User-friendly interfaces for Web GIS

Bruno Simões[1], **Giuseppe Conti**[1], **and Raffaele De Amicis**[1]
[1]Fondazione Graphitech, Trento, Trentino, Italy

Abstract— *This paper describes a geospatial information system (GIS) that can be used for several mission-critical activities, such as management of risky activities, and urban planning. The main novelty of this system, stands on the ability to analyze and present data inside a 3D media-rich geo-referenced environment. The 3D user interface suggested is based on the idea of interoperable processing units that can be directly manipulated and linked to create complex process chains; thus, freeing the operator from unnecessary information or constraints on the analytic reasoning. Additionally, users can use this interface to visually program and deploy web accessible algorithms. This is an important aspect within collaborative activities, where is often necessary to create new bridges between different knowledge areas in response to unexpected situations.*

Keywords: geovisualisation, interactive 3D graphics, service oriented architectures, user-interface paradigms

1. Introduction

Today's society is becoming everyday more rapidly vulnerable to natural disasters due to growing urban populations, environmental degradation, lack of planning, management and preparedness. This scenario is bound to worsen in the coming years, since extreme weather-related events are increasing, particularly floods and droughts [1], [2].

Spatial planning and simulation, land management and monitoring provide the missing basis for taking precautions against catastrophes or hazards. In order to face this problem we can make use of a new generation of geospatial technologies that allow experts to capture, store, process and display an unprecedented amount of information about the environment and a wide variety of phenomena [3], [4]. However, the extremely wide range of data available from large and heterogeneous datasets (e.g. high-resolution satellite imagery and digital maps that can exceed several pentabytes; and economic, social and demographic information) poses a critical challenge in operators that need to move from data to information, to awareness, to pieces of knowledge [3].

1.1 Interactive Tools For Environmental Management

Environmental control is largely based on management and processing of very complex databases providing a variety of different spatial information on the environment [4]. This information has to be improved with information retrieved from monitoring systems. An ideal system should support operators with the largest possible set of automatic or semi-automatic processes which should be capable to support them in interpreting specific data patterns, potentially triggering early warnings.

When dealing with geospatial information, be this a physical map, a web-based mapping applications, or a more interactive 3D system, the first priority is to determine where the user is looking at and where the information is located. A user-friendly framework is essential when dealing with mapping applications [4] and the latest generation of 3D interactive applications are the ideal candidates to provide the most user-friendly experience. As consequence, in the last few years we have witnessed a large market response to this demand resulting in the development of 3D applications, engineered for use by the large public, such as Google Earth [5], Microsoft Virtual Earth [6], NASA World Wind [7] as well as several other 3D geospatial solutions designed for professional use. Those applications, often referred to as Virtual Globes or 3D Geobrowsers, have been designed to provide high usability in terms of navigation and data visualization.

The ease with which the user can interact with the data and the environment they are immersed in, is essential to facilitate exchange and dissemination of spatial information among stakeholders and government agencies.

1.2 From Data to Knowledge Through Interaction

Navigation and data access must be complemented with functions that enable operators to analyse interrelations between spatial information, data patterns and environmental effects. 3D interactive interfaces can provide a fundamental support to effectively understand complex spatial relationships among information and dataset, for instance to define the most appropriate evacuation routes or to plot an emergency plan [4].

However the very nature of this process, which is intrinsically cooperative -as diverse competences are involved- underlines the importance of developing user-friendly universal interfaces, essential to achieve short training time, ease of use and fast response. Further a user-friendly interface is essential to avoid programming of simulation procedures, which notably requires extensive training and time. The use of visual programming languages can ease the definition of complex simulation process.

The main contribution of this work is to show how 3D interfaces can be used to sustain abstract visual metaphors in combination with a human information discourse in order to enable detection of unexpected events within massive, dynamically changing information spaces.

The system presented provides interactive capabilities that allow intuitive control of the data available and, most importantly, of the various processing tools required by the operator. Furthermore, the system supports visual programming and deployment of complex simulation processes. Programming is performed through the use of visual expressions, spatial arrangements of text and graphic symbols, used both as syntactic elements as well as secondary components. Visual components can be manipulated by users in an interactive way thus making simulations easier and faster to build and debug. New algorithms can be visually created without any knowledge about the service and server architecture.

2. Related Work

A vast body of work on general principles in 3D navigation, interaction and visualization can be found in literature. With specific regards to the field of environmental control, 3D GIS have received considerable attention and the number of works found in the literature on this topic is constantly growing.

Card et al [8] have a vast work showing that when information is presented visually, efficient innate human capabilities can be used to perceive and process data. Information visualization techniques that amplify cognition by the increase human mental resources, can reduce search times, improve recognition of patterns, increase inference making, and increase monitoring scope.

Thorndyke and Hayes-Roth [9], as well as many others authors [10], [11], [12], [13], [14], have studied the differences in spatial knowledge acquired from maps and from their exploration. Darken and other authors [15], [11], [16], [13] have explored cognitive and design principles and how these are applied to large virtual worlds. Furnas explored view traversability and navigability for effective navigation through large data structures [17], [13].

In 2002, Zlatanova et al. [18] edited a survey of mainstream GIS software, where they reviewed a number of systems including: ArcGIS, Imagine VirtualGIS [19], PAMAP GIS Topographer [20] and GeoMedia Terrain [21]. The authors found that some initial steps forward had been made in terms of visualization of 3D spatial data, mainly through the extension of the map metaphor to 3D [22], [23]. Further advancements found by the authors comprise the inclusion of some main requirements such as multi-resolution and multi-view representations, real-time rendering, interactivity, and high visual quality [24]. This has been possible through the adoption of algorithms and data structures specifically developed for efficient visualization of terrains which had been extensively studied in the past (e.g., multi-resolution geometry modelling [25], [26] and multi-resolution texture modelling [27], [28]).

Further examples of existing 3D research prototypes include Terrafly [29], GeoVR [30] and TerraVisionII [31]. A noteworthy system, called GeoTime [32] proposes an interesting solution to the problem of integrating timeline events into interactive GIS. The Adelaide 3D GIS planning model [33] is an example of a 3D GIS which provides 3D visualizations of social and environmental data within a 3D cityscape [34]. Two other interesting applications are GeoZui3D [35] and VGIS [36]. GeoZui3D is a 3D marine GIS that supports real-time input and draped imagery. VGIS [36] is an integrated global GIS and visual simulation system [34] and finally Heidelberg 3D which is a client that allows the user to explore and analyse the 3D city and landscape models which are streamed by the W3DS server [37], [38].

These applications, besides showing clear bottlenecks in the manipulation of 3D data and generic 3D analysis [18], often lack even simple GIS function [34] such as querying, usually performed through the adoption of standard WIMP (Windows, Icons, Menus and Pointers) metaphor [3], [39]. This clearly produces a cognitive barrier that does not allow to visualize and manipulate geographical information in a realistic and natural manner [39].

3. Application Overview

The system presented in this paper allows users to interact with large 3D geo-referenced environments. The software has been developed on top of Java World Wind [7] APIs and it can be deployed as standalone application, applet or as Java Web Start [40].

Furthermore, the system provides support for the access and automation of multiple-step procedures known as workflows by providing a rich set of tools and mechanisms to combine a series of tools into a sequence of operations using models [41].

The types of tasks to be automated can be very diverse in their nature, ranging for example from being able to predict the path of wildfire, to analysing and finding patterns in crime locations, to predicting which areas are prone to landslides, to predicting flooding effects of a storm event. Workflows can be potentially very complex sequences of elementary operations necessary to model and analyse complex spatial relationships. For that reason, the interface allows performing operations on different datasets (such as a feature classes, raster images, or database tables) and to produce a new dataset as result. Additionally, the interface also supports common geoprocessing operations, including geographic feature overlay, feature selection and analysis, topology processing, raster processing, data conversion, projecting a dataset from one map projection to another, adding a field to a table, or creating a buffer zone around features. All these functionalities, which are provided as Web Processing Services (WPS), are interactively available to the user through the user interface.

Additionally, operators can create new WPS components through the system UI without having to worry about all the issues related with the creation or deployment of a web process service. The WPS component will be compiled and deployed, at the server side, in a completely transparent way.

4. 3D Interface for Web Processing

In a previous work [42], we described a server architecture using interoperable protocols such as Web Processing Service (WPS), Web Map Service (WMS) and Web Feature Service (WFS). In this paper, we describe how to benefit from that interoperability. The only information required is the address of a web service supporting geo-processes across the web (WPS). The system will automatically update the menu with all the new geoprocessing services discovered at that service address. Their category and name is automatically retrieved from the WPS metadata and are used to structure the menu layout.

4.1 Creating Complex Process Chains

Once the user clicks on one process, the corresponding 3D process can be placed on the desired position and dragged over the terrain. To maximize usability and readability, the interface automatically sticks to the nearest terrain point, and process icons are rendered as billboards - always facing the user. Besides that, interface occlusion rarely happens due to the level of detail and to the nature of the navigation. A process can be composed up to three distinct types of components: the process controller, input and output slots.

As visible in Figure 1, the process controller has four distinct buttons. The first button (starting from the left side) is used to remove the process from the 3D environment. The remaining buttons are respectively used to execute or stop the process, as detailed in the following sections, and to obtain a process description.

Fig. 1: Process representation

Every process operates as a black box that can receive input and transmit results to a further process via its output slots. Each input and output slots is automatically created from the process descriptor exposed by the WPS, and represented using distinct icons. In this way, it is immediate the identification of its name and mime type. Being completely automatic, this architecture scales very well in case of availability of a large number of services.

In the example of Figure 1 the slot at the left represents an output (double) that can be connected to a compatible input of any other processing unit. Input slots (right) can be connected to some output (left) from another process or connected to results of user inputs (for instance when the user selects an area within the virtual scene).

The icon associated with each slot turns green if there is some data available on it (see for instance the slot on right in Figure 1). An additional button on the right side of the slot can be used, depending on the data type, to save the XML content as a file, to visualize text in a window, to represent a 3D shape within the environment, to render an image on the terrain, etc.

To create a connection between two processing units, the user has to drag the input slot over the output slot (or vice-versa) as illustrated in Figure 2.

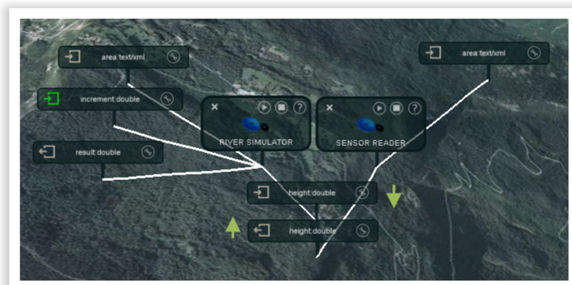

Fig. 2: Creating a connection

When the user drags an in-slot close to an out-slot they snap together and are replaced by the resulting slot shown in Figure 3. This last component is logically analogous to a bridge (both visually and practically). This way the user can create complex processing structures by connecting in- and out- bound connectors. If the user needs to break a connection, the task can be simply archived by clicking on the button indicated by the arrow on Figure 3. When this is done the process chain is broken and the original in- and out- slots become available to the user. As result, the system gets back to the state illustrated in Figure 2.

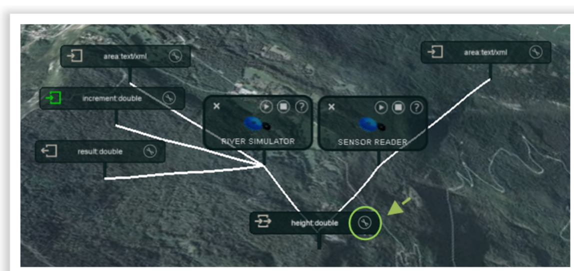

Fig. 3: Breaking a connection

As each block of the chain represents a processing service, the operators can perform very articulated web-service orchestration tasks in a completely transparent way, by just interacting with icons directly within the 3D representation of the scene. This approach has proven to be extremely user-friendly and it allows performing complex processing tasks with very little training.

Furthermore, processing functionalities can be constrained to specific areas of the territory. Therefore, users can interactively select within the 3D scene, a specific area containing the area of interest (see Figure 4). Once the process chain is created, it can be activated by pressing the relevant icon on the process controller. Most importantly, as the processing takes place at the server level, this is asynchronous by its very nature, meaning that the 3D client continues to respond in a fully interactive manner also when complex simulations are being executed.

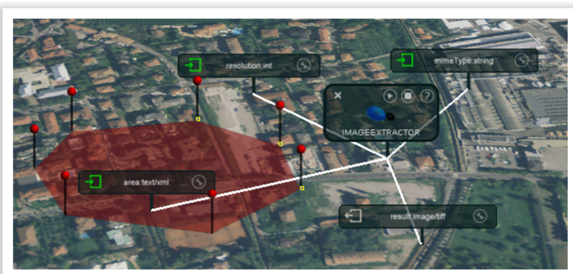

Fig. 4: Process chainning

4.2 Web Process Suggestion

The system has also the ability to suggest possible processes that can be used within the working scenario. This not only reduces search-time (supposing the process is known by the operator) but also informs the operator about new related processes, which may not have been available before. By using this feature, users can learn about new potential-useful algorithms without affecting their normal routine (Note that WPS protocol provides mechanisms to get information about processes). Besides that, the suggestion interface is only displayed when the user selects an input or output slot (see figure 5).

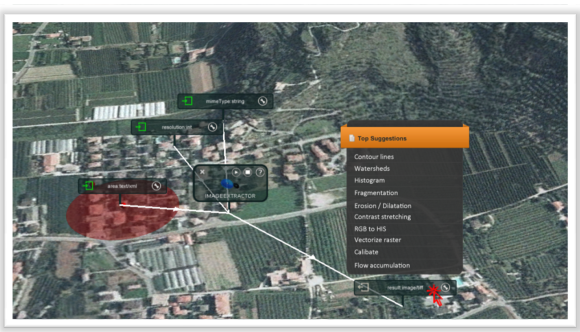

Fig. 5: Process suggestion

The algorithm used to suggest a list of processes is based on the following formula:

$$\omega(X) = \max_{10}\left\{ \frac{1}{g(\alpha)f(\alpha)} + f(\alpha)\chi(\alpha, X) \mid \forall\alpha \in \Omega(X) \right\} \quad (1)$$

where X represents the set of processes currently in use, Ω, the set of compatible processes (using type matching), $g(\alpha) \in [0, \ldots, 1]$, the usage's frequency of the process by the operator, $f(\alpha)$, the matching parameters percentage, and $\chi(\alpha, X)$, the semantic relation evaluator, which is described by the following equation:

$$\chi(\alpha, X) = \sum_{\beta \in X} 0.35\rho(\alpha, \beta) + 0.55\varphi(\alpha, \beta) + 0.10\phi(\alpha, \beta) \quad (2)$$

Where $\rho(\alpha, \beta)$ evaluates the relevance of data to be processed (e.g. complex geometry as advantage over numeric data), $\varphi(\alpha, \beta)$ gives a score to questions such as *Where*? *When*? and *What?* (In the example illustrated in Section 7, *Where* is the user selection, *When* is given by the simulation process and finally *What* is who is living there) and $\phi(\alpha, \beta)$ is a score for sharing the same computation group (e.g. simulation, image analysis).

5. Extending the WPS protocol

Currently, the WPS standard does not provide any support for hot deployment or source code retrieval. Yet, these are relevant features within collaborative activities. For that reason we will introduce and describe three new operations. The first operation is called *StoreNewAlgorithm* and it allows the user to deploy new algorithms into a web server. For this operation we have to specify a XML document that describes the process in the standard way (*ProcessDescription* element), and provides its respective source code. Eventually, the request can be associated to an expire time, after which the algorithm will be removed.

A parameter *IDGEN* should also be provided to describe how the process ID is defined, or how the response from the server is handled when the ID is duplicated (See Table 1). Note that the algorithm identifier should be unique because it is used to identify the process.

IDGEN value	Action
GenerateNew (default)	The WPS will generate a unique identifier for this algorithm.
UseExisting	In response to an operation *StoreNewAlgorithm* a WPS uses the value of the IDENTIFIER attribute as identifier for the algorithm. If the IDENTIFIER is already been used, the WPS respond with an exception.
ReplaceWhenDuplicated	A WPS client can request to the service to generate an IDENTIFIER in order to replace the identifier if duplicated. In this case, the WPS server will not respond with a duplicated IDENTIFIER exception.

Table 1: Possible IDGEN Values

The source code has to be included in the XML file, delimited by an element called *Source*, which is composed by a list of one or more sub-elements *File*. Each element *File* stores a source code block, as well as the programming language name, the original file name, and if it contains a main function.

The following example illustrates an example of a *StoreNewAlgorithm* request.

```
<?xml version="1.0"?>
  <wps:StoreNewAlgorithm
  version="1.3.0"
  service="WPS"
  idgen="ReplaceWhenDuplicated"
  xmlns="http://www.someserver.com/myns"
  xmlns:ows="http://www.opengeospatial.net/ows"
  xmlns:wps="http://www.opengis.net/wps"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.someserver.com/myns
    http://www.opengis.net/wps ../wps/1.3.0/WPS.xsd">

    <ProcessDescription
      processVersion="2"
      sourceAvaliable="true"
      statusSupported="false">
        ...
    </ProcessDescription>
    <Source>
      <File
        name="TestAlgorithm.Java"
        language="Java"
        main="true">
          R0lGODlhZABq ... base-64 ...
      </File>
      <File name="sample.bin">
        R0lGODlhZABq ... base-64 ...
      </File>
    <Source>
</wps:StoreNewAlgorithm>
```

The second operation proposed, extends the WPS protocol in order to provide access to source code. If users can have access to the source code, then it can easily be extended, fixed or improved.

However this feature may not be desired for all processes due to multiple reasons. Thus, it should only be possible to obtain the source code of those algorithms that specify the attribute *sourceAvaliable* as *true*, within *ProcessDescription*.

The user will receive an error message when isn't possible to retrieve the source code. The format in which is possible to receive the source code can also be specified through the parameter FORMAT.

An example of a request would be:

```
HTTP://www.someserver.com/path/wps?SERVICE=WPS&VERSION↩
    =1.3.0&REQUEST=GetAlgorithm&IDENTIFIER=TestProcess&↩
    FORMAT=XML
```

The last addition will be the operation to remove a process. This operation can be combined with *StoreNewAlgorithm* to perform update operations.

```
HTTP://www.someserver.com/path/wps?SERVICE=WPS&VERSION↩
    =1.3.0&REQUEST=RemoveAlgorithm&IDENTIFIER=↩
    TestProcess
```

Additionally servers can implement security measures through regular security mechanisms such as user authentication, in order to control who can deploy or remove new algorithms.

5.1 Protecting server from malicious code

Protecting servers with authentication mechanisms it is simply not enough, since they can't block users from uploading malicious algorithms. Therefore, new security mechanisms are required to address these new security flaws.

One of the mechanisms that can be used and ensure some good level of security is the use of the Java Security Manager. *SecurityManager* was mechanisms to monitor and control all the permissions of the running processes. In the most drastic case, it can deny access to any resource (be it local or remote) or the access to other objects/code.

6. Hot algorithm deployment

To facilitate the creation of new algorithms, a 2D widget is rendered in overlay. This strategy allows the operator to work simultaneously in 2D and 3D. Initially, the widget contains only one icon that represents the core algorithm. By clicking on it, the operator is able to write the algorithm according to one of the supported programming languages. When required, new input or output parameters can be added by pressing the button "plus". Whenever a new parameter is added, a new icon will become visible. This new icon can be used to define specific parameters such as name, type (input or output), data type and if applicable the default value. Depending on slot type, the icon will have different representations.

The user can inspect if the process has default values associated by checking the icon color, as previously mentioned - it will be green when some value is assigned (See example in Figure 6 - icon on the left).

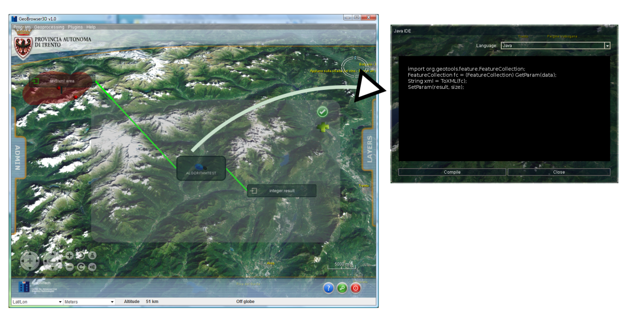

Fig. 6: Process Creation

These slots will became part of the 3D context when dragged outside the 2D widget. So, they can be connected to any selection or object, and that will automatically become the default value. As illustrated in Figure 6, the input was associated with an area on the ground, dynamically selected by the user.

A new window will be visible if the user selects the icon that represents the algorithm core. Within this new window it is possible to select the programming language, write the algorithm code and have a run-time diagnostic of programming errors. Java and Javascript are among the supported languages.

During the programming phase, the programmer has only to care about the algorithm and does not need to be aware of how it will work at the server side as a web service. Furthermore, to facilitate this abstraction, there are some macros that can dramatically simplify the code. Some examples are *FromXML(Object)* and *ToXML(Object)*, which can be used to convert XML into native objects and vice-versa. *GetParam(String)* on the other hand, can be used to obtain the value of one input parameter through its name, while *SetParam(String, String)* can be used to set the value for an output parameter.

When the entire process structure is ready, the user can write an abstract for each component (algorithm, inputs and outputs) as well as keywords that later will support catalogues against client's searches.

If the algorithm passes all tests, then it will be automatically published online, becoming at the same time immediately available to everyone.

When a newly added algorithm is selected from the menu, it will appear inside the 3D environment with all its predefined values (e.g. associated to a selection of some particular area).

7. Use Case

This section describes a practical use case and illustrates the potential of the interface. The use case consists in a simulation, where users have to raise the current river level and identify people potentially affected by the flooding.

As shown in Figure 7, a first process is used to identify and retrieve water levels from sensors, accessible through web services and within a given region. The relevant sensor data is passed as output onto a second processing unit (2nd). The second processing unit will create a flood using a detailed terrain model available at the server side, and delimited by a selected area passed as further input. The output of this process, is passed as input together with the affected area to a final processing unit (3rd), which retrieves from a spatial database the list of streets within the affected areas. Then invokes a public white pages web service to identify the telephone numbers of those potentially affected by the flood. Their address is in turn passed to Yahoo Geocoding API that returns the corresponding spatial location, used to render a green pin in the scene. Further, operators can elaborate evacuation plans taking in account attributes such as people distribution, roadworks, optimal paths, etc. using similar processes.

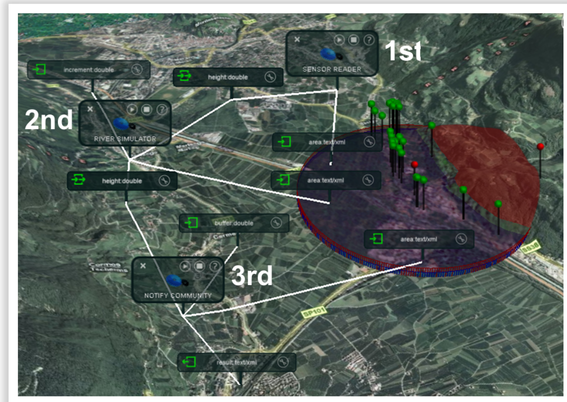

Fig. 7: Flooding Simulation

7.1 Testing And Validation

The usability test, based on the ISOMETRIC test [43], has involved 10 users with different expertise. One was younger than 25, six aged between 25 and 35, two between 35 and 45 and the last one older than 45. 70% of the sample were males, seven were students and the remaining post-graduates with no specific experience with 3D user interfaces. Only one of the users had previous knowledge in GIS systems.

Generally speaking users have clearly indicated that the functions implemented in the software support them performing their work (arithmetic mean of 5.0/5.0) and they are not forced to perform any task not related (arithmetic mean of 5.0/5.0).

In terms of self descriptiveness the test has shown that the users were satisfied with the interface. However it is clear that future effort is required to further improve the system by the possibility to easily retrieve information about a certain entry field (arithmetic mean of 3.2/5.0). The interface has proved quite clear to use (arithmetic mean of 4.1/5.0) with the user perceiving clearly the concept transmitted by the interface (arithmetic mean of 4.0/5.0).

The application can be considered appropriated for learning, since users have indicated that they did not need long time to use the software (arithmetic mean of 3.5/5.0), and it is supposedly easy to relearn to use after a lengthy interruption (arithmetic mean of 4.4/5.0). Finally, users stated that there isn't much need to remember many instructions to properly use the software (arithmetic mean of 2.5/5.0).

8. Conclusion

Today interoperability in the domain of geo-visualisation and geovisual analytics is starting to become a reality thanks to several international harmonization efforts trying to solve the problem by offering standards for management of geographical information. However, several issues regarding data access and management, 3D visualization and analysis need still to be tackled.

The work described in this paper provides an example of 3D interfaces applied to the context of visualization, analysis and processing of geographical information, and illustrates that future developments should maximize usability to ensure a profound understanding of the respective information space. Specifically, this framework gives the user the opportunity to work with any information available within the system by simply drag-and-drop connectors. This avoids not only the need to memorize object ids, which are not always known but also the need to perform unnecessary searches. Another useful contribution is an extension of the WPS protocol in a way that allows users to build new processes from scratch and share them on-fly with other users.

Besides that, our experiments have indicated that this user interface can be considered suitable and self-descriptive for a given task, breaking the complexity of most standard GIS systems and allowing any decision maker to make use of it, with virtually no training.

References

[1] V. F. Grasso and A. Singh, "Global environmental alert service (geas)," *Advances in Space Research*, vol. 41, no. 11, pp. 1836–1852, 2008.

[2] V. Grasso, G. Cervone, A. Singh, and M. Kafatos, *Global environmental alert service*. Fall Meeting, 2006.

[3] A. Gore. (1988) The digital Earth: Understanding our planet in 21st Century. [Online]. Available: http://www.isde5.org/al_gore_speech.htm

[4] T. Kotter, "Prevention of environmental disasters by spatial planning and land management," in *2nd FIG Regional Conference*, Marrakech, Morocco, 2006.

[5] Google. (2009) Google Earth. [Online]. Available: http://earth.google.com

[6] Microsoft. (2009) Microsoft Virtual Earth. [Online]. Available: http://www.microsoft.com/VirtualEarth

[7] NASA. (2009) World Wind. [Online]. Available: http://worldwind.arc.nasa.gov

[8] S. M. J. Card and B. Shneiderman, *Readings in Information VisualizationReadings in Information Visualization*. San Francisco, CA: Morgan Kaufmann Publishers, 1999.

[9] P. Thorndyke, "Differences in spatial knowledge acquired from maps and navigation," *Cognitive Psychology*, vol. 14, no. 4, pp. 560–589, October 1982.

[10] E. Hunt and D. Waller, "Orientation and wayfinding: A review," Tech. Rep., 1999.

[11] W. Robinett and R. Holloway, "Implementation of flying, scaling and grabbing in virtual worlds," in *Proc. SI3D'92*, Cambridge, Massachusetts, United States, 1992, pp. 189–192.

[12] A. W. Siegel and S. White, "The development of spatial representations of large-scale environments," *Advances in Child Development and Behavior*, vol. 10, pp. 10–55, 1975.

[13] D. S. Tan, G. G. Robertson, and M. Czerwinski, "Exploring 3d navigation: combining speed-coupled flying with orbiting," in *Proc. CHI'01*, Seattle, Washington, United States, 2001, pp. 418–425.

[14] B. G. Witmer, J. H. Bailey, B. W. Knerr, and K. C. Parsons, "Virtual spaces and real world places: transfer of route knowledge," *International Journal of Human-Computer Studies*, vol. 45, no. 4, pp. 413–428, 1996.

[15] R. P. Darken and J. L. Sibert, "Navigating large virtual spaces," *International Journal of Human-Computer Interaction*, vol. 8, no. 1, pp. 49–71, 1996.

[16] N. G. Vinson, "Design guidelines for landmarks to support navigation in virtual environments," in *Proc. CHI'99*, Pittsburgh, Pennsylvania, United States, 1999, pp. 278–285.

[17] G. W. Furnas, "Effective view navigation," in *Proc CHI'97*, Atlanta, Georgia, United States, 1997, pp. 367–374.

[18] S. Zlatanova, A. Rahman, and M. Pilouk, "Trends in 3d gis development," *Journal of Geospatial Engineering*, vol. 4, pp. 1 – 10, 2002.

[19] ERDAS. (2011) Imagine Virtual GIS. [Online]. Available: http://www.erdas.com

[20] P. Geomatics. (2008) PAMAP GIS Topographer. [Online]. Available: http://www.pcigeomatics.com

[21] Intergraph SG&I. (2008) Intergraph SG&I - Geospatially Powered Solution. [Online]. Available: http://www.intergraph.com

[22] C. Haeberling, "Symbolization in topographic 3d-maps: conceptual aspects for user-oriented design," in *Proc. ICC'99*, Ottawa, 1999.

[23] J. Döllner and O. Kersting, "Dynamic 3d maps as visual interfaces for spatio-temporal data," in *Proc. GIS'00*, Washington, D.C., United States, 2000, pp. 115–120.

[24] A. Terribilini, "Maps in transition: Development of interactive vector-based topographic 3d-maps," in *Proc. ICC'99*, Ottawa, 1999, pp. 993–1001.

[25] H. Hoppe, "Smooth view-dependent level-of-detail control and its application to terrain rendering," in *Proc. VIS'98*, Research Triangle Park, North Carolina, United States, 1998, pp. 35–42.

[26] P. Lindstrom and V. Pascucci, "Visualization of large terrains made easy," in *Proc. VIS'01*, San Diego, California, 2001, pp. 363–371.

[27] D. Cline and P. Egbert, "Interactive display of very large textures," 1998, pp. 343–350.

[28] J. Döllner, K. Baumman, and K. Hinrichs, "Texturing techniques for terrain visualization," in *Proc. VIS'00*, Salt Lake City, Utah, United States, 2000, pp. 227–234.

[29] N. Rishe, Y. Sun, M. Chekmasov, A. Selivonenko, and S. Graham, "System architecture for 3d terrafly online gis," in *Proc. ISMSE'04*, Washington, DC, USA, 2004, pp. 273–276.

[30] B. Huang and H. Lin, "Geovr: a web-based tool for virtual reality presentation from 2d gis data," *Computers and Geosciences*, vol. 25, pp. 1167–1175, 1999.

[31] M. Reddy, Y. Leclerc, L. Iverson, and N. Bletter, "Terravision ii: Visualizing massive terrain databases in vrml," *IEEE Computer Graphics and Applications*, vol. 19, no. 2, pp. 30–38, 1999.

[32] T. Kapler and W. Wright, "Geotime information visualization," in *Proc. INFOVIS'04*, Washington, DC, USA, 2004, pp. 25–32.

[33] GISCA. (2008) Adelaide model. [Online]. Available: http://www.gisca.adelaide.edu.au

[34] S. Brooks and J. Whalley, "Multilayer hybrid visualizations to support 3d gis," *Computers, Environment and Urban Systems*, vol. 32, no. 4, pp. 278–292, July 2008.

[35] C. Ware, M. Plumlee, R. Arsenault, L. Mayer, and S. Smith, "Geozui3d: data fusion for interpreting oceanographic data," vol. 3, 2001, pp. 1960–1964.

[36] D. Koller, P. Lindstrom, W. Ribarsky, L. Hodges, N. Faust, and G. Turner, "Virtual gis: a real-time 3d geographic information system," in *Visualization, 1995. Visualization '95. Proceedings., IEEE Conference on*, 1995, pp. 94–100, 443.

[37] A. Zipf. (2008) Heidelberg-3d - interactive 3d city mapping based on ogc standards. [Online]. Available: http://gislounge.com

[38] ——, "Integrating 3d, processing and location services into future sdis," in *Proc GSDI'10*, Port of Spain. Trinidad, 2008.

[39] F. Liarokapis, J. Raper, and V. Brujic-okretic, "Navigating within the urban environment using location and orientation-based services," in *Proc. ENCE'06*, 2006, pp. 7–10.

[40] Oracle. (2009) Java Web Start. [Online]. Available: http://www.oracle.com/

[41] J. Encarnacao, R. D. Amicis, G. Conti, and J. Rix, "Beyond foss 3d gis technologies: a chance for developing countries," in *Proc. FOSS4G'08*, 2008.

[42] B. Simões, G. Conti, S. Piffer, and R. de Amicis, "Enterprise-level architecture for interactive web-based 3d visualization of geo-referenced repositories," in *Proc. Web3D'09*, Darmstadt, Germany, 2009, pp. 147–154.

[43] G. Gediga and K. C. Hamborg. (2001) ISOMETRICS. [Online]. Available: http://www.isometrics.uni-osnabrueck.de/index.htm