Proposal of a U-space Service for Tactical Conflict Prediction in a Multi-USSP Environment

Abraham Garcia, Sandra Amarillo, Silvia Esparza and Juan V. Balbastre
Air Navigation Systems (SNA)
Universitat Politècnica de València (UPV)
Valencia, Spain
{agariba, sanamva, silesbe1, jbalbast}@upv.edu.es

***Abstract*—This paper develops the concept of a U-space tactical conflict prediction service in a multi-provider environment. This new service, aligned with the CORUS CONOPS Ed. 4.0 and inspired by the architecture of the *Discovery and Synchronization Service* set out in the ASTM F3411-2022a standard, allows the exchange of telemetry between U-space service providers through a centralized system that periodically updates the positions of the unmanned vehicles. Furthermore, this document assesses the implementation of the new service by integrating concepts adopted from the BUBBLES project for tactical conflict detection, with initial proposals for data structures, databases, and system interoperability requirements. Lastly, simulations using real flight telemetries during the BUBBLES project were conducted to evaluate the results. More precisely, the number of conflicts detected and the detection time were computed and compared with those obtained in a simulated single-provider environment by the BUBBLES project.**

***Keywords*—MultiUSSP; SPATIO; Tactical conflict prediction; U-space; U-space service.**

I. Introduction

Several projects carried out in recent years have focused on the challenge of detecting and solving both tactical and strategic conflicts among Unmanned Aerial Systems (UAS). All of them have collaborated in the design and creation of new U-space services. One of those projects is BUBBLES [1], which pursued mitigating the risks of mid air collisions by implementing separation volumes around the UAS commensurate to their contribution to the global risk. BUBBLES developed for validation purposes a tool capable of detecting tactical conflicts between UAS by receiving telemetry in real-time and applying methods and algorithms developed by the project. Building on BUBBLES, another project is underway: U-space separation management (SPATIO) [2], which extends the research by focusing on airspace separation management and capacity to ensure UAS operations remain safe and efficient.

This research, framed within the SPATIO project, adopts the concepts developed in BUBBLES and applies them in a multi-provider environment, so that a U-space service provider is capable of detecting all tactical conflicts affecting UAS of its subscribed operators, regardless of the service provider for each involved UAS. To this end, the concept of Discovery and Synchronization Service developed in the ASTM F3411-2022a standard [3] has been taken as inspiration for the design of a centralized service that exchanges telemetry.

All these ideas have allowed the conceptual development of the new U-space service and also to carry out an implementation test from which to obtain initial results and performance metrics. To accomplish this, a structure compatible with the tools previously developed in BUBBLES is proposed, and other approaches are explored to meet the remaining requirements necessary for the proper functioning of the service.

From the simulations carried out, which have been performed using the real flight recorded telemetry of the BUBBLES validation campaign [4], the number of conflicts detected, the execution time of the programs and the conflict detection time have been analyzed. These values have been compared to those from a single-provider scenario, also computed using the tools developed by BUBBLES. The comparison showed that the proposed service provides in a multi-provider environment results which are equivalent to those obtained in a single-provider one.

The rest of the paper is structured as follows: Section II provides a contextualization through the two main concepts on which this research is based: the mechanism developed by the F3411-22a standard and the separation management concept of operations developed by BUBBLES. After that, Section III presents the steps followed in the development of the project. First, the service to be designed is conceptually defined along with its operational scenario and functionalities. Then, implementation details are provided, describing the programs involved and the tasks performed by each of them. This implementation leads to the results presented in Section IV, which are discussed in Section V. This section also summarizes the conclusions drawn from the results and suggests the next steps to be undertaken in the future.

II. Background

In strong alignment with the purpose of this paper, the taxonomy regarding the different methods of Conflict Detection and Resolution (CD&R) outlined by [5] has facilitated a review of the existing CD&R algorithms based on their approaches to surveillance, control, obstacle types, etc. The review of these algorithms has helped to address the objective of this paper by examining different existing methods.

Regarding conflict prediction, some studies have tackled the use of enhanced techniques for trajectory prediction, tracking, and conflict resolution. In [6], an extended Kalman filter is proposed for predicting optimal 3D trajectories. On the other hand, although more related to conflict resolution (beyond the scope of this project), [7] and [8] present solutions that involve assessing the worst-case conflict between UAS through shared information, calculating the CPA[1] and the estimated time of arrival at the CPA. Another such algorithm, developed in [9], delves into generating timely alerts for each agent when small UAS are in conflict, ensuring a safe integration in the airspace.

In this paper, as will be discussed later, an octree structure is used as a representation of airspace. Such structures have already been tested for CD&R methods among UAS, as shown in [10]. In that work, the use of octrees was proposed to efficiently find 3D collision-free trajectories.

Finally, some studies have investigated CD&R in multi-agent scenarios. In [11], collision avoidance in multi-agent scenarios with trajectory tracking is examined. However, the most notable is [12], which assumes that different UAS are operated by independent U-space service providers (USSPs) in shared airspace. This study explores CD&R methods during the flight phase, applied within a CD&R service. This service is a centralized UTM service to which various active USSPs connect. Nevertheless, none of them fully addresses the objective of this paper, which is setting up a multi-provider environment where different USSPs, through their conflict detection services, can detect tactical conflicts between their subscribed UAS and those subscribed to other USSP.

To accomplish this goal, two main elements are necessary: a conflict detection service and a centralized information exchange service. To achieve this, the ideas presented in the ASTM F3411-22a standard and in the BUBBLES project have been exploited.

A. *ASTM F3411-22a*

The American Society for Testing and Materials, in its F3411-22a standard, sets out the performance requirements for the Remote Identification of the UAS [3]. The proposed scenario considers the presence of different USSPs, which have to exchange data. The interactions between USSPs are carried out through the Discovery and Synchronization Service, hereafter referred to as DSS.

The DSS is a standardized mechanism that allows a USSP to obtain relevant information from another USSP. This is fulfilled with the following steps, also represented in the Fig. 1.

1) Make Discoverable: When a USSP possesses relevant information that should be known by the other USSPs, it will share it with the DSS.

2) Discover: When other USSPs are interested in certain information, they will consult the DSS using a 4D volume that encompasses the area of interest (this volume is named as ISA). The DSS, which stores the shared information in a representation of the airspace in the form of a grid, detects which grid intersect with the ISA and returns to the requesting USSP a list of all relevant information and the USSP that owns that information. This query can be either a one-time request or through a subscription to the 4D volume, so that the USSP will be notified again if new information is shared.

3) Get Details: Once the USSP has gathered the information, it would use a data exchange protocol between USSPs to obtain the remaining relevant information.

4) Subscription Notifications: In the case of a subscription, when a new USSP shares information with the DSS that falls within the ISA of the subscribed USSP, the DSS would inform the new USSP of the existing subscription and provide the details for contacting the subscribed USSP.

The interoperability between USSPs is attained using a RESTful API[2], which authorizes the use of four different methods in order to access the resources on the server. These are: GET, POST, PUT, DELETE.

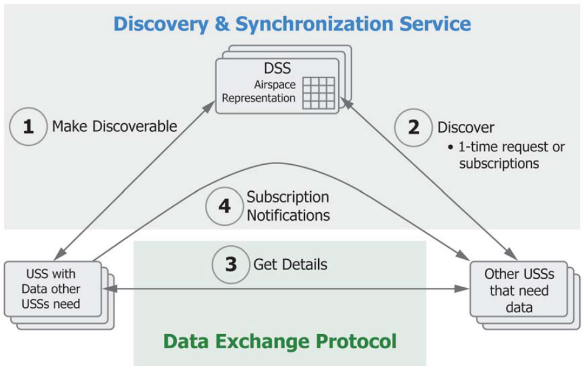

Figure 1. Representation of the DSS Data Exchange Protocol [3]

The idea of DSS is more complex than just the scheme already explained shown in Fig. 1, and is designed to operate in a U-space airspace with different DSS (that part is related with the 'Synchronization' function). However, the aim of this research is to design a centralized service that is able to exchange telemetry between USSPs so that each USSP can detect the tactical conflicts between the UAS subscribed to it and the rest of UAS flying in the airspace. To this end, the concepts briefly displayed formerly are the ones that will serve as inspiration for the design and development of the new U-space service of this research.

B. *BUBBLES Project*

BUBBLES was a SESAR2020 Exploratory Research project coordinated by the *Universitat Politecnica de Valencia* that focused on the management of tactical conflicts in the U-space airspace. This project serves as a basis for the tactical conflict detection service. In fact, this tool was already developed by the BUBBLES project for validations purposes in a single USSP scenario [13], and the goal now is to adapt it to a multiUSSP environment so that it operates together

[1] CPA or Closest Point of Approach is the point in space and time in which the range between the UAS under observation is the minimum.

[2] Application Programming Interface (API) is an interface that defines how communication will occur between different software or applications. The fact that it is RESTful implies that the protocol for information exchange takes place over HTTP, that is, via the Internet.

with the new centralized U-space service of data exchange. For that reason, it is relevant to review some concepts defined by the BUBBLES project.

The separation management depends on the operation

According to BUBBLES [14], each UAS is classified depending on its characteristics and performance. Eight different classes for unmanned aircrafts are defined, as it can be observed in Tab. I.

TABLE I. UAS CLASSIFICATION ACCORDING TO PERFORMANCE AND SIZE [14]

Traffic Performance	Cruise Sp. (m/s)	RoC (m/s)	RoD (m/s)	Size_h (m)	Size_v (m)
Open A1	5.00	4.00	3.00	0.50	0.25
Open A2	5.00	4.00	3.00	1.00	0.50
Open A3	10.00	4.00	3.00	2.00	1.00
Specific SAIL_I_II	12.00	4.00	3.00	1.00	0.50
Specific SAIL_III_IV	14.00	5.00	4.00	2.00	1.00
Specific SAIL_V_VI	15.00	5.00	4.00	2.00	1.00
Certified_No_Pass.	25.00	5.00	4.00	4.00	2.00
Certified_Pass.	25.00	3.00	2.00	5.00	2.50

The separation in BUBBLES is managed through protection volumes, centered at each UAS, so that when two volumes overlap there is a separation loss. These volumes, defined by a radius for the horizontal plane and a height for the vertical plane, change in size depending on the traffic class of the UAS. The vertical separation is constant for every class, while the horizontal separation increases with the UAS classes. The dimensions of the tactical conflict protection volumes proposed by the BUBBLES project for urban areas in uncontrolled airspace are collected in Tab. II.

TABLE II. PROTECTION VOLUME MEASUREMENTS FOR EACH UAS CATEGORY [14]

Category	Radius (m)	Height (m)
Open A1	179.22	36.22
Open A2	179.22	36.22
Open A3	337.75	36.22
Specific SAIL_I_II	401.16	36.22
Specific SAIL_II_IV	439.14	36.22
Specific SAIL_V_VI	469.03	36.22
Certified_No_Passenger	767-92	36.22
Certified_Passenger	895.1	36.22

For each scenario with two UAS, the separation minima that must prevail depends on the particular characteristics of both UAS. The separation minima are then computed as the sum of the two horizontal separations (in the horizontal plane), and the sum of the two vertical separations (for the vertical plane). The horizontal separation is represented in Fig. 2 as the sum of the radii of each protection volume. For the vertical separation, the representation is the same but summing heights instead of radii.

The calculations of the separation minima do not fall within the scope of this document, but is fully developed in the BUBBLES Concept formulation document [14]. Nevertheless, one of the distances used for these calculations, known as Near Mid Air Collision or NMAC, will be applied later. It is worth mentioning that this distance has a horizontal length of $d_{NMAC} = 25(ft)$ and a vertical length of $h_{NMAC} = 7.5(ft)$ regardless of the traffic class.

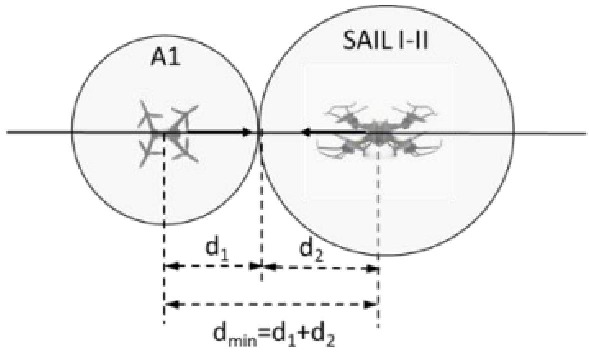

Figure 2. Horizontal representation of two protection volumes defined by radii d1 and d2 [14]

The separation management depends on time

The tactical conflict is defined as the predicted convergence of aircrafts in space and time. Thus, to detect such conflicts, not only the current position is used but also the predicted trajectory. Therefore, for a tactical conflict to occur, the distance between UAS at the CPA must be less than the separation minima, and the time to reach the CPA must be less than a pre-established threshold.

U-TraC Tool

The U-TraC is the validation platform developed by the BUBBLES project [13]; it is a tool that collects real-time telemetry from UAS and applies the algorithms and methods defined in the BUBBLES project to detect tactical conflicts.

The U-TraC is implemented in Matlab, and the data flow between this tool and the rest of the systems is carried out via an MQTT[3] server.

For validating the BUBBLES concept, in 2022 a trial with 13 simultaneous UAS flights took place on a rural area in the North of Valencia, Spain. During that experiment, the telemetry of all the UAS flights was recorded, and also the output of the U-TraC with the detected conflicts.

Furthermore, a human machine interface (HMI) was developed in JavaScript, enabling the visualization of the UAS on a map indicating the existing conflicts. All the information on this display comes from the output of the U-TraC. This can be observed in Fig. 3. All the UAS processed by the U-TraC are represented by an arrow-shaped symbol, pointing towards its orientation. The colour red indicates an existing conflict, while the colour green means free of conflict. The orange points show the last positions of each UAS. On the left hand of the screen, more information about the conflicts and the performance of each UAS is displayed.

[3]MQTT or Message Queuing Telemetry Transport is a communication protocol between devices that applies a message queue management using an online server.

Figure 3. Example of output of the U-TraC visualized through the HMI

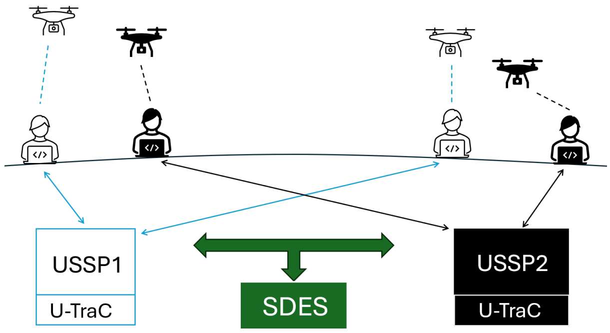

Figure 4. Operational scenario of the SDES

III. Methodology

The methodology adopted for the design and development of the new U-space service begins with the conceptual formulation of the service. To this end, the previously outlined concepts will be utilized, with certain adjustments introduced to streamline the design process. The service will be inspired in the DSS concept as defined by the F3411-2022a standard.

Following the definition of the functionalities and structure required for the service, an implementation will be undertaken to validate the concept by means of simulations. This implementation was built on the products developed within the BUBBLES project, which are specifically intended for the detection of tactical conflicts.

A. Concept Formulation

Given that the new U-space service involves the synchronous exchange of data (primarily telemetry), it has been decided to name it as **Synchronous Data Exchange Service**, or **SDES**.

This service incorporates features and functionalities of the DSS but without attempting to be a DSS as such. From all the characteristics presented in the F3411-22a standard, the following ones have been kept:

- There is just one SDES per U-space airspace, being a centralized service that does not require synchronization with others.
- The interactions between USSP and SDES will be simplified to 1) Make Discoverable and 2) Discover. The option for subscription is discarded, as subscribing to a 4D volume is deemed unnecessary when tactical conflict detection relies on real-time positions. Additionally, the step of communication between USSPs to obtain further information is also eliminated, as it is considered that this step would introduce a time delay that could be crucial for the prompt detection of conflicts.
- The information discovered by the SDES will be saved in a representation of airspace segmented into cells.
- An API RESTful will be used for the interoperability between USSP and SDES.

Fig. 4 illustrates the operational scenario of the service. In a U-space airspace, multiple UAS would be operated by their respective operators, each one subscribed to a USSP that provides U-space services just to their subscribed operators. The U-TraC would correspond to the service dedicated to the detection of tactical conflicts between UAS, currently limited to single-provider environments, where all UAS are subscribed to the same USSP that receives telemetry data from all the UAS that operate in that volume. Now, U-TraC shall receive telemetry only from its subscribed UAS, but has to detect any conflict affecting them (including those involving UAS served by other USSPs). It is the responsibility of the SDES to supply each USSP (and consequently U-TraC) with the necessary telemetry for conflict detection, regardless of the USSP that caters for each UAS.

The SDES is the data exchange service between USSPs. It collects traffic information from each UAS flying in the airspace and periodically provides it to the corresponding USSP, with a pre-established update cycle. In this way, the U-TraC of a USSP would have access to both the telemetry received from its subscribed UAS operators and the telemetry requested to the SDES.

To determine which data a USSP needs to request from the SDES, the concept of area of interest or ISA defined in [3] is used. This will be a 3D volume (since the fourth dimension disappears when working in real-time) centered on the position of each UAS, as shown in Fig. 5.

This volume will vary depending on the class of UAS making the request. The SDES segments the airspace into cells, in which it stores the received information. When the

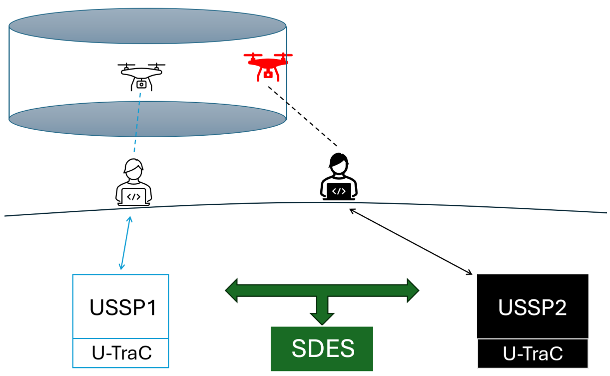

Figure 5. Representation of the ISA

area of interest intersects with a cell containing a UAS, the service returns the information stored in that cell to the USSP. In the case shown in Fig. 5, USSP1 sends a volume of interest centered on the white UAS to the SDES. The SDES analyzes the cells and, since the UAS operated by the subscriber to USSP2 is within the area of interest, it would send the telemetry of this UAS to USSP1 so that its U-TraC can check whether there is a conflict. Similarly, the same process will occur with USSP2 as the requester, allowing both USSP to detect the conflict and notify their operators.

It is important to emphasize that the SDES is not a conflict detection service; it is merely an information exchange service between USSPs. In fact, the presence of a UAS within the ISA does not imply that the two UAS are in conflict (it is the responsibility of the U-TraC to detect this). The concept of ISA and how it is calculated will be discussed later.

The cell division is more accurately depicted in Fig. 6. The SDES receives telemetry from all UAS and places them into a cell within the airspace. In the image, the color of each UAS indicates the USSP that provides the services, with the example applied to UAS 5. Its USSP makes a request to the SDES using an area of interest centered on the UAS. The SDES will analyze this volume and identify which cells intersect wholly or partially with the ISA. It will then return the information of the UAS located within these cells to the requesting USSP. In the example, the USSP would receive information on UAS 1, 2, and 3. UAS 1 is included because, although the UAS is not within the ISA, it lies in a cell that intersects with it. On the other hand, UAS 4, despite being covered by the ISA, is subscribed to the same USSP as UAS 5 and thus its telemetry is already known by the USSP.

Since the service returns information on UAS that are not directly intersected by the ISA but are within intersecting cells, it is crucial that the cell size is sufficiently small to prevent the transmission of data from UAS that are far away and are not relevant. This helps avoid overloading the processing of UAS by each USSP. The selection of cell resolution will also be discussed later.

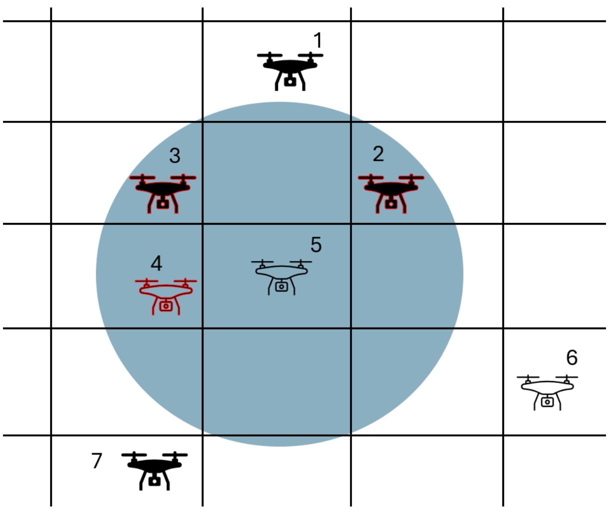

Figure 6. 2D representation of the airspace division into cells with the ISA

Finally, the interactions between the USSP and the SDES are schematically represented in Fig. 7. The USSP is responsible for providing the pilot with an application that sends the telemetry to the MQTT server. The U-TraC collects this telemetry, processes it, and makes it public to the SDES (*Make Discoverable*). At the same time, it makes the necessary requests (*Discover*) to identify potential UAS in the vicinity of its UAS. The SDES records all the telemetry data, analyzes the received ISA, and returns the relevant telemetry of nearby UAS to the corresponding USSP. Additionally, the SDES maintains a record of all the information in a database. When the U-TraC receives all the telemetry, it searches for potential conflicts and presents them to its operators for appropriate action. The interactions between USSP (U-TraC) and SDES must be done through an API RESTful.

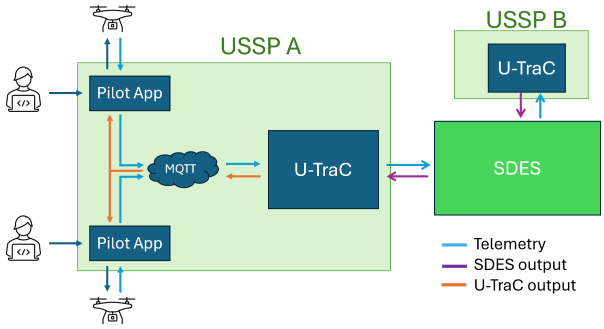

Figure 7. Scheme of the interactions USSP-SDES

B. *Implementation*

To explain how the implementation of the SDES was achieved, it is necessary to examine each step of the process, focusing on every component or link in the communication chain between the USSP and SDES. The order is as follows: U-TraC, API and SDES (including the database entries).

*1) **U-TraC**:* The U-TraC tool had to be modified to integrate it into a multi-USSP environment, but the core functionality of the program remains unchanged. The introduced changes can be summarized as:

- When the U-TraC receives the telemetry from UAS subscribed to the USSP, it generates the message that must be communicated to the SDES. This message transmits only the values required by the U-TraC to detect tactical conflicts. This approach ensures that the operational privacy of each flight is maintained, avoiding the transmission of sensitive or private information.
- In addition to the telemetry data, the message also includes the requested ISA to the SDES.
- In the same way that the functionality for sending information to the SDES is added, the step of processing the information received from the SDES is also incorporated. Once the U-TraC has all the relevant telemetry, it performs the calculations for detecting tactical conflicts. This entire process is repeated in each iteration.

a) Message format: The information sent by the U-TraC is a *JSON* message that contains the following fields:

- Telemetry data received
- UAS ID
- UAS category
- ISA, defined by a point of application, height and radius.

All the data included in the message aims to comply with the ASTM F3411 data model.

b) ISA calculation: The ISA is defined as a cylinder whose parameters depend on the category of the UAS, as the separation minima vary depending on the type of traffic. Horizontal and vertical dimensions of the tactical conflict volumes specified in the BUBBLES project have been used (Tab. II).

However, due to the meaning of tactical conflict, the ISA should be defined as the sum of the radius/height corresponding to the tactical conflict volume of the requesting UAS category plus the worst-case scenario (i.e., summing the highest volume dimensions, corresponding to the Certified Passenger category), since the category of nearby UAS is initially unknown. This method ensures that the information is transmitted when tacticak conflict volumes overlap, or in some cases even if they do not intersect, but it guarantees that when the volumes touch, the ISA will always detect the conflicting UAS.

*2) **API**:* The message exchange between U-TraC and SDES must be carried out through an API. The language selected for the design of this API is Python, and two basic methods have been implemented: GET and POST. These two methods facilitate a basic communication between services.

Fig. 8 aims to synthesize the interactions between USSP and SDES by defining the message exchange protocol. First, via a POST request to the API, a USSP informs the SDES of the telemetry data from its UAS flying in the airspace. The SDES collects this information and stores it in a representation of the airspace divided into cells. This same USSP, or another one in need of information about third-party UAS, will then send another POST request to the API with the corresponding ISA for each of its UAS (a USSP can only obtain information about UAS flying nearby if it has at least one UAS in flight). Next, via a GET request, the USSP retrieves the telemetry of the UAS within the ISA's range. This process is carried out periodically, with a frequency established by the competent authority. In this paper, a value of 1 per second has been used.

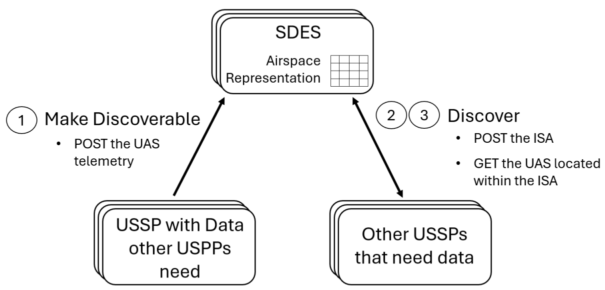

Figure 8. Representation of the SDES Data Exchange Protocol

*3) **SDES**:* The SDES has been developed in MATLAB in order to facilitate the work together with the U-TraC, and performs the following tasks in each work cycle:

1. Connection with the API of active USSPs within the airspace.
2. Cell mesh generation: First, the airspace covered by the service is defined using an ENU (East, North, Up) coordinate system to simplify operations. Although this might cause minor altitude errors, the small size of the area makes them negligible. After defining the boundaries, the cell mesh is generated using an octree.
   An octree is a data structure made up of nodes that recursively subdivides. The name is derived from the number of divisions of the nodes, as each parent node is subdivided into 8 child nodes. This method is implemented such that the entire airspace corresponds to the root node, which starts to be divided into eight octants recursively until the minimum cell size is reached. These last nodes are the leaf nodes, which do not continue dividing themselves. The minimum cell size established is the *Near Mid Air Collision* defined in BUBBLES. The structure of the octree is represented in Fig. 9.

   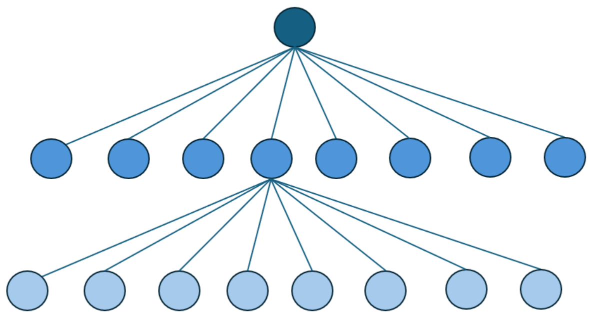

   Figure 9. Representation of the octree structure

   When the SDES is initialised, the root node is generated with the airspace size and it starts dividing once the UAS are inserted in the data structure.
3. Reading UAS: After the SDES setup, the program begins reading the UAS data from all USSPs. Data such as the number of active UAS, their UAS ID, and their originating USSP is retrieved.
4. Message Processing: For each received message, the UAS's position (usually in WGS84 coordinates) is converted to ENU coordinates, and the telemetry is saved for future verification.
5. Cell Assignment: Each pair of UAS and message is assigned to a cell in the octree, returning the ID of the cell where the data is placed.
   The UAS position is analyzed, and first is checked if it is within the root node limits. If so, it starts the recursive division of each node until the UAS is placed

in a leaf node. To enhance the process efficiency, the entire airspace is not initially divided. Instead, only the nodes containing the position of a UAS are subdivided. This can be observed in Fig. 10, where the red points represent the inserted UAS.

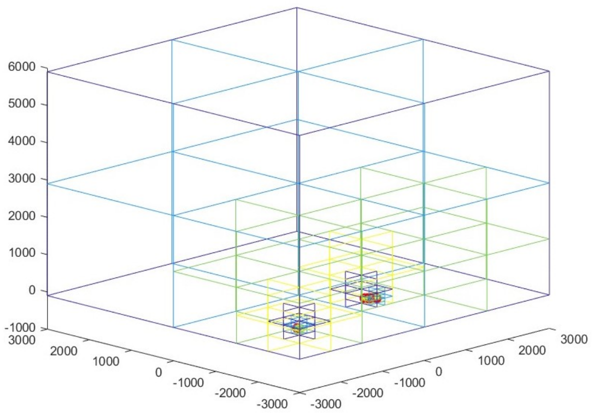

Figure 10. 3D View of the octree with inserted UAS

6. Database: Messages are saved in a database and retained during a period of time established by the competent authority. The information is saved in two different tables related by a value such that accessing one allows retrieving the data stored in the other; this is attained using a SQL database. The *missions* table gathers flight-related information, such as the mission identifier, the UAS identifier, or the flight date. On the other hand, the *telemetry* table collects the telemetry data as well as the cell identifier for that position. Telemetry is continuously updated, while mission table entries are only made once to define the mission ID. The two tables are connected through the mission ID.
7. ISA Processing: Parameters for each UAS search cylinder (i.e., the UAS ISA) are extracted from the message. The volume is processed and the SDES obtains the data of all UAS located in the cells that intersect the ISA.
8. Message Filtering: The retrieved UAS are filtered based on three factors: the requesting USSP, the USSP of the detected UAS, and ensuring that no UAS is detected more than once. This ensures that, if the requester is a USSP A, they only receive UAS data from USSP B, without duplicate entries.
9. Sending Information to USSP: After processing all requests, the SDES uses the POST method to publish detected UAS information through the corresponding API. This process occurs on a 1-second update cycle.
10. Cell mesh cleanup: At the end of each iteration, the octree is cleared by removing all inserted UAS and messages. This is done for efficiency, as the system only works with updated data. However, the implementation allows for keeping all messages if needed for continuous operation.

IV. Results

Performance tests have been conducted using the recorded telemetry from the BUBBLES validation campaign; that is, telemetry from 13 real flights of UAS carried out in 2022 [4]. From these tests, performed both by simulating a single-USSP environment (without SDES) and a multiUSSP environment, the following results were obtained.

A. Number of detected conflicts

An environment was simulated where a USSP A controls six UAS, and a USSP B manages seven different UAS. To verify the number of detected conflicts, the U-TraC output is analyzed in both scenarios.

The simulations were successful, with all the tools functioning correctly and telemetry exchange via the SDES working as expected. The HMI (Fig. 3) displayed both subscribed UAS and nearby third-party UAS, accurately detecting conflicts. Once UAS leave the ISA, they disappeared from the screen after a short delay. The tests showed that the number of detected conflicts remains consistent across scenarios, with no interference from the SDES.

B. Detection time

However, despite correctly detecting the tactical conflicts, the introduction of this new service inevitably induces a delay in conflict detection. In the single USSP scenario, the path the telemetry had to follow was shorter and simpler than the one it must follow now in a multiUSSP scenario.

To establish a reference for these new detection times, the different cycles and durations related to the new programs have been calculated. This includes analyzing the execution times of both the U-TraC and the SDES, taking into account the factor of database storage (as this task can be somewhat costly in terms of time). Additionally, the conflict detection time between UAS from different USSPs is examined. A set of Monte Carlo trials with 100 runs per trial were conducted, yielding the following results:

1) ***U-TraC execution time****:* The execution times of the U-TraC in both scenarios are compared in Tab. III, analyzing both absolute execution times and average times (time of each iteration divided by the number of active UAS) with their mean values (μ) and standard deviations (σ).

TABLE III. U-TraC execution times

	Single USSP		multiUSSP	
	Absolute times (s)	Average times (s)	Absolute times (s)	Average times (s)
μ	0.0012	8.9577 E-05	0.0345	0.0042
σ	4.3571 E-04	3.3516 E-05	0.0346	0.0046

It can be observed how the new implementation increases the execution time by 30. This is mainly due to the new connections with API and readings from new UAS. These new results also show greater variance, with atypical values occurring with some frequency. Nevertheless, execution times of 35 ms remain more than acceptable results.

*2) **SDES execution time**:* The objective of the SDES is to be a traffic information exchange service with a defined update interval. The goal was to achieve an update time of 1 second. To determine if this time was feasible, the execution times of the program were studied both with and without a database.

TABLE IV. SDES EXECUTION TIMES

	Without Database		With Database	
	Absolute times (s)	**Average times (s)**	**Absolute times (s)**	**Average times (s)**
μ	0.1667	0.0128	0.8381	0.0645
σ	0.0324	0.0025	0.1254	0.0097

Introducing the database factor increases the execution times by up to 5 times, causing iterations to last longer than 1 second. Nevertheless:

- In the 100% of the cases studied without database, the execution time is lower than the update time (1 s).
- In the 93.9% if the cases studied with database, the execution time is lower than the update time. And in the 100% of the cases, the execution time is lower than 1.3 s.

These results fulfill two rules:

- According to AMC1 Art11 (3) [15], traffic information distribution should have a latency of less than 5 seconds 99% of the time.
- According to RTCA DO-396 [16], the traffic information distribution requires an update time of 1 s in 95% of the cases (with database this point is almost achieved with 93.9% of the cases).

Therefore, it can be concluded that it is possible to implement the SDES with an update time of 1 second in both cases, achieving results within a good confidence interval.

*3) **Conflict Detection time**:* Finally, the detection time of a conflict between two UAS from different USSPs was studied, including the time since one USSP makes its UAS public until the other USSP receives it.

For the case of single USSP scenario, this detection time had a mean of 0.1852 s and a standard deviation of 0.1073 s. In this scenario the time since one USSP sends the telemetry until the other receives it is not relevant as there is only one USSP.

For the multiUSSP scenario, the results are shown in Tab. V. The UAS reception time has been measured as the time from when USSP B sends the telemetry of one UAS to the SDES until USSP A receives and registers it. On the other hand, the conflict detection time is equal to the previous time plus the interval from when the UAS is received by the new USSP until the conflict is detected.

TABLE V. RECEPTION AND CONFLICT DETECTION TIMES IN MULTIUSSP SCENARIO

	Without databse		With database	
	Receive UAS (s)	**Conflict detection (s)**	**Receive UAS (s)**	**Conflict detection (s)**
μ	1.5499	2.0824	1.8003	2.2911
σ	0.2410	0.4068	0.0546	0.1755

These results should be very similar, as despite having or not having the database, the SDES has an update cycle set at 1 second for both cases. Therefore, it is impossible to expect results lower than one second for the reception of the UAS by the new USSP. On the other hand, the difference in the average times will be caused by what was mentioned in the previous section: some SDES iterations with a database exceeded the update time, causing the SDES to take more than 1 second to make the information public in some cases. This does not happen without a database, as in 100% of the cases, the update time was respected.

V. CONCLUSION

The main objective of the research, which was to develop and implement a U-space information exchange service between USSPs, has been successfully achieved. The work presented in this paper was built on two solid foundations: the BUBBLES project and the DSS concept developed by ASTM. This enabled the conceptual development of the SDES, with a clear definition of the operational scenario and its functionalities. The implementation aimed to apply these concepts to tests that provide initial results.

These initial results have been very promising. As mentioned, the telemetry data distribution times comply with current regulations, making SDES an useful and functional service. However, its implementation is still in the early stages. It was developed to integrate with the programs created in BUBBLES, but to improve these results, it will be necessary to optimize those programs by migrating them to other software. This is one of the steps to be taken in the future, along with new ideas such as optimizing the database (as it introduces a noticeable delay) and improving the ISA application. The latter could be dynamically applied, adding the separation minimum of the highest category flying in that airspace at the moment, rather than the highest category in general.

This idea of service is not only applicable for tactical conflict detection but can also be useful for strategic conflict detection applications if applying some modifications. In the future, the goal would be to develop an application that integrates this new U-space service with other U-space services in development, as well as with the mandatory U-space services outlined in regulations.

ACKNOWLEDGMENT

This research was conducted within the framework of SPATIO and CREATE U-space projects. SPATIO project has received funding from the SESAR 3 Joint Undertaking (JU) under grant agreement No 101114674. The JU receives support from the European Union's Horizon Europe research and innovation programme and the SESAR 3 JU members other than the Union. CREATE U-space project (CIAICO/2022/044) has received funding from the Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital of the Generalitat Valenciana.

REFERENCES

[1] SESAR. *BUBBLES - Defining the BUilding Basic BLocks for a U-Space SEparation Management Service*. SESARJU.eu. https://www.sesarju.eu/projects/bubbles (accesed Sep. 12, 2024).

[2] SESAR. *SPATIO- U-Space Separation Management*. SESARJU.eu. https://www.sesarju.eu/projects/SPATIO (accesed Sep. 12, 2024).

[3] *Standard Specification for Remote ID and Tracking*, F3411-22a, ASTM International, United States, 2022.

[4] UPV. *"Burbujas de seguridad para los UAS"*. UPV.es. https://www.upv.es/noticias-upv/noticia-13547-burbujas-de-se-es.html (accesed June 28, 2024).

[5] Ribeiro, M.; Ellerbroek, J.; Hoekstra, J. *Review of conflict resolution methods for manned and unmanned aviation.* Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology. June 16, 2020.

[6] Prévost, C.G.; Desbiens, A.; Gagnon, E.; Hodouin, D. *Unmanned Aerial Vehicle Optimal Cooperative Obstacle Avoidance in a Stochastic Dynamic Environment. J. Guid. Control. Dyn.* 2011, Vol 34.

[7] Park, J.W.; Oh, H.D.; Tahk, M.J. *UAV collision avoidance based on geometric approach.* In Proceedings of the 2008 SICE Annual Conference, Tokyo, Japan, 20–22 August 2008; IEEE: Piscataway, NJ, USA, 2008.

[8] Manathara, J.G.; Ghose, D. *Rendezvous of multiple UAVs with collision avoidance using consensus. J. Aerosp. Eng.* 2012, Vol 25.

[9] Ong, H.Y.; Kochenderfer, M.J. *Markov Decision Process-Based Distributed Conflict Resolution for Drone Air Traffic Management. J. Guid. Control. Dyn.* 2017, Vol 40.

[10] Kitamura, Y.; Tanaka, T.; Kishino, F.; Yachida, M. *3-D path planning in a dynamic environment using an octree and an artificial potential field.* In Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction and Cooperative Robots, Pittsburgh, PA, USA, 5–9 August 1995; IEEE Comput. Soc. Press: Piscataway, NJ, USA, 1995.

[11] Alonso-Mora, J.; Naegeli, T.; Siegwart, R.; Beardsley, P. *Collision avoidance for aerial vehicles in multi-agent scenarios. Auton. Robot.* 2015, Vol 39.

[12] Ho, F.; Geraldes, R.; Goncalves, A.; Cavazza, M.; Prendinger, H. *Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace. IEEE Trans. Veh. Technol.*2019, Vol 68.

[13] Cecilia Claramunt Puchol et al., *BUBBLES Separation Management Environment: architecture and validation of a separation management tool for UTM*, Proceedings of the 23rd Integrated Communications, Navigation and Surveillance Conference (ICNS 2023), available online in IEEE Xplore.

[14] Cecilia Claramunt Puchol et al., *BUBBLES: A new concept of operations for separation management in the U-space*, Proceedings of the 12th EASN International Conference on Innovation in Aviation & Space for opening New Horizons, published in Journal of Physics: Conference Series (JPCS), OP Publishing,2023.

[15] *Acceptable Means of Compliance and Guidance Material to Regulation (EU) 2021/664 on a regulatory framework for the U-space*, EASA, Dec. 2022.

[16] *Minimum Operational Performance Standards for Airborne Collision Avoidance System sXu (ACAS sXu)*, DO-396, RTCA, Dec. 2022.