a (typically asynchronous) message called the event notification, and not the event itself, which is the state change that triggered the message emission. Events do not ‘travel’, they just occur. However, the term *event* is often used metonymically to denote the notification message itself, which may lead to some confusion.

This architectural pattern was used in the design and implementation for transmitting events among loosely, coupled software components and services. The IGTM event-driven system consists of event emitters/agents, event consumers/sinks, and event channels. These channels, or collection of channels, are also referred to as the message bus. The emitters have the responsibility to detect, gather, and transfer events. Event emitters are unaware of a consumer of events, when a consumer does exist the event emitters do not know how the event is used or further processed. Sinks have the responsibility of applying a reaction as soon as an event is presented. These emitter/consumers components are the Dashboard, Controller, Big Data Analytics, AOC, and the SOSS Adapter.

The Apache ActiveMQ broker fulfills the event channel role. Event channels are conduits in which events are transmitted from event emitters to event consumers. The knowledge of the correct distribution of events is exclusively present within the event channel. The physical implementation of event channels can be based on traditional components such as message-oriented middleware or point-to-point communication. The selection of ActiveMQ as the messaging broker conduit was based on its ease of configuration, its support in other third party API and its ability to handle various types of payload data. One such data payload type is JSON. In the purposes of IGTM the ActiveMQ message bus use was to simulate the NEMS/SWIM message bus (also ActiveMQ).

Mongo Server or MongoDB

NoSQL databases have emerged in recent years to provide the performance, scalability, and flexibility required of modern applications. This new wave of databases is much better suited for Big Data applications and agile software development practices than its relational counterparts. Mongo was selected because it is one of the leaders in the NoSQL arena and that it couples with other application frameworks. The framework selected for rapid application development was SpringFramework.

NoSQL databases offer many benefits, including:

- Flexible Data Model. Unlike relational databases, NoSQL databases easily store and combine any type of data, both structured and unstructured as JSON.
- Elastic Scalability. MongoNoSQL databases scale out on low cost, commodity hardware, allowing for almost unlimited growth.
- High Performance. NoSQL databases are built for great performance, measured in terms of both throughput and latency.

These advantages account for the growing popularity of NoSQL databases, and specifically MongoDB. MongoDB stands apart from its peers with its Nexus Architecture that incorporates the strengths of relational databases along with the innovations of NoSQL. MongoDB is the only NoSQL options, which offer an expressive query language, strong consistency, and secondary indexes.

IGTM chose MongoDB for this reason and that it mates easily to changes the data model and to not have to tinker with the data layer code. Mongo does all the work for you. Mongo also makes it easily possible to work with many of the Big Data Analytic tools such as Tableau, JasperSoft, OpenRefine, Knime, NodeXL, Import.io and others because of the simplicity of the Mongo Query language.

USE CASES

Two use cases, i.e., early arrival with gate conflict and gate recovery of a late arrival, of the IGTM prototype were demonstrated. Figure 6 shows a CDM display of gate conflict due to an early arrival. The IGTM Controller would identify available gates, which could be available for the estimated arrival time and required turnaround time if applicable, and display on the Gate Availability display. This will allow the user to identify and select an available option, which may require coordination among stakeholders, with minimum time lost at the tarmac, cost, and ground crew resources.

The second use case was to demonstrate a speed-up gate turnaround in order to meet the Scheduled pushback time due to a late arrival. The IGTM Controller would identify the most critical path among all critical paths and allow the user to adjust the time performance within the available resources or methods based on AOC data or BAI data. The user used the Gate-Turnaround Management display as shown in Figure 7 to speed up the Passenger Deplane rate, Passenger Boarding rate, and increase the number of cabin service (or cleaning) crew, typical methods to

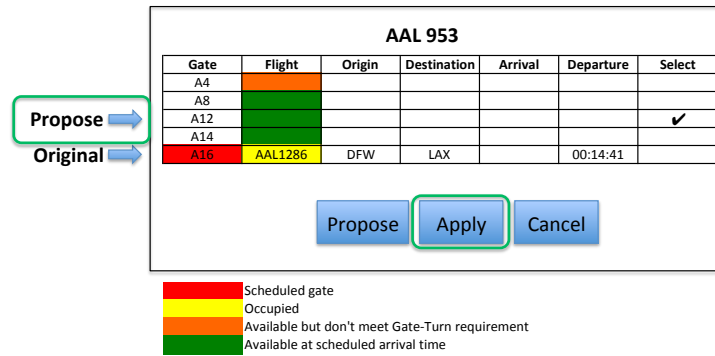


Figure 6. A Gate Availability display for gate conflict mitigation

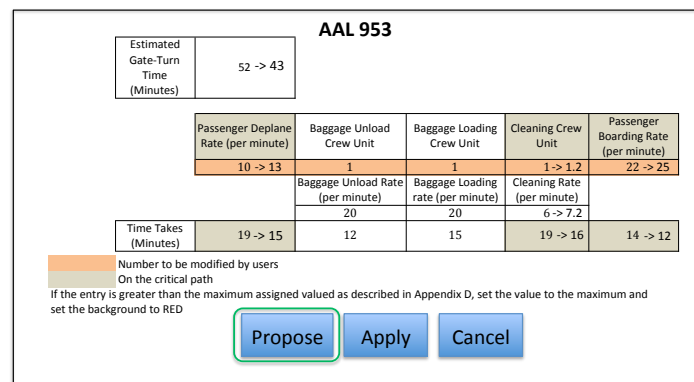


Figure 7. A Gate Turnaround Management display for time management at the gate

recovering lost time at the gate. In this case, the Estimated turnaround time was reduced from original 52 minutes to 43 minutes.

CONCLUDING REMARKS

The IGTM prototype demonstrated the concept and benefit of technologies that provide a stream of real-time analytics combined with historic archived data that bound the uncertainties in a gate turnaround NAS operational space. NAS stakeholders can share the flight information, resources, and time management tools through a common messaging network service to coordinately improve the NAS performance under the nominal and off-nominal conditions. The prototype also offers a modular design to incorporate additional Big Data and Analytics products to support future ATM research.

ACKNOWLEDGEMENTS

This work was funded by a NASA Big Data, Data Analytics, and Net-enabled ATM and Airspace Operations Project under Contract NNA15AB05C. Authors wishes to thank Deepak Kulkarni and Yao Wang of NASA Ames Research Center for their technical guidance and advice. Authors wish to thank industrial team, which include Joe Burns of XCELAR Inc., Steve Koczo and Arlen Breiholz of Rockwell-Collins, Henry Smith and Warren Qualley of Harris Corp., Randall Ho of the IBM Software Group Federal, Bruce Sawhill of the NextGen Aero Sciences, and Ben DeCosta of DeCosta Consulting LLC, for the concept development. Authors also wish to thank John Walker and Darrell Wooten of the SAIC software development group for their support in developing the IGTM prototype.

REFERENCES

Ren, L; Beckmann, B; Citriniti., T., and Castillo-Effen, M: "Cloud Computing for Air Traffic Management – Cost/Benefit Analysis" (16-20 June 2014, 14th AIAA Aviation Technology, Integration, and Operations Conference)

Ebbers, M; Abdel-Gayed, A; and et al.: "Addressing Data Volume, Velocity, and Variety with IBM InfoSphere Streams V3.0," (March 2013)

Robb, j.: "System Wide Information Management (SWIM) Program Overview and Status Update," Air Transportation Information Exchange Conference (August 2015)

Kim, S.H., et al.: "Airport Gate Scheduling for Passengers, Aircraft, and Operations," "Tenth USA/Europe Air Traffic Management Research and Development Seminar, ATM2013 (April 17, 2013).

Sandberg, M., et al.: "Airport Characterization for the Adaptation of Surface Congestion Management Approaches," Air Traffic Management Research and Development Seminar, ATM2013 (April 17, 2013)

Eshow, M. and Lui, M.: "Architecture and Capability of Data Warehouse for ATM Research," 33rd Digital Avionics Systems Conference (DASC) (October 2014)

Airbus A319 Aircraft Characteristics for Airport and Maintenance Planning. Retrieved May 2014 from http://www.airbus.com/fileadmin/media_gallery/files/tech_data/AC/Airbus-AC_A319_May2014.pdf

Boeing B752-200/300 Airplane Characteristics for Airport Planning. Retrieved February 2015 from http://www.boeing.com/assets/pdf/commercial/airports/acaps/757_23.pdf