SOFTWARE Open Access

An open source virtual globe rendering engine for 3D applications: NASA World Wind

Francesco Pirotti[1*], Maria Antonia Brovelli[2], Gabriele Prestifilippo[2], Giorgio Zamboni[2], Candan Eylul Kilsedar[2], Marco Piragnolo[1] and Patrick Hogan[3]

Abstract

Background: NASA World Wind is an open source application-programming interface for developing geographic information systems based on a virtual globe rendering engine representing a planet. NASA World Wind provides the ideal environment for scientific data, their analysis, visual representation and interaction with users, in a single platform, which can be deployed both as a Java desktop application (NASA World Wind) or a JavaScript web application (ESA-NASA Web World Wind).

Results: We give here an overview of the project, reporting details regarding current development direction, with state of the art examples. The European Space Agency is now partnering with NASA on development of the "ESA-NASA Web World Wind"; this high degree of interest from other agencies will boost future project productivity.

Conclusions: With this contribution, we want to increase awareness of NASA World Wind as a unique opportunity to foster collaboration between scientists, developers and other stakeholders, enriching knowledge of our Earth's complexity.

Keywords: JavaScript, Java, HTML5, Spatial data, Geographic information system, Virtual globe, 3D modelling

Background

The National Aeronautics and Space Administration (NASA) was born in 1958 with a motto: "For the Benefit of All". This motto is fully realized in NASA World Wind. World Wind is an API-centric SDK, an Application Programming Interface (API) Software Development Kit (SDK) that delivers virtual world for experiencing geospatial data in its native context. This visualization technology provides a multi-dimensional geographic information system (GIS) and an extensible framework for analyses and visualization of scientific data. The base model is a virtual globe, a 3D representation of a planet, enriched with its digital terrain model (DTM). A DTM is the digital representation of the elevation of a planet's surface at ground level. It provides information not only regarding elevation, but also allows one to derive other morphological features, such as slope, aspect, with the ability to model these dynamics of a planet's system processes– e.g. local solar irradiance [20]. NASA World Wind loads elevation data automatically with a multi-resolution approach, i.e. the scale of the view determines the detail of the elevation grid [4]. Custom elevation grids can also be uploaded (Fig. 1) allowing users to implement their own model in NASA World Wind.

NASA World Wind has been available as open source since 2003 in C#, since 2006 in Java, since 2012 in Android, since 2013 in iOS and since 2014 in JavaScript. The purpose of this technology has always been for developers to implement new functionalities and advance existing ones to create increasingly more useful tools for analysis and visualization of Earth and space data [6, 12, 13].

The potential to access spatial data is a common and foremost attribute of NASA World Wind. GIS interoperability is the ability for a platform to read and render multitude spatial data formats [21], from local sources and from remote public and private datasets [4] via shared open standards. The vision of NASA World Wind is to provide a high-performance open-source 3D planetary model with a large collection of components for projecting images, geometric objects and geographic shapes on the model. Its implementation is modular and

* Correspondence: ncesco.pirotti@unipd.it
[1]Department of Land, Environment, Agriculture And Forestry (TESAF) & Interdepartmental Research Center of Geomatics (CIRGEO), University of Padova, Via dell'Università 16, 35020 Legnaro, Italy
Full list of author information is available at the end of the article

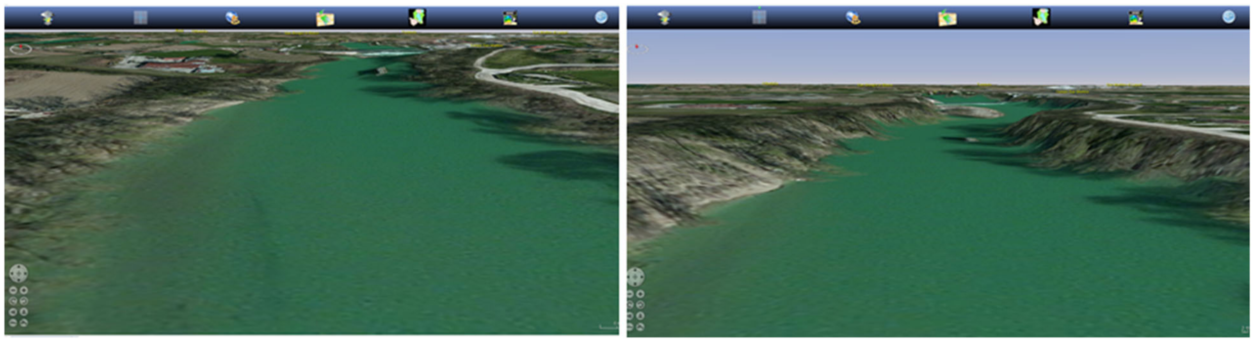


Fig. 1 Left: default NASA World Wind elevation grid (SRTM DEM 90 m resolution). Right: local elevation grid from DTM at 5 m resolution uploaded into NASA World Wind

extensible, allowing users to develop simple–to-sophisticated applications for their own requirements.

An insightful description is given in the homepage of the NASA World Wind organization website [23], which we report fully: "What is World Wind - World Wind is a free, open source API for a virtual globe. World Wind allows developers to quickly and easily create interactive visualizations of the 3D globe, mapping geographical information. Organizations across the world use World Wind to monitor weather patterns, visualize cities and terrain, track the movement of planes, vehicles and ships, analyse geospatial data, and educate people about the dynamic nature of Earth. [What are its] Features and Benefits? It's open source: [NASA] World Wind is completely open source, therefore extending the API is simple and easy to do, creating a powerful platform giving any application the means to express, manipulate and analyse spatial data. [NASA] World Wind technology can be incorporated into Windows, Mac and Linux applications, web pages and web applications, as well as mobile applications alike. Build what you want. [NASA] World Wind is different from a 3D globe like Google Earth because it is not a completed application targeted at end-users. Instead, it is an SDK (software development kit) that software engineers can use to build their own applications. SDK contains World Wind API, examples and demos. API provides the virtual globe. Examples and demos show how to use [NASA] World Wind code in order to create a wide range of projects, from satellite tracking systems to flight simulators. With World Wind taking care of the hard work of visualizing geographic data (generating terrain from elevation models, selecting and displaying images from imagery servers, etc.), software engineers are free to focus on solving problems specific to their own domains and quickly build whatever geospatial applications they choose."

All the above is available through Java, Android and JavaScript. Reusable code allows to implement functionalities such as a graphical user interface (GUI), a browser designed for the end user, with a fully extensible default layout reflecting a classical GIS application. Geographic data are retrieved, rendered and can be analysed by interacting with layers and tools designed by the developers.

NASA World Wind's open source essence, with its modular and extensible architecture makes it an ideal platform for education, research and communication on our Earth dynamics. This contribution wants to increase awareness on NASA World Wind as a unique opportunity to foster collaboration between scientists, developers and other stakeholders, enriching knowledge of our Earth's complexity.

Implementation

Architecture

The architecture of NASA World Wind provides a high-performance open source virtual globe (of planet Earth, but not limited to it). The virtual globe is a 3D model that can be draped with a large collection of images and geometric 2D and 3D objects. To this end, NASA World Wind is being developed and deployed in Java and in JavaScript, the former for loading the World Wind client locally through a Java Virtual Machine, the latter for loading the application remotely via a web browser (see Table 1 for availability and requirements). Common to both distributions are the following base objects: Layers, Globe and Tessellator (Figs. 2 and 6). The last two providing, respectively, the digital mathematic approximation of the planet's shape (an ellipsoid) and the terrain elevation grid.

Tessellator is the engine for rendering the ellipsoid by approximation, using a mesh of regular shapes, commonly rectangles or triangles. NASA World Wind developers applied a high-performance algorithm, described in Miller and Gaskins [16]. Tessellation is performed every time the terrain's resolution has to be synchronized with the current viewing state and availability of elevations [7].

Globe is rendered on screen using the tessellated representation of the ellipsoid as an approximation of a planet's surface and elevation. The ellipsoid is defined in a Cartesian coordinate system in which the Y axis points to the north pole; the Z axis points to the intersection of

Table 1 Availability and requirements of NASA World Wind's two main SDKs

Project name	World Wind	Web World Wind
Programming language	Java	JavaScript
Project home page	https://nasaworldwind.github.io/	https://nasaworldwind.github.io/
Operating system(s)	Platform independent	Platform independent
Requirements	Updated video card drivers; Windows and Linux: Oracle Java Runtime Environment (JRE) 1.6.0 update 10; Mac OS X: JRE 1.7.0	Updated video card drivers; A JavaScript-enabled web browser;
License	NASA open source agreement version 1.3[a]	NASA open source agreement version 1.3
Source code	http://github.com/NASAWorldWind/WorldWindJava	http://github.com/NASAWorldWind/WebWorldWind
Examples	https://goworldwind.org/demos/	http://explorer.worldwind.earth/ http://worldwind.arc.nasa.gov/quakehunter/ http://worldwind.arc.nasa.gov/spacebirds/ http://worldwind.arc.nasa.gov/worldweather/ http://jsbin.com/nomafey/edit?output (sandbox)

[a]Available at https://en.wikipedia.org/wiki/NASA_Open_Source_Agreement [18]

the prime meridian and the equator, in the equatorial plane; the X axis completes a right-handed coordinate system, and is 90° east of the Z axis and also in the equatorial plane [17]. There is also the possibility to use a flat representation of a planetary globe, by projecting it to a plane using the default Mercator projection. User-defined projections can be implemented as well. The default reference system for earth data geolocation is geographic, using latitude and longitude angles in a WGS84 Datum. Height can be provided as meters above sea level. Projected cartographic reference systems (e.g. UTM) and different Datum such as WGS84, NAD27 and many others are supported. Projections are supported through the Proj.4 library [9]. For a reference regarding projections and cartographic reference systems see Evenden and Warmerdam [9], in global GIS applications see Battersby et al. [3].

Layers is the class providing access to spatial data added to NASA World Wind. Most spatial data are stored as raster models, e.g. aerial or satellite imagery, temperature grids etc, and as vector models, i.e. 2D or 3D geometric objects. Some examples of spatial data visualized in NASA World Wind are in Fig. 3. Layers are linked to spatial data available from local and remote repositories. Specific implementations allow interpreting, geolocating and rendering such data on the globe. Most supported formats of spatial data are available thanks to the GDAL library [27].

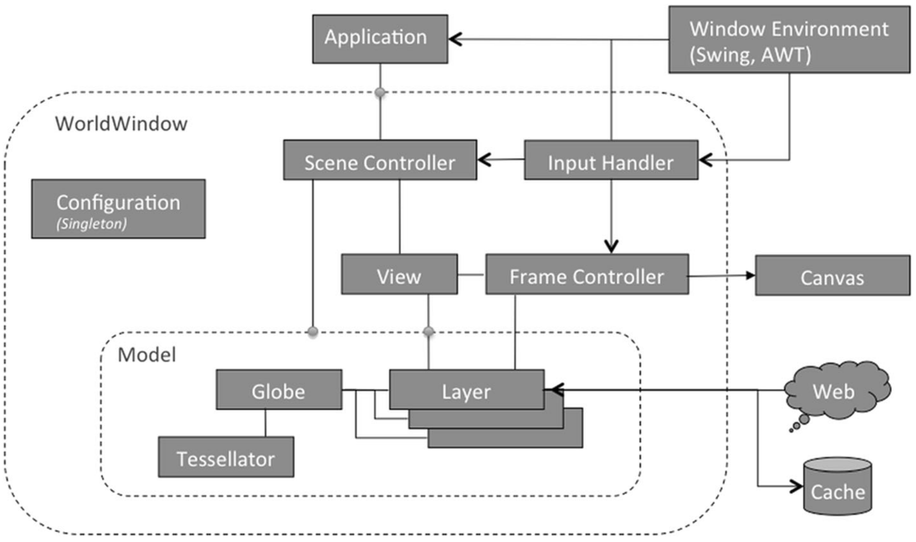


Fig. 2 Overview of the structure of NASA World Wind classes with base objects and their relationships. Image courtesy of Schubert and Collins [23]

World Wind - Java

The structure of the API is represented as a simplified diagram in Fig. 2. The Globe, Tessellator and Layer classes have been discussed in the previous section, as they are common to the Java and JavaScript APIs. These three classes provide the Model segment that is then rendered in the WorldWindow segment, allowing visualization and interaction between the data, the user and other modules. A developer can extend existing modules easily and add custom tools and plug-ins creating simple to complex applications.

Layers can be populated with raster and vector models. Vector representation can be summarized in point, polyline, and polygon objects. For a more in-depth description of vector features please consult the Open Geospatial Consortium (OGC) Simple Features specifications [11]. Points, placemarks and markers are useful to pinpoint locations and they can be enhanced with symbols, icons and labels. Polylines are useful to create paths, show a direction and delimit areas with boundaries. Figure 3 at first row depicts these two features in NASA World Wind. 3D vector objects are cylinders, spheres, cones, pyramids. 3D polygons can be created by extrusion of 2D polygons. They can be draped with images in order to create realistic complex objects (Fig. 3 second row first column). The position of the object in terms of height can be fixed on the ground or set at a user-defined height. The user can interact with vector objects by querying them or also by modifying their geometry and position manually with editing tools.

Data are retrieved from the web and cached locally for best performance [4] and organized as layers. Raster and vector models are supported. Supported raster formats are the following (for a description related to each abbreviation see [28]): JPG, PNG, GeoTIFF, JPEG2000. Supported vector formats are the following (for a description related to each abbreviation see [29]): ESRI Shapefile, Keyhole Markup Language (KML), VPF, GML, GeoJSON, GeoRSS, GPX, NMEA.

Web Map Service (WMS) and Web Feature Service (WFS) are standards from the OGC used to collect dataset from web services. Figure 3 at second row second column shows an example of MODIS Average Land Surface Temperature accessed by WMS [26]. Through the WMS service, the NASA Server also provides availability to several layers: e.g. Blue Marble Next Generation (BMNG), LandSat7 based imagery, USGS imagery. Elevation data are provided by the server as well, using a combination of sources from ETOPO2, NASA SRTM and USGS NED.

Many modules for interaction between the data and the user are available in the NASA World Wind Java API distribution. In the last row of Fig. 3, on the left, the elevation data and a user-defined location is used to calculate which points are visible in the surrounding environment; on the right, the terrain profile along a user-defined line is extracted.

Programmers can use the full potential of Java's GUI development interface (SWING) to define the look-and-feel and interaction between the user, NASA World Wind classes and custom modules. The simplest example of such interaction is a menu allowing layer management, as shown in Figs. 4 and 5; more complex examples are also available as demos (Table 1).

Web World Wind

Accessing spatial data and GIS tools through the web has been a popular solution since internet became available for private and public access [22]. NASA World Wind has recently (2014) been ported to a JavaScript API for developing interfaces which are accessible directly via a web browser, without other requirements. This API is being enriched with functionalities from the Java version. As stated in the project's webpage (Table 1): "NASA Web World Wind is a 3D virtual globe API for HTML5 and JavaScript. Web pages include Web World Wind to provide one or more virtual globes on that page. The European Space Agency (ESA) is also partnering with NASA on development. NASA Web World Wind runs on all major operating systems, desktop and mobile devices, and web browsers".

All the components available in a NASA Web World Wind application, as it can be seen in Fig. 6, are divided into external components and internal components. The blocks in red are external ones that developers typically interact with to create their personal applications.. The blocks in blue are the internal ones and usually software engineers do not interact with them, since they are using internal functionalities for rendering of the globe and editing this part of the code is a delicate task.

The WorldWindow is an object, which contains all the NASA Web World Wind functionalities and connects them to an HTML container: the canvas. Each application implementing NASA Web World Wind should have at least one canvas in its webpage, with a WorldWindow object, linked through its identifier.

The globe is the 3D representation of the Earth, a WGS84 ellipsoid containing tasselleted elevation data, though it is also capable of showing several 2D projections. The implementation of the globe and its tessellation follows what is described previously in the Section "Architecture". Usually, applications do not interact directly with the globe but implement methods to get information from it, even though it also possible to edit the globe's features.

Layers are fundamental objects to implement several features. Besides the WorldWindow, they are typically the immediate way of interacting with spatial data

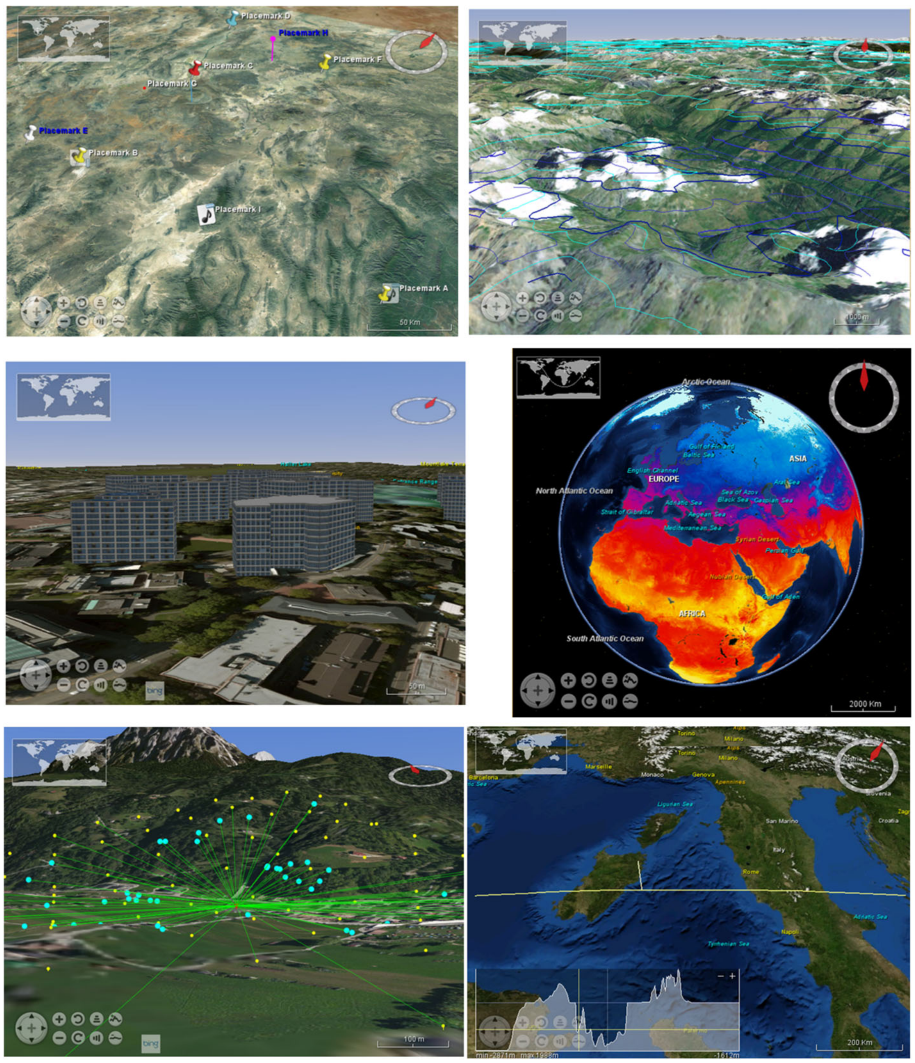


Fig. 3 Latitude and longitude reticulum over the Earth globe – on the left the menu with layers

loaded in a GIS. NASA Web World Wind can access a multitude of built-in imagery and terrain formats. It can display multi-resolution imagery, terrain and geometric data using multiple supported formats. GeoTIFF, JPEG and PNG formats are supported for raster data, and ESRI Shapefiles, KML and Collada formats for vector models (see reference to project webpage in Table 1). REST, WMS and Bing map services are supported as well. Layers allow shapes in 2D, 3D and also user-defined elevation data. Also available in the API as layers are utilities like the compass layer, graticules, coordinates display and view controls. View controls allow users to navigate in 3D using zoom, panning and tilting of the view in different directions. Upcoming features include HDF, DTED, NITF and FBX vector format support [29], WCS, WMTS and WFS service support [19]. Also, new utilities are about to be added to the JavaScript API distribution for on-screen measurements, calculation of line-of-sight, creation of analytic surfaces and subsurface visualization support.

The Navigator is responsible for converting user actions to manipulations of the globe. It monitors mouse events and user gestures and translates them to pan, zoom or tilt operations on the globe. The Navigator can also be driven by the application to change the viewpoint to a particular location or to set its tilt and heading. The Navigator is an object automatically integrated into each WorldWindow, allowing interacting with the globe to get all the information about its position. Developers can customize mouse movements and gestures to navigate the globe in different ways. The Navigator can also be controlled by mobile devices and can be personalized to assign any mobile gesture to a particular action in the globe.

Among the features there are also geocoders that permit discovery of geographic locations from a query string. NASA Web World Wind implements the one

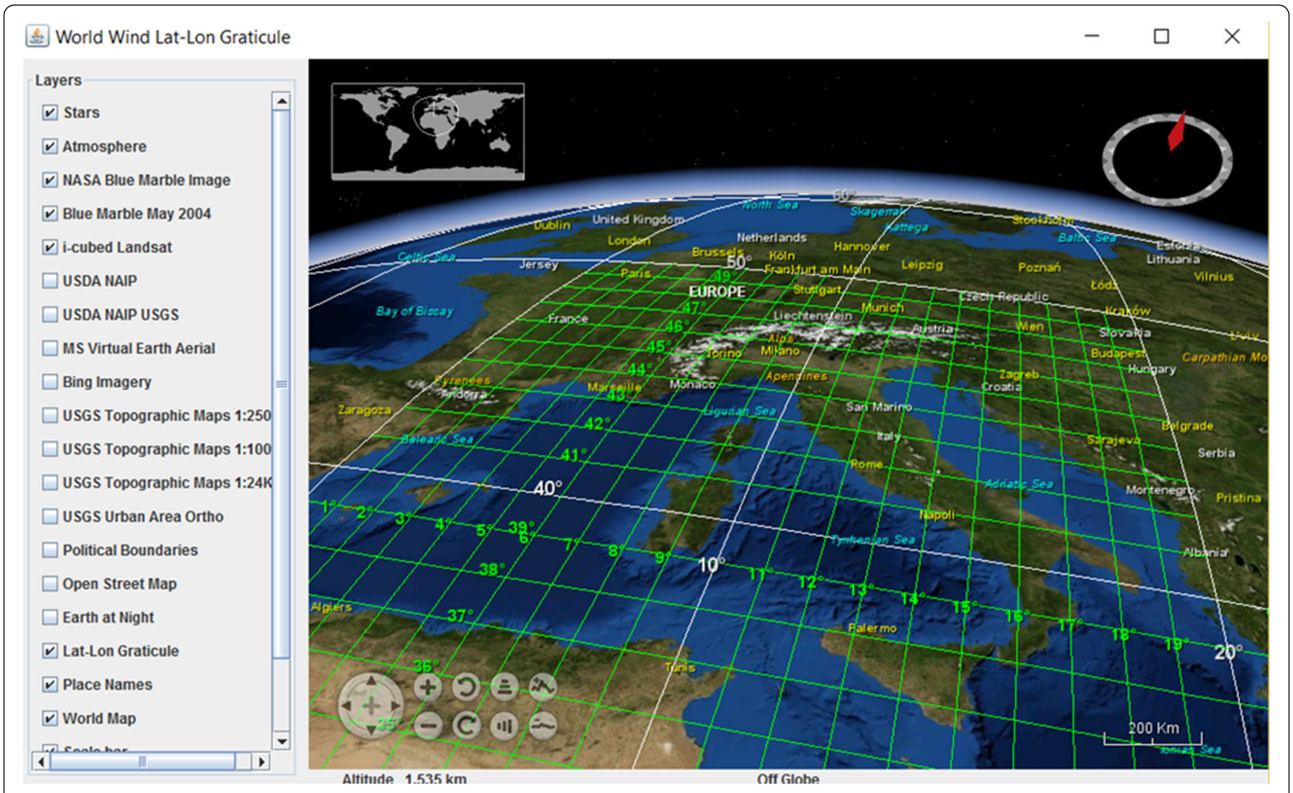


Fig. 4 On-screen layer manager

from Open Street Map's Nominatim geocoder at MapQuest [14]. Thanks to this geocoder, it is possible to retrieve information about a city, area or location from its name or coordinates and then show all the desired information in the globe.

NASA Web World Wind contains several other objects; most of them run behind the scenes, so programmers are not required to interact with them. Instead, when programmers want to extend the NASA Web World Wind features, they only need to interact with specific components, using the JavaScript API. As a result it is quite easy to extend the features already implemented or even create new ones.

NASA Web World Wind provides a rich set of features for displaying and interacting with spatial data. It is designed to be extensible; adding functionality is simple and easy to do by extending existing modules or linking new tools as plug-ins.

Results and discussion

Since its release as open source in 2003, NASA World Wind attracts a large number of contributions from the open source community. The powerful modular architecture of NASA World Wind attracted improvements to the core functionalities, implementation of add-ons, delivery of high-resolution data, and outright development of GIS solutions. Bell et al. [4] in their article report information on add-ons and high resolution data available for NASA World Wind at the time of writing their contribution.

In this section, we report three solutions developed using NASA World Wind, and three solutions developed using NASA Web World Wind. What is reported is, obviously, a very small subset of many excellent applications developed by software engineers, students and professional teams from a large international community. We attempt to represent a range of different scopes and fields: multi-dimensional data viewers, collaborative solutions, cultural heritage representations, dissemination of science data and access to historical datasets from open data services.

Multi-dimensional data viewer - EST-WA

The Environment Space and Time Web Analyser (EST-WA), is based on the netCDF format [6] and is able to visualize spatial-temporal variable distributions in a multi-dimensional interface (e.g. sea temperature or soil permeability at different depths, air pollution, temperature, humidity at different heights, etc.).

The system is composed of two parts (Fig. 7): EST-WA2D, which is the graphical 2D interface for viewing and filtering the data from a netCDF archive; EST-WA3D, which is the graphical 3D interface, based on the NASA World Wind SDK.

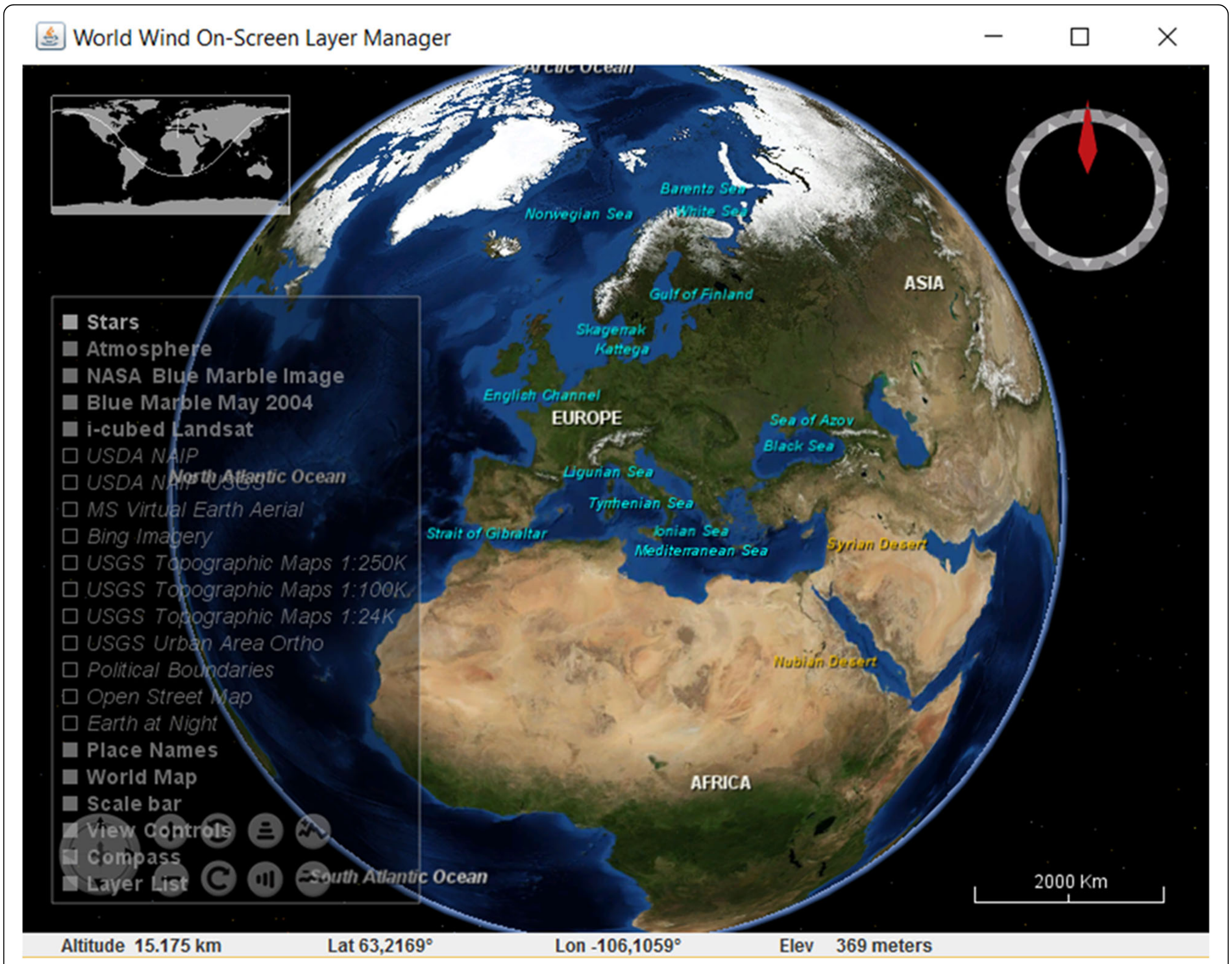


Fig. 5 From top left in clockwise direction: - vector points represented as labels and icons, - 3D polylines - raster with temperature from open datasets (MODIS) – terrain profile – view/terrain interaction (yellow dots represent a blocked view) – extruded polygons draped with images

EST-WA2D shows all main metadata related to the variables stored in netCDF files; it displays, for each variable, the data type, the measurement unit, the description, the rank and the axes defining the domain of the variable itself. For each axis, the measurement unit, the dimension, the step and maximum and minimum values are reported. For each "grid" type variable, the grid coordinate system information (i.e. the corresponding axes) is given.

By selecting each individual element (variable, domain, grid and axis), it is possible to access the second level of metadata (attribute) and to have the complete overview of the information stored in the netCDF file. Through

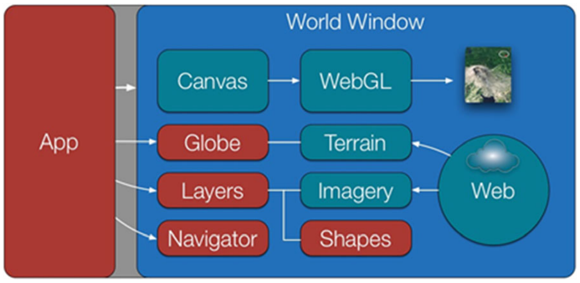


Fig. 6 Structure overview of the NASA Web World Wind application

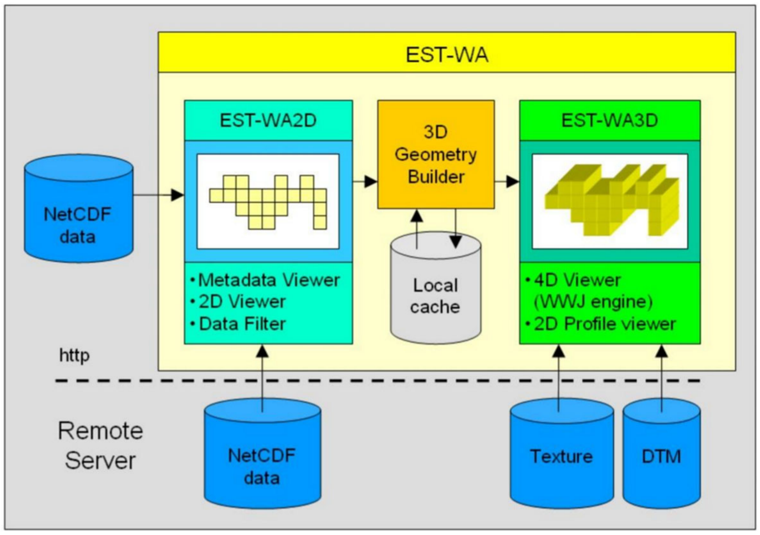


Fig. 7 EST-WA architecture

this window, after the selection of the variable, users can activate the 2D map pre-viewer. Besides the usual browsing functionalities (pan, zoom-in and zoom-out), the dynamic map allows users to interact with the content of the 4D structure and to spatially and temporally filter the data of interest. After the filtering phase, EST-WA3D is activated (Fig. 8).

The environment is similar to the default NASA World Wind interface (on which it is based) but new tools for browsing the data in time, for analysing the temporal evolution of the variable and for slicing the data model in order to see the inner doxels (dynamic voxels) of the model have been implemented.

Data can be profiled along a constant longitude, latitude or height and, in order to increase the visual analysis of the Z (height) dimension, there is a vertical shift of the model and a degree of vertical exaggeration. Moreover, there is a legend with the colour ramp corresponding to the variable values for making possible the quantitative analysis of the phenomenon.

The value of a single voxel at a certain time and location or the values of all the voxels along a profile are readable thanks to an implemented pop-up window with a 2D viewer appearing over the main map. On this main map, the graphic of the variable along the selected profile and the single values are dynamically synchronized with any movement of the scene, which can be obtained for instance by means of the cursor. The user completely controls the profile, i.e. both translation and rotation, in every direction, are possible.

Tourism and cultural heritage - PoliCrowd

PoliCrowd is a 3D/4D collaborative platform based on NASA World Wind Java. After the first version which

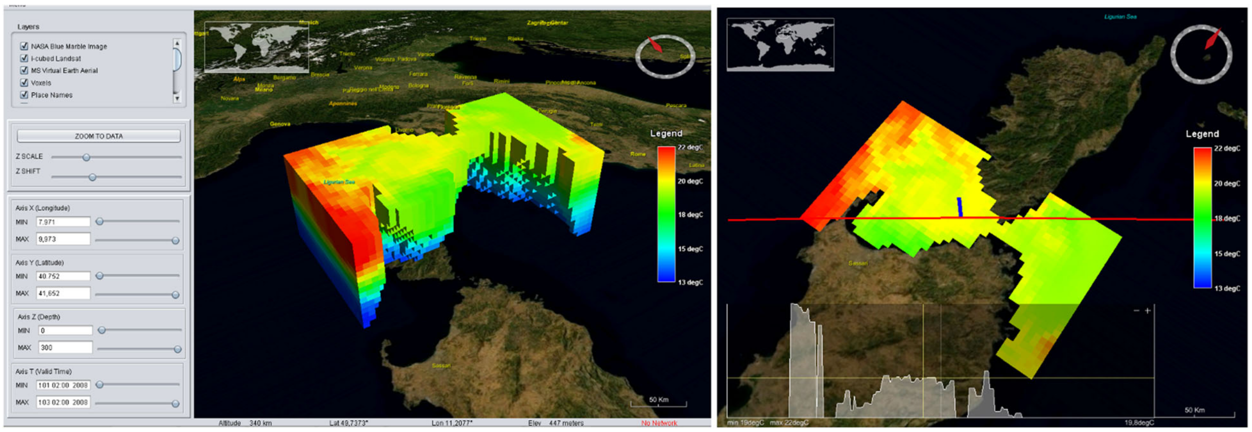


Fig. 8 EST-WA3D – Three-dimensional Viewer (left), sliced model and section profile graphic (right)

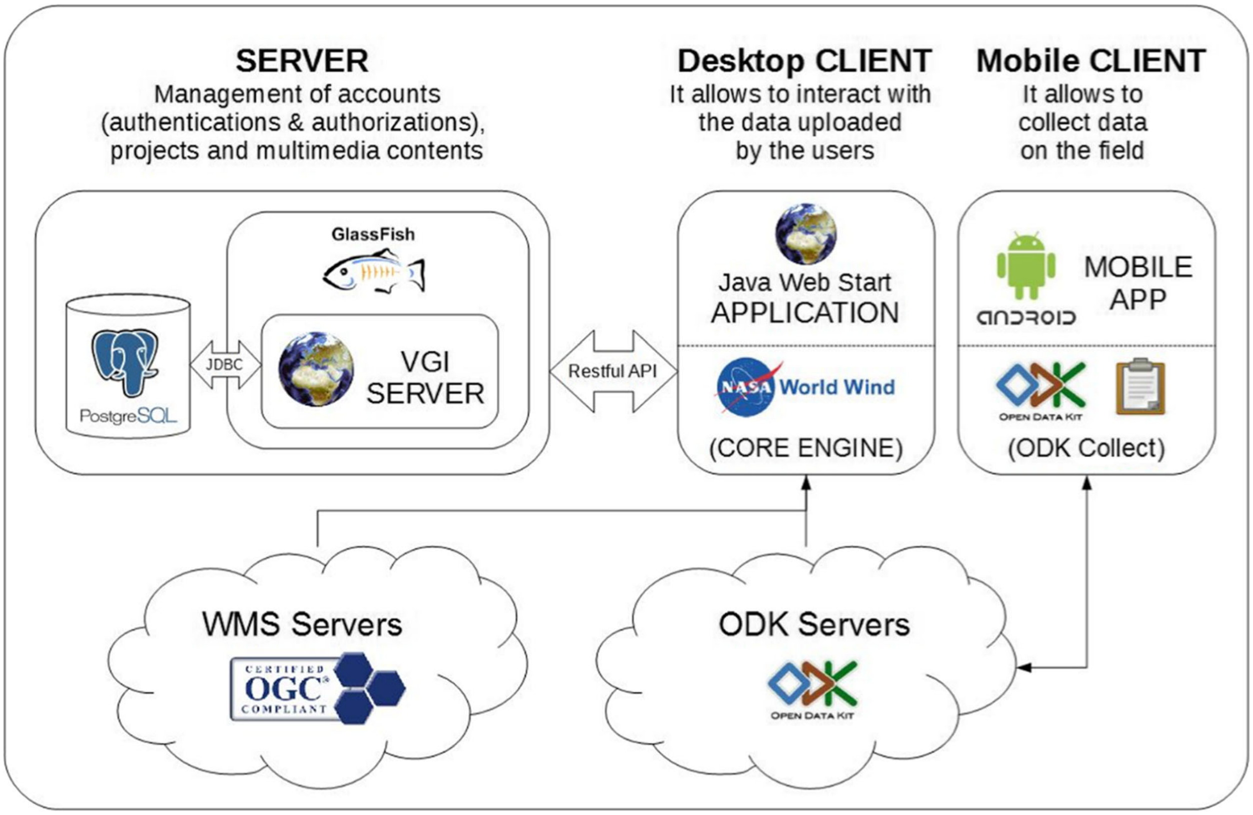
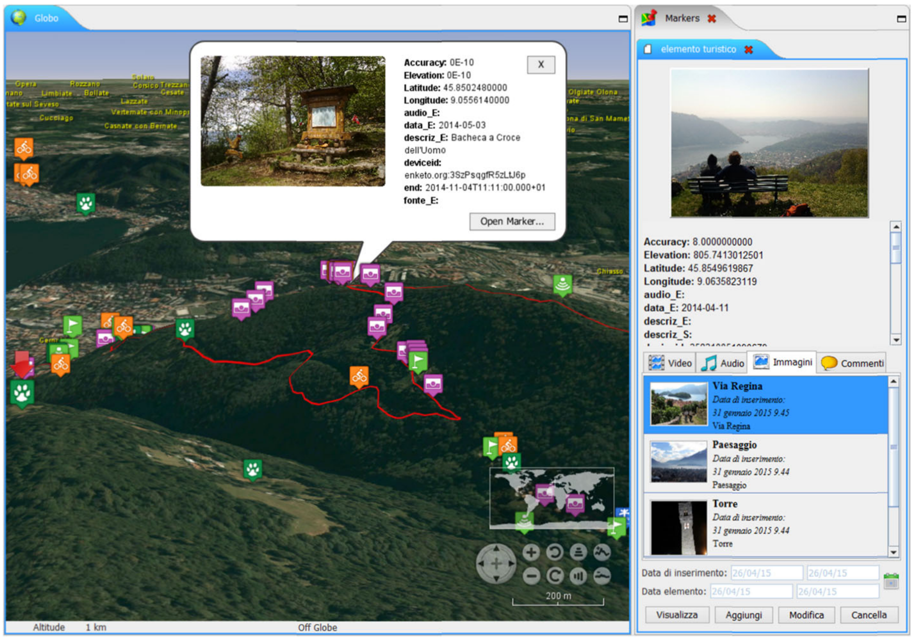

SERVER
Management of accounts (authentications & authorizations), projects and multimedia contents
Desktop CLIENT
It allows to interact with the data uploaded by the users
Mobile CLIENT
It allows to collect data on the field
PostgreSQL
JDBC
GlassFish
VGI SERVER
Restful API
Java Web Start APPLICATION
NASA World Wind
(CORE ENGINE)
android
MOBILE APP
OPEN DATA KIT
(ODK Collect)
WMS Servers
CERTIFIED OGC COMPLIANT
ODK Servers
OPEN DATA KIT

Fig. 9 The architecture of Polycrowd 2.0



Accuracy: 0E-10
Elevation: 0E-10
Latitude: 45.8502480000
Longitude: 9.0556140000
audio_E:
data_E: 2014-05-03
descriz_E: Bacheca a Croce dell'Uomo
deviceid: enketo.org:3SzPsqgfR5zLtJ6p
end: 2014-11-04T11:11:00.000+01
fonte_E:

Accuracy: 8.0000000000
Elevation: 805.7413012501
Latitude: 45.8549619867
Longitude: 9.0635823119
audio_E:
data_E: 2014-04-11
descriz_E:
descriz_S:

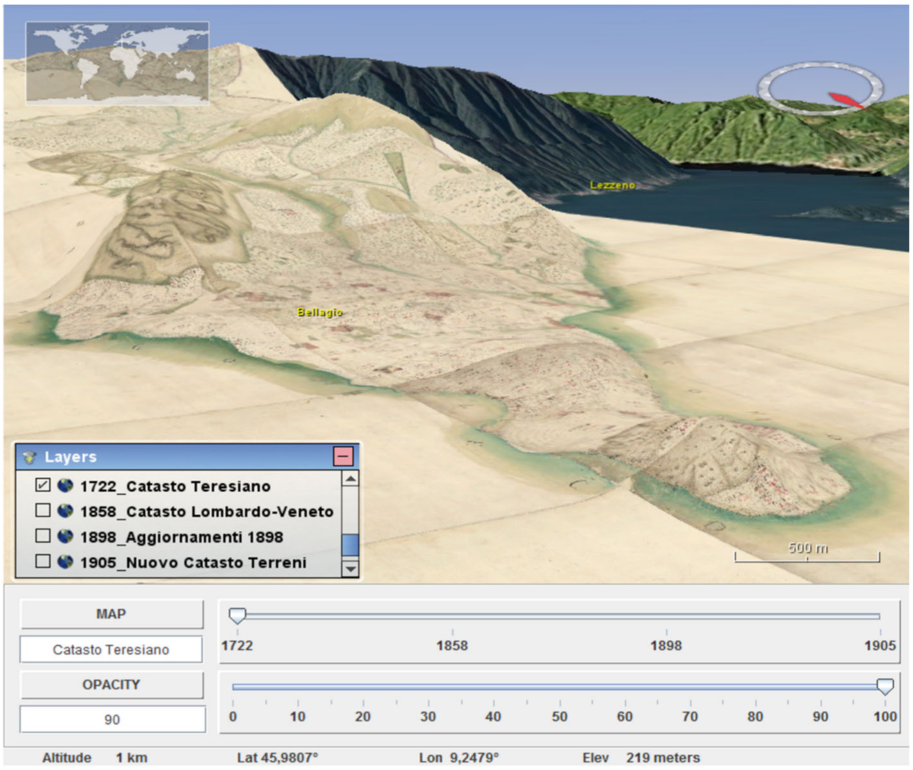
Via Regina
Data di inserimento:
31 gennaio 2015 9.45
Via Regina

Paesaggio
Data di inserimento:
31 gennaio 2015 9.44
Paesaggio

Torre
Data di inserimento:
31 gennaio 2015 9.44
Torre

Data di inserimento: 26/04/15 26/04/15
Data elemento: 26/04/15 26/04/15

Fig. 10 The display of ODK POIs as markers



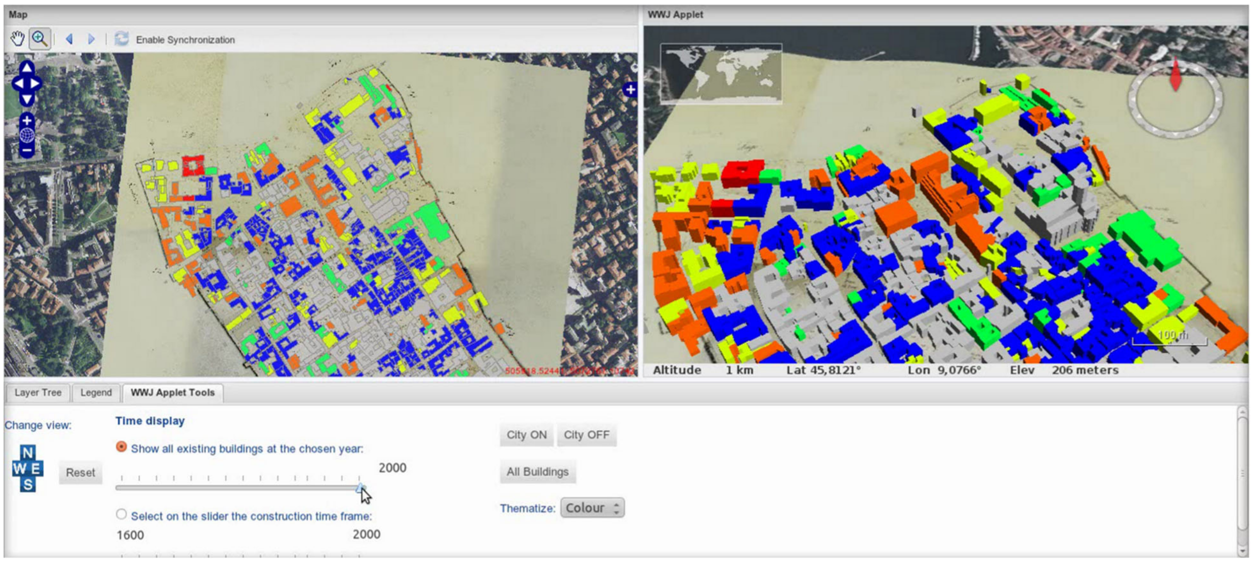
Layers
- [✓] 1722_Catasto Teresiano
- [ ] 1858_Catasto Lombardo-Veneto
- [ ] 1898_Aggiornamenti 1898
- [ ] 1905_Nuovo Catasto Terreni

MAP: Catasto Teresiano
OPACITY: 90

Altitude 1 km Lat 45,9807° Lon 9,2479° Elev 219 meters

Fig. 11 3D visualization of Theresian cadastral map

focused on tourism and cultural heritage [6], the current version, PoliCrowd 2.0, aims to be completely general-purpose and multi-thematic. It is based on the principle of Volunteered Geographic Information (VGI) to create projects according to participants' interests, as they can connect to any WMS and OpenDataKit (ODK) server [5, 10] and add information. The users can also share their projects, by saving them in a catalogue available for all the PoliCrowd users. Multimedia content (audio, photo and video) can be added to the catalogue using

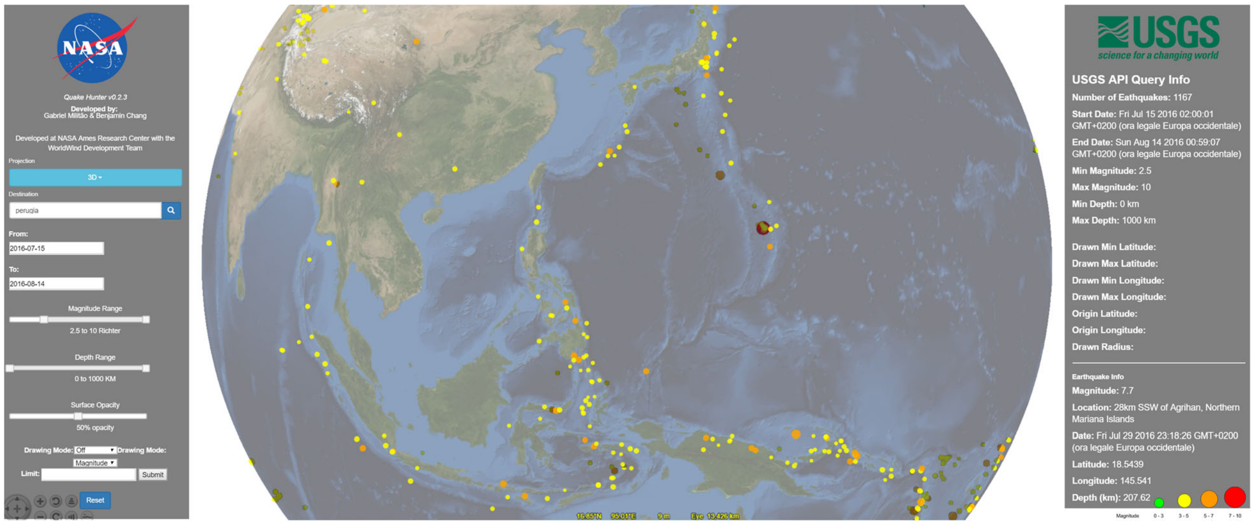


Altitude 1 km Lat 45,8121° Lon 9,0766° Elev 206 meters

Time display
- (●) Show all existing buildings at the chosen year: 2000
- ( ) Select on the slider the construction time frame: 1600 2000

Thematize: Colour

Fig. 12 Multi-dimensional platform with 2D e 3D synchronized panels



Quake Hunter v0.2.3
Developed by:
Gabriel Militão & Benjamin Chang

Developed at NASA Ames Research Center with the WorldWind Development Team

Destination: perugia
From: 2016-07-15
To: 2016-08-14
2.5 to 10 Richter
0 to 1000 KM
50% opacity
Drawing Mode: Off Drawing Mode: Magnitude
Limit: Submit

USGS API Query Info
Number of Eathquakes: 1167
Start Date: Fri Jul 15 2016 02:00:01 GMT+0200 (ora legale Europa occidentale)
End Date: Sun Aug 14 2016 00:59:07 GMT+0200 (ora legale Europa occidentale)
Min Magnitude: 2.5
Max Magnitude: 10
Min Depth: 0 km
Max Depth: 1000 km

Drawn Min Latitude:
Drawn Max Latitude:
Drawn Min Longitude:
Drawn Max Longitude:
Origin Latitude:
Origin Longitude:
Drawn Radius:

Earthquake Info
Magnitude: 7.7
Location: 28km SSW of Agrihan, Northern Mariana Islands
Date: Fri Jul 29 2016 23:18:26 GMT+0200 (ora legale Europa occidentale)
Latitude: 18.5439
Longitude: 145.541
Depth (km): 207.62

Fig. 13 Web app showing earthquake position, magnitude and depth

Android mobile devices and the data are georeferenced using the positioning technologies of the mobile device. The architecture of the system can be seen in Fig. 9.

Submitted data are stored in a spatial database and published following OGC standards. All the POIs (Points of Interest) inserted are displayed on the virtual globe as markers as in Fig. 10 and upon clicking on a marker, the information submitted is shown in a new window. ODK layers and multimedia content can be filtered by time, introducing the 4th dimension.

Historical data – WebCarte

Web C.A.R.T.E. system is a multi-frame and multi-dimensional platform, which allows users to visualize time-varying features on synchronised 2D and 3D panels [25]. The application provides access to historical maps of Como city, which are superimposed on the current local orthophoto and the topographic map showing city buildings and local DTM, i.e. the Lombardy Region DTM with pixel size of 20m and 2m (Fig. 11). A main innovation is represented by geographic synchronization of the panels, which allows users to look at the same portion of the Earth both on the 2D and the NASA World Wind based 3D viewers.

Available information about the building height, year of construction and year of demolition, allows users to both visualize the 3D city model on top of the globe and to filter the buildings visualization according to different criteria.

This means that the user can visualize all the existing buildings for a certain year, or just the buildings that were built during a specified time span (Fig. 12). A proper customization of World Wind virtual globe has therefore turned it into a working 4D Web viewer.

Data access and analysis – Quake Hunter

An earthquake viewer built using NASA Web World Wind. This web app shows time and location of the earthquake, magnitude and depth. Data are from USGS information on historic earthquakes. Users can query over 100 years of seismic data dynamically over the whole globe or on a smaller area by defining geographic boundaries over which to subset the data. The 3D capabilities allow analyses via depth and magnitude of each earthquake, for greater understanding of seismic events and their spatial correlation (Fig. 13) [15].

Education and information dissemination – Space Birds

This web app uses NASA Web World Wind to show known objects in Earth's orbit [8]. It immediately conveys the idea of how cluttered the space around Earth is (Fig. 14). SpaceBirds provides powerful query functions to identify specific platforms. Users can also inspect each object in space by interacting with the pointer.

Visual data analysis – World Weather

NASA Web World Wind has been deployed in this case to display a very broad range of weather and climate data from sources including NASA, ESA, NOAA. This web app brings together a multitude of data sources to give a comprehensive view of dynamics of the Earth's climate [24]. Figure 15 shows an example with two layers loaded, wind vectors and relative precipitation.

Current status and future developments

The NASA World Wind team is currently actively developing three platforms for the NASA World Wind application. The oldest one is the desktop implementation using Java, and was described in this article

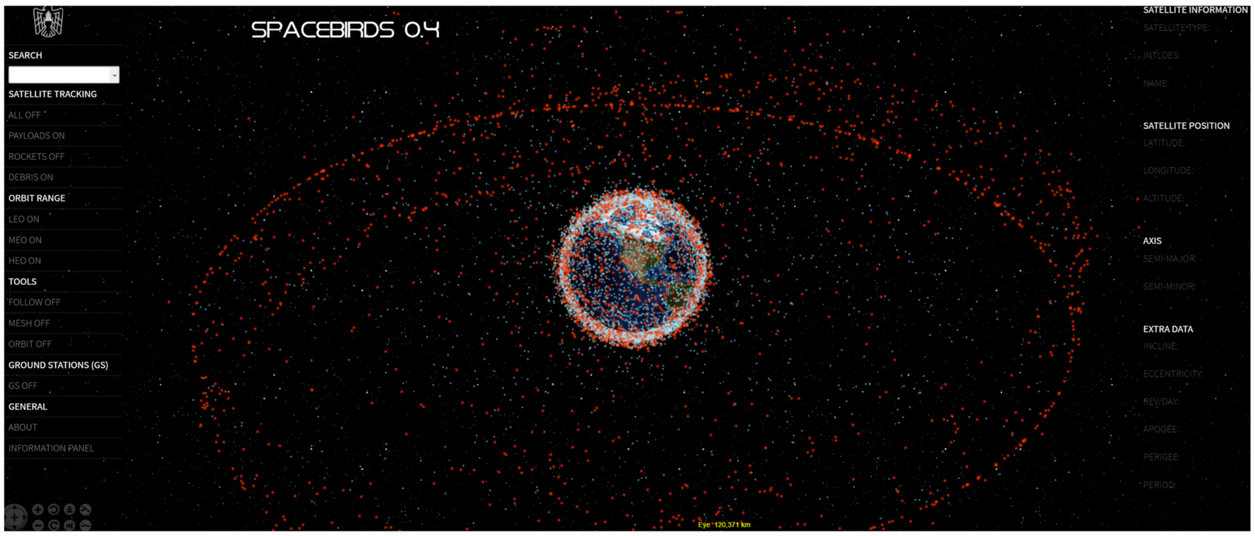


Fig. 14 Space Birds app in action: red points are payload satellites, white points around earth are debris

along with its web implementation using JavaScript. Implementations for Android and Web platforms are undergoing highly active development. Requirements and constraints of the Android and Web demos are available from the NASA World Wind's main web page https://nasaworldwind.github.io/.

Among the latest features under development, it is worth to mention the implementation of a 'picking' functionality to enable users to pick an object available in the chosen screen point. Also the atmosphere effect with a day/night cycle has been developed alongside some unit tests to allow quality assured integration for each feature and in future they could be reused in other projects.

Development is also active on the Web version of NASA World Wind, now composed by a team that involves contributors from NASA, ESA and Thales. This group is

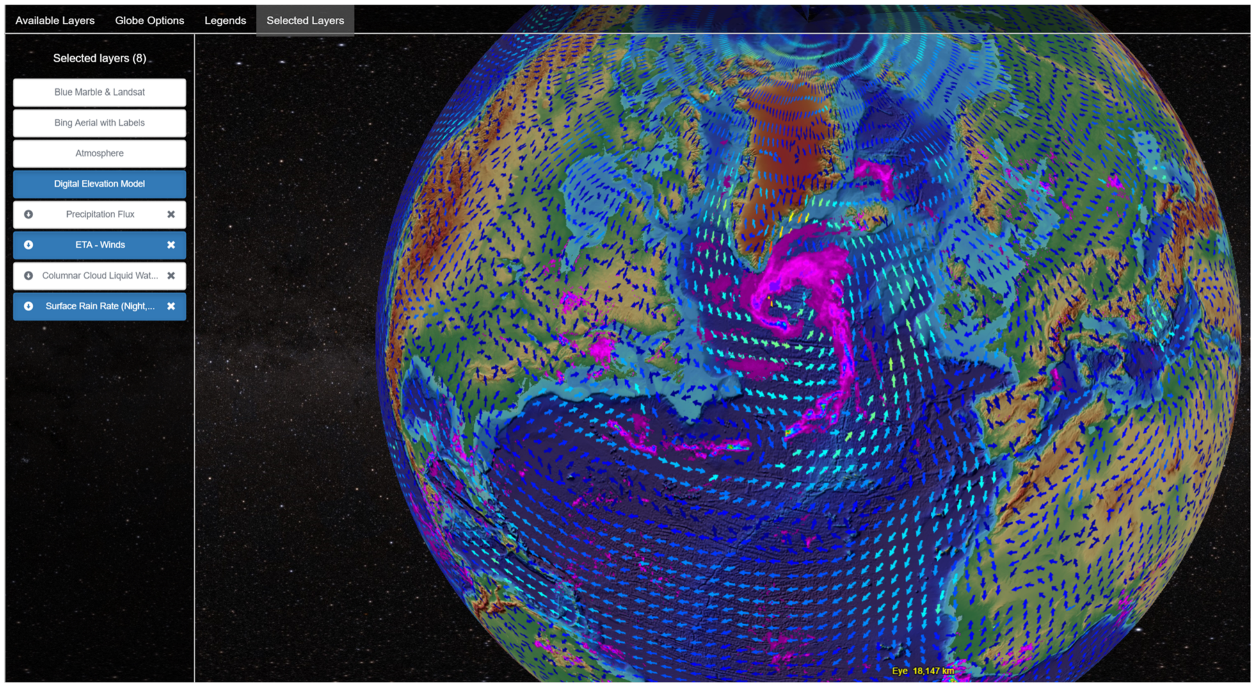


Fig. 15 World weather app with winds layer and surface rain rate layer

working on porting many of the Java features to the Web World Wind SDK. They are also focusing on the performance. To accomplish this, the programmers continually analyze the code base for performance issues, and research improvements including feedback from the user community.

At the time of writing this article some interesting features currently under development are about to be released thanks to the concentrated and collaborative effort of programmers. These new features allow importing KML, GeoJSON and Collada spatial formats. Another important feature which will soon be released is OGC's WCS service, which is still under active development.

Conclusions

In this article we have tried to provide the reader with an overview of the NASA World Wind project, outlining some existing applications and highlighting its potential for collaborative development in a wide range of applications for many aspects of science.

Participation and collaboration is a fundamental part of open source projects [1, 2], and NASA World Wind thrives through the effort of young developers and the experience and dedication of mentors.

In this era, spatial data are growing exponentially in terms of volume and variety, and are often simply referred to as "bigdata". This brings new opportunities, but also new challenges, in the ways we approach analysis and visualization of spatial data. NASA World Wind provides the platform for accurate and performant visualization of spatial data. This greatly facilitates the ability for scientists and software engineers to develop important analyses for better understanding of Earth, its environment, dynamics and complexity.

NASA World Wind improves every day thanks to a passionate community and is available across all platforms, for the benefit of all.

Availability and requirements

The following table represents the available code repositories for the different NASA World Wind implementation. In this article the two main SDKs are described, but it must be mentioned that also an Android SDK is being developed and is available.

Funding
The research for this contribution did not receive any funding.

Authors' contributions
FP carried out researching relevant information and did the main editing. MAB and GZ developed the part regarding Java-based solutions. MP worked in completing the description on implementation. GP and EK contributed to descriptions in the solutions presented in the result section. GP collaborated also in the description regarding future developments of the NASA World Wind API. PH overviewed the structure and contents of all text and contributed to introduction. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Land, Environment, Agriculture And Forestry (TESAF) & Interdepartmental Research Center of Geomatics (CIRGEO), University of Padova, Via dell'Università 16, 35020 Legnaro, Italy. [2]Department of Civil and Environmental Engineering, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italy. [3]NASA Ames Research Center, Moffett Field, CA 94035-1000, USA.

Received: 18 August 2016 Accepted: 2 February 2017
Published online: 27 February 2017

References

1. Arias de Reyna M, Simoes J. Empowering citizen science through free and open source GIS. Open Geospatial Data Softw Stand. 2016;1:7. doi:10.1186/s40965-016-0008-x.
2. Bakillah M, Liang S. Open geospatial data, software and standards. Open Geospatial Data Softw Stand. 2016;1:1. doi:10.1186/s40965-016-0004-1.
3. Battersby SE, Finn MP, Usery EL, Yamamoto KH. Implications of Web Mercator and Its Use in Online Mapping. Cartogr Int J Geogr Inf Geovisualization. 2014;49:85–101. doi:10.3138/carto.49.2.2313.
4. Bell DG, Kuehnel F, Maxwell C, et al. NASA World Wind: Opensource GIS for Mission Operations. In: 2007 IEEE Aerospace Conference, IEEE. 2007. p. 1–9.
5. Brovelli MA, Minghini M, Zamboni G. Public Participation GIS: a FOSS architecture enabling field-data collection. Int J Digit Earth. 2014;8:1–19. doi:10.1080/17538947.2014.887150.
6. Brovelli MA, Zamboni G. Virtual globes for 4D environmental analysis. Appl Geomatics. 2012;4:163–72. doi:10.1007/s12518-012-0091-3.
7. Cozzi P, Ring K. 3D engine design for virtual globes. CRC Press; 2011.
8. Del Castillo M, Stewart B. SpaceBirds: The world of satellites at your fingertips. 2016.
9. Evenden G, Warmerdam F. Proj. 4–cartographic projections library. 1990.
10. Hartung C, Lerer A, Anokwa Y, et al. Open data kit: tools to build information services for developing regions. In: Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development. 2010. p. 18.
11. Herring J. Opengis implementation standard for geographic information-simple feature access. 2010.
12. Hogan P, Coughlan J. NASA World Wind, Open Source 4D Geospatial Visualization Platform: *.NET & Java*. In: AGU Fall Meeting Abstracts. 2006.
13. Hogan P, Gaskins T. GeoSpatial Visual Analytics. Dordrecht: Springer Netherlands; 2009.
14. MapQuest I. Nominatim Search Service Developer's Guide. 2016. https://open.mapquestapi.com/nominatim/. Accessed 12 Sep 2016.
15. Militão G. Chang B. Earthquake Activity Visualizer: Quake Hunter; 2016. https://github.com/NASAWorldWindResearch/Quake-Hunter.
16. Miller JR, Gaskins T. Computations on an Ellipsoid for GIS. Comput Aided Des Appl. 2009;6:575–83.
17. NASA World Wind Developers. World Wind JAVA SDK API Documentation. 2016. https://github.com/NASAWorldWind/WorldWindJava/releases. Accessed 10 Aug 2016.
18. National Aeronautics and Space Administration. Nasa open source agreement version 1.3. 2016. https://en.wikipedia.org/wiki/NASA_Open_Source_Agreement. Accessed 10 Sep 2016.
19. OGC. OGC Web Services Common Specification. 2007.
20. Piragnolo M, Masiero A, Fissore F, Pirotti F. Solar Irradiance Modelling with NASA WW GIS Environment. ISPRS Int J Geo Info. 2015;4:711–24. doi:10.3390/ijgi4020711.
21. Piragnolo M, Pirotti F, Guarnieri A, et al. Geo-Spatial Support for Assessment of Anthropic Impact on Biodiversity. ISPRS Int J Geo Info. 2014;3:599–618. doi:10.3390/ijgi3020599.
22. Pirotti F, Guarnieri A, Vettore A. Collaborative Web-GIS design: a case study for road risk analysis and monitoring. Trans GIS. 2011;15:213–26. doi:10.1111/j.1467-9671.2011.01248.x.
23. Schubert B, Collins D. NASA World Wind organization website. 2016. https://nasaworldwind.github.io/. Accessed 29 Sep 2016.
24. Sharif K, Salah F. NASA World Weather. 2016. https://github.com/NASAWorldWindResearch/WorldWeather. Accessed 12 Sep 2016.

25. Valentini L, Brovelli MA, Zamboni G. Multi-frame and multi-dimensional historical digital cities: the Como example. Int J Digit Earth. 2014;7:336–50.
26. Ward K. Remote sensing imagery from NASA Earth Observations (NEO). 2016. http://neowms.sci.gsfc.nasa.gov/wms/wms. Accessed 12 Sep 2016.
27. Warmerdam F, Open Source Geospatial Foundation. GDAL - Geospatial Data Abstraction Library. 2012. http://www.gdal.org/.
28. Warmerdam F, Open Source Geospatial Foundation. GDAL Raster Formats. In: GDAL - Geospatial Data Abstr. Libr. 2016a. http://www.gdal.org/formats_list.html. Accessed 3 Sep 2016.
29. Warmerdam F, Open Source Geospatial Foundation. GDAL Vector Formats. In: GDAL - Geospatial Data Abstr. Libr. 2016b. http://www.gdal.org/ogr_formats.html. Accessed 3 Sep 2016.