**Macintosh HD:Users:Dependable:Shared:Documents:Templates:DCi.logo.pdf**

Sponsor: AFRL

Project No.: 144

Contract: ASTRA IST

This document is for authorized use only. Any information contained herein should be considered proprietary unless otherwise specified.

© 2023 Dependable Computing. All Rights Reserved

Dependable Computing

V0.7 1/31/2024

Top-Level Specification

Zone Alert Service

Abstract

This document provides the intended functionality of an OpenUxAS Zone Alert Service as a top-level solution specification. The specification is meant to only be as formal as is required to be reasonably accurate for all readers. The specification defines what functions/behaviors the solution provides and the model of how it provides then fitting within the OpenUxAS environment.

Changes

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Change** |
| V0.1 | 1/10/2024 | Initial draft |
| V0.2 | 1/22/2024 | Updates based on behavior validation |
| V0.3 | 1/24/2024 | Added issue tracking and issues for required OpenUxAS integration research |
| V0.4 | 1/24/2024 | Removed issue tracking in docs and switched to a global issue log as a separate artifact. |
| V0.5 | 1/30/2024 | Updated specification based on use of zone buffers rather than vehicle radius, position used for route deviation rule, defined order of precedence for notifications when both present and future violation are detected for same zone and aircraft. |
| V0.6 | 1/30/2024 | Introduced validation of the specification for correctness and completeness |
| V0.7 | 1/31/2024 | Updated functional spec to latest behavior specification. Added top-level software design architecture specification. |

.

Contents

1 Introduction 1

2 An Important Notice About Document Synchronization 1

3 Functional Specification 1

4 Software Structure Specification 4

5 Validation Analysis 5

5.1 Completeness Assessment 5

5.2 Correctness Assessment 5

1. Introduction

The Zone Alert Service is a stand-up service that can be activated as part of an OpenUxAS instance. Its purpose is to alert subscribers to when vehicles associated with the UxAS instance in one of two states:

1. Are in current violation of a zone boundary.
2. Are in danger of violating a zone boundary based on current linear velocity and a lookahead time window.

There are cases where the occurrence of both scenarios simultaneously for the same aircraft and zone must be considered. These cases are more carefully defined below as a functional specification.

1. An Important Notice About Document Synchronization

**Important Notice**: *This functional specification must be kept synchronized and consistent with the behavioral documentation provided to stakeholders.*

**Important Notice**: *This functional specification must be kept synchronized and consistent with functional and non-functional requirements, as well as further design and specification of the solution system.*

1. Functional Specification

## Definitions

1. **Iterative Service**: An OpenUxAS service that computes and delivers its functional behavior at fixed, repeated soft real-time (best effort) intervals.
2. **Function Iteration**: An iteration of an OpenUxAS iterative service.
3. **Zone**: A zone is defined a closed, simple polygon with vertexes that define its edges.
4. **Zone Violation**: A *zone violation* at time **T** for a vehicle and zone is defined as the prediction position **P** of the vehicle violating the zone and the zone being part of the vehicle’s operating region.
5. **Keep-In Zone Violation:** A position **P** is in *keep-in zone violation* with a zone if **P** is outside the edges and area of the *zone*’s defined polygon.
6. **Keep-Out Zone Violation**: A position **P** is in *keep-out violation* with a zone if **P** is on one ore more edges or in the inside area of the zone’s defined polygon.
7. **TS**: The time of the last state report from a given vehicle.
8. **PS**: The position of the vehicle in its latest state report
9. **VS**: The velocity vector of the vehicle in its latest state report.
10. **TW:** The time lookahead window, in seconds, defined for the service.
11. **Current iteration time, TP:** The time of the start of the current iteration of the service ***for the current vehicle***.
12. **Vehicle projected linear trajectory**: The projected linear trajectory of the vehicle is the vector from last reported position **PS**, along its last reported linear velocity vector, **VS**, to the position of the vehicle **PW** at future lookahead time **TS+TW**. The projected linear trajectory of a vehicle is undefined if **TP** > **TS+TW**.
13. **Projected Position P at time T**: A vehicles projected position P at time T is the predicted position of the vehicle at time T on the vehicle’s projected linear trajectory. This is ony defined if the vehicle’s projected linear trajectory is defined and **TP** <= **T** <= **TS**+**TW**.
14. **Current Projected Position, Pp**: A vehicle’s *current projected position* **Pp**is the predicted position of the vehicle at current time **TP**. This is only defined if the projected position is defined for time **TP**.
15. **Potential Zone Violation at Time T and Position P**: A vehicle is in *potential violation* with a zone at time **T** at position **P** if and only if the projected position **P** at time **T** is in *zone violation* with the zone. This is undefined if the projected position is undefined at time **T**.
16. **Current Zone Violation**: A vehicle is in current violation with a zone if and only if its current projected position **PP** is in *potential* *zone violation* with the zone at the present iteration time **TP**. Current zone violation is not defined if the vehicle’s *current projected position* is not defined.
17. **Earliest Potential Zone Violation at time TV and position PV**: Given a vehicle and zone where there is a potential zone violation at time and position **TV** and **PV**, and there is no time **T <** **TV** that is a potential zone violation with the zone and aircraft, then **TV** and **PV** represent the earliest zone violation. It is not defined if no vehicle’s potential zone violations are defined.
18. **Worst Case Zone Violation at Time TX** at position **PV**. A vehicle with a potential future zone violation has a worst-case potential zone violation time, **TX**, that is calculated by computing the earliest potential Zone violation time using the maximum sped of the vehicle’s configuration declaration, rather than its last reported velocity’s speed. It is not defined if a vehicle and zone do not have a defined earliest potential zone violation.
19. **Route**: For a given vehicle, a route is a declared route along a set of waypoints being filed by the vehicle as its current activity in a mission.
20. **Current Route**: For a given vehicle, a route is current if the vehicle’s current mission action is to follow the route.
21. **Current Route Leg**: For a given vehicle with a current route, the current leg (path from previous waypoint to next waypoint) on which the vehicle is meant to be traveling.
22. **Path Deviation Tolerance**, **ED**: The allowed distance of a vehicle with a current route leg from its route below which no potential future zone violations will be reported.
23. **Separation Distance from Current Route Leg:** The minimum distance from a vehicle with a current route leg’s current projected position, **PP**, to a point on the current route leg.
24. **Adequately Following Route**: For a given vehicle with a current route, the condition that the vehicle’s separation distance from its current route leg is less than or equal to **ED**.

## Function 1: Report Route-Aware Zone Violations

### Assumptions:

1. The zone alert service is aware of any current route that the vehicle is currently following.
2. Routes are defined by waypoints and are defined so as to not violate zones along their path for a vehicle (taking into account turning capability and velocities of vehicle configs).
3. A ‘completed’ routed path is easy to discern from an uncompleted routed path.

### Function Definition:

1. The successive iterative period of the service is never greater than **TIt**.
2. At iteration time, **TP**, for each vehicle, for each zone, notification is sent to subscribers under the following rules:
3. If the vehicle and zone have a **current zone violation**, then notification is sent.
4. Otherwise, if the vehicle has a current route leg and the separation distance from the current route leg is greater than **ED**, and there is an earliest potential zone violation between the vehicle and zone, then notification is sent.
5. The notification consists of the following fields:
6. Vehicle ID for the vehicle.
7. Zone ID for the zone.
8. Keep-in or keep-out zone flag. Boolean: True = Keep-In. False = Keep-Out
9. Time of calculation **TP**.
10. Time of predicted earliest zone violation, **TV,** where **TP**<=**TV**<=**TS**+**TW**.
11. Worst case time to occurrence, **TX**.

## Function 2: Report Unable to Issue Alerts

### Assumptions:

1. Vehicle positions are exact, without noise, in the intended simulation environment.
2. There are a limited number of zones and vehicles, so that the number of simultaneous notifications to be sent and received will never burden OpenUxAS or services.

### Function Definition:

1. If during an iteration of the algorithm, for a given vehicle, **TP >TS+TW**, then a Stale State Error message is reported to subscribers to alert them that the zone alert service cannot be provided at this iteration for this vehicle because state reports for a vehicle are too old.
2. The Notification consists of the following fields:
3. Vehicle ID
4. Time of last state report from vehicle **TS**.
5. Time of computation, **TP.**

## Function 3: Initialize Service

### Definitions:

1. **Service Initialization**: Startup of the service under the control of OpenUxAS.

### Assumptions:

1. OpenUxAS executes initialization of the properly configured service

### Function Definition:

1. The service is initialized with configuration data values of the following form:
2. **Time window** **TW** in seconds, as a real number. Values greater than 0 are allowed.
3. **Iteration Period** **IT**, in seconds as a real number greater than 0.
4. **Path Deviation Tolerance ED**, in meters as a real number greater than 0.
5. The service will immediately terminate with an appropriate reported or logged error if it is misconfigured.
6. Software Structure Specification

The software will be structured as illustrated informally in the following diagram:

A diagram of a system

Description automatically generated

In any given iteration of the Zone Alert OpenUxAS service, A Zone Alert Checking system determines when an aircraft has a projected potential violation of zone for any currently registered aircraft. It computes this based on the authoritative representation of a zone polygon model (including all semantics such as padding) shared in a common UxAS codebase.

The ZoneAlertService is a wrapper that presents the checker as a service to OpenUxAS. It contains the configuration information and subscription to necessary messages from other services. It supports subscriptions from other services.

The CMASI file additions support the ZoneAlertService messages created by the service and reported to its subscribers.

1. Validation Analysis

The specification has been assessed for whether it is a correct specification of a solution given the defined behavioral description document and demonstrations presented to stakeholders. The analysis determines this specification’s completeness and correctness.

* 1. Completeness Assessment

### Completeness of Functional Specification

### Completeness of Software Architecture Specification

* 1. Correctness Assessment

### Correctness of Functional Specification

### Correctness of Software Architecture Specification