Rapid Prototyping of IESTA, a Platform to Evaluate Innovative Air Transport Concepts

Martin Adelantado[1] and Armand Oyzel[2]
The French Aerospace Lab (ONERA), 2 Av. E. Belin, 31055 Toulouse, France

Jean-Baptiste Chaudron[3]
The French Aerospace Lab (ONERA), 2 Av. E. Belin, 31055 Toulouse, France

and

Thomas Rivière[4]
The French Aerospace Lab (ONERA), 2 Av. E. Belin, 31055 Toulouse, France

The French Aerospace Laboratory (ONERA) is carrying out an ambitious program (IESTA) to design a modeling and simulation infrastructure able of evaluating innovative air transport concepts. This paper describes a first prototype of the IESTA platform built around distributed simulation standards, acoustic modeling and earth browsers such as Google Earth.

I. Introduction

THE European Airspace is becoming more and more congested as traffic is forecast to grow steadily over the next 20 years and beyond. To meet the challenge of sustainable growth of Air Transport Management in Europe, an Advisory Council for Aeronautics Research in Europe (ACARE) was launched in June 2001. ACARE's main focus is to establish and carry forward a Strategic Research Agenda (SRA 1/2) that will influence all European stakeholders in the planning of research programs, particularly national and EU programs, in line with the Vision 2020 and the goals it identifies. Among these programs, SESAR (Single European Sky ATM Research) and its various phases (from Definition to Development and ultimately Deployment) has the ambition to face the lack of commitment from a part of the Stakeholders or from deciders in previous initiatives. The Definition Phase, jointly funded by EUROCONTROL and the European Commission, will deliver a European ATM Master Plan based on future aviation requirements, and will identify the actions needed to achieve the objectives of SESAR. At the European level, other programs are strongly linked with SESAR, including the Joint Technology Initiatives (JTI) a new instrument provided by the European Seventh Framework Programme, to support research of long duration. One of those initiatives, called CLEAN SKY, will create various technology demonstrators, including flight test vehicles that will be essential for successful market introduction.

Research into the future of the air transportation system critically depends upon the ability to evaluate the effects of the revolutionary candidate concepts across the entire air transportation system. The testing of these concepts through modeling and simulation can provide useful information on improvements on system capacity/safety and related environmental impact. In this evolutionary European context, the French Aerospace Laboratory (ONERA) is carrying out an ambitious program (IESTA) to design a modeling and simulation infrastructure able of evaluating innovative air transport concepts. This platform has to be considered as a mean to help harmonization of pan-European assessment tools for innovative research on future Air Transport Systems (ATS).

This paper firstly introduces organizational issues of the IESTA Program focusing on expected benefits of the underlying approach from the customer's point of view. We then discuss system architectural issues of the platform as well as technological features. Finally, we give a detailed presentation of a rapid prototype of the first version

[1] Senior Researcher, Long Term Research and Systems Integration Department, Martin.Adelantado@onera.fr.
[2] Senior Researcher, Long Term Research and Systems Integration Department, Armand.Oyzel@onera.fr.
[3] Research Engineer, Long Term Research and Systems Integration Department, Jean-Baptiste.Chaudron@onera.fr.
[4] Research Engineer, Long Term Research and Systems Integration Department, Thomas.Riviere@onera.fr.

(application) of IESTA, called Clean Airport, aiming at assessing both noise and chemical impact during the Landing and Take-Off (LTO) phases on airports vicinity. This prototype has been design to evaluate the relevance and efficiency of a set of well known existing tools to achieve the requirements of IESTA/Clean Airport platform, focusing on noise propagation and impact. Tools used include the ONERA CERTI HLA Run-time Infrastructure[5], X-Plane simulation outputs[6], Google Earth and related tools[7].

II. The IESTA Program

The ambition of the IESTA Program is to offer an infrastructure for evaluating future concepts for air transport systems through an advanced formalization procedure. This program then participates in the world wide effort to face the challenge of sustainable development and to achieve ambitious goals, particularly with regard to security, environment and capacity. IESTA unifies all the aeronautical skills of ONERA in physics modeling and its know-how in terms of system evaluation and simulation architecture to provide a unique service for the benefit of players in the air transport industry who are interested to evaluate the performance and impact of their innovations in a more comprehensive way.

IESTA is a 5 years Program built around 2 main projects, IESTA/AMO the responsibility of which is to deliver a set of computational modules, and IESTA/IMI aiming at integrating those modules within a given computational platform. Computational modules include noise propagation, chemical species emission and propagation, engines burning, aircraft dynamics, aircraft planning and maneuvering on the airport layout, en route flight plans etc. A set of long term thematic studies are also carried out in parallel in order to model system components and evaluation methodologies through internal and external co operations. Two applications of IESTA have been selected in a previous specification phase, each one addressing a set of relevant study cases:

1) Clean Airport (IESTA.V1 version), a set of computing facilities and models to assess the airport impact on environment in terms of noise and chemical emissions.
2) SimSky (IESTA.V2 version) addressing evaluation of innovative ATS concepts across the entire airspace.

A first version of IESTA/Clean Airport will be delivered by June 2009, while IESTA/SimSky version is expected to be available by 2011. The IESTA Program should lead to the availability of a unique platform benefiting from the ONERA aerospace research environment, a large scale computing power and the multidisciplinary teams of engineers and scientists. Exploitations plans of the platform are very similar to those of ONERA wind tunnel facilities. Customers (civil aviation authorities, aircraft industries, airlines, airport management companies etc), bring their own models and concepts to evaluate, and ONERA provides the infrastructure, the simulations facilities and any further model if necessary. The customer property on data, models and simulation results is contractually preserved.

III. System Architecture Design and Technological Issues

A. Reuse and Interoperability through Distributed Simulation

The main goal of the IESTA Program is to provide a set of open, reusable, and interoperable modeling and simulation facilities to evaluate future ATM concepts and equipment. To achieve this goal, the High Level Architecture (HLA), an IEEE standard for distributed simulation (IEEE 1516) has been selected to support interoperable simulations[1, 2, 3, 4]. Motivations for HLA are the following:

1) First of all, HLA offers a mean to increase the complexity of the modeled system by introducing additional components that can communicate through the RTI (Run-time Infrastructure), achieving then flexibility and scalability of the simulated system. Therefore, it is not necessary, in a preliminary step of design, to incorporate all the components of the simulated system. This issue is of primary importance, since Air Transport Systems are very large, dynamic and complex systems with many interacting traffic modes and numerous services that have to share the same resources
2) Secondly, HLA really offers a mean to face well-known site-proprietary constraints and to easily reuse existing models and simulations to design simulations of increasing complexity and coverage.

[5] http://www.cert.fr/CERTI/

[6] X-Plane by Laminar Research, www.x-plane.com/

[7] Google Earth is a registered trademark of Google Inc, http://earth.google.com/

3) Thirdly, HLA faces with the lack of design methodologies in previous distributed simulation standards such as Distributed Interactive Simulation (DIS). It provides a Federation Development and Execution Process (FEDEP), standardized by IEEE since 2003 through a set of recommended practices[5].

Several IESTA simulation technological solutions are very close to those adopted by the Virtual Airspace Simulation Technology Real-Time (VAST-RT) Project at NASA Ames Research Center, part of the Virtual Airspace Modeling and Simulation (VAMS) Project[6, 7]. Furthermore, the adoption of HLA for IESTA benefits from the experience gained by ONERA from the "*Airport of the Future*" previous Project[8, 9, 10].

B. IESTA Platform Architecture Design

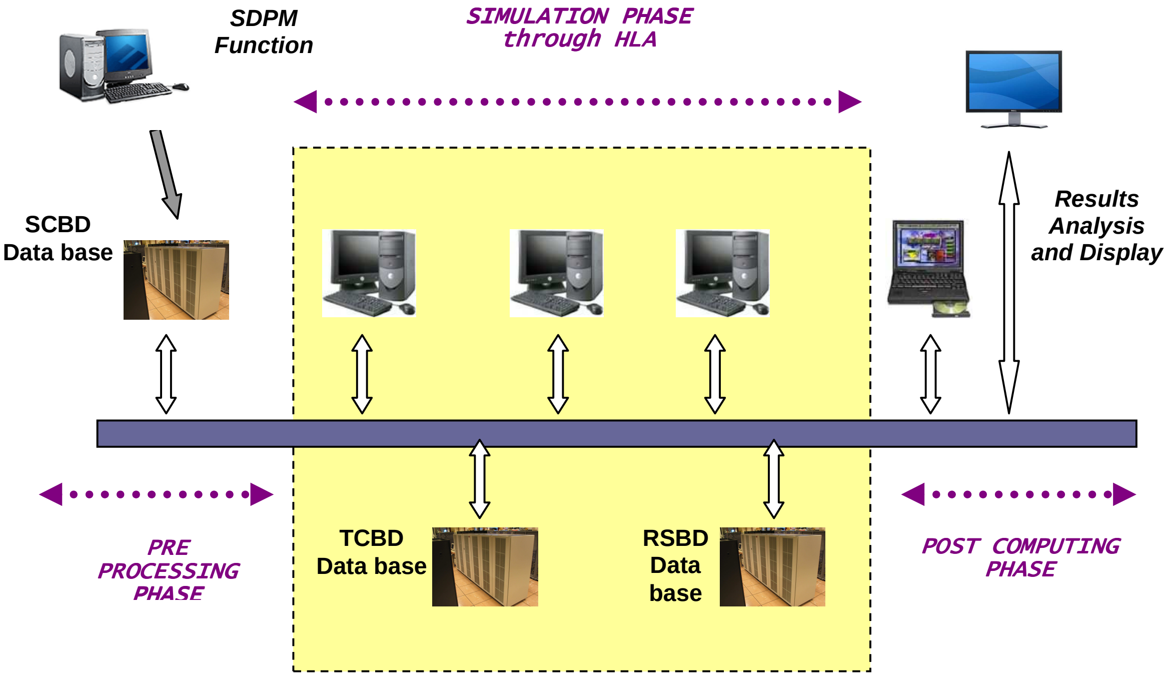


Figure 1. IESTA computational platform.

The IESTA platform architecture, depicted in figure 1, is built around a set of computing resources, software and data bases devoted to a set of given functions and phases. In the following, these functions and components are briefly described:

1) The Scenario Development and Platform Management (SDPM) function offers a set of tools to design the simulation scenario (airport selection, aircraft schedule, flights plans, fleet characteristics, etc). Other software are intended to handle the simulation phase (federates mapping, HLA federation initialization, run and completion, replay, results files handling, etc).
2) The HLA simulation phase consists of executing a given federation built around a set of participating federates using the Scenario Data base (SCBD) and the Technological Data base (TCBD). The TCBD contains all the physical data regarding aircraft characteristics, engine performances, weather, etc). Finally, simulation results are saved into the Results Data base (RSBD).
3) The last phase is devoted to results analysis and visualization using a set of plotting and gridding software tools for computing noise and chemical species contours. High resolution LCD screens are devoted to simulation visualization and replay.

IV. Rapid Prototyping of IESTA/Clean Airport Platform

A. Introduction

As stated above, IESTA/Clean Airport will be the first version of the IESTA Platform aiming at assessing noise and chemical impact of aircraft around airport's vicinity. The added value of this initiative versus well known tools as the Integrated Noise Model (INM) from FAA[8] and the Emissions and Dispersion Modeling System (EDMS)[9], deals with intensive use of the ONERA skills regarding noise propagation and chemical species dispersion modeling in the aerospace domain.

This prototype has been designed to prepare the migration to HLA of ONERA existing models written in C language, Fortran or Matlab. Only a preliminary version of the acoustic propagation model has been integrated within the prototype, since the expected chemical emission and propagation models are not mature enough. An additional motivation in designing this IESTA prototype was to evaluate the ability of efficient tools such as Google Earth to display some results either during the simulation phase or during the post computing phase.

The prototype, written in C++ language, is built with the free cross-platform *Code::Blocks* IDE[10] and runs on a PC under the Windows XP operating system.

B. Scenario Generation and Simulation Steps

1. *Organization of the HLA federation*

The simulation prototype is an HLA federation consisting of only 2 federates:

1) An Aircraft Federate (AC_F) federate publishing the aircraft state vector (latitude, longitude, altitude, pitch, roll, heading, velocity, etc).
2) An Acoustic Federate (ACOU_F) calculating the noise contours of a set of aircrafts around airport vicinity.

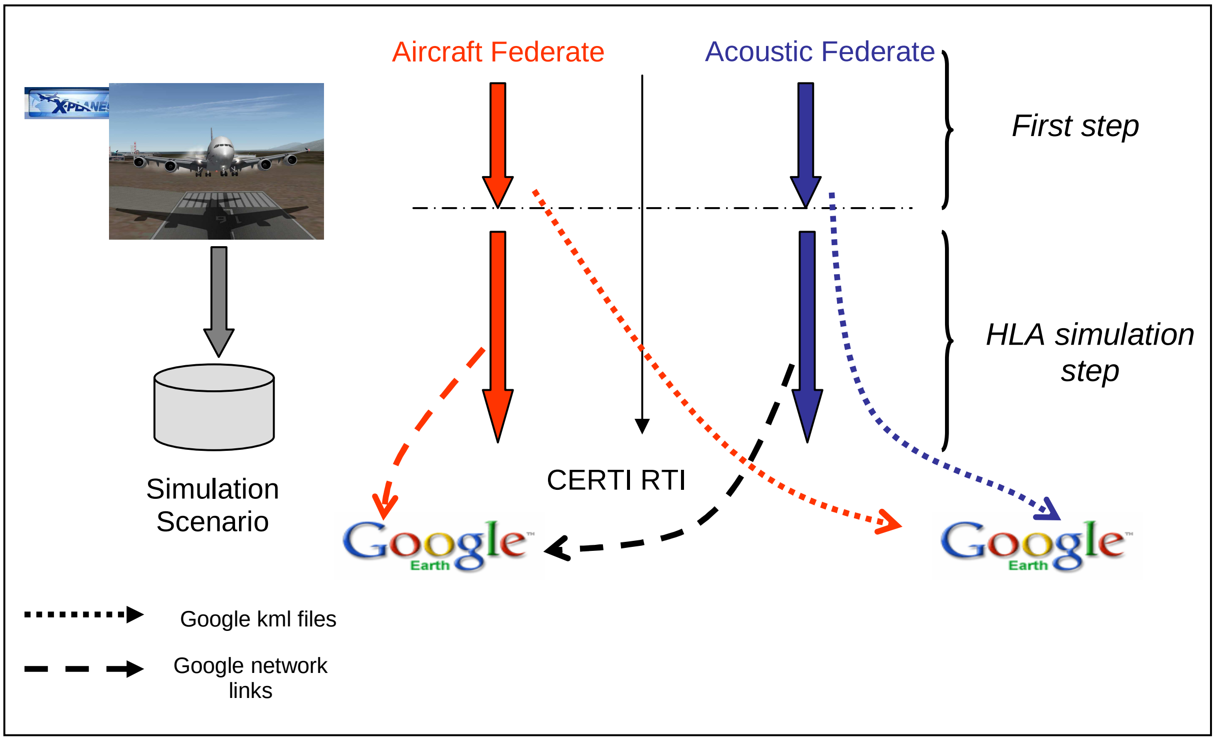


Figure 2. Scenario generation and simulation steps.

The HLA Run-time Infrastructure used is an open source RTI designed by ONERA and called CERTI, freely available for download from the Web site[11]. This RTI is US DoD 1.3 standard compliant, and is currently upgraded

[8] http://www.faa.gov/about/office_org/headquarters_offices/aep/models/inm_model/

[9] http://www.faa.gov/about/office_org/headquarters_offices/aep/models/edms_model/

[10] www.codeblocks.org

[11] http://www.cert.fr/CERTI/

to the IEEE 1516 HLA standard. Notice that the complete IESTA/Clean Airport platform will include additional federates such as ground planning/guidance models and engine burning models. These models are currently in a design state. Figure 2 provides a graphical view of both scenario generation and simulation phases.

2. Scenario generation

The first step consists of designing a simulation scenario. Currently we are using the capabilities of X-Plane flight simulator[12]. In the IESTA operational platform, ONERA custom models will be used to generate trajectories from flight plans. Those models are described in the following section.

In the prototyping phase of IESTA, a simulation scenario is a set of trajectories provided before the simulation phase using X-Plane. Each trajectory consists of a set of parameters (aircraft vector state) including aircraft type, longitude, latitude, altitude, heading, pith, roll, speed, engine status and other parameters required by the acoustic model. Refer to section VI for a detailed explanation of the noise modeling. Each aircraft vector state is a set of time stamped measures, the acquisition rate being set by the X-Plane "pilot" before simulation (a rate of 1 measure every 2 sec has been selected). Flight plans can be introduced within the FMS of the simulated aircraft in two ways: in the first one, the "pilot" introduces himself the flight plan by selecting a set of waypoints (fix, VOR, NDB, runway ILS) and providing a given speed, altitude and climb/descent rate; another alternative is to use flight planners dedicated to X-Plane simulator, such as *Goodway*[13] or the *Navigation* tool of François Fouchet[14]. For example, the *Goodway* tool combines a powerful flight planner using all of the navaids and the airports of X-Plane, and an X-Plane plug-in allowing the "pilot" to automatically set up the flight plan into the FMS.

Generation of an X-Plane trajectory from a given flight plan needs an additional free tool called Xplage developed by Chris Kern[15]. Xplage is a software utility that listens on an IP network for UDP datagrams representing the geographical coordinates of the simulated flight path of an instance of the X-Plane flight simulator, and then transforms them into a series of continuously updated KML files that Google Earth can ingest to create "moving map" displays. Section VII provides a more in depth explanation of both KML language and Google Earth display mechanisms.

The main routine of the Xplage utility launches three asynchronous lightweight child processes which respectively wait for an incoming datagram from the simulator, the end of the specified interval for updating the KML output files, or a manual interrupt from the user. When the user issues a request to the interrupt thread to exit the program, all four processes are simultaneously terminated. The primary goal of the Xplage utility is to repeatedly generate two different KML files at a refresh interval specified by the user, allowing displaying two different views of the simulated aircraft through Google Earth. The first file is used by Google Earth to display an overhead map with North at the top and an icon in the center of the satellite image to indicate the position of the simulated aircraft. In this view, the Google Earth "eye" is fixed directly above the airplane. Each refresh cycle shifts the map in the direction of the flight path so as to keep the icon centered. The second KML file is used by Google Earth to display a perspective view of the terrain ahead of the simulated aircraft. In this view, the eye may be thought of as being right in front of the airplane and pointing slightly downward. Each refresh cycle repaints the perspective based on the simulated aircraft's roll and pitch orientation.

Preparing the scenario uses an option of the Xplage command allowing the deposit of a file containing comma-separated values representing the flight track, in addition to the two KML files.

Once a set of trajectories have been generated using X-Plane flight simulator, an additional scenario file is created containing the simulation starting time and the number of aircraft participating in the simulation. The trajectories files generated by both X-Plane flight simulator and Xplage utility are enriched by the simulation starting time of the corresponding aircraft and a string identifying the aircraft path. Simulation starting time of each participating aircraft is selected in such a way that the simulation scenario is realistic given the infrastructure and performances of the target airport.

Recently a "real time" approach for generating the aircraft trajectories has been proposed by Jean-Michel Mathé[16] from ONERA. This approach avoids generating the aircraft trajectories before the simulation phase. It is supported by the design of an X-Plane plug-in linked with the X-Plane API and the RTI CERTI libraries. This X-Plane plug-in automatically launches an HLA federate when the flight simulator is started. In a similar way as Xplage, the process waits for incoming datagrams from the simulator and published them to any federation. In that way trajectories generation can be performed in real time launching several instances of the flight simulator

[12] X-Plane by Laminar Research, www.x-plane.com/

[13] http://www.xpgoodway.com/

[14] http://francois.fouchet.free.fr/Navigation/

[15] http://www.chriskern.net/code/xplaneToGoogleEarth.html

[16] Jean-Michel.Mathe@onera.fr

connected with an Ethernet network. A receiving federate subscribing to the vector state attributes of the aircraft class (in the HLA sense), can then receive the vector state of each aircraft participating in the federation and immediately invoke the physical model of noise propagation.

3. *Simulation phases*

Once a scenario has been created, the simulation phase proceeds in two steps (figure 2). During the first step, the Aircraft Federate (AC_F) creates a KML file for each participating aircraft, allowing Google Earth to display the trajectory of the aircraft and its starting location as well. Obviously, the user can seamlessly fly through the landscape with Google Earth navigation controls. Alternatively, the Acoustic federate (ACOU_F) creates a KML file to display the reference grid around the airport. Noise impact from each aircraft as well as from all the participating aircraft will be further displayed at each point of the reference grid.

Once the initial step has completed, each federate enters the main simulation loop. The Aircraft federate proceeds according to the following algorithm:

```
 For each simulation time step do
      Activate waiting aircraft if their departure time is less than the current simulation time
      For each activated aircraft do
         Read the corresponding vector state from the file provided by X-Plane and Xplage
         Send the current vector state of the aircraft
         Create a KML Network Link file to display an overhead view through Google Earth
         Create a KML Network Link file to display a perspective view through Google Earth
      End for
      Inactivate activated aircraft meeting their arrival time
End for
```

Since the High Level Architecture is used, the Aircraft Federate obviously publishes the vector state attributes of the aircraft class. Sending vector state attributes involves invocation of the RTI UpdateAttributesValues method. The time step is equal to the acquisition rate selected by the user before each X-Plane simulation. The first generated Network Link allows displaying an overhead map with an icon in the center of the satellite image to indicate the position of the simulated aircraft. In this view, the Google Earth “eye” is fixed directly above the airplane and follows the aircraft heading. The second KML Network Link is used by Google Earth to display a perspective view of the terrain ahead of the simulated aircraft.

The algorithm of the Acoustic Federate is straightforward. It subscribes to the vector state attributes of the aircraft class. Under reception of a set of state vectors corresponding to the simulated aircraft, it constructs the data structures needed by the noise model and immediately invokes the C++ method.

The federate generates then several KML Network Links to display the current noise impact of the aircraft on the reference grid. One noise intensity matrix is created for each simulated aircraft as well as a global contribution matrix corresponding to all the aircraft currently in an active state. Notice that several snapshots of the displaying facilities offered by Google Earth will be described in section VII.

Regarding the time management policy of the HLA federation, the Aircraft Federate is regulating and the Acoustic Federate is declared to be constrained. Indeed, the role of the Aircraft Federate is to synchronize its advancement according to the refresh rate of the X-Plane flight simulator in order to achieve a real time rendering of Google Earth.

Finally, the simulation is initiated in the following way:

1) Launching the CERTI HLA Run Time Infrastructure.
2) Creating the federation by starting both federates, Aircraft federate and Acoustic Federate.
3) A first synchronization point allows the simulation to wait for launching Google Earth browser.
4) A second HLA synchronization point waits for the user to type Enter before starting the HLA federation execution.

Once Google Earth is launched, the user has to open all KML files and network links and to store them in the places panel of the browser. Each place can then be enabled or disabled according to the user needs regarding the visualization of the scenario and/or the noise impact on the reference grid.

V. Trajectories Generation

If the X-Plane flight simulator is good enough to be integrated into our platform, it is only one that is used before the HLA simulation. Therefore, to be able to fly several aircraft at the same time, you need to run several flights, one after the other, before the HLA simulation starts. As IESTA should be able to allow one to evaluate new ATS systems, there will be a need to fly several aircraft at the same time (over a thousand sometimes if one tries to simulate traffic over Europe).

The aim of the *Air Aircraft* simulator, currently designed, is then to generate trajectories for one or more aircraft, from lift-off to landing. This simulator takes into account the weather (temperature, pressure and wind) in standard atmosphere (ISA). In a subsequent step, this aircraft model will be implemented as the HLA Aircraft Federate (AC_F federate). Each aircraft managed by the simulator is characterised by:

1) A departure runway and an arrival one,
2) A flight plan corresponding to a set of orders (such as "over fly a waypoint", "intercept radial", etc.)
3) A total energy flight model, provided by Eurocontrol[17] through the BADA[11] data base.

The simulator proceeds according to the following high level algorithm:

```
For each simulation time step do
  Activate inactivated aircraft if their departure time is less than the simulation time
  For each activated aircraft do
    Compute the new position taking into account the previous one and using the BADA performances table.
    Knowing the previous state vector and the new position, compute the Euler angles and the engine thrust
  End for
  Inactivate activated aircraft reaching their destination
End for
```

If the trajectories generation is done using a total energy flight model, the Euler angles and engine thrust computation is performed by using a parabolic polar with linear lift coefficient and the classic flight mechanic equations. In this case, the aircraft characteristics are also provided by BADA.

VI. Noise Propagation Model

A. Introduction

The noise propagation model aims at providing noise grids from a set of simulated aircraft maneuvering or flying in the vicinity of a given airport layout. Those grids may represent aircraft noise impact either through impact maps or videos or noise intensity levels, usually called *Sound Pressure Levels*.

B. Architecture of the HLA Noise Component (ACOU_F)

1. Acoustic software organization

The Acoustic HLA federate is built around 2 components:

1) The ACOU_S component calculating the acoustic footprint for each aircraft.
2) The CARTONOISE component encapsulating ACOU_S and producing the noise maps for each step of the simulation process.

At each time step t_i of the simulation phase CARTONOISE calculates an integrated acoustic map from all the aircraft participating in the simulation at time t_i. CARTONOISE input data are broadcast by the scenario manager, the HLA federation and the ACOU_S component.

2. The CARTONOISE model

At each time step t_i the model proceeds in 3 steps included within a loop of N aircraft (A_1, A_2 … A_K … A_N):

1) For each aircraft A_K recalling the noise footprint.
2) Integration of this noise footprint within the noise map.
3) Deletion of out of limits pixels.

[17] http://www.eurocontrol.int/eec/public/standard_page/proj_BADA.html.

The first step (*data recalling*) proceeds as follows: the FOOTPRINT_ACOU_S noise footprint corresponding to the A_K aircraft at t_i time step is produced by ACOU_S noise component and corresponds to an array of float values, the maximum dimensions of the array being (Figure 3):

NFREQ x NPOINTS x 3.

with:

1) NFREQ = 24 corresponds to the well known ICAO frequencies (Annex 16 of ICAO).
2) NPOINTS = number of sound points produced by FOOTPRINT_ACOU_S for each aircraft.
3) 3 = 3-uplet consisting of X coordinate/Y coordinate and intensity..

Figure 3 depicts an example of noise footprint with deltax (respectively deltay) is oriented to the East (respectively to the North), corresponding to an Universal Transverse Mercator Projection.

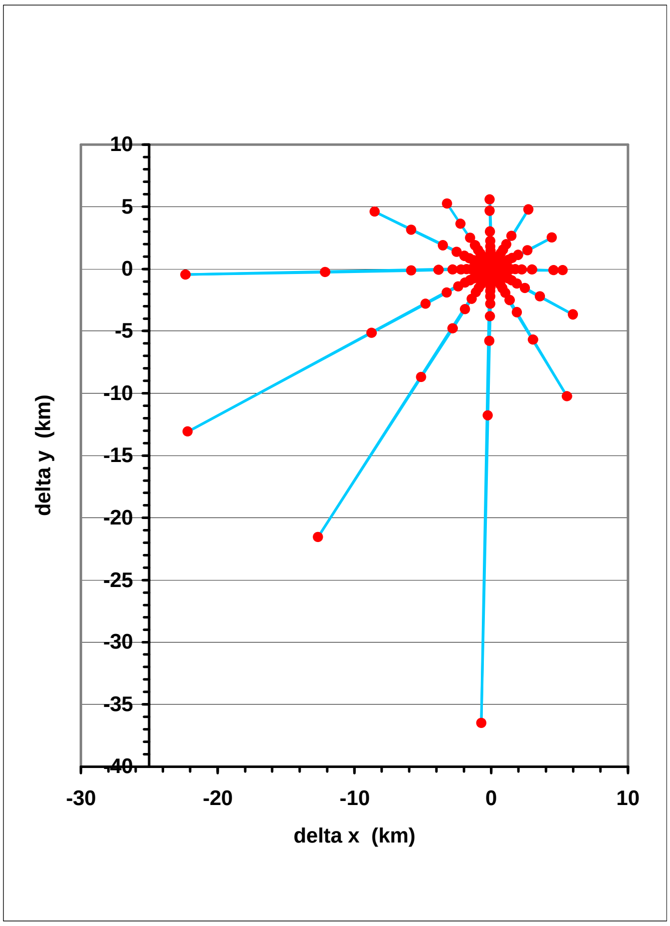


Figure 3. Noise footprint of an aircraft A_K at a given frequency

Deltax and deltay coordinates of such a noise footprint cannot produce a noise grid using constant spatial resolutions. Recalling is then mandatory using spatial resolutions from the initialization phase.

To explain in more details the recalling process, let us describe an example. Assuming that the initial spatial resolutions on ground are $\Delta X = \Delta Y = 10$ m, the output data of ACOU_S have to be located within a noise grid of identical spatial resolutions. Thus, for each (X, Y) coordinate of the noise footprint, the following transformation is applied:

$$\begin{cases} X = \text{ROUND}\left(\dfrac{X*1000}{\Delta X}\right)*\Delta X \\ Y = \text{ROUND}\left(\dfrac{Y*1000}{\Delta Y}\right)*\Delta Y \end{cases}$$

Table 1 gives this recalling transformation for the described example.

ACOU_S output data			Data after recalling		
X (km)	Y (km)	Intensity	X (m)	Y (m)	Intensity
-0,02	0,181	0,054	-20	180	0,054
-0,022	0,464	0,058	-20	460	0,058
-0,024	0,697	0,065	-20	700	0,065
-0,028	1,002	0,075	-30	1000	0,075
-0,031	1,202	0,083	-30	1200	0,083
-0,035	1,451	0,093	-40	1450	0,093
-0,041	1,781	0,108	-40	1780	0,108
-0,049	2,251	0,131	-50	2250	0,131
-0,063	3,003	0,168	-60	3000	0,168
-0,091	4,658	0,253	-90	4660	0,253
-0,104	5,569	0,301	-100	5570	0,301
0,082	0,155	0,054	80	160	0,054

Table 1. Data recalling example when spatial resolutions are 10m

The recalling process can in some cases add redundancy information on the noise footprint. Currently those redundancies are eliminated using a "weakest values elimination" algorithm.

The second step (*Noise footprint integration*) consists in correctly positioning all the coordinates of the noise footprint within the integrated noise grid using a translation algorithm. This is due to the fact that the noise footprint of an aircraft A_K is centered in a (0, 0) coordinates point. However at time step t_i the coordinates (X_{AK}, Y_{AK}) of the A_K aircraft regarding the integrated noise grid, are different. The following example explains the translation process: given a point P1 (X_{P1}, Y_{P1}) of the noise footprint of the A_K aircraft (centered in (0,0)), the translated coordinates regarding the integrated noise grid will be:

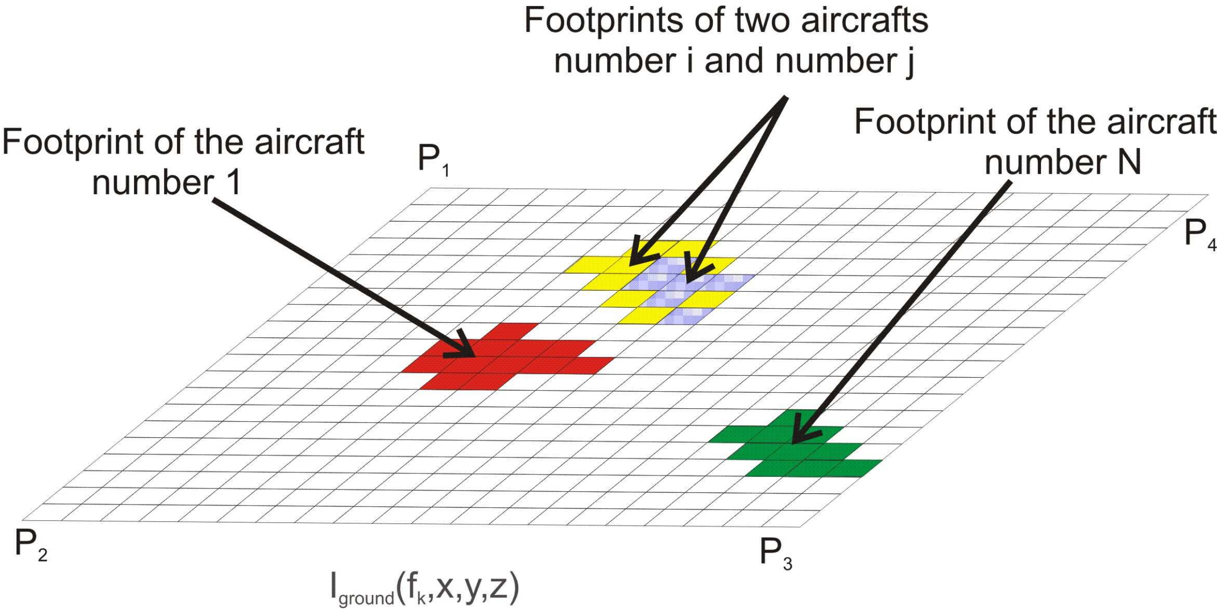
$$\begin{pmatrix} X_{P1\_NEW} \\ Y_{P1\_NEW} \end{pmatrix} = \begin{pmatrix} \mathrm{X}_{\mathrm{P1}} + \mathrm{X}_{\mathrm{A_K}} \\ \mathrm{Y}_{\mathrm{P1}} + \mathrm{Y}_{\mathrm{A_K}} \end{pmatrix}$$

Table 2 shows an integration process example considering that integrated data have been translated through $\mathrm{X}_{\mathrm{A_K}}$ = 1270 m and $\mathrm{Y}_{\mathrm{A_K}}$ = 3240 m.

Recalling data (local reference of the noise footprint)			Integrated data (local reference of the noise grid)		
X (m)	Y (m)	Intensity	X (m)	Y (m)	Intensity
-20	180	0,054	1250	-3060	0,054
-20	460	0,058	1250	-2780	0,058
-20	700	0,065	1250	-2540	0,065
-30	1000	0,075	1240	-2240	0,075
-30	1200	0,083	1240	-2040	0,083
-40	1450	0,093	1230	-1790	0,093
-40	1780	0,108	1230	-1460	0,108
-50	2250	0,131	1220	-990	0,131
-60	3000	0,168	1210	-240	0,168
-90	4660	0,253	1180	1420	0,253
-100	5570	0,301	1170	2330	0,301

Table 2. Data integration example

The third and last step (*Deletion of inconsistencies*) aims at eliminating all the noise pixels not participating in the integrated noise grid calculation regarding the aircraft locations in the local reference of the noise grid. Figure 4 illustrates how several noise footprints of individual aircraft are considered when the integrated noise grid is calculated (pixels colored in gray are eliminated).



Footprints of two aircrafts number i and number j
Footprint of the aircraft number 1
Footprint of the aircraft number N
P_1
P_2
P_3
P_4
$I_{ground}(f_k,x,y,z)$

Figure 4. Computation of the integrated noise grid from aircraft footprint

3. *The ACOU_S model*

ONERA has developed several noise simulation tools that will be integrated in the *Clean Airport* version of the IESTA platform. Among these tools, CESAR, JEDI and MINOTAURE are devoted to determination of noise sources, CESAR and BEMUSE are noise diffraction models, and SIMOUN and PARABOLE are noise propagation models. The first version of the ACOU_S model is built around CESAR and SIMOUN.

Currently, CESAR is the unique tool capable of efficiently estimating the global noise issued by an aircraft, including noise contributions from the most active sources as well as diffraction phenomena over the aircraft structure. The underlying model is based on ray tracing methods consisting of computing all the rays from a noise source and achieving a given observation location either directly or through a set of reflections on intermediate surfaces. The different noise sources of an aircraft are introduced using analytical models, propulsion noise (jet, fan) and aero dynamical noise (slats, flaps, landing gears).

Based on ray tracing, SIMOUN has been designed for studying the long distance propagation of the “Concorde” sonic boom. From the aircraft trajectory, SIMOUN computes the ray propagation as well as their subsequent ground reflections. Propagation computing is performed in 3D and uses the actual atmosphere properties.

Finally, ACOU_S handles the interface between the aircraft trajectories, CESAR model and SIMOUN code. ACOU_S has then been designed for computing the ground noise levels of a given aircraft at each time step. Therefore, computations are performed in real time.

VII. Using Google Earth for Real Time Visualization

A. Earth Browsers

In recent years new innovative ways of providing geographical data has made geo-tagging and browsing of information a popular way of sharing information on the internet. For personal use, Google Maps and Google Earth[18] can be freely downloaded and anyone can start using these powerful tools within minutes. Other well known 3D earth-browsers include NASA’s World Wind and Microsoft’s Virtual Earth.

[18] Google Earth is a registered trademark of Google Inc, http://earth.google.com/

B. Keyhole Mark-up Language (KML)

The Keyhole Mark-up Language[19] is the XML based language used by Google Earth but also by other 2D map applications such as Google Maps and 3D earth-browsers like NASA's World Wind and Microsoft's Virtual Earth.

Even if KML was originally developed by Keyhole, Inc (later acquired by Google) it is now considered a de facto standard for expressing geographic annotation and visualization. The current version, KML 2.2, has been submitted to the Open Geospatial Consortium for standardization.

A typical KML file includes a set of placemarks, images, polygons and 3D models used by applications to display geographical information on a 2D map or in a 3D earth browser.

To provide more dynamic contents the KML language provides a feature called *Network Link*. These links can be used to dynamically load other KML files provided through HTTP or a file-system. Network Links can be set up to be reload periodically at a given update rate. KML supports a feature called *ground overlays* allowing the user to include his own terrain imagery and maps, and place these over the imagery provided by Google Earth through their public servers.

As an illustration, the following KML code is a Network Link defining the virtual camera that views the scene. The tags define the position of the camera relative to the Earth's surface (longitude, latitude, altitude) as well as the viewing direction of the camera (heading, roll, tilt).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
<Document>
    <name>Perspective A320_Crusing_LFBO</name>
    <Camera>
        <altitudeMode>absolute</altitudeMode>
        <longitude>1.243650</longitude>
        <latitude>43.449104</latitude>
        <altitude>2393.581121</altitude>
        <heading>342.169586</heading>
        <roll>-18.771399</roll>
        <tilt>90.000000</tilt>
    </Camera>
</Document>
</kml>
```

Obviously, updating this Network Link file periodically, based on time and camera changes, allows refreshing the view according the simulated aircraft's roll and pitch orientation. In this way, Google Earth can display the scene view which may be thought of as being right in front of the aircraft.

A useful advantage of Google Earth Browser and KML is also to encourage the design of a lot of freeware tools allowing the user to easily generate KML files and/or to display geographical data in a very efficient way. For example, GE-Path[20] was developed to construct paths and/or draw circles and polygons with place marks. EarthPlot[21] is a data analysis and visualization tool for Google Earth. EarthPlot processes geographic based data to generate a variety of maps within Google Earth. Data can be imported from several ASCII data formats, in addition to being copied directly from a Microsoft Excel worksheet.

C. Real Time Visualization through Google earth

As mentioned above the IESTA prototype uses Google Earth to display real-time noise impact data calculated by the noise model. Recently, a very similar approach using Google Earth for scenario development and HLA federation test and integration has been described[12].

1. *Scenario Displaying*

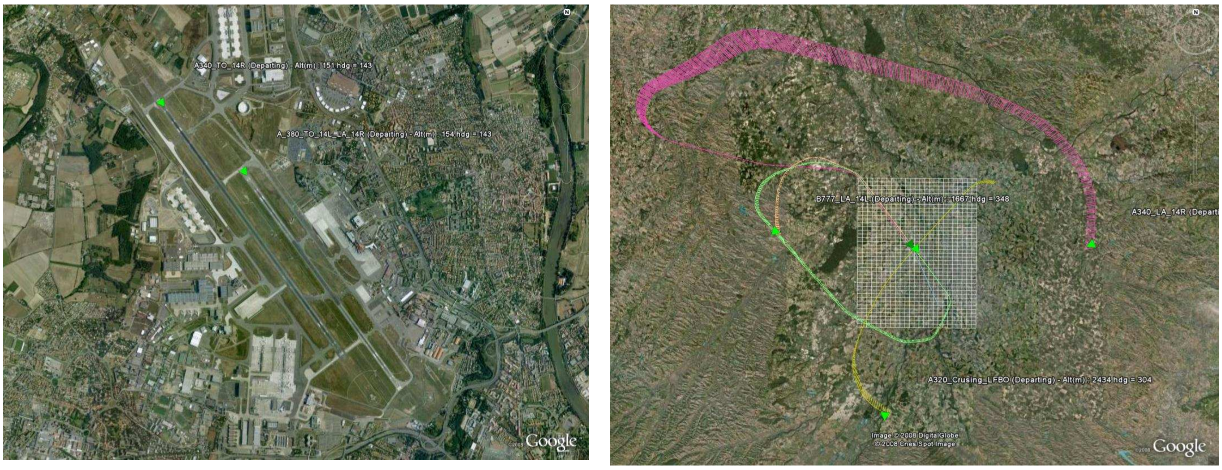
The first use of Google Earth is to display the scenario before starting the simulation phase. As an illustration, Figure 5.a shows the Toulouse Blagnac Airport with 2 aircraft waiting for departure represented by green icons. In this example, the scenario consists of 5 aircraft the trajectories of which are represented by the snapshot of figure 5.b. This figure also shows the 25km X 20km noise reference grid covering the Toulouse Blagnac Airport vicinity.

[19] http://code.google.com/apis/kml/documentation/

[20] http://www.sgrillo.net/googleearth/

[21] http://www.earthplotsoftware.com/

Aircraft trajectories are displayed by Google Earth through a set of KML files generated by the Aircraft Federate during the first simulation step. Alternatively, the reference grid is a KML file generated by the Acoustic Federate from the location of a given reference origin (the control tower of the airport). During the simulation phase, those KML files can be enabled or disabled according to the user needs.


A340_TO_14R (Departing) - Alt(m) : 151 hdg = 143
A_380_TO_14L_LA_14R (Departing) - Alt(m) : 154 hdg = 143
Google
B777_LA_14L (Departing) - Alt(m) : 1667 hdg = 348
A340_LA_14R (Departi
A320_Crusing_LFBO (Departing) - Alt(m) : 2434 hdg = 304
Image © 2008 DigitalGlobe
© 2008 Cnes/Spot Image
Google

5. a. Toulouse Blagnac airport 5. b. Reference grid and aircraft trajectories

Figure 5. Scenario display through Google Earth.

2. Displaying the simulation current state

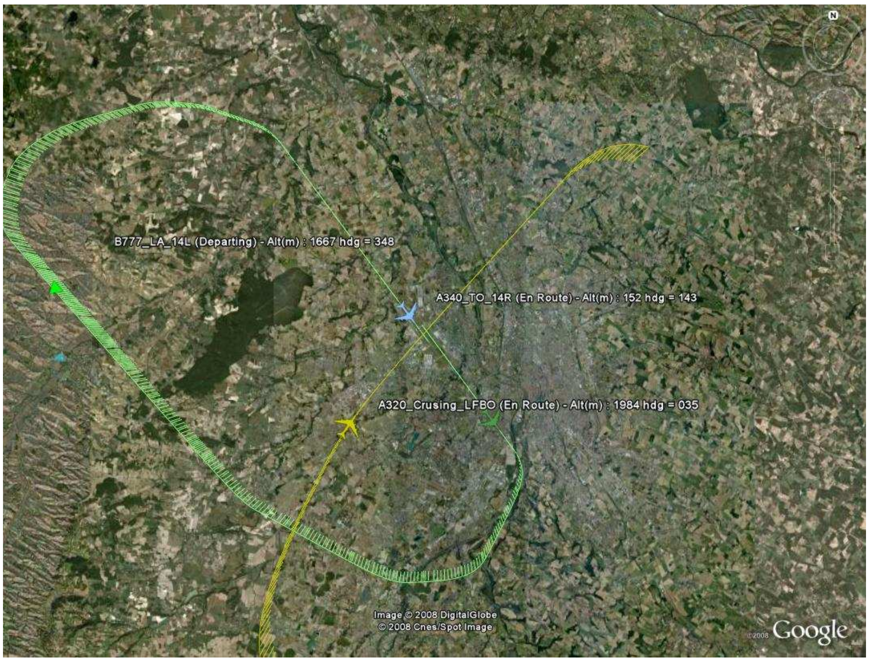

B777_LA_14L (Departing) - Alt(m) : 1667 hdg = 348
A340_TO_14R (En Route) - Alt(m) : 152 hdg = 143
A320_Crusing_LFBO (En Route) - Alt(m) : 1984 hdg = 035
Image © 2008 DigitalGlobe
© 2008 Cnes/Spot Image
Google

Figure 6. Simulation state display

Figure 6 is a snapshot displaying the simulation current state showing 3 activated aircraft represented by green, yellow and blue icons respectively. The noise reference grid has been disabled as well as the noise impact on the airport vicinity.

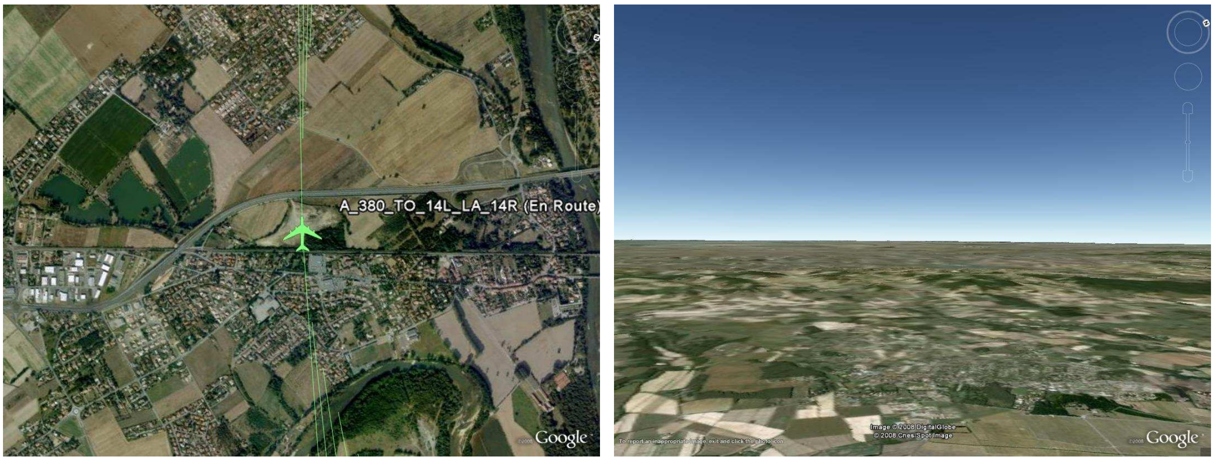


Figure 7. Overhead and perspective views

Finally, figure 7 shows 2 different view of each simulated aircraft. The left part of the figure is a snapshot of the overhead view of an aircraft, and the right part is a snapshot of the perspective view from the same aircraft. Both views are displayed by Google Earth from 2 KML Network Links generated by the Aircraft Federate every 2 sec. The left view allows following the aircraft from the "eye" of Google Earth located above the aircraft. The right part of the figure shows the terrain ahead the simulated aircraft. The refresh rate of both network links can be selected by the user from the properties menu of each network link.

3. Displaying noise impact

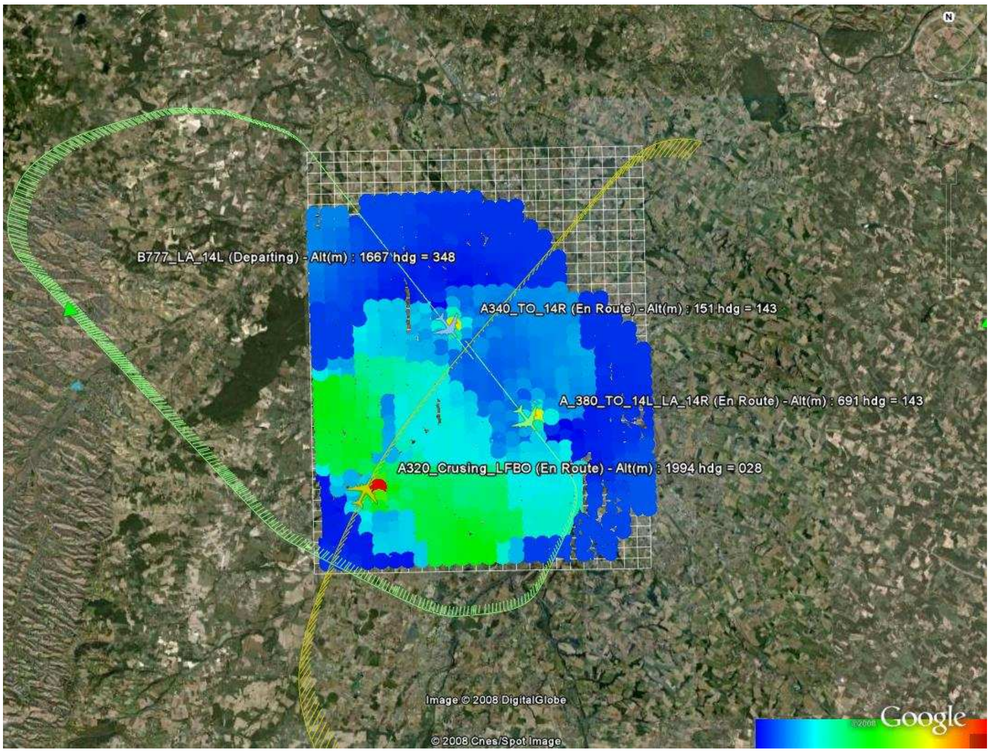

B777_LA_14L (Departing) - Alt(m) : 1667 hdg = 348
A340_TO_14R (En Route) - Alt(m) : 151 hdg = 143
A_380_TO_14L_LA_14R (En Route) - Alt(m) : 691 hdg = 143
A320_Crusing_LFBO (En Route) - Alt(m) : 1994 hdg = 028
N
Image © 2008 DigitalGlobe
© 2008 Cnes/Spot Image
Google

Figure 8. Displaying noise impact

Figure 8 shows a snapshot of the noise impact on the reference grid. The noise intensity given in db is represented through color levels according to the color ramp displayed in the bottom right of the figure. Those grids are displayed by Google Earth through KML Network Links generated by the Acoustic Federate in real time with an updated rate of 2 sec. The user can enable the noise grid from each simulated aircraft individually or the noise grid of all the contributing aircraft. This figure represents in a colored form the noise impact on the reference grid, from 3 currently activated aircraft. 2 aircraft are climbing after take off from Toulouse Blagnac Airport, and one aircraft is cruising at 1994 m altitude over the airport vicinity.

VIII. Conclusion

This paper describes a rapid prototyping phase of the IESTA platform designed to provide customers simulation tools for evaluating environmental impact of innovative air transport concepts around airports. In this paper, we have shown how commonly available tools like Google Earth can be used to gain access to geodata and to display geographical output data of both a scenario and noise impact results. Obvious advantages are that more or less any geodata is available within minutes and at a cost close to zero. Finally we have shown how implementations of SISO standards (like HLA) can work together with major, mass-market oriented commercial applications like Google Earth. More generally, our experience shows that combining easily available tools such as X-Plane flight simulator, Google Earth browser and the High Level Architecture can provide powerful solutions to display complex results and scenario in an understandable, efficient and low cost manner. Moreover, HLA constitutes an efficient architecture to facilitate reuse of additional models and interoperability between them. Next step deals with integration of all components of this prototype into an operational platform, including models of chemical species emission and propagation.

IX. Acknowledgments

The IESTA program is sponsored by several funding sources including the French “Midi-Pyrénées” Region, several Territorial Collectivities, French State and the European Fund for the Regional Development (FEDER). The authors would like to thank all contributors to the IESTA Initiative and specially the research team members of the Long Term Research and Systems Integration Department, Sébastien Aubry, Luis Basora, Muriel Brunet, Bruno Lamiscarre, Thomas Chaboud and Jean Hermetz. Jean Bulté and Jean Varnier from the Computational Fluid Dynamics and Aero acoustics Department (DSNA) are greatly acknowledged for their active contribution to noise modeling within the IESTA prototype. Finally, Jean-Michel Mathé and Eric Noulard, from the ONERA/DTIM Department have also to be acknowledged for their contribution (CERTI support and design of the X-Plane HLA Plugin).

X. References

[1] Kuhl, F., Weatherly, R., and Dahman, J., “Creating Computer Simulation Systems: An Introduction to the High Level Architecture” *Prentice Hall, ISBN 0-13-022511-8, 1999*.

[2] IEEE Std 1516-2000^{TM}, “IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) - Framework and Rules,” September 2000.

[3] IEEE Std 1516.1-2000^{TM}: “IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) - Federate Interface Specification”, September 2000.

[4] IEEE Std 1516.2-2000^{TM}: “IEEE Standard for Modeling and Simulation (M&S), High Level Architecture (HLA) - Object Model Template (OMT) Specification”, September 2000.

[5] IEEE Std 1516.3-2003^{TM}: “IEEE Recommended Practices for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP)”, April 2003.

[6] Lehmer, R. D., Malsom, S. J., “Distributed System Architecture in VAST-RT for Real-Time Airspace Simulation” AIAA Modeling and Simulation Technologies Conference, Providence, Rhode Island, August 2004, Paper # AIAA-2004-5436.

[7] Web site of the NASA VAMS Project: http://vams.arc.nasa.gov/

[8] Adelantado, M., “Rapid Prototyping of Airport Advanced Operational Systems and Procedures through Distributed Simulation” AIAA Modeling and Simulation Technologies Conference, Providence, Rhode Island, August 2004, Paper # AIAA-2004-5436.

[9] Adelantado, M., Bonnet, P., Latour, J. and Vincent, A., "Distributed Simulation for Acquisition and Analysis of Future Airport Operational Concepts", AIAA Journal of Aerospace Computing, Information and Communication, Vol.1 n° 1, January 2004.

[10] Adelantado, M., Damioli, C., and Dedieu, A., "A Reusable HLA Federation for Simulation of Airport Traffic Assistance/Automation Systems", 5th IEEE International Conference on Intelligent Transportation Systems, Singapore, September 3-6, 2002.

[11] Eurocontrol, “User Manual for the Base of Aircraft Data (BADA)”, revision 3.6, July 2004.

[12] Löfstrand, B., Möller, B., “Using Google Earth for Scenario Development and HLA Federation Test & Integration”, 2008 Spring Simulation Interoperability Workshop, Providence (RI), April 14-18, 2008.