ASSISTANCE

Adapted situation awareneSS tools and tailored training curricula for increaSing capabiliTie and enhANcing the proteCtion of first respondErs

European Commission

Project co-funded by the European Union within the Horizon 2020 Programme

Project Ref. N°	ASSISTANCE H2020 - 832576
Start Date / Duration	May 1, 2019 (36 months)
Dissemination Level[1]	PU (Public)
Author / Organisation	Eduardo Villamor/ETRA

Deliverable D5.2

ASSISTANCE SA advanced modules development

31/01/2021

[1] PU: Public; PP: Restricted to other programme participants (including the EC services); RE: Restricted to a group specified by the Consortium (including the EC services); CO: Confidential, only for members of the Consortium (including the EC services).

ASSISTANCE

Nowadays different first responder (FR) organizations cooperate together to face large and complex disasters that in some cases can be amplified due to new threats such as climate change in case of natural disasters (e.g. larger and more frequent floods and wild fires, etc) or the increase of radicalization in case of man-made disasters (e.g. arsonists that burn European forests, terrorist attacks coordinated across multiple European cities).

The impact of large disasters like these could have disastrous consequences for the European Member States and affect social well-being on a global level. Each type of FR organization (e.g. medical emergency services, fire and rescue services, law enforcement teams, civil protection professionals, etc.) that mitigate these kinds of events are exposed to unexpected dangers and new threats that can severely affect their personal safety.

ASSISTANCE proposes a holistic solution that will adapt a well-tested situation awareness (SA) application as the core of a wider SA platform. The new ASSISTANCE platform is capable of offering different configuration modes for providing the tailored information needed by each FR organization while they work together to mitigate the disaster (e.g. real time video and resources location for firefighters, evacuation route status for emergency health services and so on).

With this solution ASSISTANCE will enhance the SA of the responding organisations during their mitigation activities through the integration of new paradigms, tools and technologies (e.g. drones/robots equipped with a range of sensors, robust communications capabilities, etc.) with the main objective of increasing both their protection and their efficiency.

ASSISTANCE will also improve the skills and capabilities of the FRs through the establishment of a European advanced training network that will provide tailored training based on new learning approaches (e.g. virtual, mixed and/or augmented reality) adapted to each type of FR organizational need and the possibility of sharing virtual training environments, exchanging experiences and actuation procedures.

ASSISTANCE is funded by the Horizon 2020 Programme of the European Commission, in the topic of Critical Infrastructure Protection, grant agreement 832576.

Executive Summary

This deliverable provides a comprehensive description of the development and implementation process of four innovative Situation Awareness capabilities, materialised into four software modules: Augmented Video Fusion Module, Chemical Hazard Module, Damaged Assets Location and Routing, and Portable SA platform.

The internal processes of each module are widely described in the body of the document, together with a description on the carried-out effort on implementing each of the FRs user requirements and providing a user-friendly graphical interface. The document follows the structure below:

Section 2.1 provides the background, based on the ASSISTANCE system architecture presented in previous deliverables, and presents the context where each module will work.

Section 2.2 shows the implementation of the Augmented Video Fusion module, which allows ASSISTANCE SA platform to overlap real-time video flows from UAV on the HMI.

Section 2.3 details the Chemical Hazard module innovate features, with an emphasis on the calculation and display of uncertainties related to both position and size.

Section 2.4 explains the Damaged Assets Location and Routing module, allowing to identify damaged assets and provide re-routing mechanisms for optimal evacuation planning.

Section 2.5 provides a description on how the Situation awareness platform (SAP) has been adapted, with a dedicated user interface, to run in mobile devices and enhance accessibility.

Finally, section 3 closes the document with the main conclusions.

List of Authors

Organisation	Authors
ETRA	Eva Muñoz, Santiago Martínez, Virgilio Gómez, Eduardo Villamor
UPVLC	Federico Carvajal, Manuel Esteve, Israel Pérez
TNO	Reinier Sterkenburg, Tina Mioch, Erik Vullings, Tatjana Beuker, Marit Leijnse, Ward Venrooij
UC	Javier González, Daniel Alvear, Arturo Cuesta, Gemma Ortiz, Adriana Balboa, Orlando Abreu

Change control datasheet

Version	Changes	Chapters	Pages	Date
0.1	First draft	All	17	16/11/2020
0.2	Revised ToC	All	21	26/11/2020
0.3	Added UPV's contribution on Portable SA Platform. General updates and section 1 added	All	36	17/12/2020
0.4	Added TNO's contribution on the Chemical Hazard Module	All	53	07/12/2020
0.5	Updated ETRA's contribution on the Damaged Assets Location and Routing Module	All	54	15/01/2021
0.6	Added UPV's contribution on Augmented Video Fusion Module and updates on Portable SA Platform	All	83	18/01/2021
0.7	First version completed	All	91	19/01/2021
0.7.1	Minor updates before review	2.3	91	21/01/2021
0.8	Updates after internal review	All	91	27/01/2021
1.0	Final version completed	All	92	29/01/2021

Content

List of Figures

List of Tables

Acronyms

AB	Advisory Board
ASSISTANCE	Adapted situation awareneSS tools and taIlored training curricula for increaSing capabiliTie and enhANcing the proteCtion of first respondErs
CBRN	Chemical, Biological, Radiological and Nuclear
CHM	Chemical Hazard Module
D#.#	Deliverable number #.# (D1.1 deliverable 1 of work package 1)
DALR	Damaged Assets Location and Routing
DDP	Distributed Data Protocol
DoA	Description of Action of the project
EC	European Commission
EM	Evacuation Management
EU	European Union
FR	First Responder
GA	Grant Agreement
GIS	Geographic Information System
GUI	Graphical User Interface
H2020	Horizon 2020 Programme for Research and Innovation
HMI	Human Machine Interface
HQ	High Quality
IPR	Intellectual Property Rights
M#	#th month of the project (M1=May 2017)
MMM	Mission Management Module
NATS	Neural Autonomic Transport System
PC	Project Coordinator
PDA	Personal Digital Assistant
PIC	Project Implementation Committee
PSB	Project Security Board
PSC	Project Steering Committee
REST	REpresentational State Transfer
RS	Routing Service
SA	Situation Awareness
SAP	Situation Awareness Platform
SAS	Sensor Abstraction Service
SVP	Smart Video Player
TL	Task Leader
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle

UxV	Unmanned vehicle of any type
VM	Video Manager
VRS	Video Recording System
VS	Video Server
WP	Work Package
WPL	Work Package Leader

1. Introduction

1.1. Purpose

The purpose of this document is to report the activities performed under Task 5.2, comprising the development of the four innovative SA modules in the four respective subtasks (5.2.1, 5.2.2, 5.2.3 and 5.2.4). The document focuses on the functional scope of these SA advanced modules and outlines the resulting architecture that has been implemented.

The main objective of the deliverable is to provide a technical description of each module and highlight how each of the FR requirements have been fulfilled in the development process, preparing the ground for the integration of the modules in subsequent tasks.

1.2. Scope

Four cutting-edge SA software components have been developed in the context of this deliverable, namely Augmented Video Fusion, CBRN Hazard Evolution, Damaged Assets Location and Routing, and Portable SA platform modules. Hereafter, the document consolidates the efforts and close collaboration of each technical partner - ETRA, UPVLC, TNO, PIAP and UC, to meet FRs real needs with innovative technical solutions, allowing for an expanded situation awareness.

The document includes a detailed description of each of the modules and the synergies and communications among them.

1.3. Relationship with other work packages

The architecture of the innovative modules follows the overall instantiation of the high-level architecture provided as a result of T2.4 and described in D2.4: System and Network Architecture Design. The technical requirements that were elicited in T2.2 are considered when developing the different components and therefore a discussion on their fulfilment will be also provided, being a first step to further analyse and test the functionalities of the ASSISTANCE system as a whole in T7.2: Integrated System Test Bed.

To sum up, this deliverable draw information from the following tasks:

- Task 2.2 User requirements gathering analysis and tracking
- Task 2.4 System and Network Architecture Design
- Task 3.1 Sensor Abstraction Service Adapted Interfaces Definition

- Task 3.2 Sensor Abstraction Service Adapted Interfaces Implementation
- Task 5.1 ASSISTANCE SA platform adaptation

The outputs from this T5.2, namely the four SA advanced modules, will contribute the development of the following tasks:

- Task 5.4 Advanced Modules, SAS & Communications Infrastructure Integration in ASSISTANCE SA Platform
- Task 7.2 Integrated system test bed

2. SA advanced modules

2.1. Overall system

The ASSISTANCE system is a novel Situation Awareness platform that joins multiple innovative services thanks to its advanced software modules/components. Among the various modules that are being developed in the project, this deliverable will focus on four strong contributions to the SA paradigm:

1. Augmented Video Fusion - UAV video real-time overlapping on the GIS, showing the status of the area under surveillance.
2. CBRN Hazard Evolution, Chemical Hazard Module (CHM) - Toxic footprints forecasting based on chemical sensors data, images and meteorological data.
3. Damaged Assets Location and Routing (DALR) - Video streaming from drones' cameras, allowing to locate damaged assets and provide re-routing mechanisms for optimal evacuation planning.
4. Portable SA platform - Portable version of the SA platform that can run in mobile devices, such as tablets and rugged PDAs.

The architecture of each module presented below has been derived taking the overall ASSISTANCE architecture as reference - provided in D2.4. The Sensor Abstraction Service (SAS), a system provided by ETRA, is the middleware that enables interoperability and synchronization between all data exchanges among ASSISTANCE systems and modules. It works asynchronously through publish and subscription methods. Each module collects data from different sources - sensors, cameras, UAVs, etc; in order to execute their internal processes and provide actionable insights to the SA platform through the SAS.

Data flows for each module are described in the following sections of this document. The integrated architecture, where a common infrastructure will be defined for the four modules, will be presented in D5.4: Final SA Platform Integration.

2.2. Augmented Video Fusion

2.2.1. Overview

The augmented video fusion is an innovative capability completely developed during ASSISTANCE project that allows the SA platform to overlap real-time video flows from an Unmanned Aerial Vehicle (UAV) on the HMI's Geographic Information System (GIS), projecting the field of view on a 3D terrain. This information will increase FRs' Situational Awareness during their mitigation activities.

This SA advanced capability description is stated in the following sections, apart from this overview section, which describes overall the Augmented Video Fusion:

In 2.2.2 section a description of all components will be performed as a matter of introduction for the following ones. In the next section, the main sensors involved in the video draping process have been also stated.

In sections 2.2.4, 2.2.5 and 2.2.6 the main model of the Augmented Video Fusion has been described. This overall model has been split in different sections in order to make the text more readable.

Section 2.2.4 will introduce the mathematical model used to extract the needed parameters from the raw data, telemetry and frames.

Section 2.2.5. will describe how for each frame the data is consumed and processed. Also, the creation of the virtual cameras (the object that will be the basis for the projection process) from each frame will be explained.

In Section 2.2.6. the overall model computation will be provided to represent the video projection in 3D and in real-time over a Geographic Information System.

In section 2.2.7. Augmented Video Fusion Module inside the SAP will be displayed.

Finally, in sections 2.2.8 and 2.2.9, the operational procedures and the requirements identified for the Augmented Video Fusion in D2.2 are stated.

2.2.2. Description of components

Before describing the main components used for implementation of the Augmented Video Fusion Module on the SAP, an outline is presented in the following diagram:

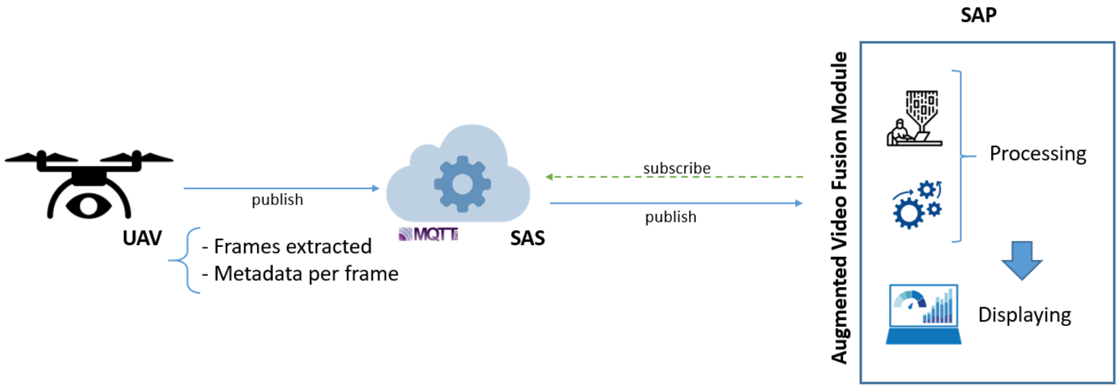


Figure 1 - Augmented Video Fusion components.

The rest of this section will be describing shortly, as a matter of introduction, key components. A deeper review of each component will be done on following sections.

UAV

UAV will transmit live-video to the SAS during its mission in those zones indicated by the Mission Management Module (MMM). This module provides support in definining and communicating mission planning activities to the asset's operators deployed on the incident. A more detailed explanation about this module will be presented in D4.6.

Therefore, during an emergency scenario, the UAV will capture real time video flow as well as raw data from several sensors mounted on-board. To capture the current status of the area under surveillance and the UAV location, the key elements are camera and the GPS.

For ASSISTANCE demonstrations and drills, an UAV from the Matrice DJI family will be used. This is the UAV model used to illustrate the images in section 2.2, see Figure 2.

Figure 2 - Matrice DJI UAV.

UAV position is determined using a GPS receiver and the corresponding subsystem at the UAV platform provides the altitude. Data is collected for each video frame and will be received through the telemetry[2] message. In Figure 3, orthogonal Yaw, Pitch and Roll axes have been represented for both UAV and camera.

[2] See 2.2.5. Telemetry for further information about the telemetry message.

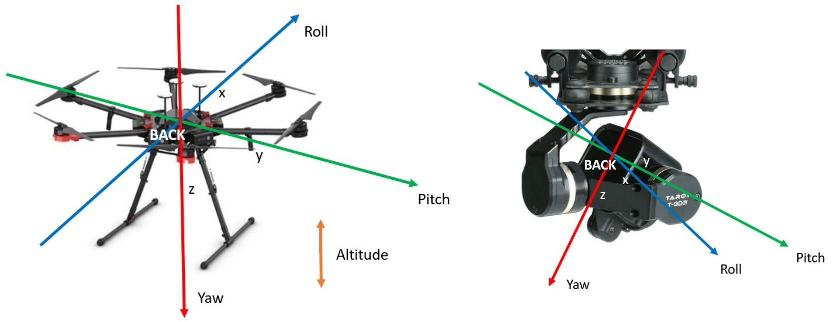


Figure 3 - Yaw, Pitch and Roll axes represented over both systems: an UAV and a camera mounted on the UAV.

Details of the nomenclature and the angles are summarized as follows. For representation simplification purposes, a naked and compact camera is used as a free object in the space. Same motions can be extrapolated to the UAV.

Yaw: Is the motion along the vertical axis. It is expressed as in decimal degrees and varies from 0 to 360, where 90 is due east, 180 is due south, 270 is due west, and 360 is due north. It is similar to azimuth, still relative to movement direction.

Figure 4 - Different yaw values

Pitch: Is the motion along the lateral axis, perpendicular to roll and yaw axis. It is expressed as a degree number and varies from -90 when it is pointing the nadir (to the ground) to 90 degrees when it is pointing the zenith.

Figure 5 - Different pitch values

Roll: Is the motion along the longitudinal axis, the direction of the drone. It is expressed as a degree number and it can vary from -90 to +90, where 0 is the horizon, 90 is full roll right and -90 full roll left.

Figure 6 - Different roll values

Sensor Abstraction Service (SAS)

The communication between the UAV and the SAS is handled by Message Queuing Telemetry Transport (MQTT) interface. It is a basic messaging system based on the publication and subscription of topics. SAS database is structured in layers that contains the topic as different collections.

UAV drones will send encoded images and its telemetry synchronized for each frame by MQTT. SAS receives this information from the UAVs connected to the system and publishes it for being processed by the Augmented Video Fusion modules on near real-time.

Situational Awareness Platform (SAP)

The Situational Awareness Platform (SAP) is a component of the ASSISTANCE platform that provides Situational Awareness to their users (both in the HQ and in the mobile versions) by providing, in real time, information about sensors, units location, messaging, video flows, etc. It will obtain the data from the frames and its synchronized telemetry through the SAS subscription and will process the data following the next phases:

Data Processing

This processing takes place in order to match the real-world data captured by the UAV to the visualization module needs. The parameters needed for creating each new image projector will be the following: Eye Point, Reference Vector, Up Vector, Field of View (FOV) and Aspect Ratio Range, which is the distance between Far and Near distance that will be explained in detail in 2.2.5Processing.

Displays visualization

At this point the processed data has been computed and will be shown in the corresponding interface of the HMI. Providing a boost on Commanders' Situational Awareness.

As stated, the following sections describe more specifically each component.

2.2.3. Sensors

In this section, sensors and some characteristics and parameters that define them and their accuracy will be approached.

2.2.3.1. GPS

The Global Positioning System (GPS) is a part of the Global Navigation Satellite Systems (GNSS). GNSS is based on measure pseudo-distances between at least 4 satellites to get an accurate position -including height. Geometrically speaking, if the satellites are close together in the sky, can be stated that has a poor geometry and the Dilution of Precision (DOP[3][4]) value is high.

DOP components are:

- GDOP = Geometric dilution of precision
- PDOP = Position Dilution of Precision (3-D), sometimes the Spherical DOP.
- HDOP = Horizontal Dilution of Precision (Latitude, Longitude).
- VDOP = Vertical Dilution of Precision (Height).
- TDOP = Time Dilution of Precision (Time).

Thus, a low DOP value it is translated as a better geometry positional and the precision due to this increase.

Other scenario that can cause low accuracy is when even with a good satellite position and a good GDOP other elements interfere in their visibility, as might happen in urban scenarios or in narrow valleys. Also, GPS signal interferences might diminish the accuracy and may be caused by transmission lines or telecommunication towers.

The GPS uses the World Geodetic System (WGS84) as its reference coordinate system. It is made up of a reference ellipsoid, a standard coordinate system, altitude data, and a geoid. The error of WGS84 is estimated to be less than 2 cm into its origin, the Earth's centre mass.

[3] The term DOP is a value of probability for the geometrical effect on GPS accuracy.
[4] http://www2.unb.ca/gge/Resources/gpsworld.may99.pdf

Ground-derived coordinates (latitude and longitude) are determined with measurements and calculations on mathematical reference models. These models represent the shape of the Earth in a particular geographic region known as geodesic datums.

Therefore, UAV position is determined using a GPS receiver, using the European datum and with certain error margin. This position is sent to the SAS within the telemetry message as latitude, longitude and height in its correspondent datum. As mentioned, the accuracy of the UAV position depends on the signal received and its capability to acquire signal from the satellite platform. A well-designed GPS can achieve 3 meters or less horizontal accuracy, and 5 meters or less vertical accuracy. Augmented GNSS systems can provide submeter accuracy. For our purposes a normal GPS accuracy and a low DOP would be enough.

2.2.3.2. Video Camera

During the mission, the video stream is captured by a video camera mounted on a UAV gimbal. A gimbal is a gyro-stabilizer that allows to stabilise camera payloads on a UAV. It integrates an Inertial Measurement Unit (IMU) that responds to motion activating separate motors to keep the camera steady on each axis. The gimbal is attached on the UAV payload.

The geometry of the video frame varies, depending on the optic distortion, but also because the possible camera movement dictated by the camera operator. The gimbal movements are captured to be included in the model collecting the rotation in all axes: Yaw, Pitch and Roll. This movements were described on Figure 3 - Yaw, Pitch and Roll axes represented over both systems: an UAV and a camera mounted on the UAV.

Frames can be classified depending on the angle of the optical axis or tilt of the sensor with regards to the ground in three categories: nadir, high oblique and low oblique[5], as shown in Figure 7.

[5] https://www.e-education.psu.edu/geog480/node/444

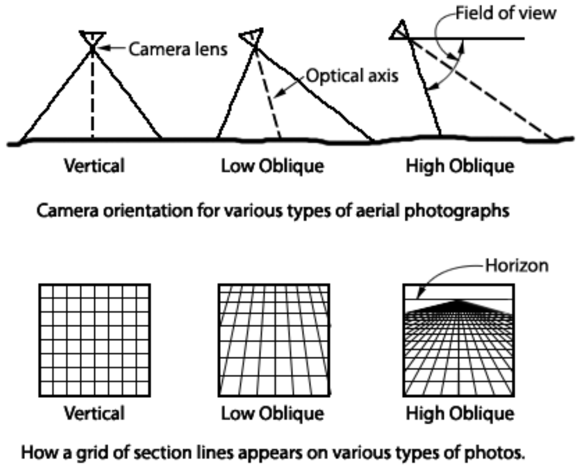


Figure 7 - Camera orientation and scale effects for vertical and oblique aerial photographs.

These precise contrasts are computed utilizing the position and altitude of the sensor and its position in the real world.

Real examples are show in Figure 8. Changes can be observed between pictures since the UAV is turning. The camera is pretty much stabilized.

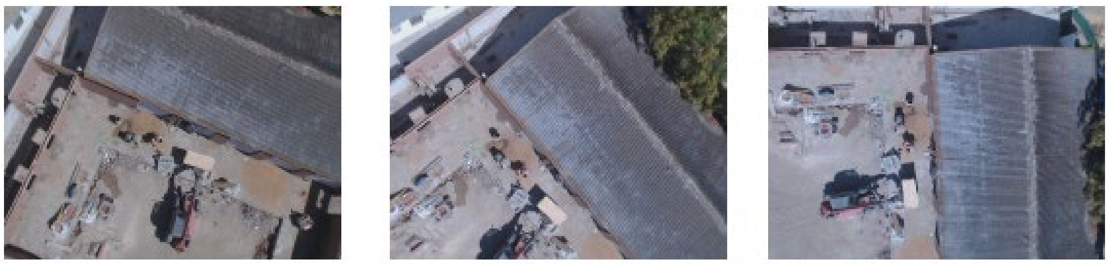


Figure 8 - Frames extracted from the video flow

2.2.4. Coordinate Transformation Pipeline

Several transformations are needed to transform the frame captured by the camera and eventually display it in our Canvas, Figure 9. These transformations are involved in the acquisition of some parameters as EYE, REF and UP.

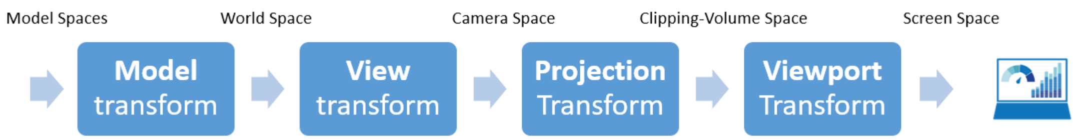


Figure 9 - Coordinates Transform Pipeline

As, it is well known, a N-dimensional vector space is defined by N base vectors, in our case, three orthogonal vectors. These can be scaled and added together to obtain all the other vectors in that given space.

Our frame captured by the camera belongs to one specific vector space; this is called Model Space and all the points are relative to the origin of the Model Space.

Then, to relate different model spaces spatially, transformations are needed to put them into a common space, the one used on the GIS, also called World Space. See Figure 10.

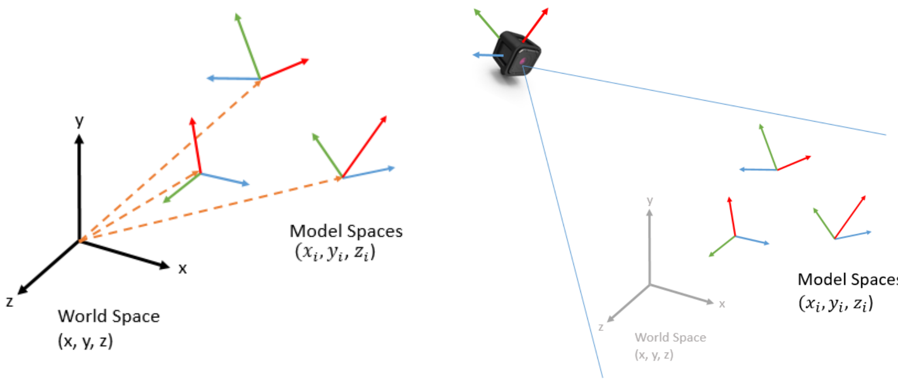


Figure 10 - On the Left spatial relation between different model spaces and the world space; On the right the model spaces and the camera in world space.

Every frame has its own position and orientation in the world, depending on where the virtual camera is, being the position defined by replicating where it was facing to, and the UAV location in that moment. Each frame has a different Model Transform. This step will be performed using the projection matrix afterwards.

Model Transform

In Figure 11, the world space is represented by an orthonormal basis, meaning that the unit vectors are orthogonal to each other, with an origin at $O = (0,0,0)$:

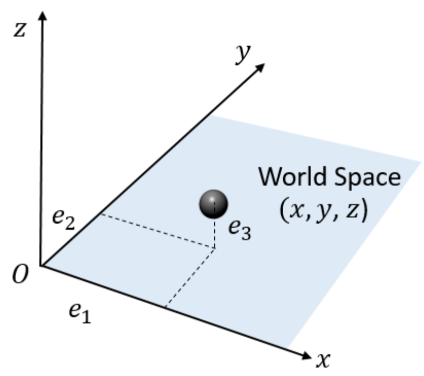


Figure 11 - Multiple Spaces Represented

$$\{e_1 = (1,0,0), e_2 = (0,1,0), e_3 = (0,0,1)\}$$

$$(x, y, z) = xe_1 + ye_2 + ze_3$$

The View Space known as Camera Space has also an orthonormal basis (x_c, y_c, z_c), with origin at EYE=$(e_x, e_y,\ e_z)$. A more detailed definition on EYE can be found on section 2.2.5.3.

In order to express all the coordinates in the camera space, we translate the EYE to the Origin and then rotate the axes.

EYE is converted to a (1x4) matrix, known as homogeneous coordinates, see Figure 12, in order to multiply it to a (4x4) matrix. Therefore, EYE would be EYE = $(e_x, e_y,\ e_z, w = 1)$. Afterwards every component of the vector is divided by the w component itself, going back to Cartesian (1x3) vector.

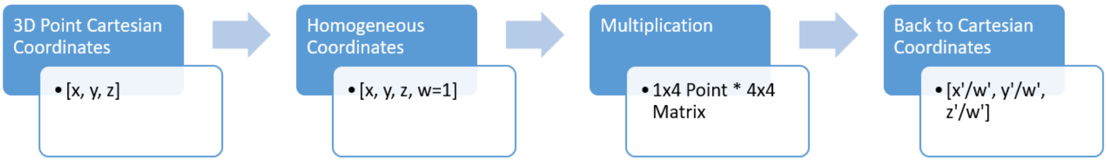


Figure 12 - Steps involved for multiplying a 3D point to a 4x4 matrix

View Transform

The camera is translated to the Origin and in order to simplify our figures, its Z axis points down, but a second transformation is needed, in terms of rotation. This space is called View Space or Camera Space, and the transformation moves all the vertices from World Space to View Space. In Figure 13 world space is represented to help visualize the transformation.

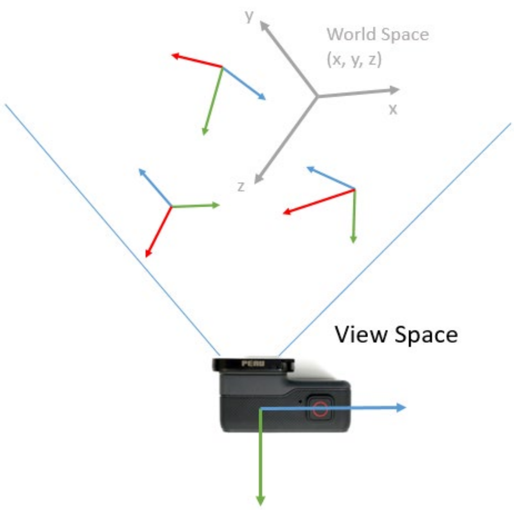


Figure 13 - Everything is transformed into View Space.

As mentioned, view transformation involves two steps:

- Translation: Moving the EYE to the Origin.

$$T(-EYE) = \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation:

$$R = \begin{bmatrix} x_x^c & x_y^c & x_z^c & 0 \\ y_x^c & y_y^c & y_z^c & 0 \\ z_x^c & z_y^c & z_z^c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where:

$$x_c = \begin{bmatrix} x_x^c \\ x_y^c \\ x_z^c \end{bmatrix}, \quad y_c = \begin{bmatrix} y_x^c \\ y_y^c \\ y_z^c \end{bmatrix}, \quad z_c = \begin{bmatrix} z_x^c \\ z_y^c \\ z_z^c \end{bmatrix}$$

C refers to Camera Space (View Space), following the convention.

Hence, the View Matrix is:

$$V = R\,T = \begin{bmatrix} x_x^c & x_y^c & x_z^c & 0 \\ y_x^c & y_y^c & y_z^c & 0 \\ z_x^c & z_y^c & z_z^c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_x^c & x_y^c & x_z^c & -e_x \cdot x_c \\ y_x^c & y_y^c & y_z^c & -e_y \cdot y_c \\ z_x^c & z_y^c & z_z^c & -e_z \cdot z_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With the camera positioned and oriented, then frustum is described.

Frustrum

The view frustum represents the region of space that is projected onto the viewing plane. It defines the field of view of the virtual camera. Same size objects nearer to the EYE appear larger than objects further to the EYE.

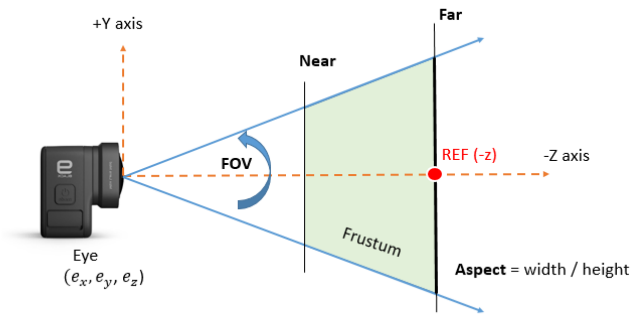


Figure 14 - Parameters that define a frustum on the View Space

As Figure 14 illustrates, four parameters define a frustum:

- FOV: is the vertical angle of the camera’s lens. Normal values range from 30 to 60 degrees.
- Aspect Ratio: is the width divided by the height of the canvas window, it determines the relative x direction field of view. It should match the actual aspect ratio of the view window.
- Near and Far distances: can be any positive numbers always being *Near* smaller than *Far*. *Near* would be as far away from the camera and *Far* as close to the camera possible. An object outside Near and Far is not visible to the camera.

Projection Transform

A projection matrix is applied to transform the view frustum into the final space, the Projection Space. This space is an axis-aligned cuboid clipping volume of 2x2x1 centred on the Near plane. The near plane has z=0, whereas the far plane has z=-1. Anything outside the cuboid range is outside the camera view area, and will not be projected. As Figure 15 shows on the right-hand side, the planes have dimensions of 2x2, with range from -1 to +1.

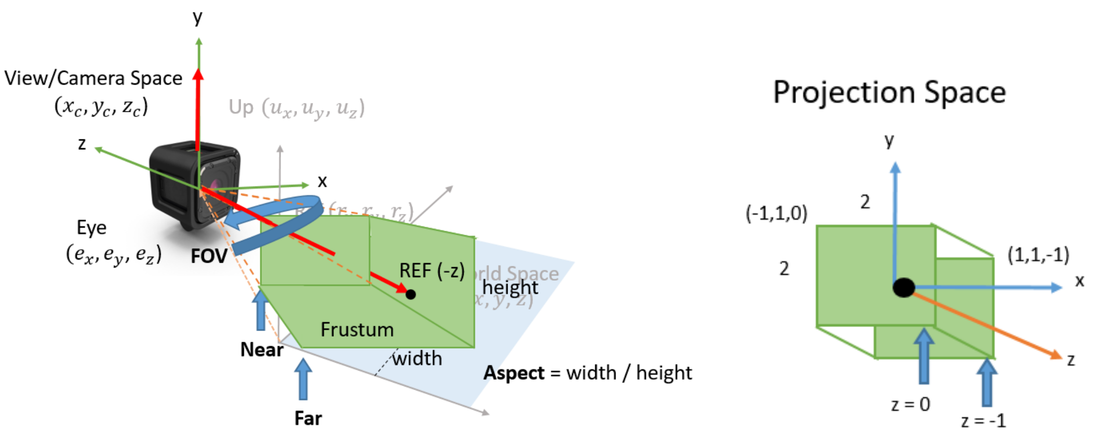


Figure 15 - On the left the View Space with Frustum parameters is represented;
On the right is represented the Clip-Volume, the Projection Space

Projection Matrix

Projection matrix is the last step to go from the View Space to the Projection Space. As described before, the frustum or area that the camera can see is defined with the Width and Height for X and Y axis, and Near and Far for the Z axis. The matrix that follows remaps the Frustum into the cuboid.

$$M_P = \begin{bmatrix} \frac{cot\ (FOV/2)}{aspect} & 0 & 0 & 0 \\ 0 & cot\ (FOV/2) & 0 & 0 \\ 0 & 0 & -\frac{z_{far}}{z_{far} - z_{near}} & -\frac{z_{near}\ x\ z_{far}}{z_{far} - z_{near}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

The last row of the matrix is no longer [0, 0, 0, 1], therefore we need to normalize the resultant homogeneous coordinates (x, y, z, w) to (x/w, y/w, z/w, 1) to obtain position in 3D space. Homogeneous coordinates can be used for translation, as well as the perspective projection.

Every frame vertex is transformed and positioned in the clipping-volume cuboid space, the relationships between vertices are not considered.

Hereafter, in the processing stage, this pipeline would be referred as create a Video Model Projector, meaning positioning both: the video frame and the camera, virtualized in the same Projection Space, the GIS.

2.2.5. Processing

In this section the telemetry message and the processing of real data are discussed. It must be reminded that this is the message that the SAP system retrieves from the SAS, where it has been injected by the UAV platform. At the end of this section an output sample is shown.

2.2.5.1. Telemetry

The telemetry message describes the position, speed and altitude from the UAV during its mission - shown in Figure 16. This object will have the following attributes: “timestamp” that indicates the time when the telemetry is taken, “speed” that indicates movement speed in three dimensions, “location” that indicates location of an unmanned vehicle. Finally, “Pitch”, “Roll” and “Yaw” attributes indicate the three axes of an unmanned vehicle values. As described before, camera has its own pitch, roll and yaw.

```
{
    "_id": "WuPJE6JZyma3M6iAL",
    "resource": "RSRC00000001",
    "context": "CTXT20200101100000",
    "mission": "MSN20200101100000",
    "remainingAutonomy": 1,
    "geometry": {
        "type": "Point",
        "coordinates": [-0.4444444, 39.4444444]
    },
    "height": 1,
    "speed": 1,
    "attitude": {
        "pitch": 1,
        "roll": 1,
        "yaw": 1
    },
    "cameras": [{
            "_id": "CAMERA_ID",
            "url": "http://192.168.0.2",
            "attitude": {
                "pitch": 1,
                "roll": 1,
                "yaw": 1,
                "offset": [1, 1, 1]
            }
        }
    ],
    "timestamp": "2020-01-01T10:00:00.000Z"
}
```


Figure 16 - Telemetry sample object JSON formatted

Altitude data is extracted from the inertial sensors (gyro-scopes and accelerometers), widely used because they represent a cost-efficient compact solution. Since their measurements may suffer from several errors, as compensations from the gimbal, we will be using it together with the GPS tracked data to avoid divergence from position and altitude estimates.

In order to process all the information, for each frame, all these parameters have to be collected and gathered using the frame itself and the telemetry. The expected format for each frame is the following:

- File and path location of the given frame
- EYE in ECEF [6] of the given frame
- REF in ECEF of the given frame
- UP of the given frame
- Near and Far of the frustum
- FOV of the sensor
- Aspect (ratio) of the given frame
- Yaw, Pitch and Roll of the camera orientation for the given frame in degrees.

[6] ECEF is described further in this section. It is a conventional terrestrial coordinate system.

Earth-Centred, Earth-Fixed coordinates (ECEF) is a Cartesian coordinate system which represents positions as X, Y and Z coordinates in meters. Its centre (0, 0, 0) is the mass centre of Earth. This is the terrestrial coordinate system used in the GIS selected for ASSISTANCE project.

GPS Geodetic coordinates (latitude ϕ, longitude λ and height h) have been converted to ECEF (X, Y, Z) at some point using the following equation:

$$X = (N(\phi) + h)\ cos\phi\ cos\lambda$$

$$Y = (N(\phi) + h)\ cos\phi\ sin\lambda$$

$$Z = \left(\frac{b^2}{a^2} N(\phi) + h\right) sin\phi$$

Where $N(\phi)$ is the prime vertical radius of curvature:

$$N(\phi) = \frac{a^2}{\sqrt{a^2 cos^2\phi + b^2 sin^2\phi}} = \frac{a}{\sqrt{1 - e^2 sin^2\phi}}$$

And:

a is the semi-major axis estimated as 6378137.0 meters.

b is the semi-minor axis estimated as 6356752.3 meters.

e^2 is square of the ellipsoid eccentricity estimated as 0.0818191.

As it was anticipated, for being able to create a Video Projected for each frame, the following parameters - previously defined, are needed. Real data will be used for describing the output from each of them.

2.2.5.2. File and path location

To process the raw data, file and path locations are needed. For clarity purposes, an example is provided below with the way a pointer to the correct file is created and a way to synchronize it with the metadata.

sevilleEmergency/frame0000.jpg

2.2.5.3. EYE

Eye Point (e_x, e_y, e_z) - This is the position of the camera in the world. It is extracted from converting the longitude, latitude and height position of the camera to ECEF.

e_x =*5059899.750798772 m*

e_y =*-521200.23439180484 m*

e_z =*3835129.069074831 m*

2.2.5.4. REF

Reference Point (r_x, r_y, r_z) - This is the point at which the camera is pointing over the Earth surface. The Reference Point is obtained using the coordinate transformations principles stated previously. Translating and rotating the Eye Point, intersecting it with the Digital Elevation Model (DEM).

r_x =*5059899.700452008 m*

r_y =*-521200.22935046325 m*

r_z =*3835129.030664666 m*

Each height is saved in a collection of Heights, and contrasted with the (DEM) in a given area, in order to be always projecting upon the surface of the earth. In case the emergency happens on an orographic terrain, REF will always be draped over the DEM surface.

2.2.5.5. UP

Up Vector (u_x, u_y, u_z) - This is a normalized vector, relative to the eye point, which defines the "up" direction in the camera's local coordinate frame. UP is roughly orthogonal to REF, as in Figure 11, but not necessary. As UP and REF define a plane, we can construct an orthogonal vector to REF in the view/camera space. It is another value derived from the View Transformation calculation, from REF to EYE.

u_x = *-0.067705331404472848*

u_y = *-0.9968172894999748*

u_z = *-0.042086570937762094*

2.2.5.6. Near and Far

Stated as positive number, "Near" is always smaller than "Far". The range has been widened to gather the minimum and high enough for any flight, even with high altitude military UAVs. "Near" value is extracted from the height collection, and it changes if a new minimum distance is obtained. Typical used values are the following:

Near = 5.9857158352807165 m

Far = 6.370928158012614E7 m

2.2.5.7. Field of View (FOV)

FOV - This is the camera's vertical field of view, in degrees. FOV values do not vary at a given camera and, therefore, is a fixed value provided specific optics.

FOV = 40.0

2.2.5.8. Aspect Ratio

Aspect Ratio - This is the ratio of the projected image's width and height. The frames received from the specific camera configuration used in these tests are 800x600 pixels.

$$\text{Aspect Ratio} = \frac{width}{height} = 1{,}33$$

2.2.5.9. Yaw, Pitch, Roll

These angles are explained with more detail on 2.2.2 Section. As a reminder:

- Yaw: affects the vertical axis.
- Pitch: affects the lateral axis, perpendicular to roll and yaw axis.
- Roll: affects the longitudinal axis, the direction of the drone.

For the processing: the yaw, pitch and roll values are gathered from the camera gimbal turns.

Yaw = 268.68427083494555°

Pitch = -89.84671070478763°

Roll = 0.3007114045668224°

With these results it can be assumed that the camera is pointing West (yaw ≈ 270º) in a quasi-Nadir way (pitch ≈ -90º) and parallel to the ground (roll ≈ 0º).

A sample with a couple of frames pre-processed data is shown in Figure 17.

31_frames/frame0000.jpg ← *From 2.2.5.3 section: File and Path location*

5059899.750798772 -521200.23439180484 3835129.069074831 ← *From 2.2.5.4 section: EYE*

5059899.700452008 -521200.22935046325 3835129.030664666 ← *From 2.2.5.5 section: REF*

-0.06770533140472848 -0.9968172894999748 -0.042086570937762094 ← *From 2.2.5.6 section: UP*

5.9857158352807165 6.370928158012614E7 ← *From 2.2.5.7 section: Near and Far*

40.0 ← *From 2.2.5.8 section: FOV*

1.33 ← *From 2.2.5.9 section: Aspect ratio*

268.69097125052144 -89.8701627996331 0.31051987638007794 ← *From 2.2.5.10 section: Yaw, Pitch and Roll*

31_frames/frame0001.jpg

5059899.772979872 -521200.2277516999 3835129.057439585

5059899.7092125155 -521200.22139961127 3835129.008792283

-0.06720737810167167 -0.9968518053379142 -0.0420671668078578

5.9857158352807165 6.370928158012614E7

40.0

1.33

268.68427083494555 -89.84671070478763 0.3007114045668224

Figure 17 Sample of frames information gathered for processing

2.2.6. Visualization

When selecting a GIS, a key aspect when choosing between GISs is the processing power. In that sense, Luciad GIS offers an above-average performance through the usage of the computer graphics processing unit (GPU) (high-end ones) using an Open Graphics Language (OpenGL) with versions greater than 2.0. This makes possible to speed up calculations to several orders of magnitude above what a CPU can achieve, enabling to process close to real-time tasks, as in the real-time augmented video fusion.

Luciad does not offer final products as such, but a set of tools for the development of geospatial applications tailored to customer needs. The core of most of its suite is based on LuciadLightSpeed technology, a powerful modular component-based API written in Java. This is the API that has been used for the Augmented Video Fusion Module.

The Augmented Video Fusion interface will be display a GIS system on to the SAP with the following inputs from the surveillance area:

- Orthoimage as a background image for the video projection. An orthoimage is an aerial photograph orthorectified, geometrically corrected.
- Digital Elevation Model (DEM). This will be used as a raster surface grid for draping the projected image captured by the UAV.

Based on the data processed, the following layers are added to the view:

- A latitude/longitude Grid. It is a wireframe grid that varies the increments based on the map's current scale.
- Video Panel, providing a simple UI interface for interacting with the model. It includes a play/pause and reset button, as well as a search-slider.
- UAV layer, showing an UAV icon positioned on the point of the current Virtual Camera (EYE). It helps to see flights in context.
- Projected Image layer, showing the frame from the Virtual Camera projected on the 3D terrain, matching the DEM surface.
- Frustrum layer, showing the outlines of the projection.
- Pointer layer, showing an icon on the map, corresponding to the position of the mouse pointer over the Video Panel, as shown in Figure 18.

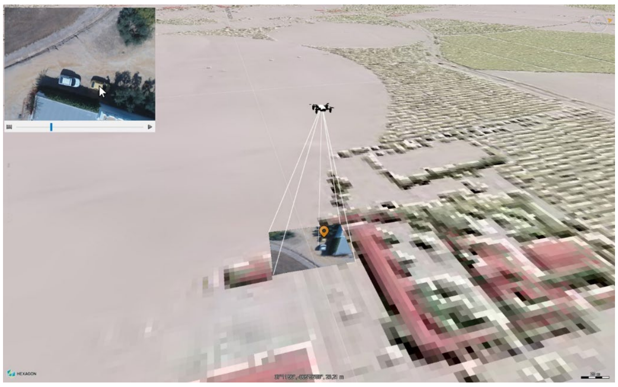


Figure 18 - Screenshot of the module showing the mouse pointer placed over the Video Panel, and its representation over the frame projected

On the interface a toolbar will appear for the given view, which provides controls for working with the given view, for example drawing shapes.

An overall picture of the Augmented Video Fusion visualization with a delimited perimeter is shown in Figure 19.

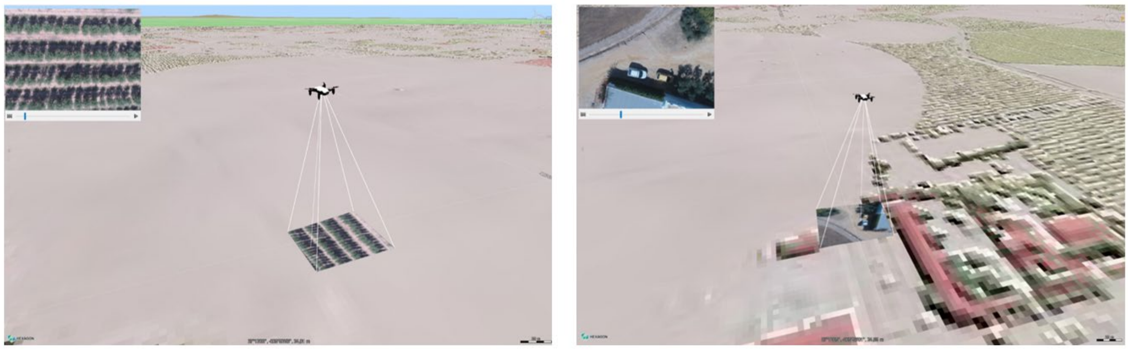


Figure 19 - Assistance Video Draping Module Screenshots Samples

2.2.7. Integration with SAP

As it was mentioned on previous sections, the Augmented Video Fusion Module is a software component that is part of the SAP. It will be in charge of processing and displaying the solution through a tailor-made interface.

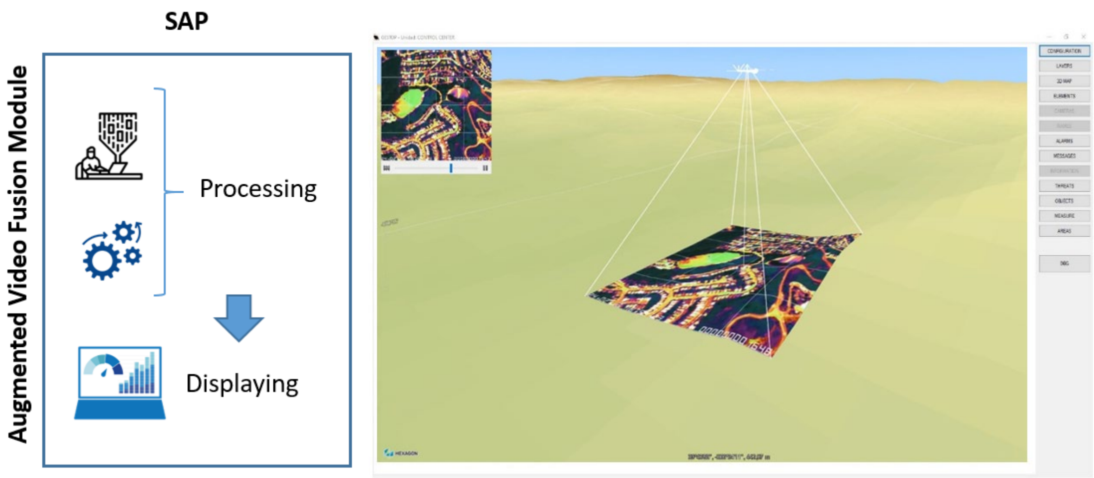


Figure 20 - SAP integration diagram and SAP Augmented Video Fusion Module interface.

2.2.8. Operational procedures

After an emergency occurs, the commander will request a mission to the MMM operator in order to evaluate the selected area. This procedure will send a UAV to the selected zone and will start to capture data from its sensors and to send that raw data. As far as this module is concern, the UAV will send the video flow and its related metadata through the SAS.

Once the Augmented Video Fusion is enabled, it will start to consume those data from the SAS and it will be enabled on the SAP. This way a real time video from UAVs displayed on the GIS will show the current status of the area under surveillance in the command room.

From the Command and Control room, an operator shall draw synthetic data extracted from the real video such as delimiting the current fire location and perimeter in case that this was requested by the commander. After this point, all the FRs will have this available at their devices for performing more secure and more accurate mitigation activities, as this information will be exchanged, seamlessly, among SAP nodes.

2.2.9. Requirements fulfilment

One requirement was identified in D2.2 to be realized with regards to the Augmented Video Fusion module.

Id. req	**Description**	**Fulfilled**	**Specific actions**
SA_020	SA_020 ASSISTANCE SA application should provide augmented video fusion capabilities for overlap real-time video flows from cameras mounted in drones on the emergency area GIS displayed in the SA application HMI.	√	This requirement encompasses the whole section 2.2

Table 1 - Augmented Video Fusion requirements

2.3. CBRN Hazard evolution

2.3.1. Overview

The Chemical Hazard Module (CHM) calculates potential hazard footprints, presented on a map, based on user input describing the type of hazard. The output is continuously adapted and adjusted based on real-time information from sensor data, meteorological data, and input by FRs.

It produces and shows the boundaries of areas on the map where the gas concentrations exceed certain levels of toxicity and can forecast the evolution of such gas cloud. It shows the most likely position and size of the cloud (current and predicted), but also the uncertainties in position and size, which are due to uncertainties in e.g. the amount of released gas, the wind direction and wind turbulence. This helps the FRs to know where to be careful to avoid the toxic hazard and where to do measurements to establish the actual gas concentrations that occur.

Furthermore, gas measurements, available from the SAS, are used by the module to recalculate the position and size of the gas cloud.

To give a general overview of the different research topics that were investigated to build this model to optimally support the SA of the FRs with regard to hazard prediction, we place these topics in the context of an example scenario. The global architecture is depicted in Figure 22 and will be detailed below. However, before diving into the technical components, we will give a brief outline of a typical workflow involving a chemical hazard situation and indicate the situation awareness support that we have made.

1. *An incident (fire/accident/leakage) occurs and is reported to the emergency hotline (e.g., 112 or 911).*
2. *FRs arrive at the incident location and estimates the seriousness of the incident.*
3. *The FR concludes that a serious chemical hazard has occurred and initiates the standard safety measures (setup safety perimeter, call for backup, etc.).*
4. *The FR requests information about the chemicals (type, quantity, etc.) e.g. from the owners, local managers, or truck driver.*
5. *The FR estimates the amount of chemicals that is being spilled (usually based on experience or available information).*

   Chemical Hazard Module:

   (a) FRs can input all available information via the user interface into the model calculation. For more details on the user interface, see Section 2.3.4.1.

   (b) In addition, for fluid spills, an estimation can be calculated by taking several images of the fluid with a (smartphone) camera. These images can be used to estimate the surface area of the fluid spill. After a few minutes, a second set of images can be taken, and run through the algorithm described above to determine the release rate. This can add valuable input for FRs and the chemical hazard module regarding the release rate of chemicals and the spilled amount and area. For a more detailed description of this research, please see Section 2.3.2.2.

6. *The FR estimates the impact of the chemical spill based on pre-defined hazard modules (typically, elliptic shapes that are offered as plastic models, for a few wind conditions). Based on the actual wind direction, a shape is drawn on the map and used as the basis for the incident response.*

   Chemical Hazard Module:

   (a) Based on the estimated chemical spill area, release rate, and properties of the fluid or gas (or other agent), a dispersion calculation is done for up to 1 hour in the future. The toxicity levels are computed too. For more details on this research, see Section 2.3.2.3.
   (b) This map layer can subsequently be shared via the SAS with other FR applications. See Section 2.3.2.1 for more details.
   (c) Sensor data from the SAS is used to reduce the inherent uncertainty in these calculations even further. See Section 2.3.2.3 sub 3 for more details.
   (d) In addition, the calculated position of the gas cloud and toxicity level, with its uncertainty information, is shown on the (digital) map. For more details on our research on effective visualization of uncertainty, see Section 2.3.4.4.

7. *The incident response area is shared within the team before they start their incident response activities.*

   Chemical hazard module:

   (a) The locations of the FR are related to the estimated hazard area, so effective warnings can be given to the FR in the field. For more details on the technical implementation, see Section 2.3.2. In Section 2.3.4.3, our research on teaming and interaction patterns is described.

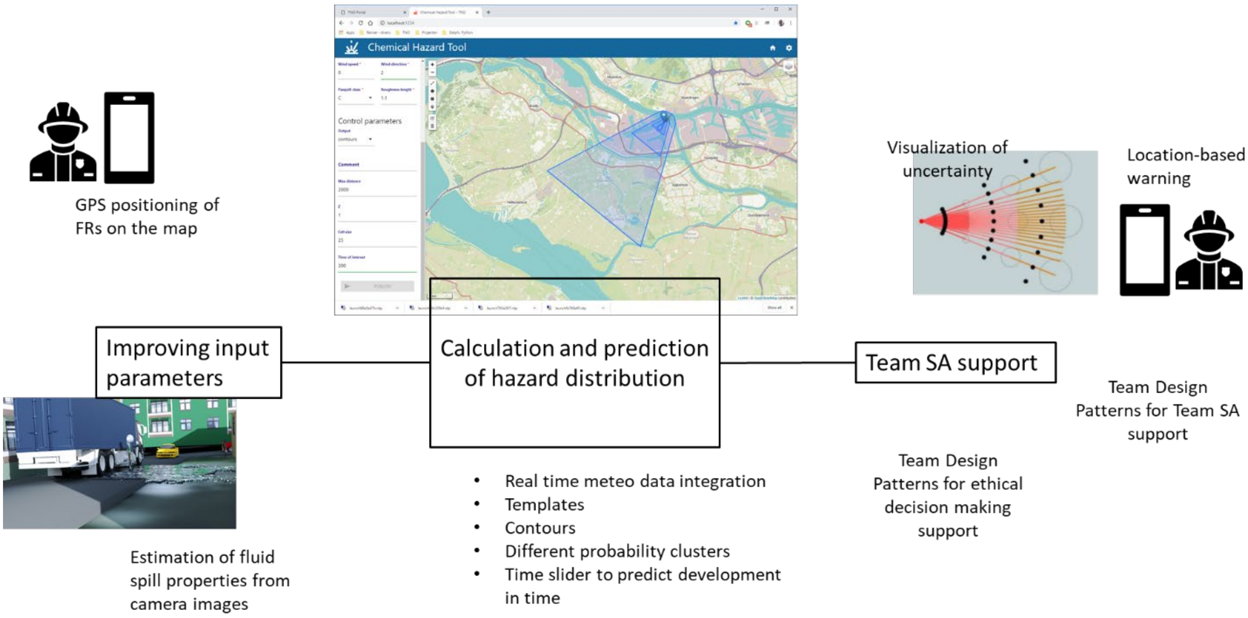


Figure 21 - Overview picture of the different aspects involved in relation to the Chemical Hazard Module.

2.3.2. Description of components

2.3.2.1. Architecture

To realize these improvements, the architecture as depicted in Figure 22 was created. From an end-user's perspective, there are two applications: a [Traccar](https://www.traccar.org) [https://www.traccar.org] background service for Android phones and iPhones, and the chemical hazard web application.

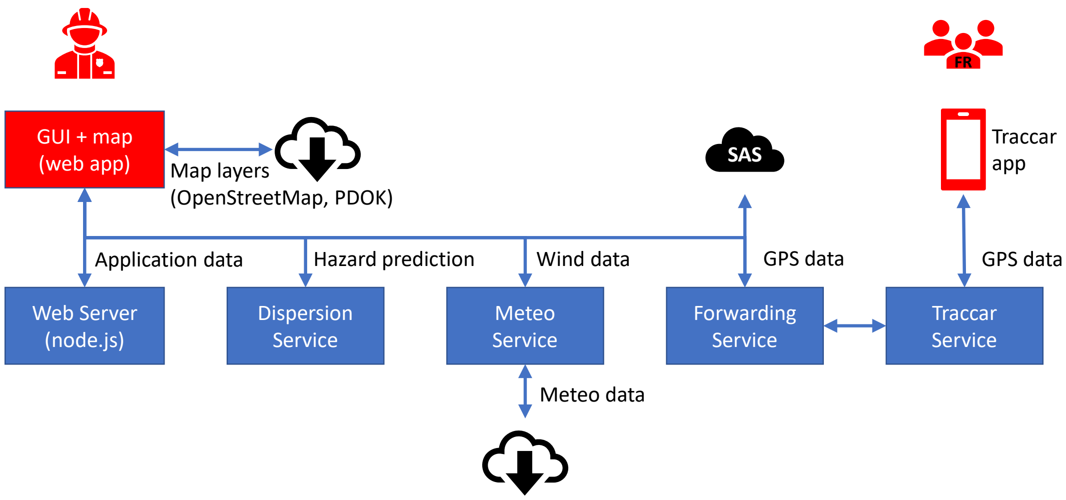


Figure 22 - Global architecture of the chemical hazard and GPS tracking modules

In the backend, several services are running:

- Traccar service: Traccar is an open source GPS Tracking Platform. It provides a Docker image at (https://hub.docker.com/r/traccar/traccar) so you can run the service locally without installing anything (besides Docker itself). They also provide a Traccar application in the Android and iPhone stores, so you can install it on your own smartphone. All that is required are the IP address of the Traccar service, running in Docker, and a unique identifier of the FR.
  Furthermore, the end user can control how frequent the GPS location is shared, and can disable it, for example when the FR is off duty. The Traccar service also provides a forwarding feature, so all GPS data it receives can be forwarded to our forwarding service. A more detailed explanation of the GPS service was provided in D4.4: Wearable sensors integrated into the system.
- Forwarding service: The forwarding service converts the Traccar GPS data format to the required SAS data format and uploads the GPS data in the correct format to the SAS. It also provides an endpoint for the GUI application, so it can show the locations of the FR on the map.
  It is developed by TNO in Python 3 and runs in Docker.
- Meteo service: The meteo service gets, upon request, the actual meteorological weather and wind data from the openly accessible meteorological data server of the German Weather Service (Deutsche Wetterdienst). It uses data from the high resolution COSMO-DE model when the area of interest lies in the area of validity of that model and from the COSMO-EU model for the rest of Europe. The data is subsequently uniformly formatted so it can be used by the Dispersion service and the GUI.
  It is developed by TNO in Python 3 and runs in Docker.
- Dispersion service: It predicts the actual chemical hazard by combining the provided input from the FR end user with the actual weather data from the meteorological service. Furthermore, it uses sensor data from the SAS to reduce the inherent uncertainty in its prediction. As this model is at the heart of the hazard prediction module, it is explained in more detail below.
  It is developed by TNO in Python 3 and runs in Docker.
- Web server: It provides the backend for the GUI application and serves a Single Page Application (SPA). Communication between the GUI and the services occurs via REST (for creating new incidents) and web sockets (for real-time updates of the positions of FR or of the hazard areas).
  It is developed by TNO in Node.js (TypeScript) and runs in Docker.

The Dispersion Service is the most crucial part of the CHM. It consists of three components that communicate with each other over a network (Internet), as illustrated in Figure 23.

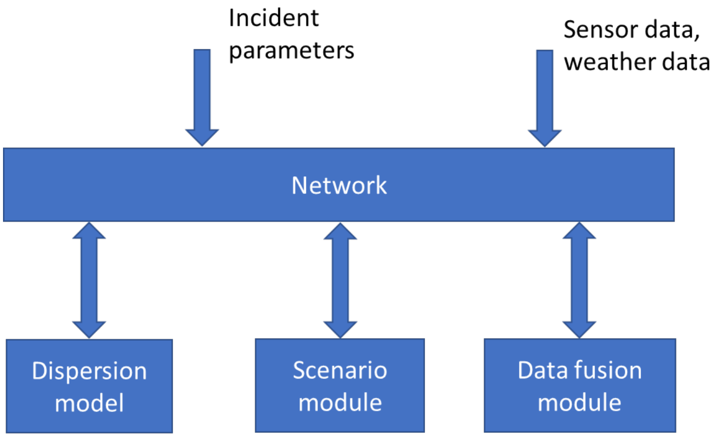


Figure 23 - Components of the Dispersion service

More information on what these components do and how they work is in Section 2.3.2.3.

2.3.2.2. Automatic estimation of input parameters

As described in Figure 23, the CHM can receive different types of input to calculate the gas dispersion, including incident parameters such as location of gas leak and sensor data such as gas measurements and weather data. Most parameters are optional for the calculation of gas dispersion; a first calculation can be done with only the location of the gas leak. However, the more parameters are given, and the more accurate these parameters are, the better the prediction of the model. The main focus of our research was the (dynamic) integration of sensor measurements by UAVs and UGVs and infield rescuers into the module to continuously improve the prediction that is shown to the FR (as is described in Section 2.3.2.3), improving the SA of the FR and their understanding of the unknown and uncertainties of the prediction.

During workshops and tests with FRs, they reported that even when being at the location and seeing chemical spills, it is very difficult to estimate some of the optional (and not directly measured) parameters, with large deviations in estimation. A particular hard situation is when a toxic chemical fluid spill is involved in the incident, for example a leaking tank. The impact and the temporal development of such a spill is very hard to accurately estimate. A more accurate (automatic) estimation of these parameters would lead to better input parameters and thus a more accurate calculation of the predicted consequences. To become valuable for the first responders in the field, this estimation should be done in a time sensitive manner, while also being minimally intrusive to the first responders' current workflow.

Based on this input from the FRs, we researched the possibility of determining the area of a fluid spill from a single image, and release rate from two images (Smet, n.d.); (Otten, n.d.) as an additional capability of the module. This prototype algorithm may then be used for example on images captured using a first responder's bodycam. This algorithm estimates the scale of the camera image and uses deep-learning techniques to determine the fluid spill region, and combines the two to determine the size of the fluid spill in square meters.

The reference points to determine the scale of the camera images were automatically found using a custom algorithm based on instance segmentation. This process of automatically estimating camera properties requires no additional setup. The lack of additional setup makes this system minimally intrusive, with little of the first responder's time being used. Using the estimated camera properties, it is possible to estimate the real-world dimensions of objects on the image.

Next, using instance segmentation, it is possible to extract the dimensions of the fluid spill on the image. Combining this with the aforementioned algorithms, it is possible to estimate the fluid spill's area based on one image. The release rate can then be estimated by taking two pictures consecutively, summing the areas of the fluid spills and then dividing by the difference in time between the two photographs.

While the initial results of each individual algorithm were promising, the results of the prototype in its entirety were not (yet) accurate enough to be actually used as input for the model. Each individual algorithm added some measure of uncertainty, and thus, error to the prototype. More research is needed to have estimates that are accurate enough to be applied in real-life situations and on real body-cam images of FRs. The challenge that remains in the current algorithm is the sampling method used to estimate camera properties. As this research was an exploratory research regarding an additional capability to support the FRs own estimation of an optional input parameter, it does not affect the overall working of the other capabilities of the module.

2.3.2.3. Chemical hazard model calculation

Model calculations are done at 3 levels:

1. Core dispersion calculation (calculation of a single plume; "Dispersion model" in Figure 23)
2. Ensemble calculations (taking uncertainty in inputs into account; "Scenario module")
3. Data fusion (using measure data to reduce uncertainty; "Data fusion module")

1. Core dispersion calculations.

The model used for the dispersion calculations could be called a semi Lagrangian neutral gas gaussian plume model. It calculates plume axis trajectories based on the (non-uniform) wind field as valid (predicted) at the times of release or as provided by local measurements. The dispersion parameters of the plume are calculated according to (Yellow Book, 2005), which are also used in other well-known models like EFFECTS.

Based on the given inputs about the source, the gas properties and the circumstances, contour lines are calculated for 3 levels of toxicity: the life threatening (LBW) level, the potentially harmful (AGW) level and the discernible level (VRW). These levels are taken from the Dutch RIVM's list of *interventiewaarden* (RIVM, 2020).

2. Ensemble.

In principle, all input variables that are used to calculate the contours have a certain inaccuracy. Many of the source parameters, like the release rate or released amount, the location and the time when the release happened or started, are estimations made by the one who reported the incident. Other parameters like the wind speed, direction and atmospheric stability are obtained from model predictions that also have an inherent inaccuracy.

For each of the input parameters, a distribution type (e.g. linear, gaussian, frequency distribution table) and distribution parameters (e.g. min, max, mean, standard deviation) are assumed. These distribution specifications for each parameter are then used to generate sets of N values that represent the distribution. From these sets of values, combinations are made to form N random input scenarios. Each of the scenarios is calculated by the dispersion model and results in a set of plume trajectories with associated contours of the hazardous concentration levels. The N sets of contours form an ensemble of equally likely realizations of where (and how large) the gas plume may be. An example of an ensemble with 3 realizations is shown in Figure 24.

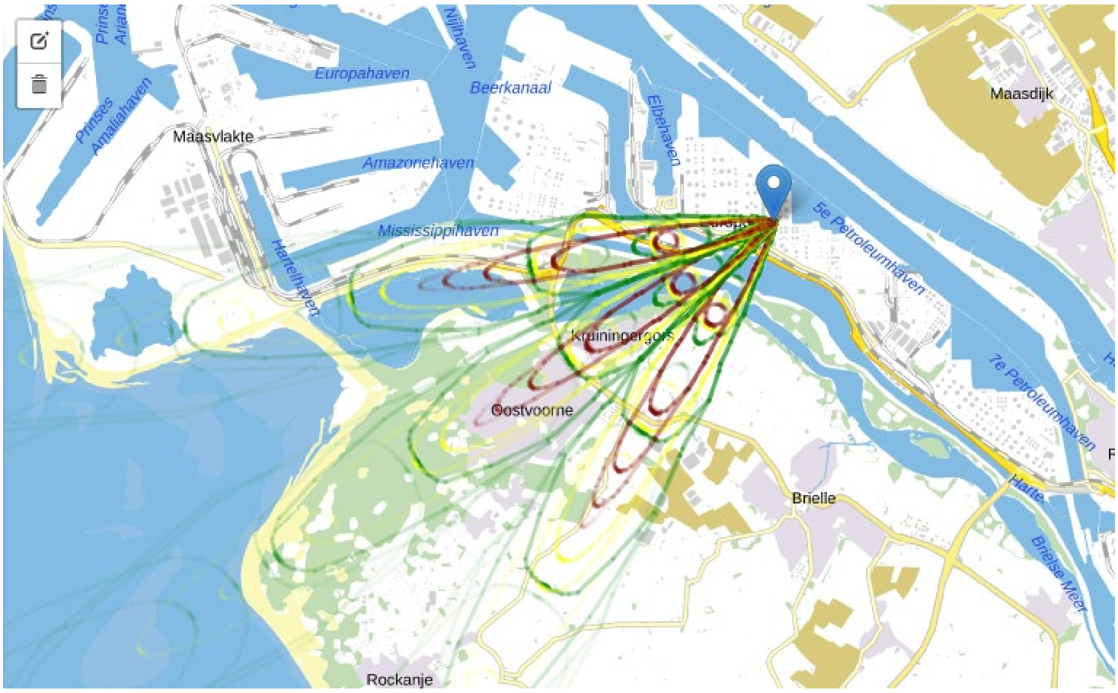


Figure 24 - An ensemble of gas plumes, which shows the operator that there is uncertainty in the direction in which the gas cloud can be moving.

Based on the calculated ensembles, it is possible to show not only the most likely (expected) position and size of the cloud, but also the areas within which the cloud will be with a certain level of confidence, e.g. 90 % certainty. The concept of ensemble based calculations and showing probabilistic hazard areas is innovative and perhaps hard to understand for operators. How best to visualize this kind of information and how to learn operators to interpret them correctly is a subject of our research. See Section 2.3.4.4.

3. Data fusion

When concentration measurements are made in the field, these can be used to modify or improve the results of model calculations. For each concentration measurement value, reported for a certain position and time, for each member of the ensemble (see previous Section) a 'likelihood of correspondence' value is be calculated, based on the difference between the calculated and the measured concentration at that location and time.

This results in the assignment of probability values to all ensemble members and if the probability values are low, they can be eliminated from the ensemble set. The remaining ensemble members (and their associated input parameter values) then get a higher probability.

2.3.3. Integration with SAP

The following ways of communication between the CHM and the SAP can be described:

- CHM connects via NATS with the SAS to receive the message to start calculating.
- CHM connects via REST with the SAS to
  - store the calculated plume (so the SAP can receive it) and
  - receive sensor information and information about the positions and status of own assets like personnel and vehicles.
- The SAP receives the plume information from the SAS via NATS

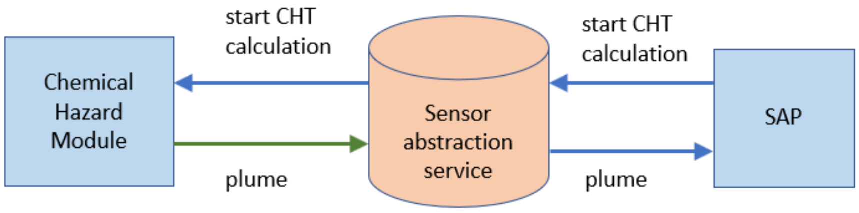


Figure 25 - Integration of CHM with the SAP module.

In Figure 25, the blue arrows are NATS messages; the green arrow indicates that the plume is sent through REST to the NAS, is stored there, and the SAP gets the data through NATS. Communication of sensor information and position of assets is not shown in this picture because they are not exchanged between the two modules but originate from other modules.

2.3.4. Operational Procedures

In this section, the interaction between the FRs and the CHM is described. First, we describe the display, followed by the overview of the iterative design cycle and a usability test with end users. Subsequently, our ongoing research on team design patterns is described, further specifying design choices regarding interaction and cooperation aspects between FRs and the system, and research on the visualization of uncertainty.

2.3.4.1. Display

This Section describes the User Interface and how the operator interacts with it. It was designed to help the operator carry out his or her tasks accurately and efficiently. In the (iterative) process of designing it, we have involved the end users and paid attention to (and are doing research on) the aspects of team design patterns and visualization of uncertainty. Those aspects are described in more detail in Sections (2.3.4.2, 2.3.4.3 and 2.3.4.4).

The graphical user interface (GUI) consists of two parts: a map and an input panel. By clicking on the map, the chemical source location can be selected. Next, in the input panel, all other relevant settings to determine the hazard cloud can be specified, such as the quantity and source type, to name but a few. In addition, the user can choose to either use an online weather service to set the wind conditions or enter them manually. The latter is especially useful for estimating the impact during training.

In the very first phase of an incident, detailed information on the source of a gas leak is not yet known. In order to immediately get a quick impression of where a gas cloud can go and how fast, the operator can enter the estimated location and calculate the 'template', which shows, with a wide error margin, the area that is at risk of being affected in 5 minutes, 15 minutes and 1 hour. The wind direction and speed are taken from the on line weather service model. This is illustrated in Figure 26.

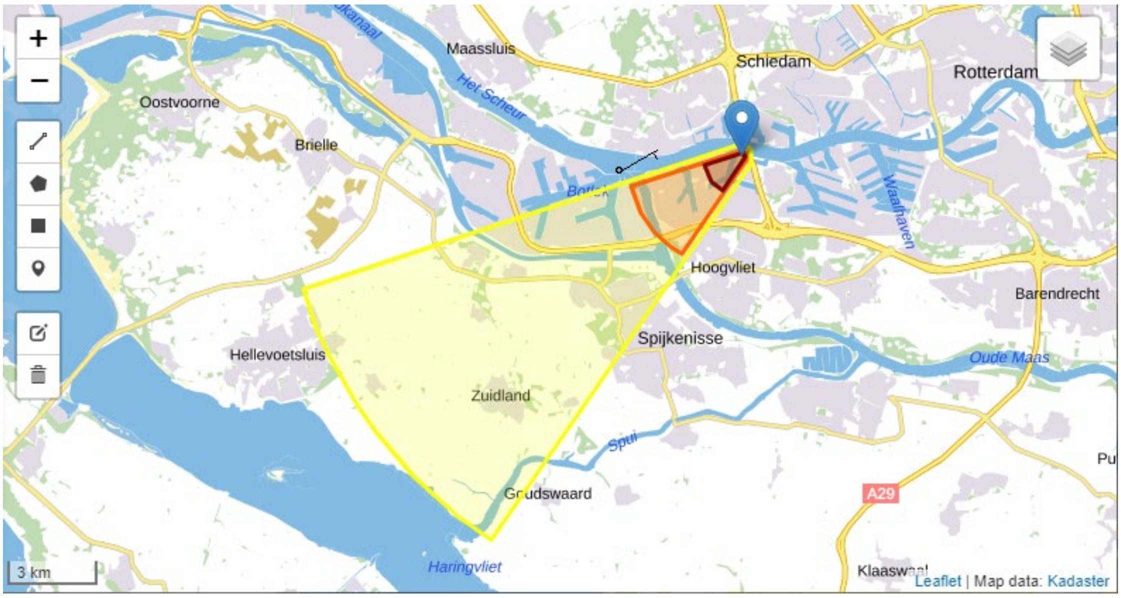


Figure 26 - Template on map that shows areas at risk

After entering more information about the accident, a dispersion calculation can be started and a more accurate picture showing the length and width of the gas plume is shown. Important dispersion parameters like atmospheric stability and surface roughness can be modified and have significant influence on the calculation results. Figure 27 illustrates this.

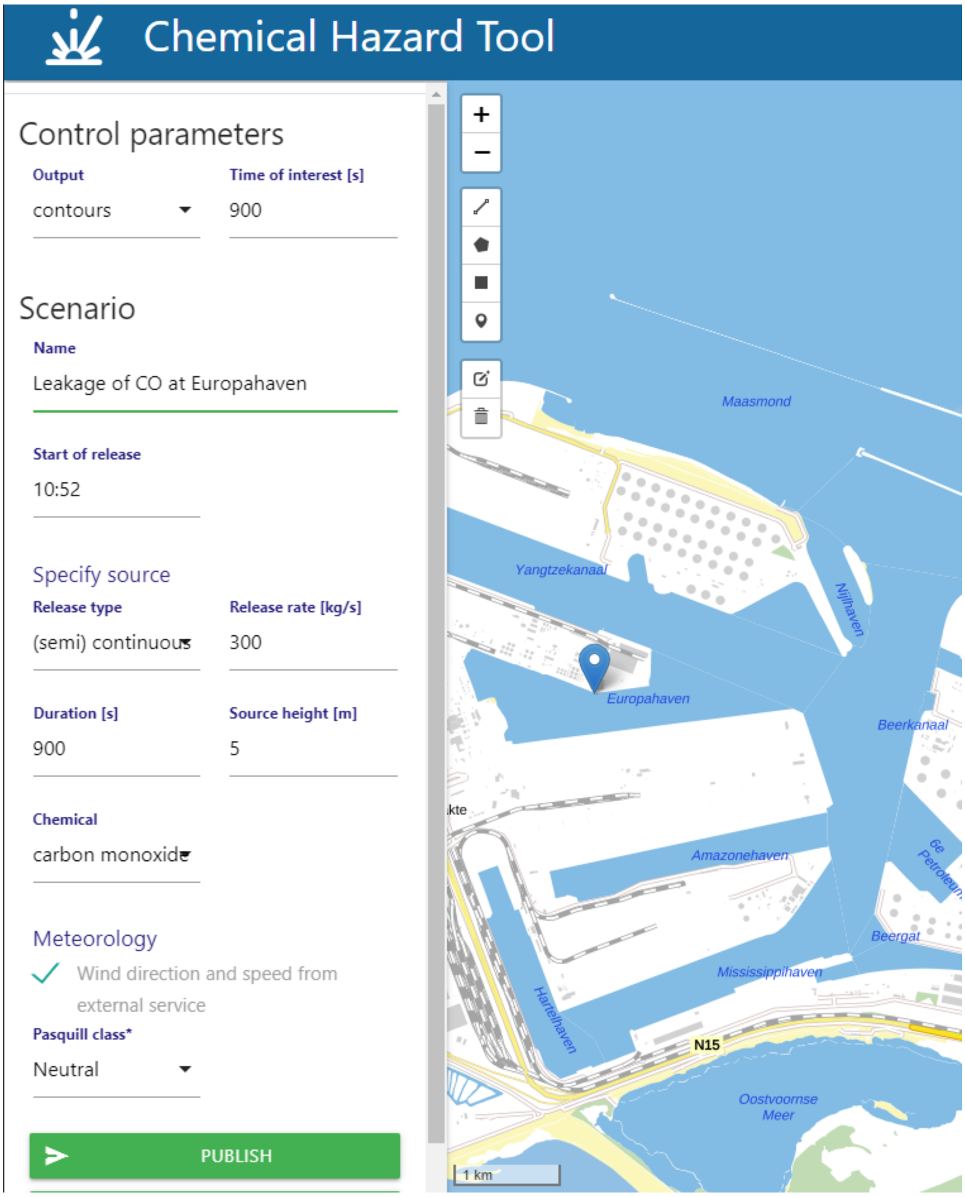


Figure 27 - Screenshot of input parameters.

When the above mentioned information has been entered, a new computation is performed, and the expected hazard cloud is drawn on the map. The expected development of the cloud as a function of time is calculated. Since it is difficult to distinguish hazard clouds at early times and later times, because the contour lines overlap, a time slider can be used to highlight a specific time, as shown in Figure 28. In this Figure, the image on the left shows no time slider used. In the middle, the image displays the short-term effect, while the right image displays the long-term effect.

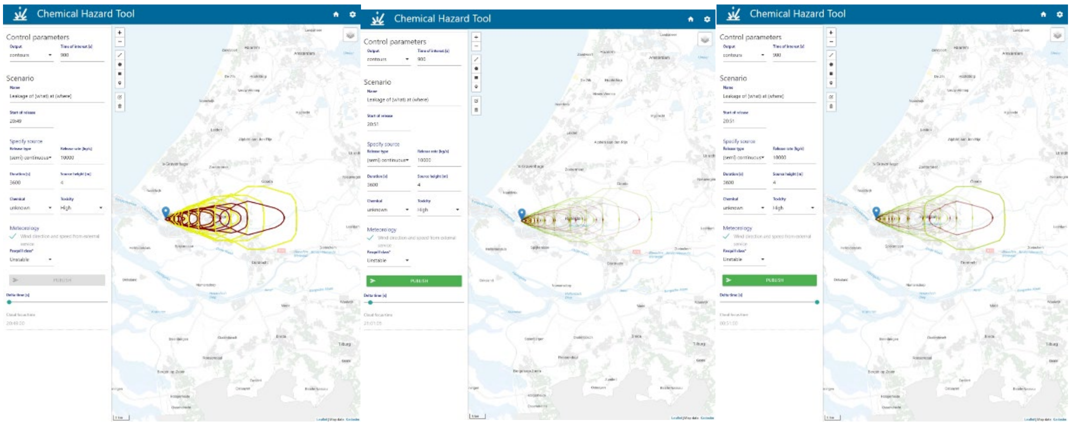


Figure 28 - Using the time slider, the cloud can be highlighted at different times

On the map, also the positions of first responders can be displayed: this is especially useful to warn them in case they get too near the hazard cloud.

Furthermore, when sensor measurement values are published through the SAS, these measurements will appear on the map as coloured dots, on which the operator can click or hover to see the measured value and properties like the sensor ID and measurement time. As more measurements are received, a more accurate picture of the plume can be obtained by starting a recalculation that uses the data fusion functionality (as described in Section 2.3.2.3).

The map also shows the locations of vulnerable objects like hospitals, elderly homes and kindergartens. The operator can highlight specific locations on the map that may require special attention because they lie in the path of an approaching gas plume.

Also, the operator can send out several types of messages to personnel or to the commander, for example advice to wear protective gear, to take measurements at certain locations or advice about warnings that must be issued to the population in the area.

Uncertainty (see also Section 2.3.4.4)

Most of the input variables for the calculations have uncertain values. When an accident has just occurred, it is very hard to estimate how much gas is being released, how long it will take to stop the release, and also the wind direction and speed are only known with limited accuracy. Therefore, the operator can also choose to display an ensemble of gas clouds. This is a visualisation which takes into account the uncertainties in input parameters, including the uncertainty in wind direction. This feature is experimental and has been subject of our research into how to present information that has inherent uncertainty to operators so that they can understand the information and that it helps them in their decision making. An example of a calculated ensemble of gas clouds is shown in Figure 24.

2.3.4.2. Iterative design and End-user involvement

During development of the Chemical Hazard Module, end users were regularly involved to develop a module that is in line with the FRs' needs and use context.

Several workshops and interviews were held with different groups of fire fighters, for example fire fighter teams that would be handling the situation on location; hazmat experts that are stationed at a distance and would be involved in large-scale accidents from a distance; and commanders and team leaders that would be mostly stationed in the command post at location. These different groups of FRs were introduced to the researchers' vision of the chemical hazard module and the ASSISTANCE goals, and the concept of the chemical hazard module. During these sessions, the researchers asked the FRs about their current practice of handling accidents with dangerous substances, what their information needs are (which differ for the different roles), needed communication, and their wishes regarding the module. In addition, the module was shown and the FRs could work with it, add new requirements and wishes regarding functionality and display possibilities.

Below, the usability test with the latest version is described in more detail:

Usability test

An interim usability test was conducted to assess the current work, shed light on possible improvements, and determine next steps. Two FRs (both male, one of them a hazmat Officer, the other an on-scene commander) were present, one of which was using the tool. Throughout the meeting the FRs were thinking aloud and stating their thoughts towards the module.

Method

First the tool was explained while the user was trying it out and asking questions. Then, five tasks were given to verify that the tool was understood. These tasks included calculating a gas cloud for a location, changing automatic meteorological data to manual, deciding a safe way to get to the site, and change the visualization of the cloud. This was followed by playing a scenario in which a gas leak was reported. The FRs were asked to verbalize what they would do at each step with information. As the scenario evolved, it was assessed how the system was used and what issues arose. After the scenario was played out a last question round was performed in which the FRs were able to state what could be improved.

Results

The usability test showed two aspects: (1) the FRs used the chemical hazard tool as intended, as a mean to gain situation awareness of the gas cloud and the predicted development of the gas distribution, and (2) the FRs need and wish a good situation awareness to better decide on follow-up actions.

Gaining situation awareness

The FRs appreciated the module and different views on the hazard prediction. They understood how to interpret the visualization and were happy with the different possibilities. However, they stated that there is a need for differentiating between a standard and an expert mode, as some of the outputs might be interpreted the wrong way by a non-expert. In the standard mode, some of the more advanced functionality should be unavailable, while it should be enabled in the expert mode, for users that have a higher level of training and education in the area of managing accidents with toxic chemicals. For example, the ensemble view which displays three clouds as possible deviations in the prediction of hazard development (see Figure 24) has been wrongly interpreted by non-expert FRs to mean the general expected distribution of the gas cloud. As the missions are often stressful and dynamically evolving, there is no room for misunderstandings and explanations and non-experts should have enough information on the easier-to-understand template view (see Figure 26).

Actionable and interpretable awareness

The FRs used the chemical hazard tool as an assistant to identify a safe route to reach the incident site and to make assumptions about possible actions regarding vulnerable locations (e.g., hospitals or residential buildings). As the Hazmat Officer put it: ``The whole reason that you want to see the cloud on a map is to see vulnerable spots to decide whether we should take action and for example evacuate particular parts of the surroundings.'' Moreover, the displayed cloud distribution was used to decide where to send FRs of the regional teams to do measurements in the affected areas. The Hazmat officer mostly used the 'contour' visualization of the cloud to aid his decisions (see Figure 28).

Furthermore, the results of the usability test have been used to analyse the current status of the module and additional user requirements have been identified and existing requirements have been further detailed. This has resulted in a list of (small) adjustments of the user interface and additional functionalities that have been implemented into the current module, for example to add a ruler to make distance measurements easier; use the terms Stable, Neutral, Unstable for atmospheric stability instead of the term Pasquill Class and the letters A..F; give notifications if values are missing.

2.3.4.3. Team Design patterns

Design Patterns provide generic (re-usable) solutions for recurring problems. They can be implemented again and again but never look the same (Alexander, Ishikawa, & Silverstein, 1977). Hence, they are like templates that can be used later on to solve similar problems. Moreover, different patterns should be seen as related to each other as they can be combined to larger patterns. A variety of fields have picked up design patterns as a tool because they offer a good way of sharing and structuring design knowledge (e.g, interface design patterns (Van Welie, Van Der Veer, & Eliëns, 2001)). The field of Human-Agent Teaming (HAT) has also started to use patterns to design how humans and artificial agents should collaborate, they are referred to as Team Design Patterns (TDPs) (Van Diggelen & Johnson, Team design patterns, 2019). This project focuses on patterns in first response teams that consist of human first responders and artificial intelligent (AI) agents.

The benefit of using TDPs when designing HAT is that it offers a tool for identifying, specifying, exploring and transferring of desirable collaborative and interactive. Furthermore, these TDPs can be programmed into the AI agent which enables it to engage in team processes (Van Diggelen, Neerincx, Peeters, & Schraagen, 2019). An additional benefit is the graphical language that (Van Diggelen & Johnson, Team design patterns, 2019) proposed. It facilitates communication about design choices among different fields. Concerning this purpose, (Van der Waa, van Diggelen, Siebert, Neerincx, & Jonker, 2020) argues that the intuitive nature and simple graphical language of TDPs facilitate discussion about human-machine teamwork solutions among a wide range of stakeholders including non-experts. The TDP language can represent different types of work, different degrees of engagement and different environmental constraints. Doing so, critical aspects of teamwork can be captured which "enables a holistic view of the larger context of the teamwork".

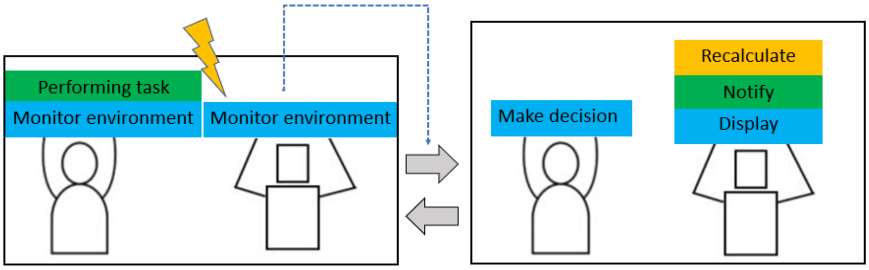
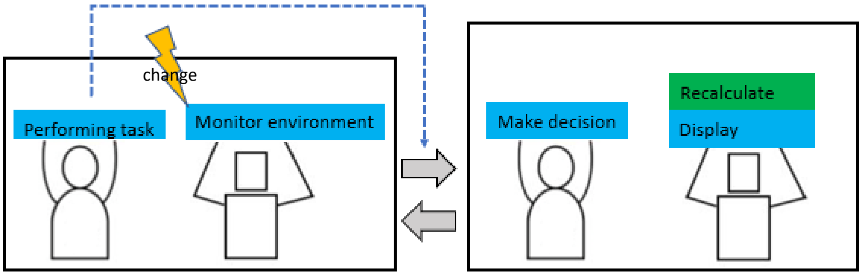
TDPs are especially profitable for this project because the team looking for solutions in ASSISTANCE consists of experts from a variety of fields and TDPs guide discussions about HAT solutions. Also, it is an optimal medium to communicate and present design intentions to the stakeholders (first responders). Additionally, as first responders work in safety critical environments where their decisions and actions have great implications, implementing an intelligent system should be done with caution. It is important to avoid unintended interaction failures due to not considering possible human factors issues or not recognizing interdependencies within the team. In TDPs roles, responsibilities and elements of teamwork can be made explicit which helps to prevent possible issues.

Team Design Patterns are especially useful to describe dynamic task allocation in a human-agent (as well as humans-only) team context. Whereas humans can learn and develop teaming skills, AI agents need to be programmed for this purpose. TDPs assist in identifying the functional requirements (incl. the communication requirements) for the AI agent to show the team behaviour. They specify how roles and responsibilities are to be construed for a team task in first response teams that consist of human agents and Artificial Intelligent agents.

Over the past few decades, a lot of attention has been given to the use of Artificial Intelligent agents to advance the momentary situation awareness of first responders in dynamic, uncertain demanding environments. Artificial Intelligent agents are increasingly deployed for a wider variety of tasks with the goal to improve safety and performance. However, to be able to improve momentary situation awareness Artificial Intelligent agents should complement human capabilities and relieve them from demanding tasks (Neerincx, van Diggelen, & van Breda, 2016). It therefore is important to explicate how roles, responsibilities and tasks are to be divided between human and Artificial Intelligent agents in the envisioned teamwork.

We investigated how this collaboration between the AI agent and FR should be designed in situations that require moral decision making and situations that need adaptation towards dynamic changes during the mitigation of an incident. A moral dilemma for example occurs when the FR has to decide between an action that will safe civilians but is potentially harming for the FR. Working agreements are proposed to harmonize the work and information processes to the context (e.g., when, who in which way will be informed about a change in risk of exposure to hazardous substances). We investigated how working agreements can be specified in the TDP to consider possible emerging moral issues and define beforehand how the AI agent will support the FR in situations that bring forward such issues. Dynamic changes during the mitigation of a disaster are constantly present, examples are wind direction changes and progress or regress in mitigating the source.

For dynamic changes, TDPs can be used to make the responsibilities explicit within a human-agent team for recognizing, communicating and initiating the need for adaptations towards the changed conditions. For example, the AI agent could have the responsibility to monitor the environment, inform the FR of any changes that it registers and display the effect it has on for example the gas cloud distribution. The FR then decides how to react accordingly to these changes. How this interaction should be displayed in concrete scenarios is defined by deriving Interaction Design Patterns from the created TDPs. In *Table 2* an example TDP is provided that illustrates the teaming aspect of the developed module and the FR, and how it provides assistance in an environment with everchanging conditions. *Table 3* show an according interaction design pattern could look like. An evaluation of the created TDPs will be performed by presenting scenarios to the FR which need to be solved in collaboration with an AI agent that acts according to the implemented TDP.

Name:	**Information Support**
Description	The human agent is performing his/her task when some change in the environment happens. The AI agent recognizes the change and initiates a recalculation of the SA model. It then displays the consequence and notifies the human agent. The human agent interprets the implications of the new situation and considers actions and decisions to take. A variation is that the human agent notices the change and initiates a recalculation.
Structure	 OR 
Example	a) The commander is delegating tasks to the firefighters based on the calculated gas dispersion. The wind direction changes, and the agent recalculates the gas distribution and notifies the firefighter about the change. The firefighter investigates the changed situation and decides on how to adapt his action (e.g. if a region needs to be warned; if firefighters should be sent to other areas that should be measured; if other resources or back up is needed). b) When approaching the incident location, the firefighters notice that the wind vanes are pointing towards a different direction then indicated by the module. The firefighter requests a recalculation indicating the changed wind direction.
Requirements	The *Human Agent* has to make decisions about his/her next actions. The *Artificial Agent* needs to be able to monitor the environment.

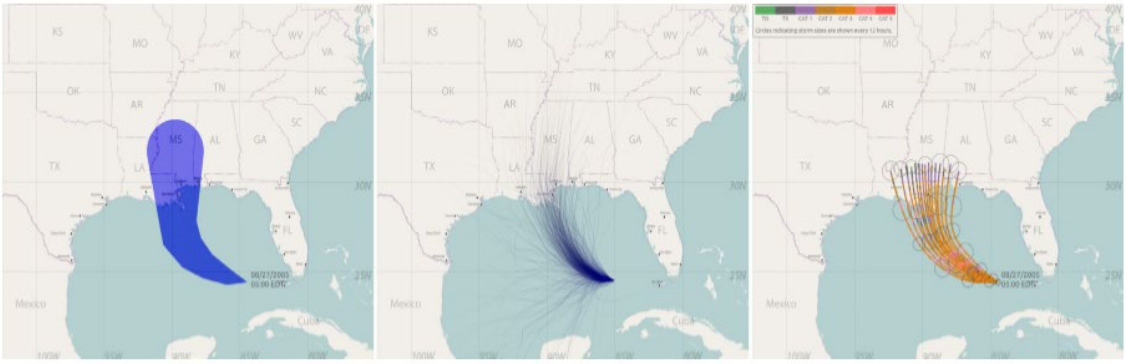
	The *Artificial Agent* has to be able to recalculate and display changes that had happened in an understandable way. The *Artificial Agents* presents information in a manner that it induces accurate trust calibration.
Advantages	+ *Human Agent* is responsible for all decisions + *Artificial Agent* is not required to interpret changes in the environment with regard to actions that should be taken.
Disadvantages	- *Human Agent* might be overloaded.

Table 2 - Team Design Pattern- Information Support

Name	**Communicate Changes and Consequences**
Problem Summary	The FR needs to be notified about changes that affect the distribution of the visualized gas cloud. These changes can be a deviation in wind direction or new measurements that reduce the uncertainty of the distribution. The FR should be able to recognize the altered situation.
	
Usage	Use when it is important that the user can compare the previous state with the new state.
Solution	Display an explanation to what has happened and why there was a change of state and recalculation of the display. Make sure that the old state is still visible, thereby increasing the observability of the change.

Table 3 - Interaction Design Pattern - Communicate Changes and Consequences

2.3.4.4. Visualization of uncertainty



(a) explicit: uncertainty cone (b) implicit: ensemble of tracks (c) selectively sampled annotated tracks

Figure 29 - Cones, tracks, and annotated tracks as studied by (Liu, et al., 2017).

In case of a toxic gas related emergency, first responders quickly need to be aware of the predicted direction and intensity of the gas cloud. Current gas visualization methods are often using cones to represent the spatial uncertainty of the location of the cloud (see for an example Figure 30, left). Previous studies show that cones could be misleading ((Ruginsky, et al., 2016), because people confuse the size of the cone with the predicted size of the cloud. Previous work on hurricane visualization show that sampled annotated tracks can solve this misinterpretation (see Figure 30, right, (Liu, et al., 2017)).

For the ASSISTANCE project we developed, based on literature mentioned above, the two visualizations for gas related emergencies, see Figure 30.

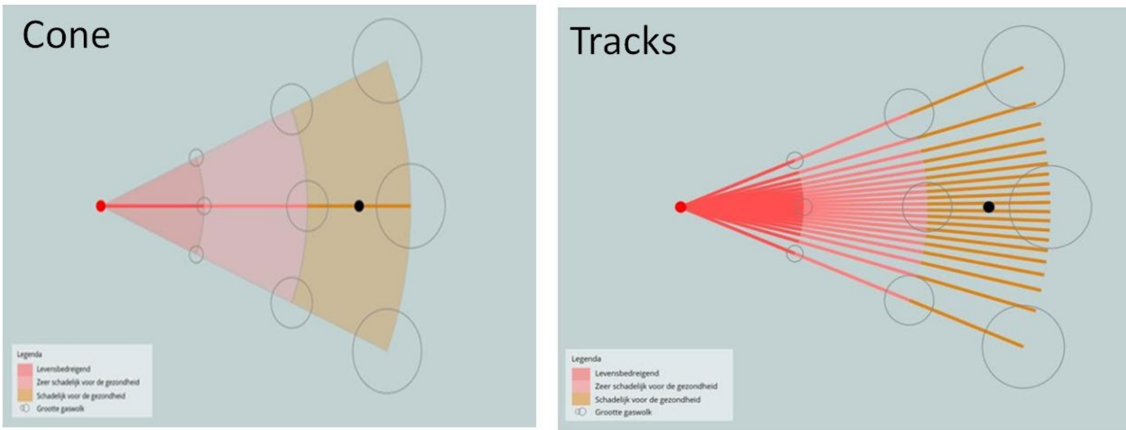


Figure 30 - Cone and Tracks visualizations for the ASSISTANCE chemical hazard module display.

We conducted a pilot user study with 54 non-expert participants to compare the performance between the cone and tracks. In the experiment asked participants to estimate the risk on a particular location on a 1-7 scale. We varied the locations in terms of intensity level (high-medium-low) and distance-to-center (0 = the middle, 4 is the most distant point) (see Figure 31).

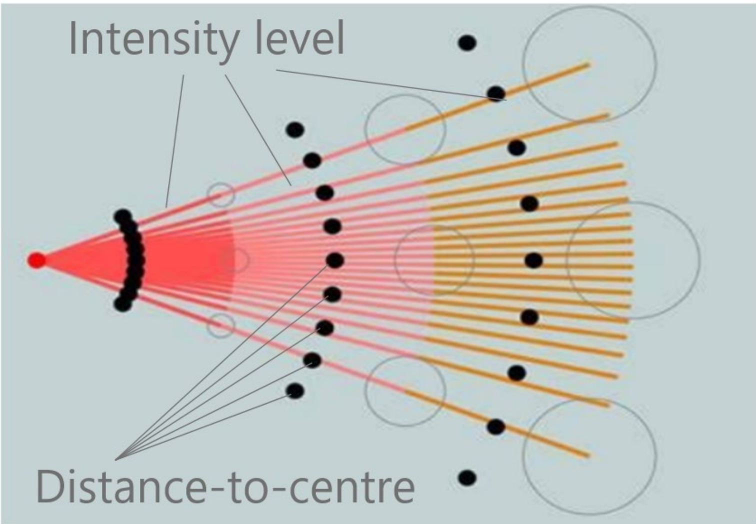


Figure 31 - Track visualisation with risk evaluation points for different intensity levels and distance-to-centre.

We found that for both cone as tracks, there was significant decrease in risk estimation for the lower intensity levels. We also found for both visualizations that there was a significant decrease in assigned risks when the distance-to-centre was larger. There was no difference found in risk estimation between the visualizations (Ticheler, 2020); (Venrooij, n.d.).

Based on this pilot experiment it can be concluded that both visualizations could be used as effective visualizations of gas clouds. More research is needed to determine whether this also holds for experts and how temporal uncertainties should be included in this visualization.

2.3.5. Requirements fulfilment

A total of 14 requirements were identified in D2.2 to be realized with regards to the Chemical Hazard Module (CHM).

Id. Req	**Description**	**Fulfilled**	**Specific actions**
CBR_001	The CHM should listen to a central data bus (the SAS).	√	See Section 2.3.2.1.
CBR_002	The system can present sensor information in real-time.	√	See Section 2.3.2.3.
CBR_003	The end user shall be able to locate new gas measurements on the map.	√	See Section 2.3.2.3.

CBR_004	The CHM dynamically predicts the position of the hazard footprint based on real-time meteo information, a realistic landscape, and real-time sensor information.	√	See research described in Section 2.3.2.3. The module only takes a realistic landscape into account when this data is openly available (as is for example in the Netherlands).
CBR_005	The user should easily understand the visualisation of the gas measurements on the map.	√	See research described in Section 2.3.4.4.
CBR_006	The CHM dynamically calculates the current position of the hazard footprint based on real-time meteo information, a realistic landscape, and real-time sensor information.	√	See research described in Section 2.3.2.3.
CBR_007	The CHM should be suitable for training.	√	It is possible to train operators and FRs by running different simulated scenarios, see Sections 2.3.2.3. and 2.3.4.1.
CBR_008	The CHM can determine a danger zone, including highlighting vulnerable places such as hospitals.	√	See Section 2.3.2.3.
CBR_009	The CHM can warn the first responders about approaching the danger zone, in all phases of the emergency.	√	The system can display the position of FRs and the operator can send warnings to those FRs.
CBR_010	The CHM can localize and position all people and critical assets close to/in the danger zone	√	Fulfilled (for people). Localization and positioning of people will be described in D4.4: Wearable sensors integrated into the system
CBR_011	The CHM can calculate the uncertainty of the gas cloud position.	√	See Section 2.3.2.3.
CBR_012	The CHM can calculate the optimal sensor position based on current prediction and measurements to gain more certainty about the position of the gas cloud.	√	The module helps the operator to determine optimal sensor positions to gain more certainty. Additional support by the module will be integrated in the next month and further described in D5.4: Final SA Platform Integration.
CBR_013	The CHM can calculate the fall-out area.	-	The CHM calculates the hazards of toxic gases. Gases have no fallout (dry deposition is not significant). Adding the modelling of smoke plumes with fallout of particulate combustion products is considered as an option to add later.
CBR_014	The system shall generate a static visualisation of the situation with the	√	This is just one of the ways the User Interface can show the

	following information: Title mentioning the name of the gas; Subtitle mentioning the time of visualization generation; Map with current/predicted levels of danger		results of calculations. The UI is interactive.

Table 4 - CBRN Hazard evolution requirements

2.4. Damaged Assets Location and Routing

2.4.1. Overview

The Damaged Assets Location and Routing (DALR) module leverages real-time images provided by cameras and drones/UAVs to support decision making processes of first responders. This module implements three main functionalities:

- Locate damaged assets and/or infrastructures or impassable areas.
- Provide a routing mechanism to obtain intervention or evacuation routes avoiding identified damaged assets.
- Assist the evacuation management process providing intervention and evacuation times based on appropriate optimization and decision algorithms.

All these functionalities together with the joint work and data obtained by other modules (e.g. sensor data, CBRN module, etc), stored in the Sensor Abstraction Service (SAS), lead to a remarkable innovative step thanks to the real-time interaction between first responders and simulation modules.

2.4.2. Description of components

DALR module has two main components described in the following subsections. The unified architecture is shown in Figure 32.

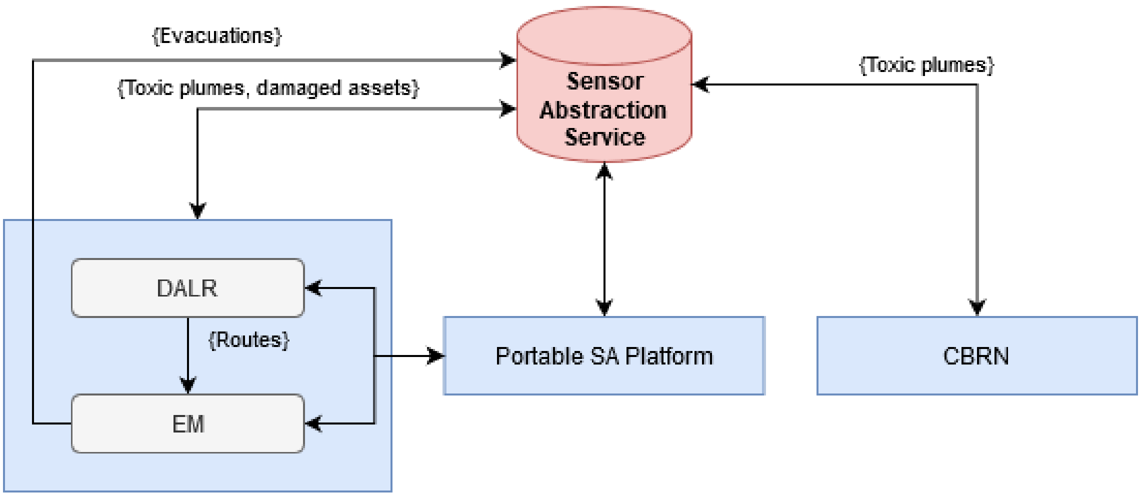


Figure 32 - Unified DALR architecture.

2.4.2.1. Damaged Assets Location and Routing

DALR is a high-end application that combines real video images with 3D cartography and map manipulation tools. These features allow users to: 1) analyse the video recorded by a UAV, 2) detect damaged areas or any element that may influence the management of an emergency, and 3) interact with the map by overlaying geometric objects and classifying them so that other services such as Evacuation Management (EM) take them into account when making their calculations.

The main component of this module for the operator is its **GUI**. It has a small main menu on the top to select the needed parameters to use the application: a context, a resource (e.g. a specific UAV), either a mission to use its start and end dates or manually selecting the desired dates with the corresponding date pickers, and the camera from which the operator wants to reproduce the video. The rest of the interface is divided into two differentiated zones.

On the left side we found the **Smart Video Player (SVP)**, an advanced video player capable of reproducing several files as a single one (as explained below, to play the video recorded between two dates may involve several video files). It also includes usual features of video players like loading only the necessary piece of video or seeking the desired moment.

The right side displays the 3D map and its dedicated menu to interact with it. These two zones of the interface can be resized, so the user can expand the left zone to make the video bigger or reduce or even hide that zone to focus on the map and the damaged areas management.

Some screenshots of the DARL GUI are shown below in Figure 33, Figure 34, Figure 35. Since no real videos were available during the development stage, tests were carried out using the open-source film (Big Buck Bunny, 2008).

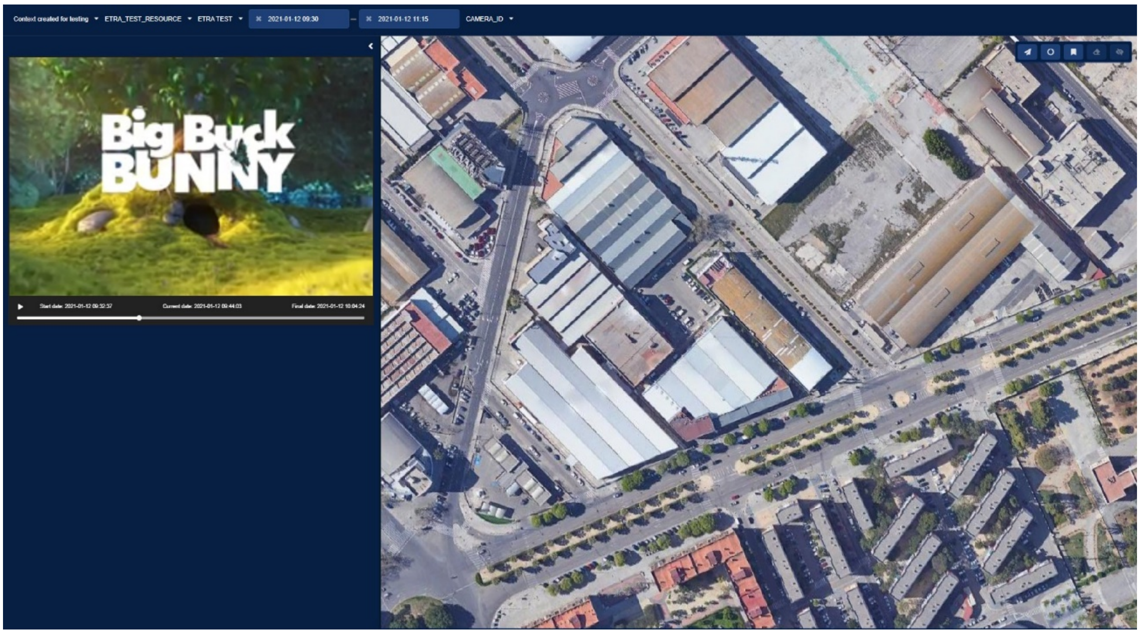


Figure 33 - DALR GUI playing a video.

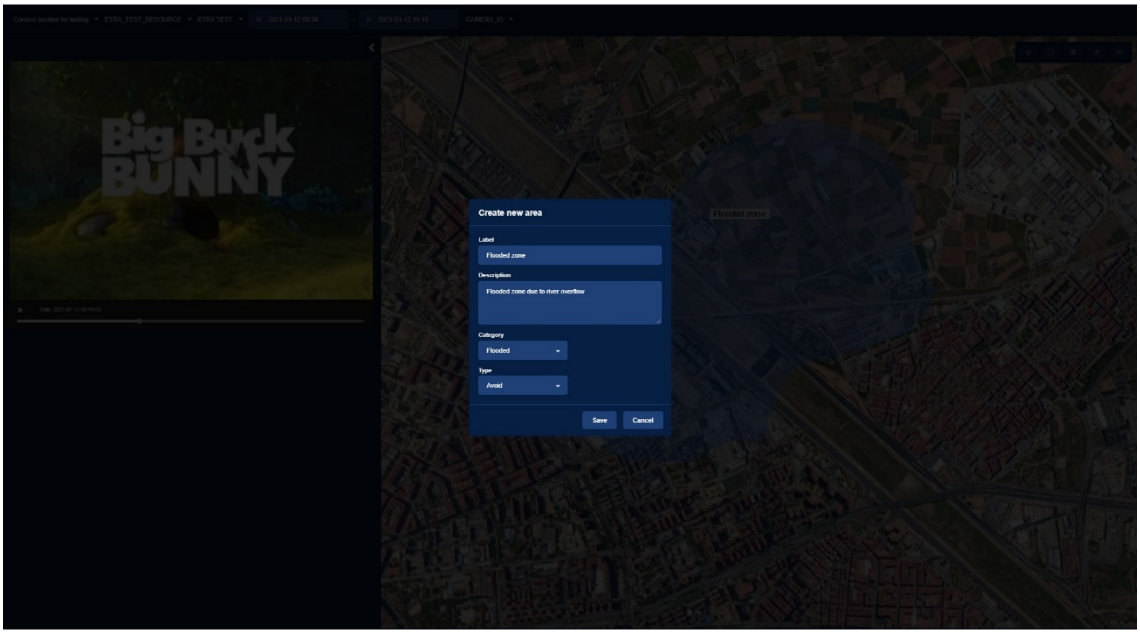


Figure 34 - DALR GUI creating an area to be avoided.

Figure 35 - DALR GUI with left side hidden.

To manage the video streamings, DALR has an essential subcomponent at the server side: the **Video Manager (VM)**. It is composed by two core elements:

- **Video Recording System (VRS)**. It deploys an instance of Shinobi Open Source CCTV software (https://shinobi.video/) to process the video streaming signals and transform them into video files. If a whole streaming broadcast by a resource were saved in a single file, it could become very large and difficult to manage, so it is divided and saved in several files according to the configured maximum duration. That is why it was commented before that the reproduction of the desired video can imply the access to multiple files. While it would be possible to develop a dynamic configuration service, it would require a huge effort to create such a system which would not be cost-effective. Currently a manual configuration is required for each streaming to be processed setting mainly its URL and codec.

- **Video Server (VS)**. This service is in charge of managing the configured streamings in the VRS and serving the video files. It exposes a RESTful API used by the SVP to request the corresponding files data (name, start, end) and to receive the video to be played.

In parallel, the **Routing Service (RS)** is available as a backend application (to be used mainly by the EM) to process requests for routes from a departure point to a destination point. It is connected to the NATS messaging broker exposed by the SAS listening to the topic “assistance.mapping.routing” to process the received messages and generate a query to send to Openrouteservice (https://openrouteservice.org/) for the routes calculation. It then returns an array containing one or more routes trying to apply all the parameters of the request. This means that in some cases it might not be possible to comply with a particular restriction, such as not passing through an avoid area when there is no other possibility.

The architecture overview is shown in Figure 36:

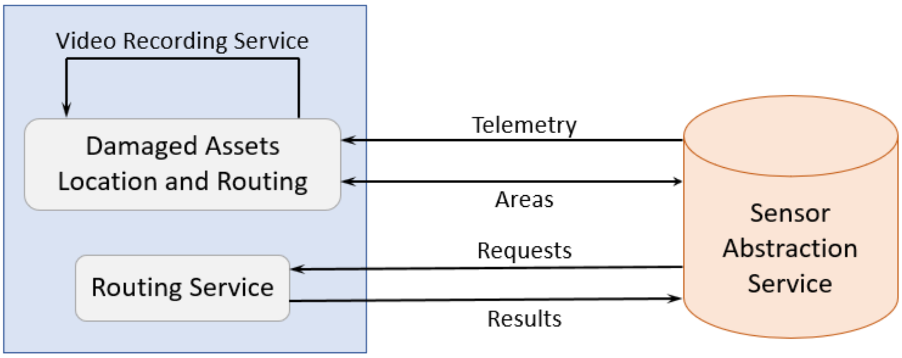


Figure 36 - DALR components and communication overview.

2.4.2.2. Evacuation Management

The Evacuation Management is able to provide comprehensive evacuation strategies through processing geographical information including evacuation areas, damaged assets location and toxic plumes. The main variables used to characterize a particular evacuation strategy are: 1) assembly point locations, 2) shelters locations, 3) evacuation routes and 4) traffic simulation results. This module is composed by the following components (see Figure 40):

- Graphical User Interface (**GUI**): it allows the operator to manage the active evacuations via Geographical Information System (GIS), providing an intuitive and visual interface that shows real-time status of the existing damaged assets and toxic plumes. The operator can modify the evacuation parameters and re-simulate the evacuation to explore alternative evacuation strategies - see Figure 37, Figure 38 and Figure 39.
- Assembly Points Model (**APM**): it processes the input geometry of the evacuation to provide a set of assembly points considering the number of evacuees, the points of interest in the evacuation area and the maximum recommended distances from starting pedestrian locations to assembly points.
- Shelter Points Model (**SPM**): it takes as input active evacuations, damaged assets and toxic plumes to provide a set of feasible shelters located at the required distance far from dangerous areas.
- Routing Model (**RM**): it uses dedicated services to provide a routing plan ensuring a uniform distribution of evacuees in the shelters. Also, this model deals with likely interactions between routes (e.g. road section used by more than a route or distribution of vehicles in an intersection).
- Pedestrian Simulation Model (**PSM**): this model is used to simulate the pedestrian response behaviour and movement through the proposed evacuation routes to predict the potential results of the simulated process.
- Vehicular Simulation Model (**VSM**): this model is used to simulate the vehicular stage in the evacuation process to obtain estimated route parameters (e.g. traffic density or average speed).

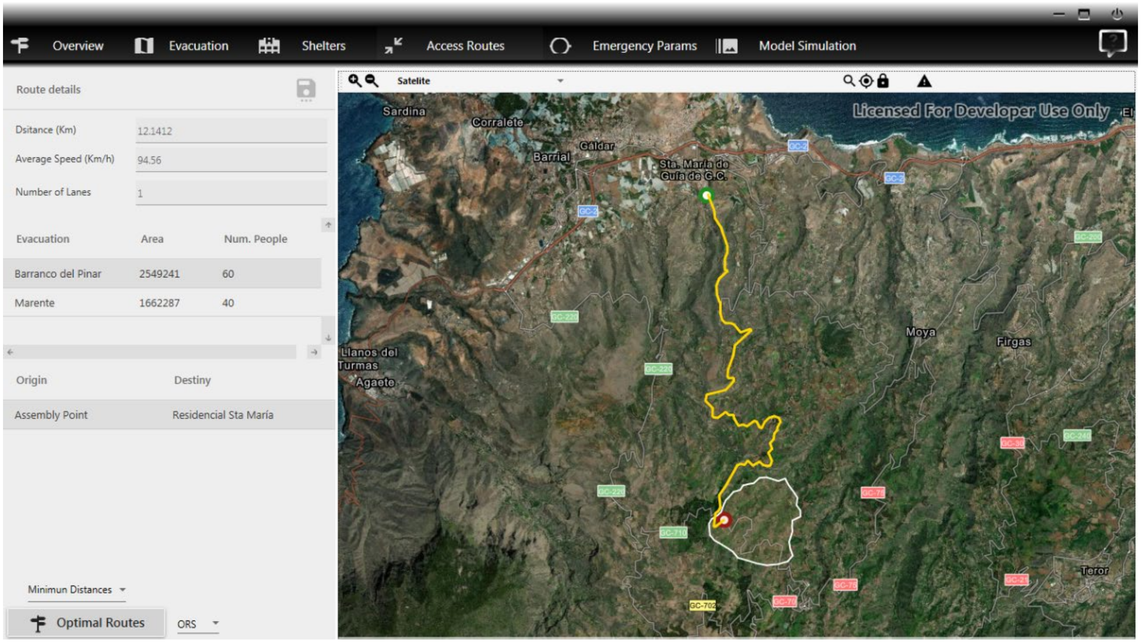


Figure 37 - EM module GUI. Evacuation route to Assembly Point

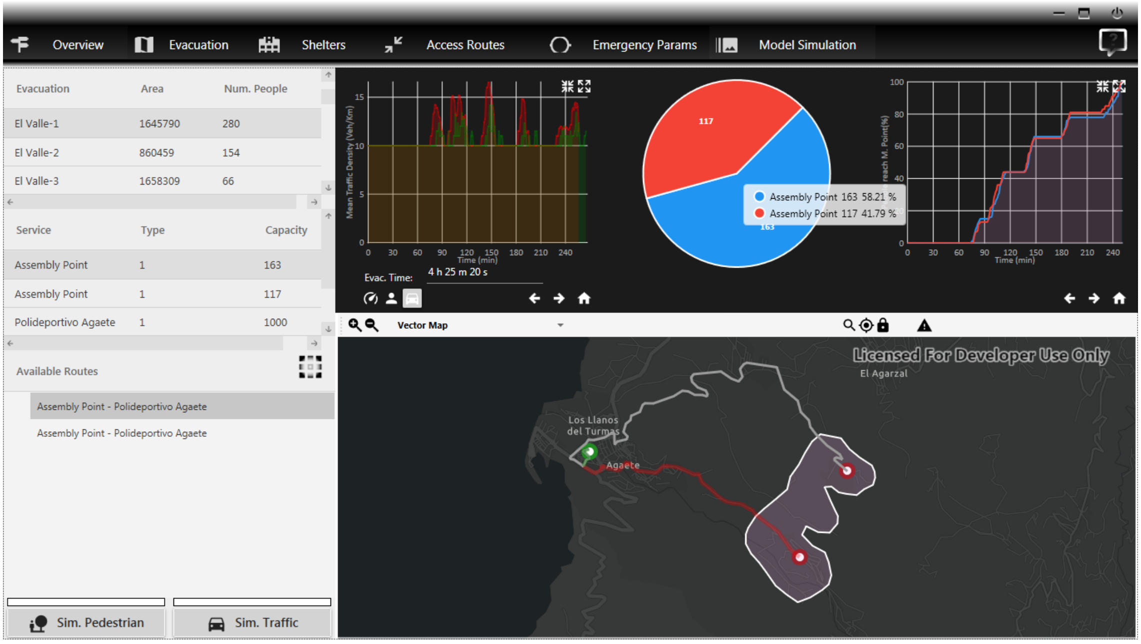


Figure 38 - EM module GUI. Decision Support Dashboard

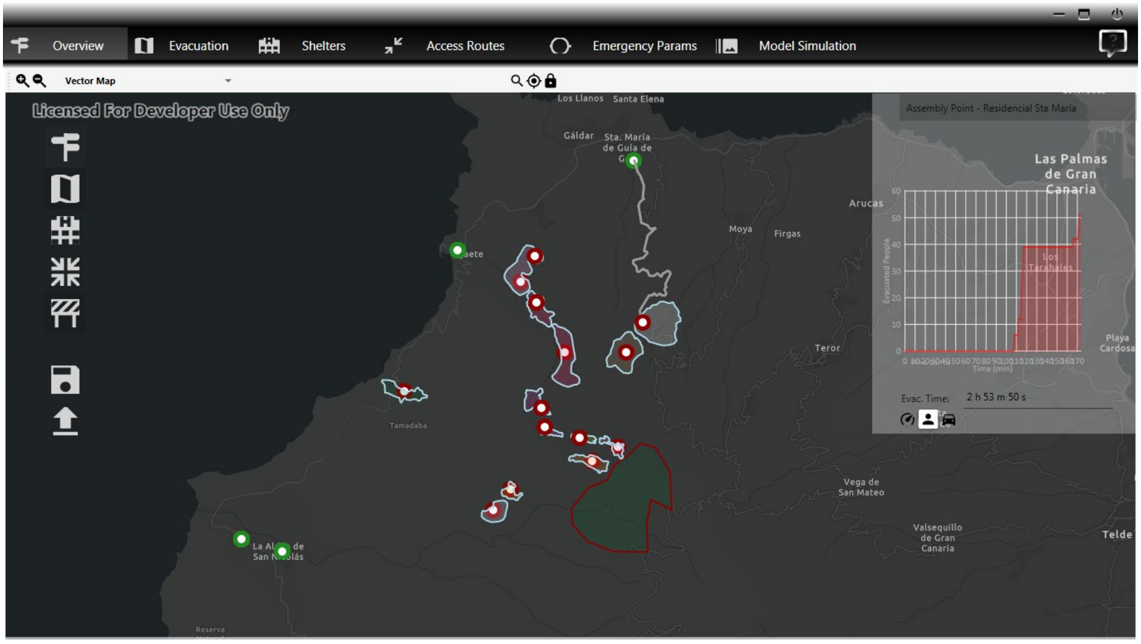


Figure 39 - EM module GUI. Population evacuation progress

The architecture of this module will be client-server with the GUI at the client side and the other detailed modules operating in the server side, providing an API REST interface.

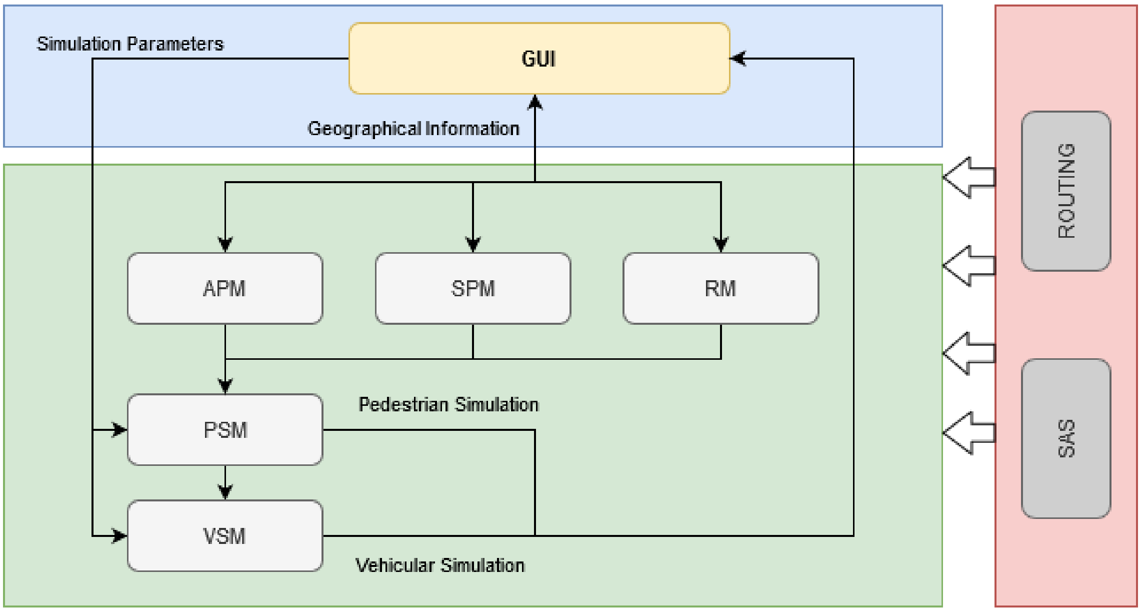


Figure 40 - Functional architecture Evacuation Management Module.

2.4.3. Integration with SAP

2.4.3.1. Damaged Assets Location and Routing

We can differentiate two ways of communication with the DALR within the SAP:

- DALR connects via DDP with the SAS to a) get basic data from contexts, missions and resources so the user can select the desired video; b) retrieve the telemetry data necessary to synchronize the video and the cartography; and c) manage the areas data (create, update, delete).
- RS connects to NATS to receive requests by any module (it is only expected to be used by the EM but it is available for any other that might need it).

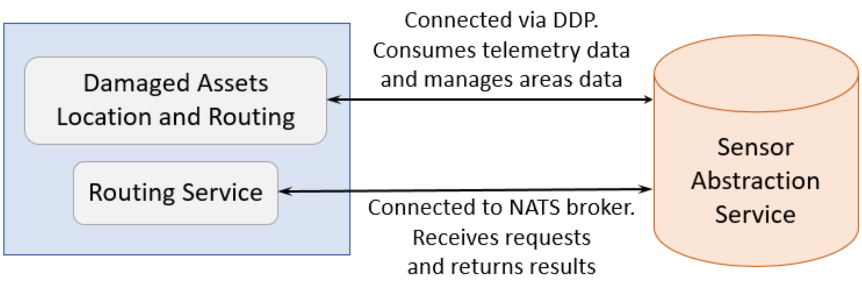


Figure 41 - Integration of DALR within SAP.

2.4.3.2. Evacuation Management

All the capabilities defined in the architecture follow a client-server architecture. For this purpose, a set of interfaces will be developed in Task 5.4 following a RESTful API design over HTTP protocol. A possible approach of the interfaces implemented is shown in Figure 42.

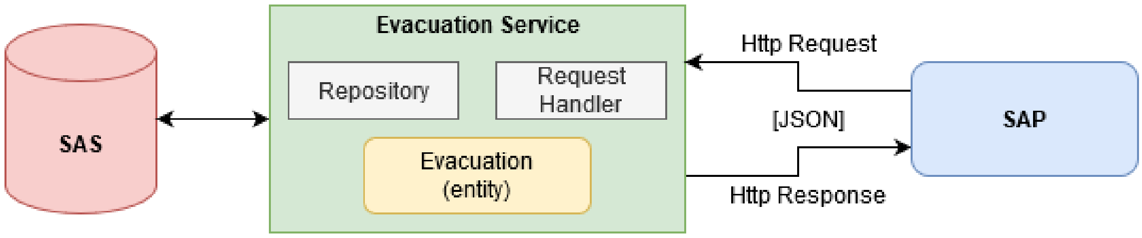


Figure 42 - SAP Integration architecture.

2.4.4. Operational procedures

2.4.4.1. Damaged Assets Location and Routing

The first step to make the whole system work is to configure the Video Recording System to receive the corresponding streaming emitted by each resource and store it into video files. Once configured, resources must command the system to start recording the streaming when emission is on (for example, when a UAV takes off) and to stop the recording when no longer needed (for example, when landing).

The entry point of the DALR is its GUI, where the operator selects needed parameters: context, resource, dates and camera. The Smart Video Player sends then this data to the Video Server, which communicates with the Video Recording System via its internal API, and responds with a list of objects containing the name, start and end dates and duration of the corresponding video files. The SVP processes this data to show a unified seek bar to move through the videos as they were a single file: the operator sees a regular video player playing the video recorded between the selected dates but internally it is requesting to the VS the necessary fragment of the video corresponding to the instant that it is being reproduced.

While playing the video, DALR synchronizes the shown map area with the video image, retrieving the corresponding UAV telemetry and moving the virtual camera on the map accordingly. This way, the camera flies in a simulated way over the map replicating the position, height, orientation, etc. of the real flight of the UAV. It is possible to seek to the desired moment and/or pause the video at any time to analyse the image and highlight an area.

Creation of overlay areas is done either by drawing a polygon point by point or creating a circle by selecting its center and radius. After the desired zone is selected, a popup menu allows the operator to introduce data such as a label, a description and a category to assign a risk level. Finally, the area is stored into database so that other modules can extract this data and process it. For example, the EM can request the avoid areas for its algorithms to know where it is not possible or it would be risky to pass.

Regarding the Routing Service, as it was documented before in section 2.4.2.1, it is connected to NATS waiting for incoming requests. When a message is received, it parses the body to validate the parameters and creates a query to send to OpenRouteService. Then it validates the result and transforms it to a specific data model. This allows to change the third-party service in charge of the routes calculation without modifying the structure of the incoming messages or the results returned.

2.4.4.2. Evacuation Management

Particular characteristics of a disasters (Gravley, 2001) (Montz, Tobin, & Hagelman, 2017) and the factors that shape evacuation evolution have to be considered to create appropriate models that do not contradict the untreated evolution of natural, technological and/or man-made disasters (Schneiderbauer & Ehrlich, 2004). In fact, disasters have common attributes that can be redefined mathematically to better understand the proposed models.

- A disaster is noted as $D_i = \{E_i, A_i\}$ where $i \in \mathbb{N}$ is the number of active disasters at the same time.
- $E_i = \{e_1, e_2, \dots, e_j\}, j \in \mathbb{N}$ is the set of active evacuations related with the disaster D_i.
- $A_i = \{a_1, a_2, \dots, a_k\}, k \in \mathbb{N}$ is the set of damaged assets, impassable areas or toxic plumes resulting from the disaster D_i.
- Elements contained in E_i and A_i are also defined as a set of geographical coordinates $e_j, a_k = \{(\phi_1, \theta_1), (\phi_2, \theta_2), \dots, (\phi_l, \theta_l)\}$ where $l \in \mathbb{N}$ is the number or coordinates that are used to define the evacuation/damaged area.

A mixed evacuation model was selected, including a first pedestrian phase and a second vehicular stage. Regarding to the strategy: staged counterflow evacuation was selected since there are studies that indicate it is the most optimal evacuation strategy for major disasters (Kaisar & Parr, 2012). This staged methodology divides the evacuation area into small zones which are evacuated sequentially according to the proximity dangerous area. On the other hand, the counterflow strategy optimizes traffic flow by making use of both lanes simultaneously in the same direction. This process can be divided into two main procedures.

1) Evacuation Routing.

A route can be defined primarily by an achievement of geographical points, starting from an origin and reaching a destination. Therefore, for the calculation of the evacuation routes, it is necessary to know three main factors.

Assembly points behave as route origins and are generated creating division zones according to characteristics previously defined.

Hence, a hexagonal tiling strategy is followed where the evacuation area is completely covered and, in turn, an assembly point $C_i = (C_\phi, C_\theta)$ is defined in the centre of each hexagon, which is, as a general rule, a location known by the nearby population, see Figure 43. For the adjustment of the optimal assembly locations to widely known public places, the external service OpenRouteService (ORS) (https://openrouteservice.org/) is used. This external service contains a database of places and points of interest classified in such a way that they can be filtered according to the intended use. These hexagonal zones ensure two fundamental properties: 1) every evacuee in the evacuation area is located at a maximum distance r (circumradius) from any assembly location, 2) the evacuation capacity of the assembly point does not exceed the expected population load in each hexagonal zone.

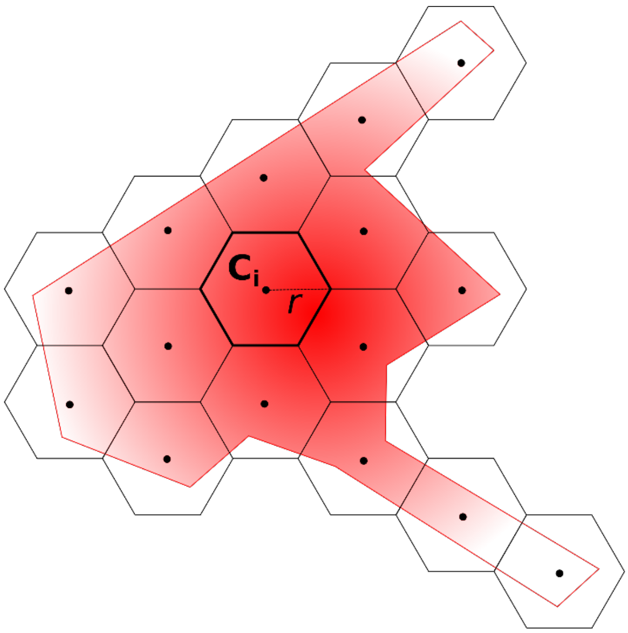


Figure 43 - Hexagonal tiling (black) over evacuation area (red).

Destination shelters of the evacuation have to be defined as geographical coordinates. In our case the considered shelters are emergency or temporary based on three criteria: 1) shelter purpose (Bashawri, Garrity, & Moodley, 2014), 2) facilities and equipment (Shelter-Centre, 2012) and (Huang, Yin, & Chu, 2019) and 3) spatial location (Boonmee, Arimura, & Asada, 2017) due to the intended real-time application. Three fundamental principles are required to fulfil the spatial location:

- $\forall\, e_j \in E_i\ \exists\ S_{ij} = \{s_1, s_2, \dots, s_m\}$ where S_{ij} represents all feasible shelters which distance to e_j geographical boundary ranges in $[d, d+\mu]$ where $d+\mu$ is the maximum distance.
- S_{ij} is systematic uniform random (UR) generated on the circumference with radius d ensuring well distributed locations,

$$\alpha_0 \ \sim \mathrm{UR}(0, 2\pi)$$
$$\alpha_l = \alpha_0 + l \cdot T$$
$$T = 2\pi/m$$

where m is the total number of shelters, see Figure 44.

- $\widehat{S_{ij}} \subset S_{ij} \mid \forall\, s_m \in S_{ij}, \forall\, l_n \in (E_l \cup A_l), d \leq Dist(s_m, l_n) \leq d+\mu$ where function $Dist(s_m, l_n)$ is the minimum distance between shelter geographical location s_m and damaged asset or alternative evacuation l_n geographical boundary (see Figure 44).

These fundamental principles ensure the existence of uniformly distributed shelters within a safe distance apart of any other emergency event. The geographical locations of these shelters are slightly modified using ORS to well-known Points of Interest (Pois) nearby the optimal location surroundings to fulfil sheltering purposes and facility requirements.

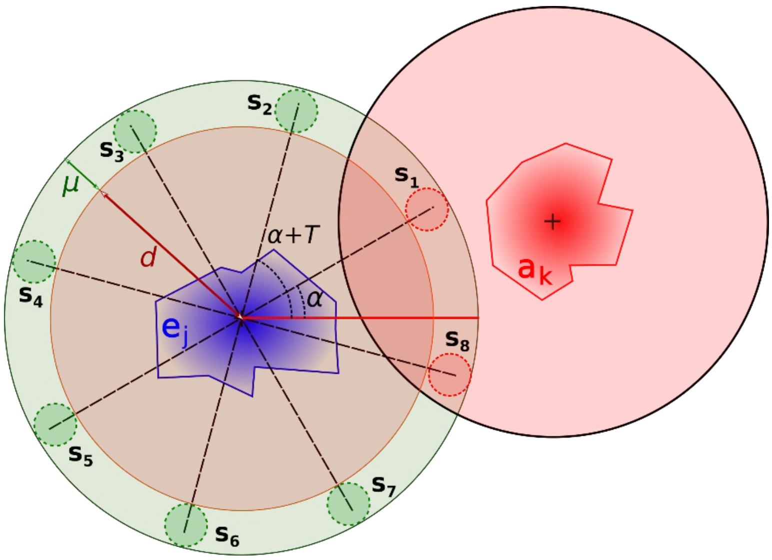


Figure 44 - Uniform spatial distribution of shelters avoiding damaged assets and alternative evacuations.

Evacuation routes are obtained using dedicated routing service provided by DALR from assembly points to destination shelters avoiding A_i damaged assets, giving place to a matrix of routes. These routes are filtered prioritizing those with better characteristics (number of lanes, maximum speed allowed, route overlapping) reducing them as much as possible while ensuring two mandatory conditions: 1) at least one route starts from every assembly point, 2) the distribution of evacuees is uniform for shelters defined as destinations, see Figure 45.

The set of routes for a given evacuation e_j can be defined as $R_i = \{r_1, r_2, \ldots, r_n\}$ where a particular route is defined by geographical coordinates $r_i = \{(\phi_1, \theta_1), (\phi_2, \theta_2), \ldots, (\phi_l, \theta_l)\}$ where first and last coordinates overlap both assembly point and a destination shelters.

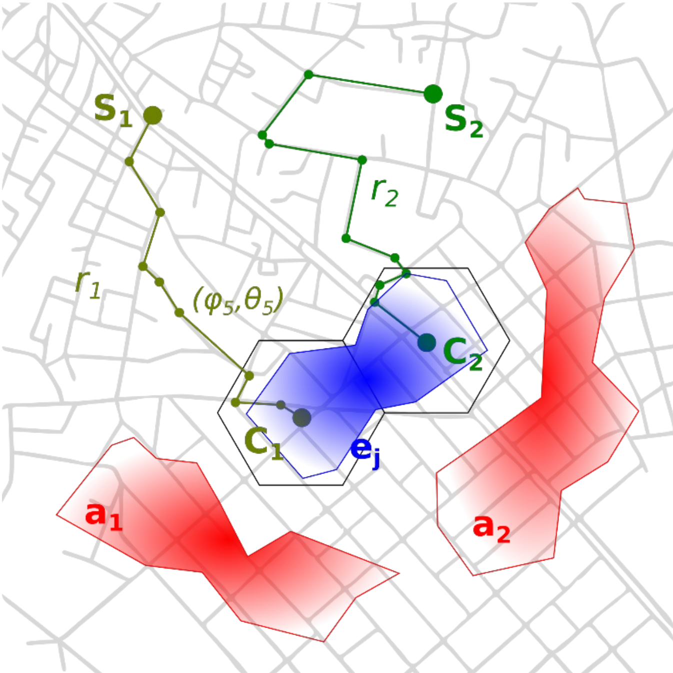


Figure 45 - Evacuation routes representation avoiding damaged areas.

2) Evacuation time estimation

Based on the routing model, the need arises to evaluate evacuation times, thus being able to determine the optimal route(s) among all alternatives. This also helps to detect potential conflicts between routes and whether the distribution of shelters and assembly points is appropriate. For that purpose, two independent models are developed.

Pedestrian model estimates the individual evacuation times from evacuee's location to nearest assembly points. Whereby a pedestrian (p) is modelled using three variables: 1) Initial geographic location (l_p), 2) Pre-movement time (t_{pre_p}) and 3) Walking speed (v_p). These variables are modelled following particular distributions (see Table 5) widely studied in the literature (Chandra & Bharti, 2013), (Purser & Bensilum, 2001) and (Lovreglio, Kuligowski, Gwynne, & Boyce, 2019).

Var	Distribution	Details
l_p	Uniform.	$\phi = u_1 \cdot \lvert\phi_{max} - \phi_{min}\rvert$ [Lng] $\theta = u_2 \cdot \lvert\theta_{max} - \theta_{min}\rvert$ [Lat] $L_2 - Evac.\ Polygon$ $\mathrm{l_p} = (\phi, \theta) \cap L_2$
v_p	Normal.	$\mathrm{r} = \sqrt{-2 \cdot Ln(U_1)} \cdot Sin(2\pi U_2)$ $v_p = \mu_v + \sigma_v \cdot r$
t_{pre_p}	Log-Normal	$\mathrm{r} = \sqrt{-2 \cdot Ln(U_1)} \cdot Sin(2\pi U_2)$ $t_{pre} = e^{\mu_{pre} + \sigma_{pre} \cdot r}$
$U_1 \sim UR(0,1), \quad U_2 \sim UR(0,1)$		

Table 5 - Pedestrian parameters estimation.

Through these variables it is feasible to estimate the evacuation time t_{r_p} for a particular evacuee as follows:

$$t_{r_p} = t_{pre_p} + \frac{\min_i\{Dist(l_p, C_i)\}}{v_p}$$

where function $Dist(l_p, C_i)$ is the minimum distance between evacuee location and assembly point C_i. This process follows a stochastic approach, simulating several sets of pedestrians to obtain average evacuation times. The number of pedestrians in the area is obtained using WorldPop (https://www.worldpop.org/) project data which uses a mixed model integrating census data and aerial imagery via satellite to estimate the population.

Vehicular model estimates for each route in R_i the traffic density, average speed and population evacuated in each temporal moment. This model uses the pedestrian individual evacuation time, vehicles occupancy and embarquement time as inputs to simulate via cellular automata the current traffic status of the routes and the simulated vehicles contained following a microscopic modelling approach (Azlan & Rohani, 2018). The routes calculated by the previous model are split in road sections with common characteristics. These road sections are produced solving the graph problem, erasing duplicated instances used by more than one route at the same time and calculating the vehicles interactions emanating from different routes that converge in the same road section and the distribution of vehicles when a road section is divided, see Figure 46.

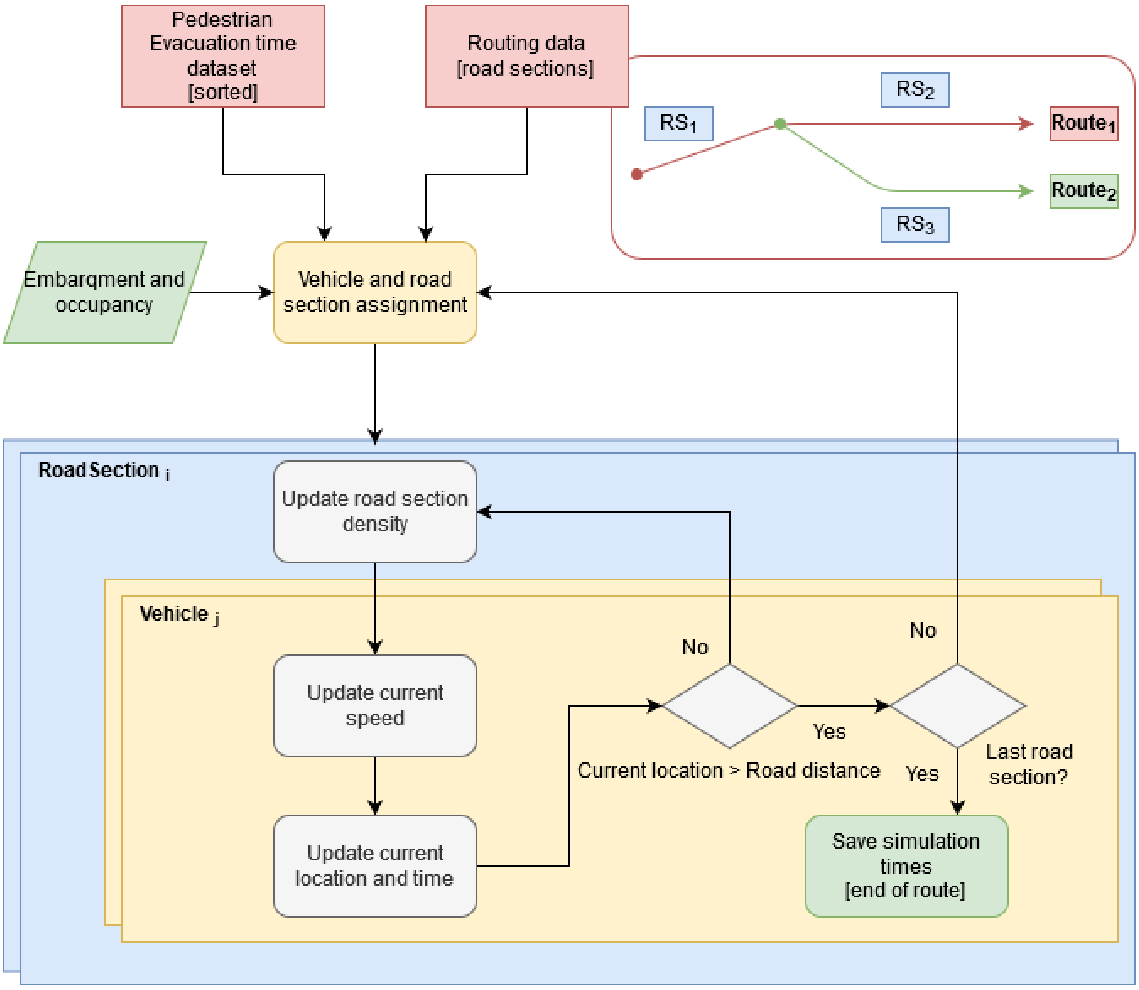


Figure 46 - Vehicular model flow chart.

Interaction between vehicles in the same road section is produced by three fundamental factors: 1) Traffic density k (current), k_c (critical) and k_j (jam), 2) Average speed u, u_c and u_j and 3) Route maximum density k_{max}. These variables follow a particular function previously studied (Notley, Bourne, & Taylor, 2009).

$$u = u_c - \frac{(u_c - u_j) \cdot \ln(\frac{\min(k, k_{max})}{k_c})}{\ln(\frac{k_j}{k_c})}$$

Bear in mind this relationship, two important constraints emerge that have to be addressed. First the lane changing problem in our particular case is solved using a random factor to increase or decrease the vehicle current speed allowing to pass between vehicles when the road conditions are favourable. Second, the gap acceptance is another factor to restrict the number of vehicles in a stretch solving intersections problems when the road capacity is exceeded. This problem was solved considering a queue of vehicles with time priority until exceed the road capacity moment in which current status of other vehicles in the queue is preserved emulating a traffic jam.

2.4.5. Requirements fulfilment

A total of 10 requirements were identified in D2.2 to be realized with regards to the Damaged Assets Location and Routing module.

Id. req	Description	Fulfilled	Specific actions
ALR_001	The tool will have a user-friendly, intuitive Graphical User Interface.	√	See section 2.4.2.1
ALR_002	The tool will allow users to input emergency parameters (type and location), evacuation areas and shelters (location and capacity) and damaged infrastructures (location, damage type and risks).	√	See section 2.4.2.1
ALR_003	The tool will have a GIS-based system.	√	See section 2.4.2.1
ALR_004	The tool will be able to calculate possible safe evacuation routes and safe access routes for emergency services to critical areas.	√	See section 2.4.2.2
ALR_005	The tool requires FRs status information (location, available units and type) to calculate dynamically safe routes.	√	See sections 2.4.2.2 and 2.4.3.2
ALR_006	The tool provides real time results.	√	See section 2.4.2.2
ALR_007	The tool will allow users to explore fictitious emergencies to develop previous plans.	√	
ALR_008	The tool will calculate approximate evacuation times using emergency particular parameters and historical demographic data.	√	See section 2.4.2.2
ALR_009	The tool will calculate in real time routes status and access times to the emergency points.	√	See section 2.4.2.2
ALR_011	The tool should allow changes in the scenario depending on the emergency time evolution.	√	See section 2.4.3.2

Table 6 - Damaged Assets Location and Routing requirements.

2.5. Portable SA platform

2.5.1. Overview

The portable SAP is another ASSISTANCE innovative module that will allow to increase the Situation Awareness of the FRs deployed on the field. This application has been adapted to run in mobile devices (e.g. rugged tablets) in order to give the FRs on field the possibility of accessing and using the majority of the general SAP features, which is used in the C2/emergency management room. On the other hand, in the portable SAP buttons are bigger than in the general SAP, since these buttons have to be used on field and maybe with gloves. This capability was a user requirement stated during the project meeting and have been developed accordingly (see next screenshots).

In addition, this mobile application automatically sends the GPS location provided by the tablet to the SAP in order to locate in the system the FRs who is managing this device.

2.5.2. Description of components

The different information exchanges performed by the portable SAP are described in Figure 47. There is an internal information exchange among the general SAP and all portable SAP devices (see blue links). Through this internal information exchange several data such as; internal messages, areas, point of interest, perimeters, etc) are replicated among the different SAP instances in order to provide the more updated information to all FRs and commanders that are using both the general SAP and the portable SAP.

On the other hand, in the same way that general SAP do, the portable SAP has its own interfaces for receiving information directly from the SAS (e.g. sensors information, UxV information, alarms, etc). (See red colour links)

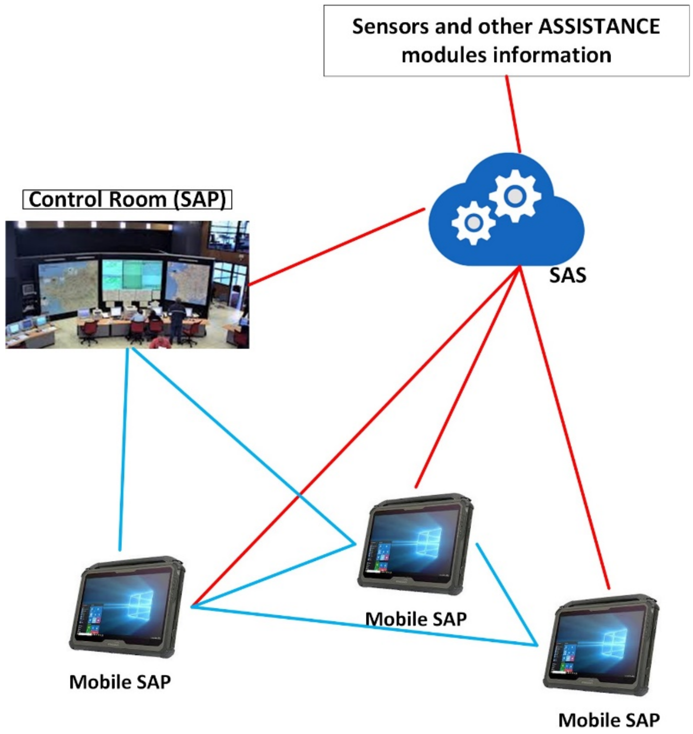


Figure 47 - Mobile SAP information exchanges

The portable SAP includes the following features and capabilities:

Adapted and customizable Human Machine Interface (HMI): In the same way that the general SAP, the portable SAP is capable of selecting the information that is shown in each moment through the HMI according to the needs of the FRs at this moment. It can be seen in the following figure:

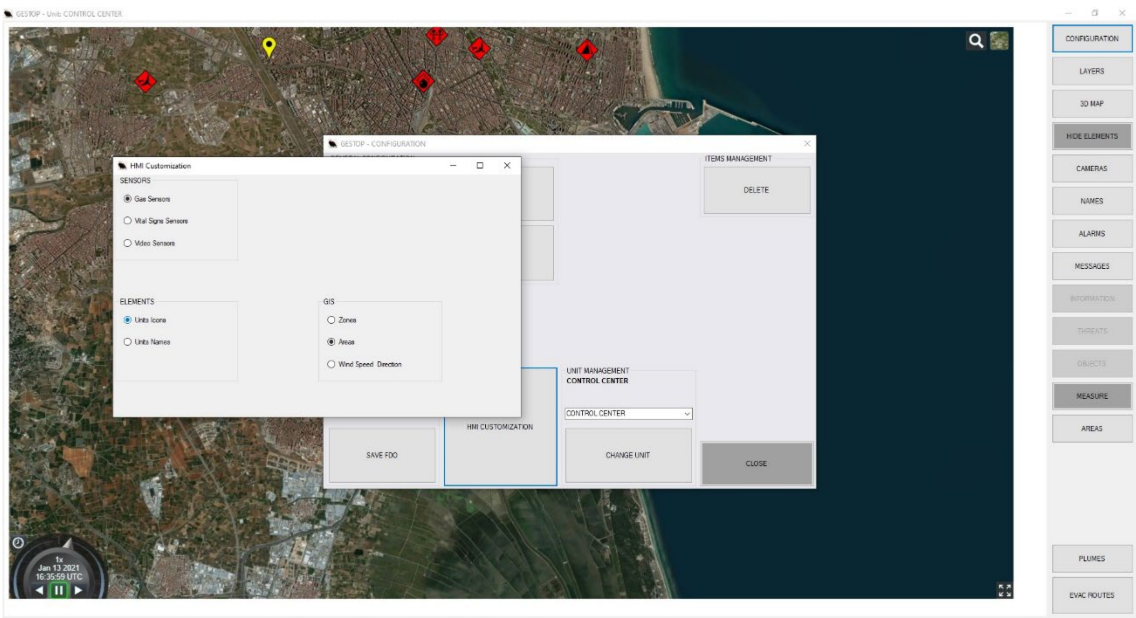


Figure 48 - HMI customization

This way FRs on the field can customize dynamically and filter the information they want to see in the HMI depending on the evolution of the situation.

In addition, the portable SAP HMI present some adaptations suggested by the FRs in order to be able to be used easily during the operations, for example, the menu buttons are larger than the general SAP buttons, since in some cases the mobile application could be used with gloves by the FRs.

Messaging: All SAP instances (general and portable) have their own internal messaging service that allows to send and receive text messages in an easy and secure way during the operations. In the following figures, some messaging capabilities are shown:

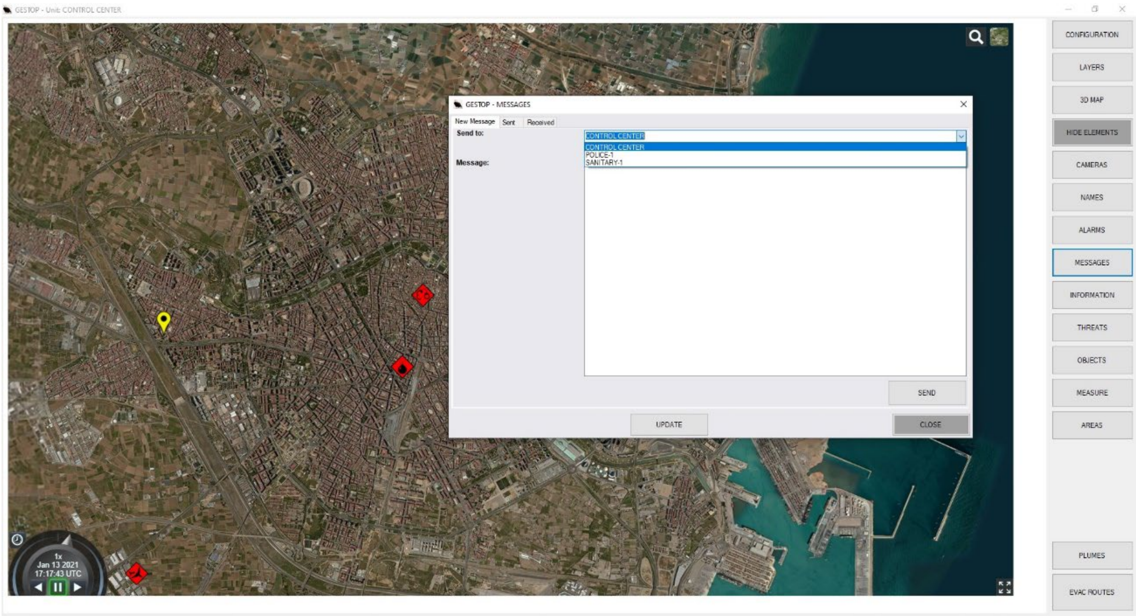


Figure 49 - Messages capture 1

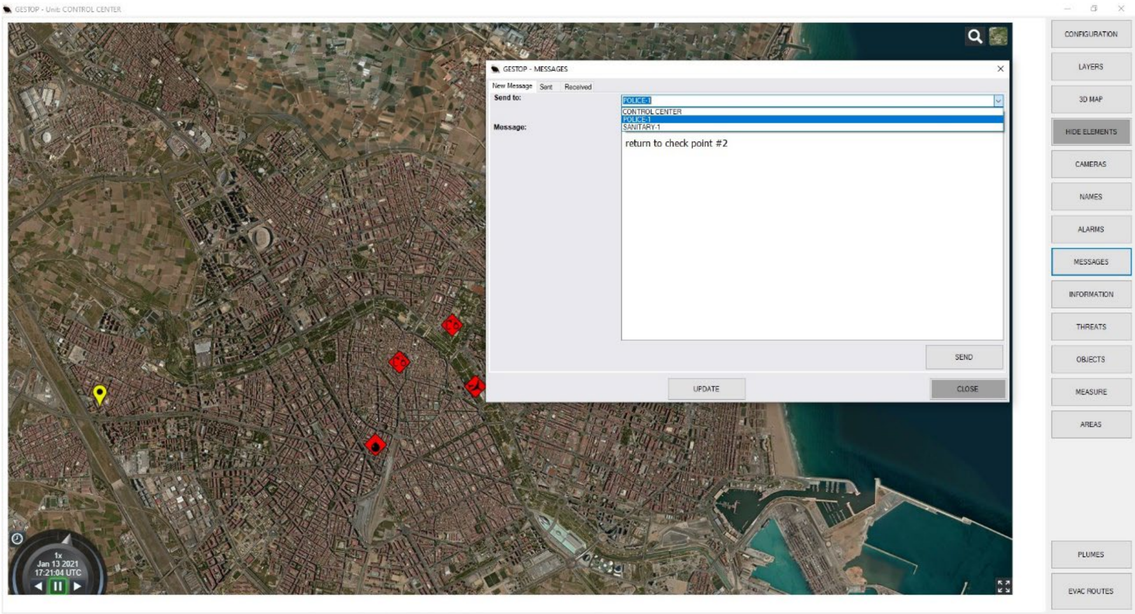


Figure 50 - Messages capture 2

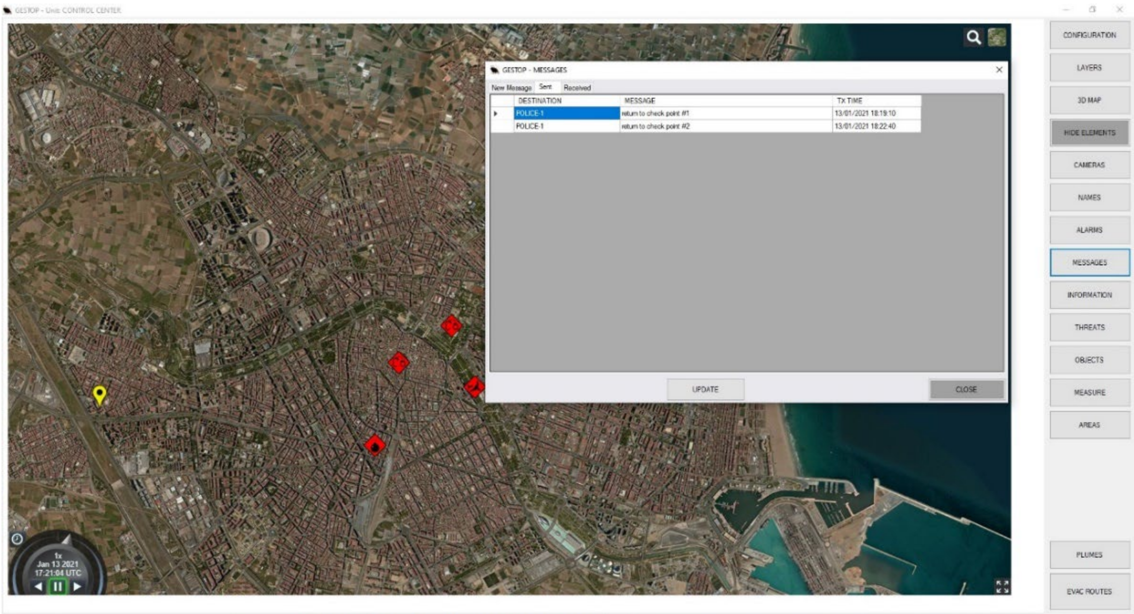


Figure 51 - Messages capture 3

Geo-tools: The mobile SAP has a complete set of Geo-tools. These capabilities allow to perform actions on the maps in an easy and quick manner in order to help FRs and their commanders during the planning or mitigation of a large disaster. These actions would be the following:

- Quick distance measure between two points on the map.
- Terrain 3D view

- Terrain profile between two points on the map.
- Synthetic perimeter representation

In the following figure, the user clicks into the map in two points for obtaining the existing distance measurement:

Figure 52 - Distance Measurement 1

Then, SAP system returns the measured distance:

Figure 53 - Distance Measurement 2

Also, the SAP system provides the capability to show the GIS and the data in 3D (with Digital Elevation Model included). It can be seen in the following figure:

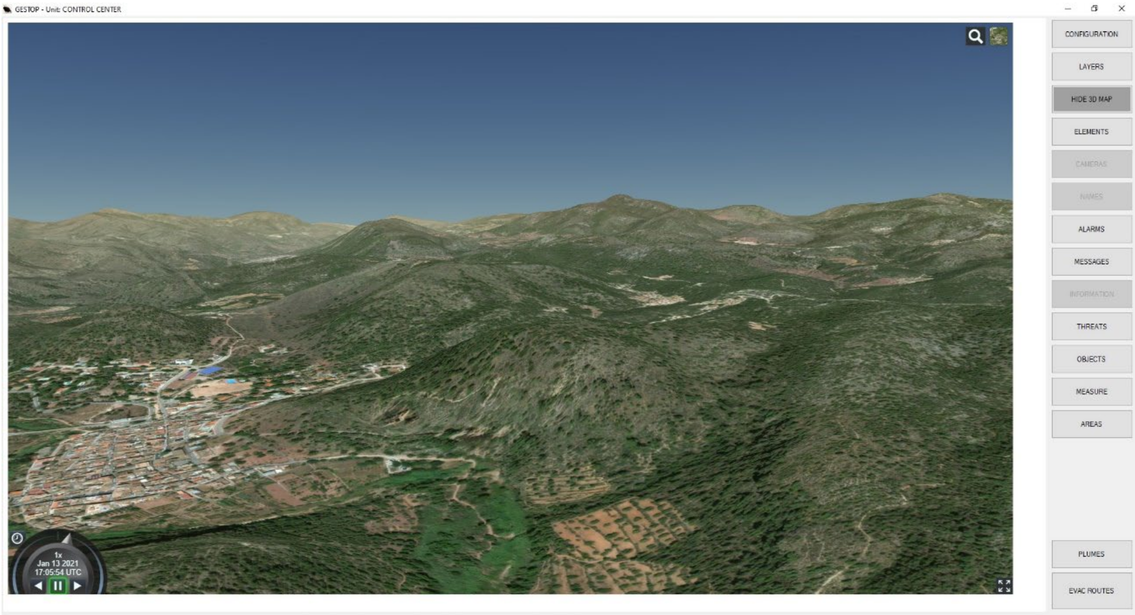


Figure 54 - 3D View

Regarding the perimeter visualization, it can be seen in the following figure:

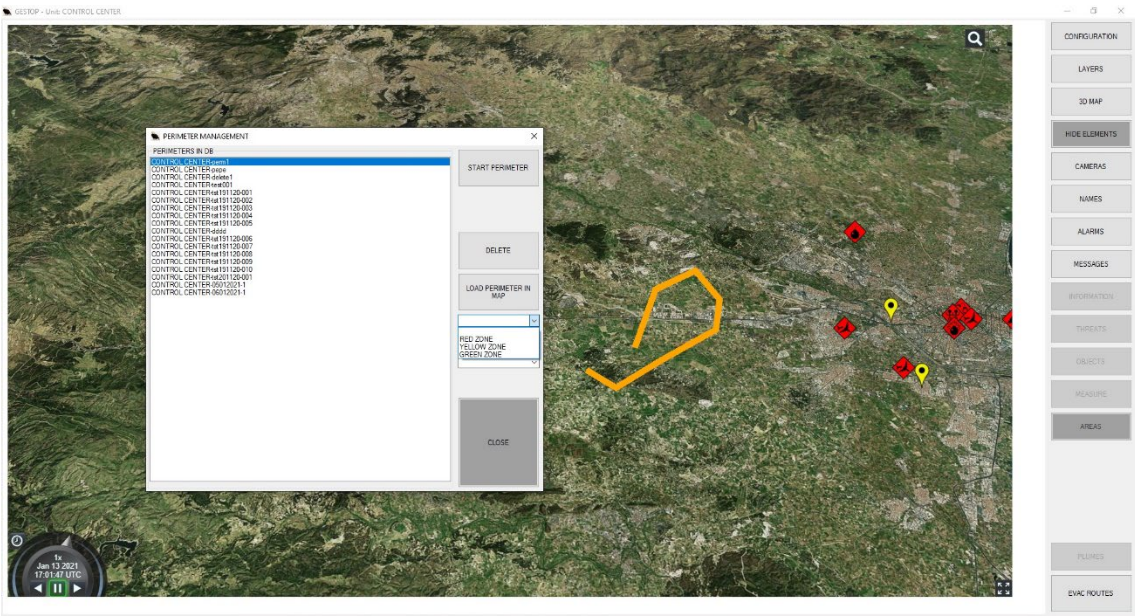


Figure 55 - Perimeter Visualization

Security & Privacy: The portable SAP follows the same security and privacy protocols that are used in the general SAP, which guarantees the security in the communications and the privacy of the data stored. Database data is encrypted and password accessed by SAP tool and all communications are done by means of TLS (Transport Layer Security).

CBRN Hazard Evolution results: The portable SAP can access anytime the results offered by the CBRN Hazard Evolution module and also receive alarms from this module if the FR location matches that of a dangerous area. In addition, the FRs can send through the portable SAP Points of Interest to the CBRN module, to be used in its calculations.

This way FRs on field can ask directly to the SAS for the last results of the CBRN module in order to increase their SA and also their safety during the operations.

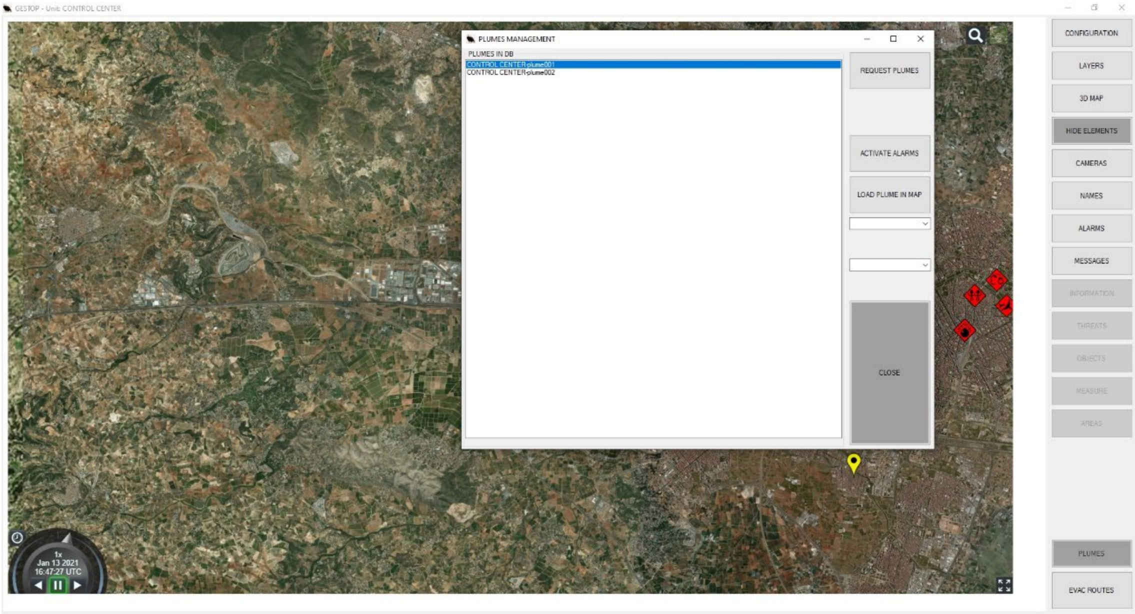


Figure 56 - Plume visualization from Hazard Module 1

Damage assets location & routing (DAL&R): The portable SAP has the possibility of requesting evacuation routes to the DALR module in each moment as the general SAP.

This fact increases also the FRs safety on the field since they can have in real-time evacuation routes from a determinate location or from an area.

If the user clicks in the evacuation routes button in order to see how to escape from the dangerous area, the following interface appears:

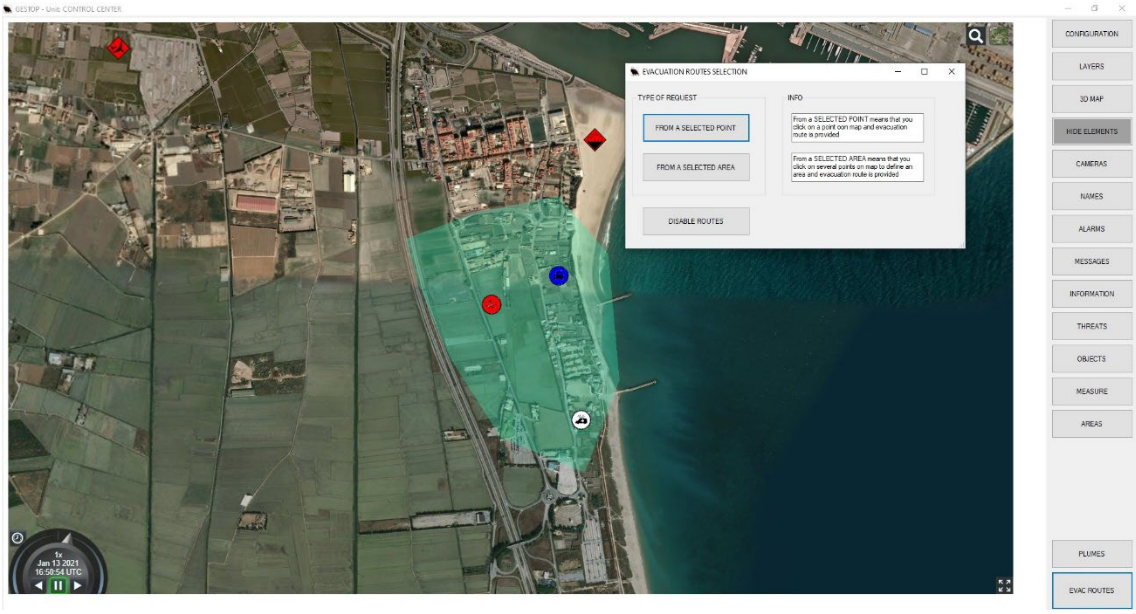


Figure 57 - Evacuation Routes 1

Then, the user can select the type of request for evacuation. The user can ask an evacuation route from a single point clicked (See yellow point) on the map or from a polygon and then the system shows the corresponding evacuation route:

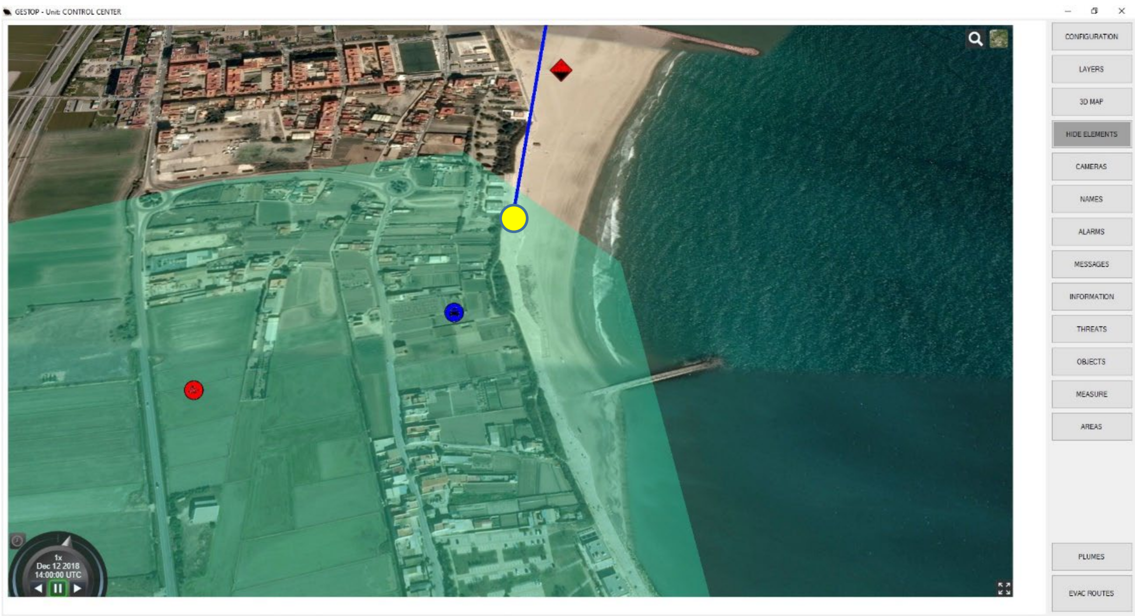


Figure 58 - Evacuation Routes 2

If the user zooms in can see a 3D model of an emergency vehicle:

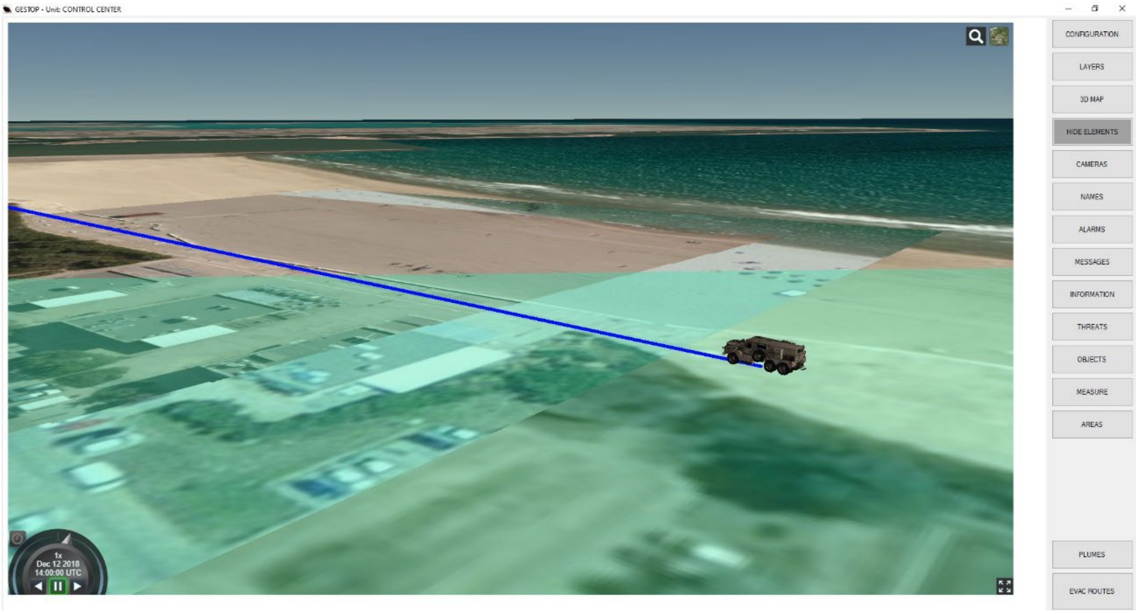


Figure 59 - Evacuation Routes 3

And then, if the user clicks in the clock in the GIS, an animation shows the vehicle moving along the evacuation route:

Figure 60 - Evacuation Routes 4

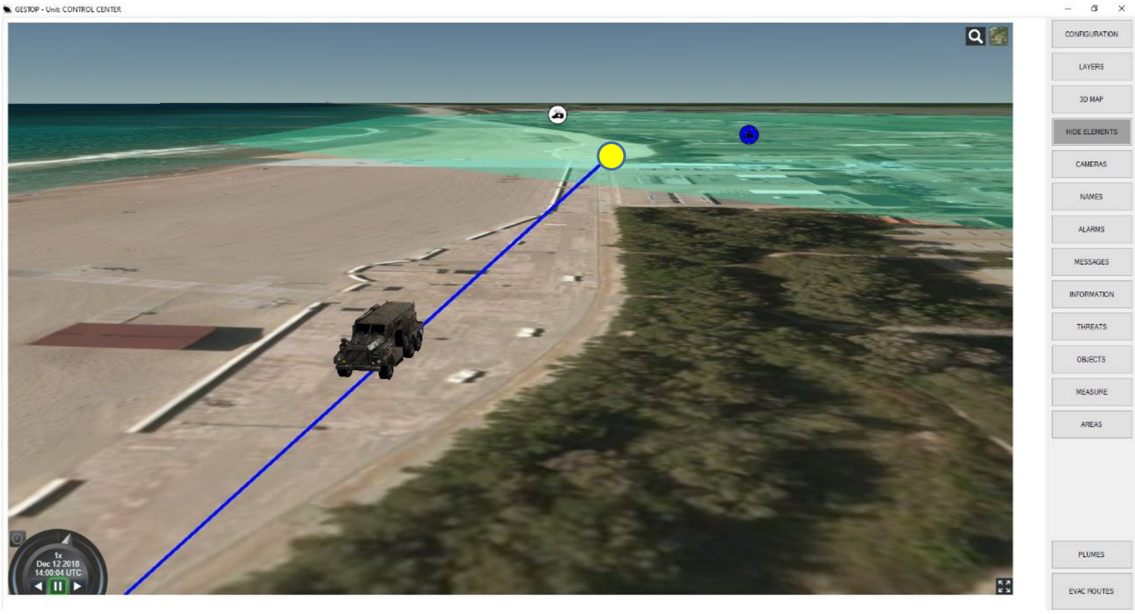


Figure 61 - Evacuation Routes 5

Sensors Mounted on Unmanned Platforms (UAVs, Drones and Robots): The portable SAP can receive automatically from the SAS the same information on sensors mounted on drones and robots that the general SAP.

This way the FRs on field have the same information that is available in the control room, which increases again their SA and also their safety. The sensors information that can be seen through the portable SAP are as follows:

- Video Cameras
- IR Cameras
- CBRN Sensors
- FRs Wearable Sensors
- GPS Sensors
- Personal Video Cameras
- Carbon monoxide detectors (CO)
- Vital Signs information

This information can be also tailored from the rugged tablet in the same way than in the general SAP. This feature is widely described in D5.1.

The screenshots below show the video flows visualization:

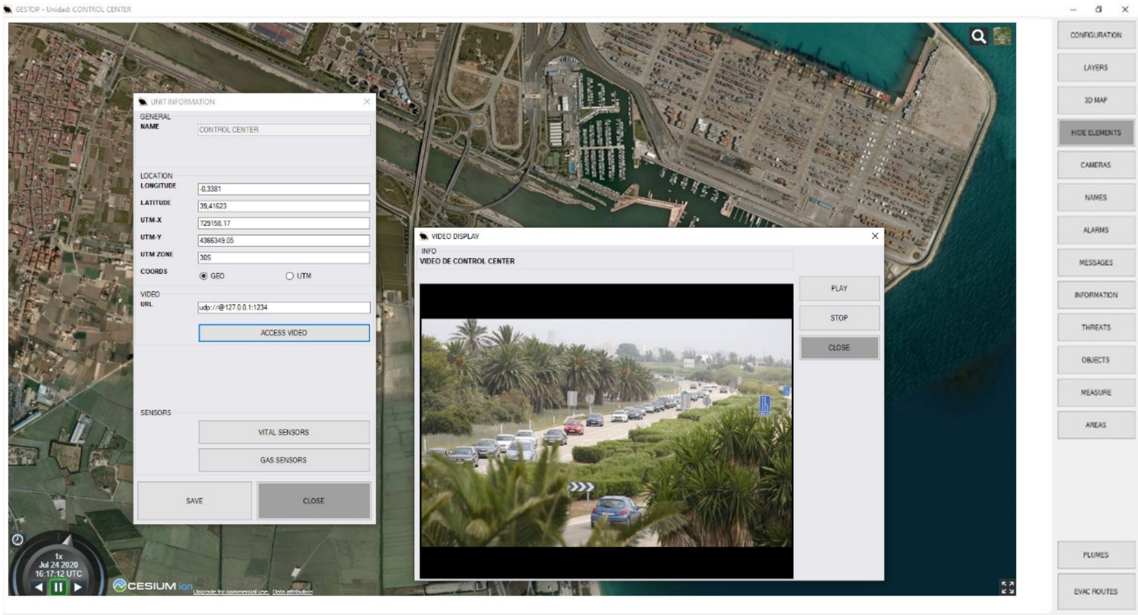


Figure 62 - Video Flow Visualization

The FR wearable sensors visualization:

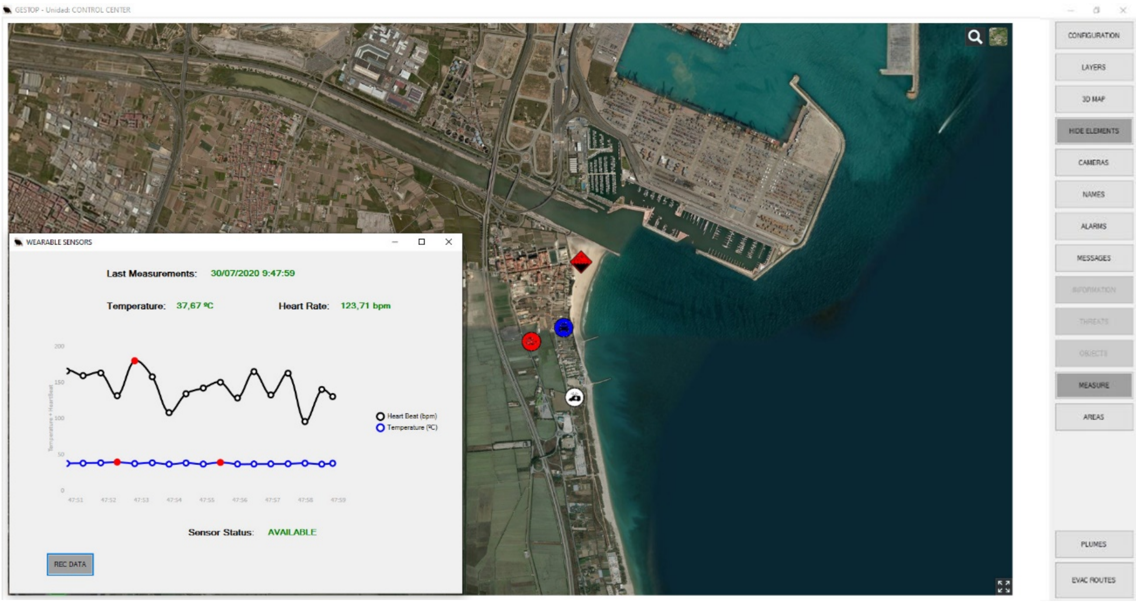


Figure 63 - Wearable Sensors Visualization

The GPS information is also received and shown in the mobile SAP HMI as shown in Figure 63.

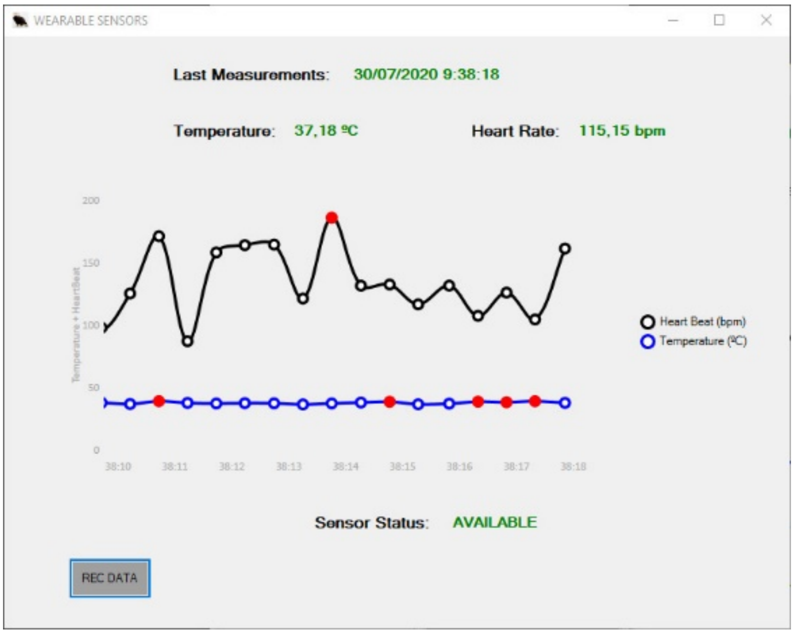


Figure 64 - Detailed Wearable Sensors Visualization

UAV measures visualization:

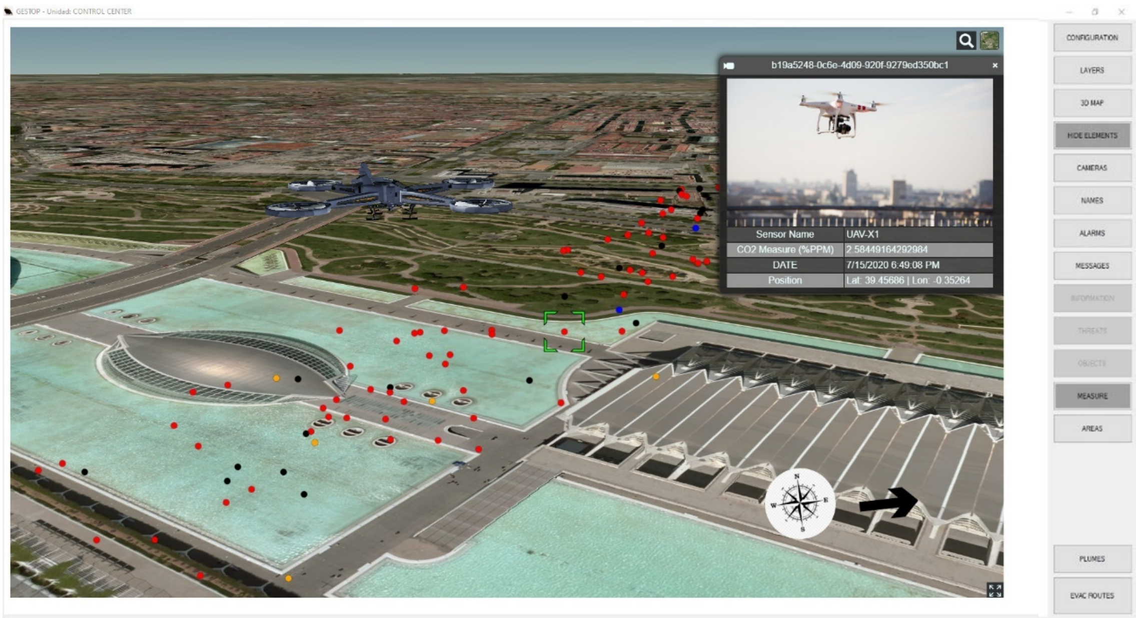


Figure 65 - Gas Sensor Visualization

2.5.3. Integration with SAP

The integration with the general SAP is native since basically both portable SAP and general SAP have the same core and share the same database, therefore they share many features and capabilities. The replication mechanism is also native and allows to replicate any information introduced in the database from any SAP instance (Point of interest, perimeter, alarm, etc) in the rest of SAP instances connected. In addition, the messaging capability used by the portable SAP is the same that the general one, so it is not necessary any integration for this feature.

The integration with the rest of the modules and sensors is performed through the SAS and using the same interfaces that the general SAP uses for receiving information from sensors or other ASSISTANCE modules.

2.5.4. Operational procedures

The portable SAP can be used on field by the FRs deployed, just by connecting the rugged tablet to the network used by the members of the different FR teams that are working together in the emergency mitigation.

Once the tablet is connected all information from the different mobile platforms, sensors and other modules is available in the portable SAP.

2.5.5. Requirements fulfilment

The following list states the requirements that were identified for the general SAP in D2.2 and that are also fulfilled by the portable SAP including some general requirements of ASSISTANCE system.

Id. req	Description	Fulfilled	Specific actions
999_001	ASSISTANCE should produce a complete physical situation awareness for the different FR organizations connected	√	This requirement encompasses the whole section 2.5
999_002	Access to ASSISTANCE system should be done by means of a secure authentication process	√	See section 2.5.2
999_005	ASSISTANCE system/applications should work in common COTS (Commercial off-the-shelf) hardware	√	Mobile SAP runs in a commercial rugged tablet.
SA_003	ASSISTANCE SA application HMI should provide discriminate information access depending on the FRs profile connected to the system	√	See section 2.5.2
SA_004	ASSISTANCE SA application should be executed in mobile devices (e.g. tablets) and adapt its performance to these devices.	√	Mobile SAP runs in a commercial rugged tablet.
SA_005	ASSISTANCE SA application should show real time video flows from the connected cameras (including	√	See section 2.5.2

	the ones mounted on mobile platforms) depending on the needs and restrictions, for instance bandwidth.		
SA_006	ASSISTANCE SA application should integrate IR cameras video flows (including IR cameras mounted on mobile platforms, if any) depending on the needs and restrictions, for instance bandwidth.	√	See section 2.5.2
SA_007	SA application should integrate the following wearable sensors for being installed on demand in some FRs uniforms depending on their protection needs. (GPS Sensors, Personal Video Cameras, Vital signs sensors)	√	See section 2.5.2
SA_009	ASSISTANCE SA application should allow messaging capabilities from/to any SA application node	√	See section 2.5.2
SA_010	ASSISTANCE SA application should give in real time and with high precision location of own resources (personnel and vehicles) including mobile platforms location (if available).	√	See section 2.5.2
SA_011	ASSISTANCE SA application should properly store all data received by the system from sensors and external sources in order to ensure the availability of all information stored in the database for being shown to the FRs where necessary.	√	See section 2.5.3
SA_012	ASSISTANCE SA application should shown near real-time evacuation routes for helping the FRs for moving the victims in a secure and quick way and for FRs evacuation of the area quickly in case of mayor incident.	√	See section 2.5.2
SA_013	ASSISTANCE should provide layers management of information capabilities on a GIS to foster the possibility to turn off or on information according to specific needs stated by the FRs.	√	See section 2.5.2
SA_017	ASSISTANCE mounted and wearable sensors data (e.g. toxicity measurements, etc) should be visible on the main SA application HMI and in each ASSISTANCE SA application node (including mobile devices e.g. tablets)	√	See section 2.5.2
SA_018	SA application HMI should allow map selection, distance measurements, zooming and scrolling	√	See section 2.5.2

SA_019	No SA application HMI action should require more than 4 clicks	√	The test performed with the SAP and mobile SAP ensure the accomplishment of this requirement.
SA_024	ASSISTANCE should be equipped with 3D mapping functions to provide terrain model information to raise situation awareness	√	See section 2.5.2

Table 7 - Portable SA platform requirements

3. Conclusions

The foundations of this deliverable are the comprehensive user requirements database derived from FRs real needs and the system architecture defined and agreed by the consortium. This background was used in each subtask in T5.2 and fed with each partner scientific and technical expertise to materialise the algorithms behind each SA module, keeping a strong collaborative approach in the process.

The information provided above shows novel tools and methodologies implemented to solve critical FRs needs, such as the Evacuation Model or the Dispersion Model in the CBRN module. An extensive work on Graphical User Interfaces has been also performed with the aim of maximising user-friendliness and efficacy during operations. Each requirement has been successfully implemented and preliminary tests have been performed during the development to verify each module reliability, opening the floor to more comprehensive testing during the project - e.g. T7.2 Integrated System Test Bed.

The software tools obtained from this work will become a subject of further research in subsequent tasks and beyond the project, being a key development to enhance FRs decision-making in emergency scenarios, and ultimately save lives.

References

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *Pattern Language: Towns, Buildings, Construction.* Berkeley, California: Center for Environmental Structure series: Oxford University Press, USA.

Azlan, N., & Rohani, M. (2018). Overview of application of traffic simulation model. *MATEC Web of Conferences, 150*.

Bashawri, A., Garrity, S., & Moodley, K. (2014). An overview of the design of disaster relief shelters. *Procedia Economics and Finance, 18*, 924-931.

Big Buck Bunny (2008). [Motion Picture]. Retrieved from www.bigbuckbunny.org

Boonmee, C., Arimura, M., & Asada, T. (2017). Facility location optimization model for emergency humanitarian logistics. *International Journal of Disaster Risk Reduction, 24*, 485-498.

Chandra, S., & Bharti, A. K. (2013). Speed distribution curves for pedestrians during walking and crossing. *Procedia-Social and Behavioral Sciences, 104*, 660-667.

Gravley, D. (2001). Risk, hazard, and disaster. *University of Canterbury, New Zealand*.

Huang, Y., Yin, Z., & Chu, H. (2019). Suitability Assessment of Emergency Shelters Based on GIS: A Case Study in Urban Function Optimization Area of Shanghai. *IOP Conference Series: Earth and Environmental Science, 234*(1), 1-7.

Kaisar, E., & Parr, S. (2012). Traffic Simulation Modeling for an Urban Chemical Disaster: Emergency Evacuation Development and Case Study. *Latin American & Caribbean Journal of Engineering Education, 6*(1).

Liu, L., Boone, A., Ruginsky, I., Padilla, L., Hegarty, M., Creem-Regehr, S., . . . House, D. (2017). Uncertainty Visualization by Representative. *IEEE Transactions on Visualization and Computer Graphics, 23*(9), 2165-2178.

Lovreglio, R., Kuligowski, E., Gwynne, S., & Boyce, K. (2019). A pre-evacuation database for use in egress simulations. *Fire safety journal, 105*, 107-128.

Montz, B. E., Tobin, G. A., & Hagelman, R. R. (2017). *Natural hazards: explanation and integration.* Guilford Publications.

Neerincx, M., van Diggelen, J., & van Breda, L. (2016). Interaction design patterns for adaptive human-agent-robot teamwork in high-risk domains. *International conference on engineering psychology and cognitive ergonomics* (pp. 211-220). Springer, Cham.

Notley, S. O., Bourne, N., & Taylor, N. B. (2009). *Speed, flow and density of motorway traffic.* Wokingham: Transport Research Laboratory.

Otten, J. (n.d.). Toxic fluid modelling. Unpublished research report.

Purser, D. A., & Bensilum, M. (2001). Quantification of behaviour for engineering design standards and escape time calculations. *Safety science, 38*(2), 157-182.

RIVM. (2020, December). *Intervention levels: Interventiewaarden voor de incidentbestrijding*. Retrieved from RIVM (Dutch Institute for Public Health and the Environment): https://rvs.rivm.nl/sites/default/files/2021-01/Interventiewaarden_ed_2020-1_alfabet.pdf

Ruginsky, I., Boone, A., Padilla, L., Liu, L., Heydari, N., Kramer, H., . . . Creem-Regehr, S. (2016). Non-expert interpretations of hurricane forecast. *Spatial Cognition & Computation, 16*(2), 154-172. doi:10.1080/13875868.2015.1137577

Schneiderbauer, S., & Ehrlich, D. (2004). Risk, hazard and people's vulnerability to natural hazards. *A review of definitions, concepts and data. European Commission Joint Research Centre. EUR, 21410*, 40.

Shelter-Centre. (2012). *Transitional shelter guidelines.* International Organization for Migration.

Smet, C. W. (n.d.). *Estimating characteristics of toxic spills using machine learning.* University of Applied Sciences Utrecht. Unpublished research report.

Ticheler, A. (2020, July). Non-expert risk assessment using gas cloud visualizations, and the influence of user characteristics. MA thesis. Department of Psychology, Utrecht University.

Van der Waa, J., van Diggelen, J., Siebert, L. C., Neerincx, M., & Jonker, C. (2020). Allocation of Moral Decision-Making in Human-Agent Teams : A Pattern Approach. *Springer Nature Switzerland AG 2020, 2*, 203–220.

Van Diggelen, J., & Johnson, M. (2019). Team design patterns. *HAI 2019 - Proceedings of the 7th International Conference on Human-Agent Interaction*, 118–126.

Van Diggelen, J., Neerincx, M., Peeters, M., & Schraagen, J. (2019). Developing Effective and Resilient Human-Agent Teamwork Using Team Design Patterns. *IEEE Intelligent Systems*, 34(2), 15–24.

Van Welie, M., Van Der Veer, G. C., & Eliëns, A. (2001). Patterns as tools for user interface design. In C. Farenc, & J. Vanderdonckt, *Tools for Working with Guidelines* (pp. 313–324). Eds. Springer.

Venrooij, W. (n.d.). Non-expert risk assessment using gas cloud visualizations. Unpublished draft.

Yellow Book. (2005). *Methods for the calculation of physical effects CPR 14E (Publicatiereeks gevaarlijke stoffen 2)* (Vols. Third edition, Second revised print). (C. van den Bosch, & R. Weterings, Eds.)