**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.

© 2024 Dependable Computing. All Rights Reserved

Dependable Computing

1/13/2024

Use Cases

Zone Alert Service

Abstract

This document provides use cases / behavior specification of the OpenUxAS one Alert Service.

.

Contents

1 Introduction 1

2 Basic Behaviors 1

3 Complex Behaviors 4

4 Satisfactory Parameters 4

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
2. Are in danger of violating a zone within a future time window.

These cases are more carefully defined in the next section.

1. Basic Behaviors

## Case 1: No Reporting Regarding Operating Regions

The system does not respond or care about operating regions. Any vehicle entering or leaving an operating region or existing inside or outside of an operating region has no impact on the zone alert service.

## Case 2: Violation of a Zone Boundary

If at any time the reported location of a vehicle is in violation of a zone boundary, the service will alert subscribers to the zone boundary violation. The above graphical example shows the classic examples of violating boundaries of a keep in our keep out zone, respectively. The service sends one notification for each zone that is violated. The service sends a separate notification for each vehicle for each zone that the vehicle is currently violating. It is assumed that 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.

### Violation Region

Each vehicle defines a radius around it considered its ‘collision radius’ as part of its entity configuration in OpenUxAS. This radius in combination with the vehicle’s current reported position is used to compute violation. If any part of this defined volume intersects any part of keep-in zone’s interior or keep-out zone’s exterior, then a zone violation is occurring.

### Notification Timing

Notification occurs during the iteration of the service in which violation is detected by the latest state reports of the vehicles. It is **not** reported during an iteration when the vehicle will enter the violation before the next iteration. It **is** reported in the iteration in which vehicle position report satisfies violation conditions.

### Case 2A: Don’t Report Again While Still the Same Violations

If the service is configured with a repeat interval of -1, then the service will never repeat notifications for the same zone violation by the same vehicle while that violation continues to persist. If the violation terminates, and then the vehicle reviolates, then the notification will be resent. See the image below.

### Case 2B: Report Again Later When it Reoccurs

If the service is configured with a positive repeat interval, then the service will repeat notification of each continuing zone violation at the configured time interval. See the example graphic below.

### Case 2C: Boundary-Skimming Behavior

If a vehicle’s flight path is erratic or a zone has a highly complex geometry, then it is possible that a vehicle skimming along a boundary may frequently enter and leave violation with a zone boundary. Such rapid oscillation of violation state is allowed by this algorithm and will result in a rapid succession of violation events. See the figure below.

### Case 2D: Position Noise Behavior

OpenUxAS is currently applied to the AMASE vehicle simulator. If OpenUxAS were to be used with real-world reporting and there is noise in vehicle position data over time, then the behavior of the violation alerting behavior is as follows:

**TBD.**

## Case 3: Report Potential Zone Violations Based on Current Linear Trajectory

### The service will alert subscribers when a violation is predicted to occur in a future time window based on the current position and linear velocity of each aircraft. This basic algorithm is meant to match the linear projection models used for DIADLUS [cite].

### Projected Violation Model

Violations are detected from the last received vehicle state report, Ts, to a future time limit Tw, in the parametric range for time, t, as Ts<=t<=Ts+Tw.

The model of projected violation is based on the linear projected trajectory of the vehicle from its state-reported position to the final position in the time window along vector Pv, which has length Tw\*Vs where Sv is the current speed of the vehicle, and direct Vn which is the normalized vector of the vehicle’s velocity in the state report.

A projected violation is reported if any point of the configured ‘collision radius’ around a vehicle (described above for violation detection) would be on or inside a keep-out zone or on or outside a keep-in zone for any given projected future position of the vehicle along the vector of vehicle position, described above along the projected position vector for time Ts<Tp<=t<=Ts+Tw as defined by variable time t. If a violation is projected to occur, the time of projected first occurrence is included in the notification sent to all subscribers.

Below are illustrated two example cases. In the first, no future violation is reported as none is projected in the time window, in the second, a violation is reported as projected to occur.

### Past Projected Violations Are Not Reported

Note that this vector does not include positions along the vector for t<Tp. We will not report projected violations that only occur in the past relative to the current iteration time of the service. Below is a figure showing two example projections of violation, but one is in the past from Ts<=t<Tp. Because it occurred before this iteration only, it is not reported.

### No Re-Reports of Projected Violations

If a vehicle continues to project violation in successive iterations, those are not reported as additional notifications. This is shown in the fiture below.

### Projected Re-violations Are Reported

However, if at least one iteration occurs where the violation would no longer occur, and then a future iteration occurs where it would again occur, this is reported to subscribers a second time. See the figures below for examples.

### Stale Reporting Errors

If the present time of the iteration of the service Tp is past the timewindow for projection of the state vector, the service will report an error message to all subscribers. It will do this for any vehicle with a vehicle position report that is thus so stale. It will not repeat the message unless a new status of the vehicle is received. If that report is still stale beyond the time window, a new error report will be produced.

Thus, if continuous message lag exceeds Tw error messages will be continuously delivered to all subscribers.

### Potentially Report While Following a Route

When a vehicle has filed a path route and is currently executing it, it will only have projected zone violation warnings sent out if it deviates too far from its planned route.

The figure below shows the expected behavior of this feature. A vehicle is approaching a keep-out zone, but is following its filed path route closely. So long as it stays on the route, the projected future keep-out zone violation will not be sent to subscribers, as illustrated.

However, should the vehicle deviate too far from the line segment between its present waypoints, by tolerance distance to the line segment Ed, then notifications will be sent to the vehicle, as illustrated in the figure below. In that figure, a vehicle deviating within Ed does not report the future violation. To the right in the figure, if a projected violation occurs while deviating from route by distance Ed, or if distance Ed is exceeded while a projected violation is detected, subscribers will be notified. The notification will include a flag indicating that a route is filed but that the vehicle is exceeding the allowed distance tolerance.

### Behaviors of Service with Noise in Vehicle Entity Report Measures

**TBD**

1. Complex Behaviors

It is the intent for the service to always observe the above basic behaviors regardless of live state of vehicles in OpenUxAS. Therefore, no combinatorial behaviors are expected. Examples of this should behavior should be demonstrated in this section.

TBD: Good examples of multiple vehicles interacting.

1. Satisfactory Parameters

The system is parameterized by

* Vehicle collision radii
* Path deviation tolerance Ed
* Lookahead time window Tw

Here we demonstrate that these values, configurable in launch of the service, can be set to values useful for the applications of OpenUxAS.

**TBD**