Ground Risk Assessment Service Provider (GRASP) Development Effort as a Supplemental Data Service Provider (SDSP) for Urban Unmanned Aircraft System (UAS) Operations

Ersin Ancel
Aeronautics Systems Analysis Branch
NASA Langley Research Center
Hampton, VA, USA
Ersin.Ancel@NASA.gov

Tyler Helsel
Aeronautics Systems Analysis Branch
Analytical Machanics Associates (AMA), Inc.
Hampton, VA, USA
Tyler.Helsel@NASA.gov

Christina M. Heinich
Information Management Branch
NASA Langley Research Center
Hampton, VA, USA
Christina.M.Heinich@NASA.gov

***Abstract*—NASA's Unmanned Aircraft System (UAS) Traffic Management (UTM) project aims to enable the integration of new aviation paradigms such as Unmanned Aircraft Systems (UAS) while providing the necessary infrastructure for future concepts such as On-Demand Mobility (ODM) and Urban Air Mobility (UAM) operations in the National Airspace System (NAS). In order to do so, the UTM project has developed an architecture to allow communication among UAS operators, UAS Service Suppliers (USS), Air Navigation Service Providers (ANSP), and the public. As part of this framework, the Supplemental Data Service Providers (SDSP) are envisioned as model and/or data based services that disseminate essential or enhanced information to ensure safe operations within low-altitude airspace. These services include terrain and obstacle data, specialized weather data, surveillance, constraint information, risk monitoring, etc. This paper highlights the development efforts of a non-participant casualty risk assessment SDSP called Ground Risk Assessment Service Provider (GRASP) which assists operators with pre-flight planning. GRASP is based on the previously introduced UTM Risk Assessment Framework (URAF) and allows UAS operators to simulate and visualize potential non-participant casualty risks associated with their proposed flight. The risk assessment capability also allows operators to revise their flight plans if the casualty risks are determined to be above acceptable thresholds. GRASP is configured to account for future improvements including servicing airborne aircraft as part of NASA's System-Wide Safety (SWS) project.**

***Index Terms*—unmanned aircraft systems, casualty risk**

I. Introduction

This paper describes the details of the Ground Risk Assessment Service Provider (GRASP) Supplemental Data Service Provider (SDSP) development effort within the Unmanned Aircraft Systems (UAS) Traffic Management (UTM) Project under NASA's Airspace Operations and Safety Program (AOSP). The UTM Project focuses on developing concepts and initial implementations for safely integrating and managing small UAS (sUAS) into the low altitude airspace, paving the road for widespread commercial UAS applications in the future. In order to do so, NASA has developed an architecture and associated interfaces and protocols that allow communication between traditional Air Navigation Service Providers (ANSP), UAS operators, and the public (Fig.1). A key component of this architecture is the UAS Service Supplier (USS), which acts as a communications bridge between UAS operators, SDSPs, and the ANSP when necessary. Within this architecture, the SDSPs directly supply essential enhanced services (e.g., terrain and obstacle data, specialized weather data, surveillance, constraint information, etc.) to both USS networks and to UAS operators. The structure also allows data communication and coordination among the SDSPs for disseminating data and information for integrated services [1].

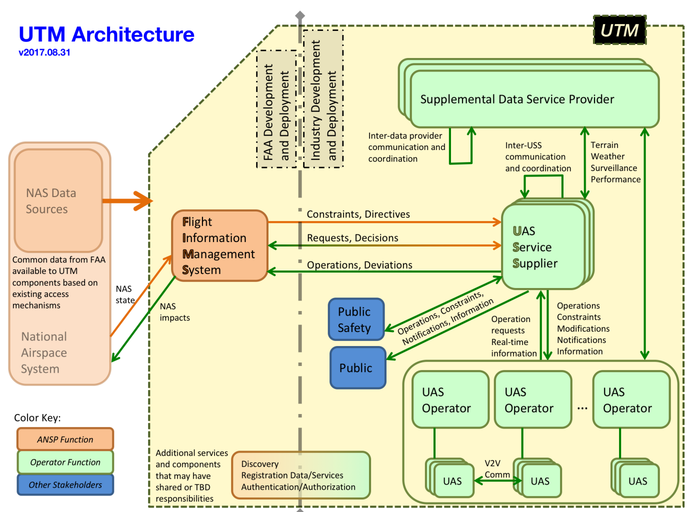

Fig. 1. UTM Architecture.

The UTM Project was planned with a phased approach, comprised of four Technical Capability Levels (TCLs) where each level is demonstrated with integrated flight tests with increasing complexity [2]. The GRASP SDSP described in this

Research funded by NASA Unmanned Aircraft System (UAS) Traffic Management (UTM) Project

paper is designed to support the TCL4 flight demonstration. The TCL4 is focused on operations in a densely populated urban environment and includes the handling of high density UAS operation environments, large-scale off-nominal conditions, vehicle-to-vehicle communications, detect-and-avoid technologies, communication requirements, public safety operations, airspace restrictions, and other related goals [3].

Additionally, besides the UTM project, the GRASP SDSP development effort supports the System-Wide Safety (SWS) project, also under the AOSP. One of the goals of the SWS project is to develop and demonstrate research tools, innovative technologies, and operational methods to monitor, assess, and mitigate safety risks expected with the introduction of emerging low-altitude urban aircraft operations. The SWS project assumes the UTM ecosystem as the underlying infrastructure to enable the safe integration of such emerging applications like sUAS, On-Demand Mobility (ODM) and Urban Air Mobility (UAM) operations. Within SWS's In-time Safety Assurance Concept, SDSPs play a crucial role in safety-relevant data generation and dissemination in all phases of flight to support highly autonomous low altitude urban operations. In particular, SDSP-supported risk assessment functions are performed a) during the pre-flight phase of the flight to assist with the flight planning phase, b) throughout the flight via in-time services such as vehicle prognostics, battery health monitoring, and onboard non-participant risk assessment capability to ensure safe flight [4], [5], and c) the post-flight phase where observed/recorded safety data is uploaded to the relevant SDSPs to update databases and validate the services and relevant models [6].

Within the UTM and SWS project scopes given above, the GRASP SDSP described in this paper provides a pre-flight risk simulation capability for UAS operators, UTM partners, and NASA's USS. The GRASP SDSP was developed based on the UTM Risk Assessment Framework (URAF) effort which provides a modular non-participant casualty risk assessment architecture for small unmanned aircraft [5], [7]. The GRASP SDSP was placed on a public facing website where the casualty risk to people on the ground can be simulated, visualized, and revised if the risk is above a certain acceptable threshold. In order to assist UAS operators as well as USSs, the website was designed to be accessible via command line/RESTful requests in addition to the website access.

To perform pre-flight risk assessment, the GRASP code requires a proposed flight trajectory, flight date/time, aircraft characteristics, and a vertical wind profile, if available. It then simulates the proposed flight and provides potential impact points and respective non-participant casualty probabilities along the route where an unpowered/uncontrolled descent and subsequent crash is assumed as the worst-case scenario[1]. It is important to note that due to the uncertain nature of the failure modes, environmental factors, and actual population density movements, the casualty probability values provided by the GRASP code are intended to be used as a comparative tool between alternative flight plans to assist GCS operators. By querying the GRASP server, operators can simulate multiple flight plans by varying aircraft, mission, and environmental parameters such as projected winds to minimize non-participant risk. Section II provides an overview of the main components of the GRASP code. Section III discusses the implementation of the GRASP as an SDSP server on a externally-facing, NASA hosted web server. Section IV presents sample GRASP SDSP outputs simulating representative TCL4 flights, and finally, Section V discusses conversion of the GRASP SDSP to serve in-flight operations and planned improvements which are considered as future work.

[1]Given that the flight is simulated using only the flight trajectory file, pertinent flight parameters necessary to simulate other failure modes caused by partial aircraft system failures is not feasible for pre-flight risk analysis.

II. GRASP Code Components

The GRASP SDSP and the underlying risk assessment code employ a variety of models, executed sequentially, to estimate the probability of casualty following an unpowered descent and crash. In order to do so, the GRASP SDSP code first ingests user-supplied inputs regarding the mission and then executes a mission setup routine followed by trajectory and impact point estimation model, casualty estimation model, and finally JavaScript Object Notation (JSON) output file generation (Fig. 2). More specifically, the flight mission is established by parsing the waypoint file and by initializing the population density data and map according to flight time and location provided by the user. Next, an off-nominal trajectory and impact point prediction model is used to simulate an assumed failure for a predetermined step size throughout the mission. A casualty estimation model employing impact dynamics and population density data is used to estimate the non-participant casualty risk of the proposed flight. Finally, the output parameters including the probability of casualty, impact location and sampled casualty areas for each waypoint are provided as a JSON output file. The GRASP code is written in C++ and was built, tested, and deployed on a Linux server using GCC and GNU make[2] utility.

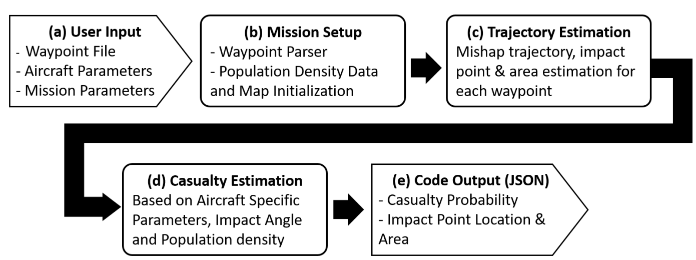

Fig. 2. GRASP Code Components.

A. User Input

The current implementation of the GRASP SDSP simulates flight missions using a waypoint file commonly generated by Mission Planner software. Mission Planner is a ground control station software which is part of the open source

[2]See https://www.gnu.org/software/make/

ArduPilot unmanned vehicle autopilot software suite[3]. Besides the waypoint file, the user is required to input vehicle-specific parameters such as multicopter aircraft weight, radius, drag coefficient and associated reference area which is used for trajectory and casualty estimation models[4]. Additionally, flight location, date and time are required to identify and develop a representative population density data and associated map. Finally, the user can submit an expected wind profile for the time of flight, if available. The users can provide the input in two different ways which will be discussed in Section III-A.

B. Mission Setup

1) Waypoint Parser: The waypoint parser is the first subroutine within the GRASP code and it is responsible for interpreting the waypoints and simulating the flight (Fig 2). In order to do so, the parser reads in the user-provided waypoint text file which provides a set of navigation commands that control the movement of the vehicle including performing takeoff/landing, changing speed/altitude, and moving to additional waypoints among other functions. Each of these commands have associated data such as latitude and longitude, altitude, speed, time delay, etc. that the waypoint parser uses to build a internal vector of waypoints. Additionally, the parser interpolates new waypoints for approximately every 10-15 meters to construct a continuous simulated flight.

2) Population Density Data and Map Initialization: As part of the pre-flight non-participant risk estimation, the GRASP code employs population density data which constitutes a crucial component of the casualty approximation. Often, lack of a good quality population density data is the bottleneck in estimating non-participant ground risk since studies employ median population density values for rural, suburban, and urban settings. This approach tends to misrepresent dynamic aspects of movement of people, especially considering missions conducted in urban settings where population density can vary greatly throughout the day (e.g. during commuting and lunch periods) or grow substantially due to an open air assembly (e.g. sporting events and concerts) [5].

In order to capture and demonstrate the dynamic aspects of the densely populated urban environments anticipated within UTM TCL4 flight demonstrations, population density datasets for Reno, NV, Corpus Christi, TX, and San Francisco, CA, were acquired from AirSage, Inc. company which specializes in population analytics and location-based data solutions. This commercially available population activity density data provides movements of population within the area of interest at a 10m x 10m resolution in hourly increments, typically with a two month processing delay. However, it is important to note that although the population density data remains historical (i.e. not real-time), it provides a dynamic and higher resolution representation of expected population density.

Using the user-supplied date, time, and flight location, the GRASP code locates and imports the corresponding population density data in comma-separated values (*.CSV) file format. Given that population activity density data is only available for select dates[5], GRASP code interpolates the given date using a modulo algorithm to find an accurate representative date and the day of the week from the available data for the proposed flight. Finally, the corresponding population density map is located and parsed to a more refined area in order to reduce computation times.

[3]See http://ardupilot.org/ardupilot/index.html

[4]At the time of writing, the GRASP SDSP only supports multicopter flights.

C. Off-Nominal Trajectory and Impact Point Estimation

As previously discussed, the off-nominal trajectory and impact point prediction model simulates an unpowered descent trajectory (e.g., battery failure or complete motor failure) and estimates the subsequent impact point and potential casualty areas for each simulated waypoint. Casualty area is defined using the vehicle radius and glide/approach angle during impact and presence of a person within the area is assumed to have sustained a casualty (Fig. 3). To estimate the impact point and associated casualty area, a Point Mass model which includes 3 Degrees-of-Freedom (3DoF) equations of motion that considers atmospheric effects such as wind and drag, is executed for every simulated waypoint. In order to represent uncertainties associated with the 3DoF trajectory model and the unknown nature of the environmental effects, the Monte Carlo technique was applied by varying the aircraft heading and wind parameters. Based on the number of variables considered for the Monte Carlo formulation, numerous impact points and associated casualty areas are generated. The histograms of casualty areas for each waypoint are then employed as part of the casualty estimation model. For trajectory and impact point estimation, other higher fidelity solutions could be used if additional forces and aircraft state vector data are available, as highlighted in [7] and [5].

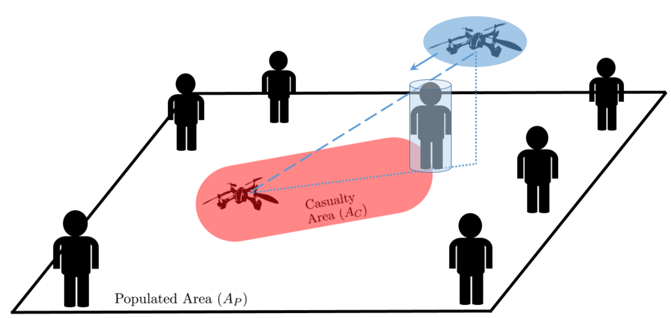

Fig. 3. Casualty Area Schematic.

D. Casualty Estimation Model

The casualty estimation model provides the probability of casualty for one or more persons in the event of a UAS crash. As with the previous URAF-developed risk assessment approaches, given the hard-to-predict nature of population dynamics and the large number of uncertainties, a probabilistic

[5]A month for each season (i.e. January, April, July, and October) was acquired to represent the entire year.

model was utilized for this task. The casualty estimation model employs the population density data and map (provided in Section II-B2), the potential casualty area histogram (described in Section II-C), and the impact characteristics including impact angle, velocity, and consideration of sheltering effects (Fig. 3). Casualty estimation methodology and its formulation employed in GRASP SDSP application were given in detail in [5], and sheltering effects[6] were previously demonstrated in [7].

E. GRASP Code Output

Following the execution of the GRASP code, the output parameters are written into a JSON file and provided to the web server interface. The JSON output file contains the latitude and longitude for each interpolated waypoint and their respective impact points, impact points radius in meters, and associated probability of casualty for each impact point. The output file is used to develop visualization or made available for direct download as discussed in Section III-B.

III. GRASP Code Implementation

As part of the UTM architecture construct within the TCL4 environment, the GRASP SDSP is developed to assist UAS operators with performing pre-flight risk assessment and revising flight plans to minimize potential non-participant casualty risk. In order to do so, the GRASP SDSP web application was embedded on a public facing website which is maintained by the Office of Chief Information Office (OCIO) Information Management Branch (IMB) and is currently being hosted on an Apache server in the OCIO Data Center (ODC)[7]. The website is a PHP interface that handles passing requests to the GRASP C++ parser and returns the results to the user. The previously discussed user data (e.g. aircraft specifications, flight details, and environmental parameters) is transferred to the GRASP code and outputs are provided back to the user in varying ways based on the method of access.

A. Methods of Access

Users can access GRASP through two methods; a) by submitting a web form by visiting the GRASP website in a web browser (https://grasp.larc.nasa.gov) (Fig. 4), or b) by submitting a REST API call to the server. Both methods require the user to provide a passphrase to authenticate with in addition to the flight mission information. Regardless of the access method of choice, the user supplied data is passed into the GRASP application via the web server which then will return the results of the assessment. Figure 5 highlights the GRASP SDSP data sequence diagram, the two access methods, and output formats.

[6]Sheltering effects were not included in the current iteration of GRASP code.

[7]To facilitate access, accommodate a larger user-base, and to eliminate future computational constraints, the application will be moved to a cloud-based solution in the next iteration phase.

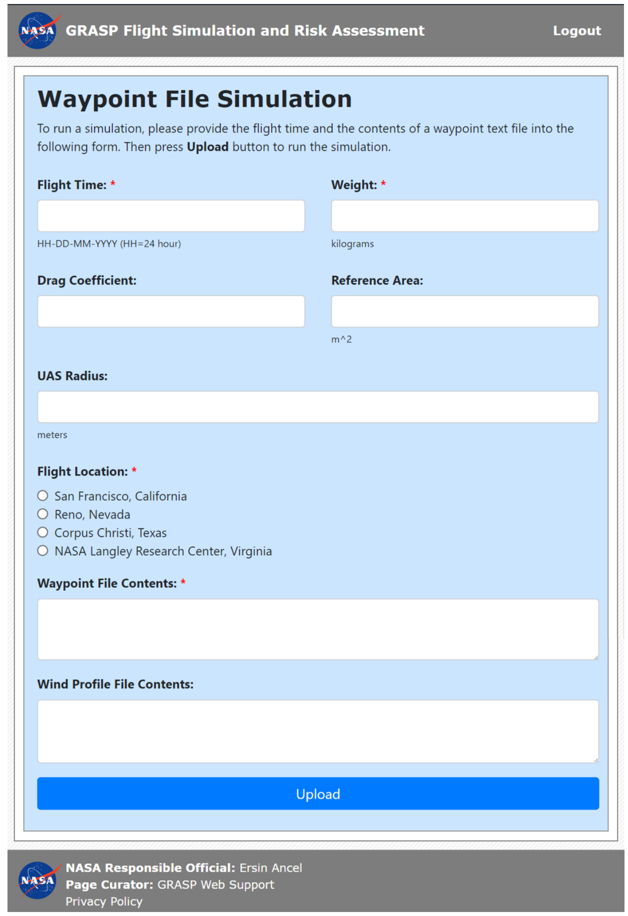

Fig. 4. GRASP Website GUI.

1) Website Access: Access through the website is implemented as a web form. Upon visiting the GRASP SDSP URL, the users are shown an online form that allows them to fill in various aircraft and mission parameters. The web form forces users to provide the required fields using the `required` attribute. After all fields have been filled out, the users can submit the form. The web application will then ingest those parameters, run various filtering/checks on the input, saves the waypoint text as a temporary file, and then creates a system call to the GRASP C++ executable installed in the web application. If the GRASP code was successful in running an assessment, it saves the JSON output file on the server. The website will then provide links for the user to download the data or to visualize the risk assessment output on another webpage. The details of the visualization are provided in Section III-B.

2) REST API/Programmatic Access: Besides the website access, the GRASP SDSP was also made available via command line access to facilitate communication with NASA and external USSs as well as other SDSPs as previously defined within the UTM architecture. The command line access also allows supporting simulated TCL4 flights within the Multi Aircraft Control Systems (MACS) simulation environment at

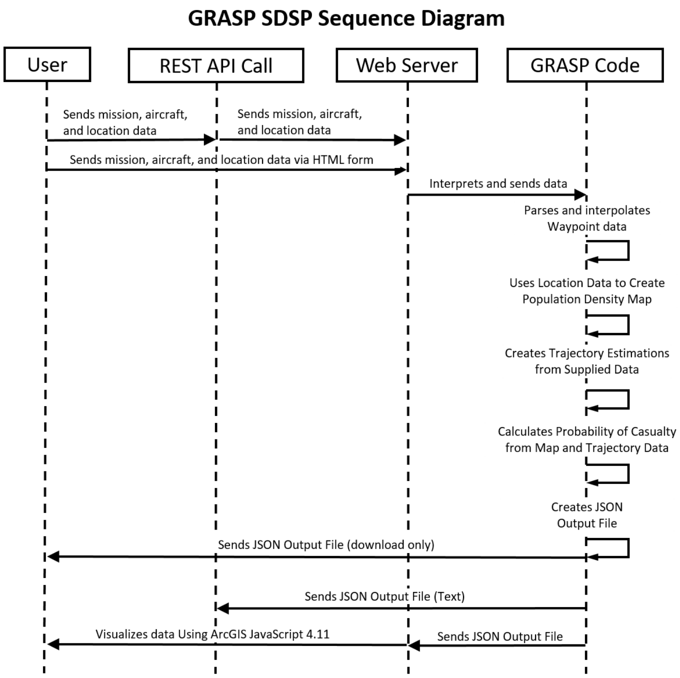

Fig. 5. Sequence diagram illustrating the GRASP SDSP Architecture.

NASA Ames' Airspace Operations Lab (AOL)[8].

Representational State Transfer[9] (REST) is a way of designing Application Programming Interfaces (API) to a web service that allows users to make queries to a web service to perform tasks like getting or posting data. Hypertext Transfer Protocol (HTTP) requests/responses are the de facto standard way to represent data flow through the API. For GRASP SDSP, this represent an ideal method for users to interact with web applications programmatically as opposed to manual web forms. To perform a task, users choose a HTTP method (GET/PUT/POST/DELETE), a URL, and parameters, if any. For GRASP implementation, given that the sole interaction is to pass the flight mission and aircraft data to the web server and receive non-participant risk in return, the most suitable method is to use POST. POST calls are used to create new resources. In that interaction, the user POSTs their flight data to the server (or creating it on the server), after which the server performs calculations and returns those results back to the user. Practically, GRASP users make HTTP POST requests to https://grasp.larc.nasa.gov/api/simulate.php, providing flight parameters. The web application then takes those parameters and performs the same tasks as the web form: it validates the data, creates and executes a system call, and saves the output as a JSON file. The web application then returns an HTTP response containing the JSON contents of the output file to the respective platform used for programmatic access.

[8]MACS is a software platform that allows for rapid prototyping, human-in-the-loop air traffic simulations, and evaluation of current and future air/ground operations for the National Airspace System.

[9]See https://restfulapi.net/http-methods/ for further information on HTTP methods within REST API.

B. Visualization

As part of the web server access to the GRASP service, the non-participant casualty risk along the simulated route and associated population density data are visualized within the web browser. The visualization of the JSON output file is done via the ArcGIS Javascript 4.11 API while jQuery 3.4.0 is used to read in and parse the JSON file. The output file is converted to a graphics layer and assigned a renderer, which is then converted into a feature layer to be displayed in the web browser. The same process is used to display the potential impact point area and casualty probabilities for each simulated waypoint on a separate feature layer. In order to assist in decision making and to provide additional insight, the population density data for the given flight time/location is also displayed as an additional layer to demonstrate areas with potentially high population activity around the flight path. In summary, the contents of the output file, i.e., color coded waypoints, heat map representation for both potential casualty areas and surrounding population density data for the given time, are all superimposed.

Additionally, several standardized ArcGIS widgets are implemented for user convenience. For instance, the Basemap widget allows the user to change the style of the map on which the probability of casualty data is displayed. Moreover, a Search Bar Widget allowing the user to find locations on the map and a Ruler Widget allowing the user to measure distances on the map were added to the interface. Finally, a layer view widget is provided to allow the user to show and hide specific layers. The visualization features discussed in this section are demonstrated in Section IV.

C. Security and Input Validation

Given that the current GRASP iteration resides on a NASA server and allows data upload from external users, precautionary steps were taken to prevent malicious actions. Specifically, the primary point of concern is the system call to the C++ executable, which, if designed poorly could potentially allow a malicious user to execute arbitrary commands on the server, similar to a SQL injection attack[10]. However, since the data being uploaded is very limited (i.e., it is either numeric or follows a very specific format), potential complications are alleviated by removing any non-alphanumeric characters and by verifying that the input is in the expected format during the filtering performed in the input validation step. Also, additional standard PHP filters and file permissions are put in place to further protect the server and embedded data. Finally, the completed site was subjected to thorough security and penetration scans by the OCIO Red Team[11] prior to making the site external.

[10]See https://www.owasp.org/index.php/SQL_Injection

[11]As part of the NASA IT Security (ITS) Division, the OCIO Red Team evaluates all public facing websites for security threats and ensures that the website is compliant with all NASA directives and procedural and technical requirements.

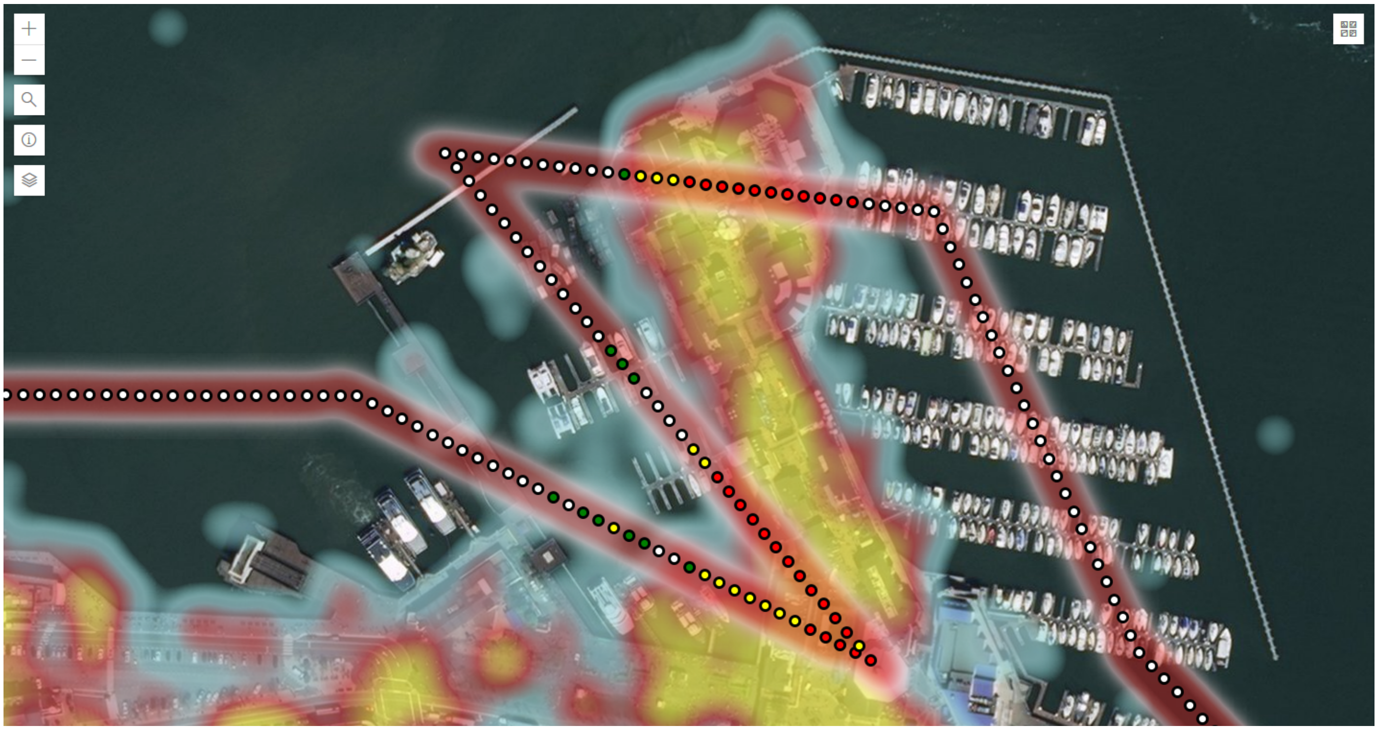

Fig. 6. Flight Over Fisherman's Wharf in San Francisco on July 4th, 2017 at 7PM.

IV. Example GRASP Service Outputs

The TCL4 demonstrations include flight tests conducted in Reno, Nevada and Corpus Christi, Texas in May and August 2019, respectively. These flight tests are designed to leverage capabilities obtained throughout the TCL3 demonstrations and focus on UAS operations in higher density urban areas for tasks such as news gathering, infrastructure inspection, and package delivery [3]. In order to demonstrate the visualization and user interface of the service, two waypoint files were developed and simulated using the GRASP SDSP.

Figure 6 provides the ArcGIS based visualization output for a hypothetical flight planned over San Francisco's Fisherman's Wharf/Pier 39 area around 7PM during the 4th of July to represent a photography mission over a heavily populated open air assembly. The simulated multirotor aircraft was assumed to have a drag coefficient of 0.2 and radius of 1 meter. The representative aircraft takes off East of the Pier 39 and then heads North/Northwest along the pier and then the aircraft turns West and flies over and along the pier at 25 meter altitude in no-wind conditions. The simulated waypoints are depicted with color-coded dots based on the casualty probability values and the red-colored path below the waypoints represent the potential areas of impact throughout the flight. As expected, the casualty probability is relatively low during the first part of the flight over the marina. As the aircraft approaches the Pier and crosses over the heavily populated area (visualized as population density heatmap, ranging from 25 to 60 people per $100m^2$), the casualty probability quickly rises to 1.0 due to population density and aircraft kinetic energy on impact as a failure is simulated at each waypoint (Fig. 7).

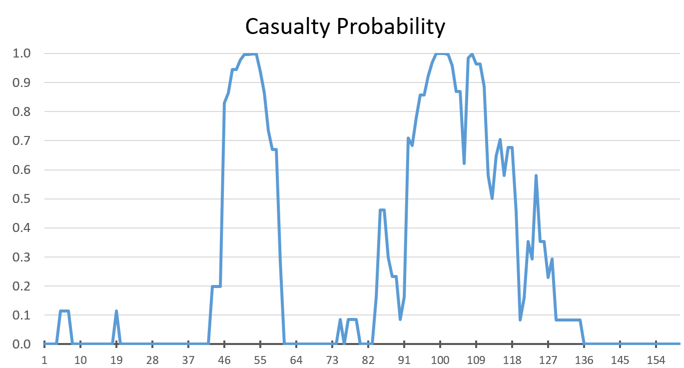

Fig. 7. Casualty Probability Values for downtown San Francisco, CA Flight.

A second flight was simulated over Reno, NV, starting at around downtown Reno followed by a brief fly-over of the Greater Nevada Field to represent a news gathering/photography mission during a minor-league baseball game on June 1st, 2018 between 7PM and 10PM (Fig. 8). The aircraft parameters and flight conditions were kept identical to the flight described above. Figure 9 provides the casualty probability values obtained from the flight where the non-participant casualty probability only increases during the flight over the stadium where the population density varied between 10 and 48 people per $100m^2$.

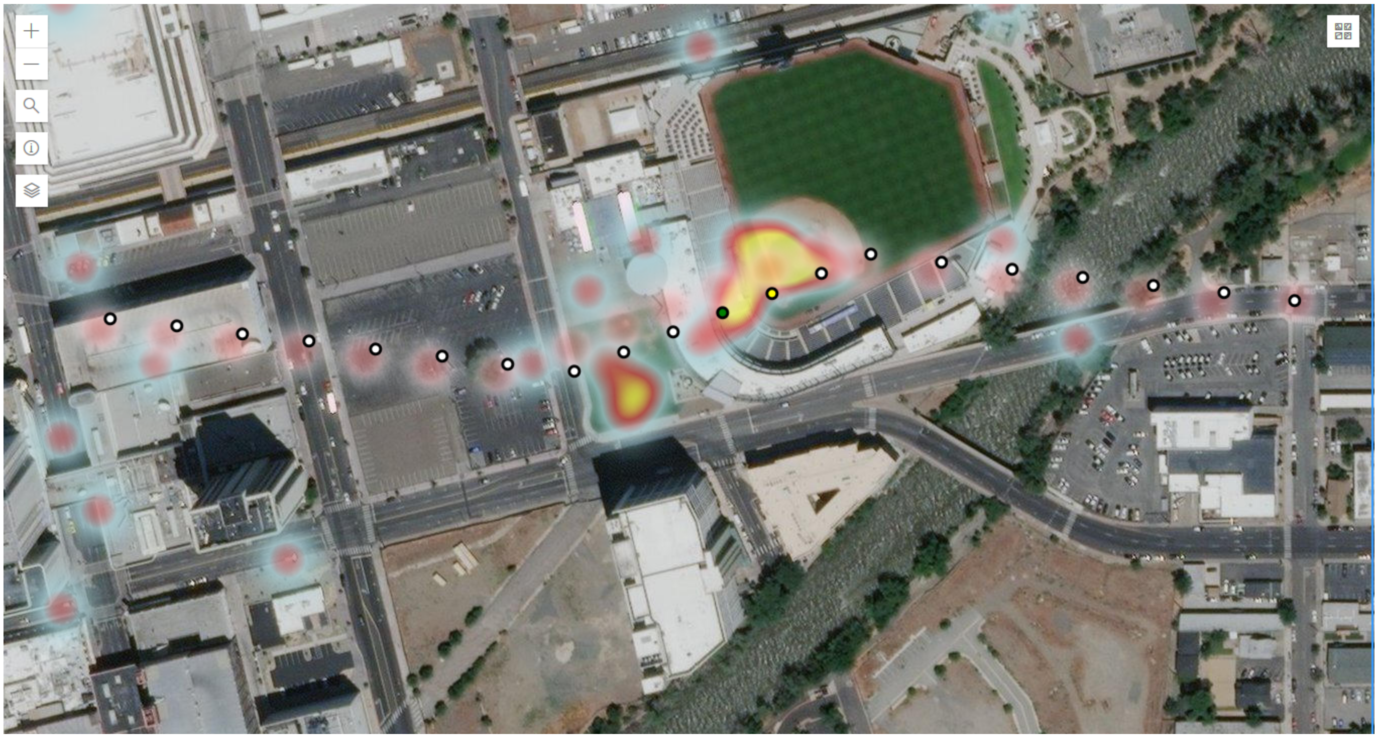

Fig. 8. Flight Over Downtown Reno, NV - Greater Nevada Field on June 1st, 2018 at 8PM.

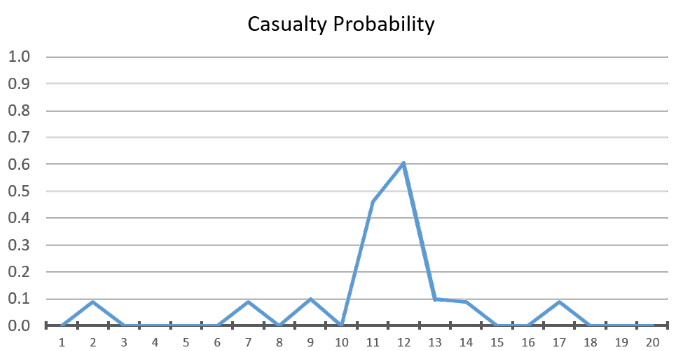

Fig. 9. Casualty Probability for Downtown Reno, NV Flight.

V. Concluding Remarks and Next Steps

As the demand for operations by small unmanned vehicles in low-altitude urban airspace increase, assuring the safe implementation and operation of these aircraft will continue to play a significant role for widespread adoption of these systems. NASA's UTM project aims to facilitate the integration of sUAS while providing the infrastructure necessary for the implementation of future concepts such as ODM and UAM. Within the proposed UTM architecture, SDSPs play a crucial role in providing support to USSs as well as UAS operators by supplying essential or enhanced services such as terrain/obstacle data, specialized weather data, non-participant casualty risk assessment, etc. This paper highlighted the development efforts for the first iteration of the ground risk assessment service provider which was configured as an SDSP within the UTM architecture. This implementation was designed as a pre-flight planning tool that allows comparison of alternative flight trajectories and flight dates/times to assist GCS operators. Future development efforts will focus on providing the service to airborne aircraft to support the in-time risk mitigation and decision making functions. Additionally, a detailed uncertainty analysis of the underlying risk assessment function to help identify the sensitivity of the input parameters and to provide confidence intervals for the estimated risk values is currently underway. Finally, the future iterations of the GRASP SDSP will support ODM/UAM missions as part of NASA's SWS project.

Acknowledgments

The work presented here was supported by NASA System-wide Safety (SWS) and Unmanned Aircraft System (UAS) Traffic Management (UTM) projects.

References

[1] Federal Aviation Administration, "Unmanned aircraft system (UAS) traffic management (UTM) concept of operations v1.0," 2018.

[2] M. Johnson, J. Jung, J. Rios, J. Mercer, J. Homola, T. Prevot, M. D., and P. Kopardekar, "Flight test evaluation of an unmanned aircraft system traffic management (UTM) concept for multiple beyond-visual-line-of-sight operations," in *12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)*, June 2017.

[3] J. L. Rios, I. S. Smith, P. Venkatesan, V. Smith, D. R.and Baskaran, S. K. Jurcak, S. M.and Iyer, and P. Verma, "UTM UAS service supplier development: Sprint 2 toward technical capability level 4,," NASA TM-2018-220050, 2018.

[4] J. Watkins, C. Teubert, and J. Ossenfort, “Prognostics as-a-service: A scalable cloud architecture for prognostics,” in *11th Prognostics and Health Management (PHM) Conference*, Prognostics and Health Management Society, Sept. 2019.
[5] E. Ancel, F. Capristan, J. V. Foster, and R. Condotta, “In-time non-participant casualty risk assessment to support onboard decision making for autonomous unmanned aircraft,” in *AIAA Aviation Forum 2019*, no. AIAA 2019-3053, American Institute of Aeronautics and Astronautics, June 2019.
[6] S. D. Young, C. P. Quach, K. Goebel, and J. Nowinski, “In-time safety assurance systems for emerging autonomous flight operations,” in *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, Sept 2018.
[7] E. Ancel, F. Capristan, J. V. Foster, and R. Condotta, “Real-time risk assessment framework for unmanned aircraft system (UAS) traffic management (UTM),” in *Aviation Technology, Integration, and Operations (ATIO) Conference*, no. AIAA-2017-3273, American Institute of Aeronautics and Astronautics, June 2017.