GIS WEB SERVICES AND SATELLITE IMAGING AS A FRAMEWORK FOR ENVIRONMENTAL MONITORING: THE DESIGN AND IMPLEMENTATION OF A VEGETATION INDICES CALCULATION TEST CASE

D. Kotzinos [a, *], N. Chrysoulakis [b]

[a] Computer Science Department, University of Crete, P.O. Box 2208 71110 Heraklio, Greece - kotzino@csd.uoc.gr
[b] Institute of Applied and Computational Mathematics, Foundation for Research and technology - Hellas, P.O. Box 1527, 71110 Heraklio, Greece – zedd2@iacm.forth.gr

KEY WORDS: Spatial Information Sciences, GIS, Internet/Web, Interoperability, Imagery, Satellite, Global-Environmental-Databases

ABSTRACT:

GIS web services bring to the Geographic information arena the promise to deliver algorithmic implementations and data services to the broader community, following a simple yet powerful paradigm. Under this paradigm, a solution to a problem is sought by providing individual services that can be used either in combination or standalone, as part of a greater application to solve the bigger problem. The main advantage is that the service can be used remotely without the user's actual knowledge and intervention and by multiple users at the same time, eliminating the need for constant updates to locally installed software. Moreover it minimizes the network traffic, since data do not need to be transferred to the client in every step of the operation. This overall framework is currently being developed and tested and the one major test application presented is the development as a web service of an algorithm that calculates vegetation indices using low spatial resolution satellite images. To achieve the overall result two separate web services are developed: a) One handling the acquisition and manipulation of the satellite images being used. A low-resolution satellite ground receiving station is used to provide real time satellite data. b) A second one implementing a vegetation index calculation algorithm, using results form the first web service. It's worth noting here that the system has the potential to be expanded by more algorithms (e.g. for surface temperature estimation, for aerosol spatial distribution monitoring, for perceptible water calculation) that will still use one or both of the already developed web services, leading to a services-chaining effect. Web services can by nature be used by both web-based and desktop applications alike so, as part of this work, two small client applications are developed – one web-based and one desktop, in order to showcase the applicability and availability of the framework. End users will use the two already mentioned clients and consequently the final result, implementing as a side effect a user interface, since web services provide only a programming interface.

1. INTRODUCTION

A great deal of progress has been made in Earth Observation (EO) over the last years. This has been driven largely by the realization that the observation of global climate processes requires the type of spatial and temporal coverage only afforded by satellite remote sensing. In spite of the real time accessibility of EO data such as NOAA (National Oceanic and Atmospheric Administration) images offered by ground receiving stations, most potential users refrain from using them because either the format of the raw data and the processing procedures are complex, or already processed data offered by institutions or companies are not always tailored to the exact needs of the end-user. Web-based geospatial technology provides the means to enhance the dissemination of satellite-derived information and to assist potential users with the processing of raw satellite data. In the field of environmental information science one should take into account the number and diversity of the organizations dealing with the production, storage and manipulation of geographic related data. The volume of available geographic-related environmental data is increasing rapidly, but technical and semantic non-interoperability prevents easy sharing and integration of geospatial information. The internet revolution, through the use of internet GIS technologies (Kotzinos and Prastacos, 2001) although expanded the possibilities, has added one additional problem: large amounts of data need to be moved among users and providers, so as to enable the first to perform their designated tasks. Since geographic-related data are usually large in size this exchange becomes more and more difficult despite the improvements in communication technology.

The last two years a new paradigm has entered the computing world and is changing the way all these issues are handled. This new paradigm is web services, which are based on the use of universally accepted standards and bring data and services together and both of them closer to the end user. Web services are trying to turn the web from a medium of simply accessing information to one that will bring together data discovery, data access and data processing along with information retrieval. GIS web services are a more profound paradigm of web services due to the diversity and size of both geographic data and software, as mentioned earlier. Using the new paradigm users will be able to access both looking only for the particular service or data-set they need. They will not be concerned about format, availability and price, since they will "pay" and access only for what they ask for.

In this paper an effort has been made to design and describe a framework of GIS web services tailored to the needs of the

* Corresponding author.

environmental modelling community. The framework will support functions that can be used by desktop and web-based applications alike in order to perform environmental modelling tasks using satellite images. In the next section a more detail presentation of the value, applicability and functioning of GIS web services is presented and reviewed. Section 3 describes the methodology used to calculate the Normalized Difference Vegetation Index (NDVI), along with the necessary satellite images. Section 4 presents the design and implementation of the proposed web services framework and describes in detail the interconnections among the different pieces of software and hardware. Finally, section 5 concludes the work presented in this paper by summarizing the research conducted so far and suggesting future work.

2. GIS WEB SERVICES

2.1 Web Services

Web Services is a constantly emerging technology that allows many diverse internet based applications to interact in order to exchange data and software. They are one, rather significant, instance of the new service based computing paradigm. Web services are actually providing application-to-application communication over the internet, although applications involved do not have to be explicitly web based. Software developers do not have any more the restriction of only local or restricted remote access to software modules. Applications can now automatically search and discover other software modules that can be used to perform specific tasks. These modules are well-described following technologies and standards that, although still in the standardization process, are well and widely accepted. Both data and software can be provided through a service. The way the service operates is transparent to both the developer, who only describes his own needs, and the end user.

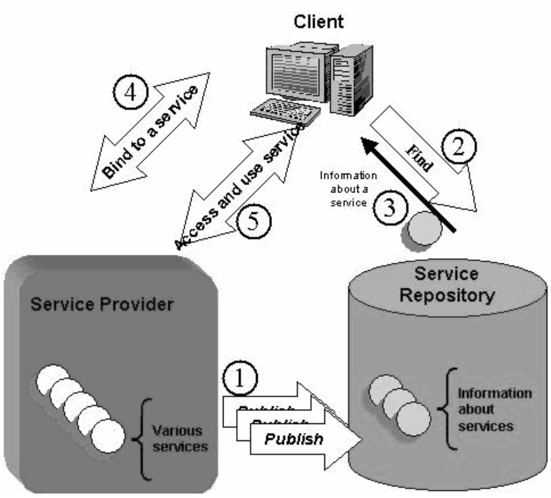

Figure 1. The web service access model - taken from (Saganich, 2001).

Web services are based on XML (eXtensible Markup Language), which is a description language widely used (xml.org, 2004). Many XML based technologies are used to describe, communicate with and finally use a web service. These technologies described in the next paragraph have become the de facto standards in the area and enjoy almost global adoption even form rival companies (like Microsoft, SUN, IBM, etc.). Having described that, one can point out a twofold benefit situation for developers: The architecture is both (a) language and system neutral and ignorant and (b) software and/or data neutral (in the sense described above).

But how do web services work? As it can be also seen in Figure 2, there are three major players involved in every web service interaction. There is first of all the client who issues the request describing the required service (the *requestor*). This request is passed to a *Service Repository*, which handles it. Depending on implementation, the repository will either provide a link to a suitable service, if it finds one, or if it does not, it will either pass it to another known service repository or just inform the requestor that such a service is not available. The link that the service repository returns to the client corresponds to the service provided by the *service provider*, who is the last vertex of the triangle. At the provider's location the requestor gets information on what to expect form the service and how to actually call it from within its own code.

2.1.1 Chaining of Web Services: Finally one of the biggest advantages of web services is their ability to be chained. By creating a web services chain one can expand the possibilities of the developed application. Many different and diverse information sources and software can be combined by forcing the one to use as input the output of the other, providing in that way an application (or broader speaking an information source) with extreme value added.

2.2 Related Technologies

There are many different technologies and languages – mostly based on XML - involved in the construction, publication and use of a web service. The technologies discussed herein are not related with the actual construction of a web service per se. For that purpose any programming language can be used (and in this case *any* is literally used). Most of these technologies are self-descriptive and they describe the usage of the service or the messages being passed among the services and the applications. Each one plays a specific role and each one is both agnostic and transparent to the other. These technologies include:

- **UDDI** (Universal Description, Discovery and Integration) (uddi.org, 2004), is used by services in order to register themselves at one or more repositories. It is absolutely cross-platform and provides a platform independent open framework for describing, discovering and integrating services.
- **WSDL** (Web Services Description Language) (W3C, 2004), is an XML based file which is used to describe network services as a set of points exchanging messages that can be either data or procedure oriented.
- **SOAP** (Simple Object Access Protocol) (W3C, 2004), which is the communication protocol of choice – although it is not the only one used. The biggest advantage of SOAP (apart the fact that it is XML based and thus universally and seamlessly used) is that it can use various different transport protocols as the messages transport mechanism, including HTTP.

The previously described technologies allow the seamlessly integration of services, information about services and use of services. They are playing a crucial role in the proper use of the service and the dissemination of the information or the procedure. Taking into consideration fact that the same service can be used by both desktop and web based applications, one can see that the role of the description, discovery and invocation of a web service is equally important to the service itself.

2.3 GIS Web Services

Geographic Information Science has a lot to benefit by the adoption of the service computing model. As mentioned also in the introduction geographic information comes from different and diverse sources and in different formats. This is especially true for the environmental related information which has to combine not only data form different sources but also models and software. The GIS web services initiative is mainly driven by the Open GIS Consortium (OGC) (OGC, 2004). OGC has provided specifications for many web based applications like the OpenGIS Web Map Server and data descriptions like GML (Geography Markup Language). But this only makes the integration easier. GIS web services require the use of technologies described in section 2.2 in order to become truly available to users.

Environmental monitoring applications in particular, like the one described in this paper, are extremely suitable to be developed under the web services model. These applications are based upon the availability of large volumes of data sets in the form of both a large enough number of satellite images, each one having a size of several megabytes, and a number of intermediate images produced by the specific algorithms used. In a application developed under the traditional software model, one (but everyone) would have been obliged either to download and keep all these data in the local machine (in the case of a desktop application) or use a web-based application with a more or less restricted functionality.

3. SATELLITE IMAGERY AND VEGETATION INDICES

Considering vegetation as a functional equivalent of terrestrial ecosystems, it follows that changes in vegetation structural dynamics provide important indications for physical process in space and time. A widely used parameter to investigate vegetation conditions is the Normalized Difference Vegetation Index (NDVI), which uses the strong reflection of living vegetation in the near infrared region of the electromagnetic spectrum and the relatively low reflection in the visible red wavelength (Lillesand & Kiefer, 1987). Thus, the NDVI is the difference of near-infrared (NIR) and visible (VIS) reflectance values, derived from multispectral satellite images, normalized over the sum of the respective image channels: (NIR-VIS)/(NIR+VIS). Several works have processed and made available global or regional NDVI data sets for subsequent use, either directly in modelling applications or in studies to extract land cover information (Los et al., 1994; Sellers et al., 1994; El Saleous et al., 2000).

3.1 Satellite images

The AVHRR (Advanced Very High Resolution Radiometer) onboard NOAA satellites is a radiation detection imager that can be used for remotely determining the vegetation status of the earth surface (Cracknell, 1997). AVHRR was a 4-channel radiometer, first carried on TIROS-N (1978). This was subsequently improved to a 5-channel instrument (AVHRR/2) that was initially carried on NOAA-7 (1981). The latest instrument version is AVHRR/3, first carried on NOAA-15 (1998). AVHRR/3 records the incoming radiation in 6 spectral bands: 0.580 - 0.680 μm (Channel 1), 0.725 – 1.100 μm (Channel 2), 1.580 – 1.640 μm (Channel 3A), 3.550 – 3.930 μm (Channel 3B), 10.300 - 11.300 μm (Channel 4) and 11.500-12.500 μm (Channel 5). Channels 1 and 2 are the VIS and NIR chnnels, respectively, whereas both 4 and 5 are thermal infrared (TIR) channels. Each AVHRR pass provides a 2399 Km wide swath with a ground resolution of 1.1 Km at nadir from the nominal orbit altitude of 833 Km.

NOAA satellites orbit the Earth 14 times per day. The AVHRR provides on board collection of data from all spectral channels. These data are coded in 10 bits and transmitted to ground receiving stations. The FORTH (Foundation for Research and Technology - Hellas) ground receiving station provides all required data to the developed GIS web services. FORTH station carries the advantage of near real time AVHRR image acquisition from NOAA 12, 14, 15, 16 and 17 satellites. All images used by the GIS web services are pre-processed using station's capabilities (Dartcom, 2002). Pre-processing includes the following three steps: a) Polar navigation and re-projection to a 1.1 x 1.1 Km cell grid (Hugget and Opie, 2002). b) Subset and conversion to a 5-layer raster with no change to the original pixels' Digital Numbers (DN). c) Export to the application's database.

3.2 The NDVI algorithm

The NDVI is calculated for cloud free land areas taking into account the received radiation in AVHRR channels 1 and 2:

$$NDVI = \frac{Channel2 - Channel1}{Channel2 + Channel1} \qquad (1)$$

Eq.(1) produces values in the range of -1 to 1, where increasing positive values indicate increasing green vegetation and negative values indicate non-vegetated surface features. AVHRR channels 1, 2 and 5 used by the NDVI algorithm. Channels 1 and 2 are used to implement Eq. (1), whereas channels 1 and 5 are used for cloud masking.

In practice, the NDVI algorithm consists of the following steps:

a) Calibration of channels 1, 2 and 5 to convert pixel's DN to radiance values ($Wm^{-2}\mu m^{-1}sr^{-1}$), using the pre-launch calibration coefficients issued by NOAA (Kidwell, 1998; Goodrum et al., 2000).
b) Production of a cloud mask (Chrysoulakis and Cartalis, 2002).
c) Production of an intermediate NDVI image using the Eq. (1).
d) Production of the final NDVI image combining steps b) and c).

3.3 The Implementation of the NDVI algorithm

The implementation of each algorithm step is performed by a different module of a Java-based in house software. In particular, each module consists of one or more programme classes, which have been designed using the Java2 programming language. AVHRR image manipulation is achieved with the use of the immediate mode imaging model of Java 2D API (Application Programming Interface.), which is a set of classes for advanced two-dimensional graphics and imaging (Pantham, 2000). AVHRR channels 1, 2 and 5 are converted to “BufferedImage” objects using BufferedImage class functionality. A BufferedImage object contains two other objects: a "Raster" and a "ColorModel". The Raster class provides the image data management, whereas the ColorModel class provides a colour interpretation of the pixel data provided by the Raster. The Java2 code implementing the NDVI algorithm consists of 5 different modules, which are activated remotely by the GIS web services:

a) Image Viewer Module: Used to view the produced images. Not used for the web service implementation.
b) Calibration Module: It consists of three different classes performing the image calibration with the use of pre-launch calibration coefficients. It produces a new Buffered Image for each AVHRR channel.
c) NDVI Module: This module uses the calibrated channels 1 and 2 and produces the intermediate NDVI image according to the Eq. (1).
d) Cloud Mask Module: It uses the calibrated channels 1 and 5 and produces the cloud mask image according to Chrysoulakis and Cartalis (2002).
e) Export Module: This module combines the cloud mask image and the intermediate NDVI image and produces the final NDVI image. In the latter, all pixels corresponded to clouds have been masked as shown in Figure 2. Finally, the NDVI product is stored in the application’s database.

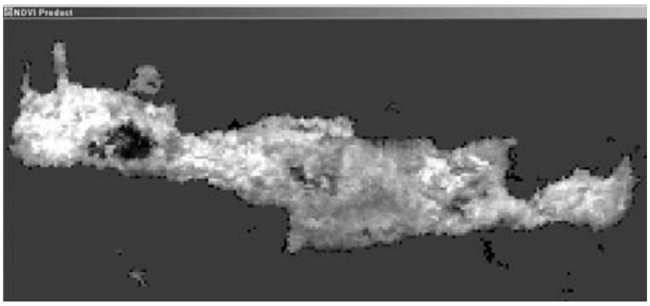

Figure 2. The NDVI product for Crete, Greece on June 18, 2003 (NOAA17/AVHRR).

4. DESIGN AND IMPLEMENTATION OF THE ENVIRONMENTAL WEB SERVICES

In this section an overall description of the design of an environmental web services framework is provided. Moreover both the implementation and deployment of the services are discussed. Specifically the tools and the framework utilized in the implementation are presented, along with the developed services. The NDVI algorithm was chosen as a demonstration because it is a simple algorithm and rather simple to implement using only satellite data, since our focus is on the services of the framework. Furthermore, it can be used to calculate other parameters like land surface temperature.

4.1 Servers and services

As described in Section 2, there are various actions taking place in the server side of the web services interaction process. Mainly the “service” component is a server based component, which inevitably will return some results to the requesting client. Many tools are available in order to facilitate building of web services. Our goal was to focus on open source tools and environments on the one hand (making sure beforehand that all the required functionality will be there) and on easily deployed and used services on the other, since we are targeting a non computer science audience.

To create a server environment for the proposed framework a simple LINUX box is adequate to perform the server role. The Apache™ Web server (Apache Foundation, 2004), along with TOMCAT™ (Apache Foundation, 2004) as the servlet container, was used to facilitate web server support. TOMCAT is a full blown servlet container that provides support through AXIS (Apache Foundation, 2004) for web services. It provides a mechanism to accept and dispatch SOAP messages in order to facilitate communication among the server and the requesting client.

The overall implementation for both the server and the client is done in Java. Java consolidates almost all of the required capabilities for such a system, being both platform and system independent, having an extensive background support and use in the web services arena, being extensively used and supported by the open source community and finally having already in place the required APIs for such a task.

For the services implementation the J2EE (SUN Microsystems, 2004) platform was the underlying toolkit. On J2EE is based the implementation of AXIS, that is used for supporting web services creation and discovery. AXIS provides also tools for automatic generation of the description XML based documents for each service. These documents describe in WSDL the required inputs and expected output of the service. The overall architecture of the framework is depicted in Figure3.

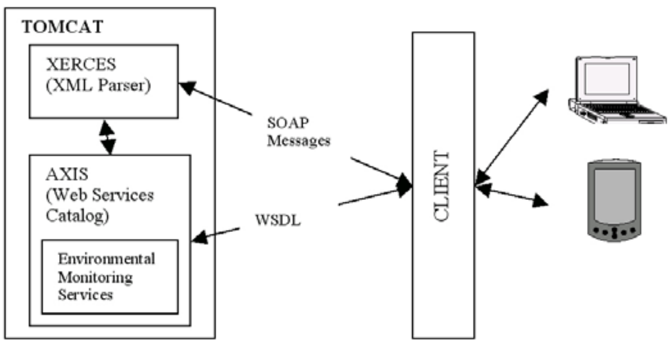

Figure 3. The overall web services’ architecture.

4.2 Imaging service

The first building block of the framework is the imaging web service. This service takes as input NOAA satellite images as described in Section 3. These images are undergoing a specific transformation procedure, making them available for use by the NDVI calculation web service. The satellite imaging service can also be used as a standalone service in case users would like to take into their hands not the processed NDVI images but the raw satellite data, with or without a performed transformation.

This service handles satellite images in order to provide them either to the user directly or to the service following this one. It can participate in services chaining usually being the ring in the chain to be formed. The necessary controls have been implemented in the service’s code to check for the availability and suitability of the images. It also configurable if the image has to be sent to the end user or should be "forwarded" to another service (saving that way time and bandwidth, especially if the next service is running out of the same server, minimizing interaction with the end user). Users can also select the required area. The way this is done is left to the client application (the web service cares only about the bounding rectangle's coordinates and not of the way they were chosen).

4.3 NDVI calculation service

This is the web service that actually implements online the NDVI calculation algorithm described in Section 3. This is a chained web service: it needs the satellite imaging web service in order to get the satellite images in the required format. The algorithm produces in turn a series of intermediate images, which can also become available to the end user by user's request through the web service interface. The output of the web service is a processed NDVI image. The distinct steps of the calculation algorithm can be transparent or not depending on the user's request. The whole processing takes place at the server and the user is only receiving the final image, thus reducing the overall downloading times and wasted network bandwidth.

There are several open issues in the area. It seems the ability to provide a caching scheme for the produced images. Users might select again and again the same images, so such a scheme would have saved redundant and repeating computations. On the other hand a balance has to be implemented in order for the system to actually take into account the proposed use of the service in order to decide if it needs to reproduce the same image (for example one image for download purposes only might be treated differently than an image that needs to be used in NDVI calculations). Another interesting issue is the ability to formally describe the produced NDVI images. This introduces the necessity of a semantic description language like RDF/S or OWL and the production of a corresponding ontology.

4.4 Publication and Discovery of services

Although the need to publish the web services in a service repository is planned in the framework design, so as to allow of easy and seamless discovery of the services, it is not implemented yet. The small number of the implemented web services does not make it necessary for such a publication since a link can be provided to the requesting client through the Web Service Description Language (WSDL).

But the lack of such a universal repository makes publication and discovery rather a difficult task at the moment. This does not affect the functionality of the proposed framework is planned for future development, being associated with the fact that the overall design calls for the broader possible availability of the services. The AXIS web services framework supports the addition of publication and discovery catalogues at a later stage and this step will follow. The necessary messages (SOAP messages) are already being produced by the framework, making the adoption to the use of a cataloguing service straightforward.

4.5 Client Applications

Clients are being developed in conjunction to the proposed web services framework in order to provide a demonstration means of it. Clients are showing the ability of almost anybody to connect to the framework by calling and using the appropriate web service. The developed so far clients are targeted the expert environmental modelling community, instead they are just demonstrating the feasibility and availability of services. Anybody is invited to built its own client, either desktop or web based, upon the framework, which is and will remain open to the environmental community.

Two kinds of clients to the environmental monitoring web services are being developed. The first one is a web based client, which presents the information to the end users through a standard web site. Users can supply through web forms the required input (like the area of interest, the index they want to be calculated, etc). The processed information is returned to them through one or more web pages. The web-based client application does not require any installation from the user side and only allows the user to save the resulting image.

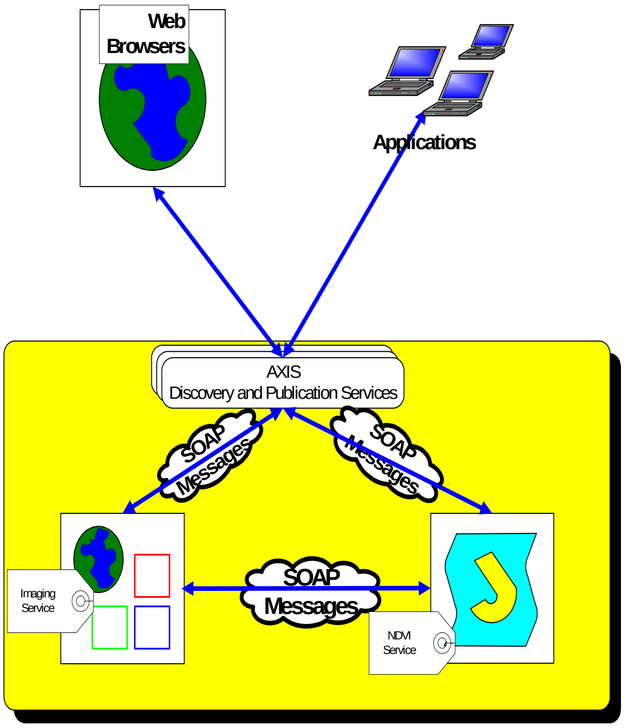

Figure 4. The chained web services communicate inside the framework using SOAP messages.

The local client application is "calling" the web services as it would have done with any software library bundled with it. The results are downloaded through the internet to the local application and extra processing can occur at the local level. This application can combine these data with locally stored data or with data form other external sources or web services. This shows that superior functionality can be incorporated to a local client application but on the other hand such an application is more demanding form the client machine. Usually a desktop application requires some kind of installation and of course an internet connection in order to access the web services and retrieve the results. It is important to note here that the client application can be developed totally independently from the services framework and can be in any language that supports use of web services. The overall approach is based on the production and exchange of the corresponding SOAP messages that describe both the necessary results and the communication between the services themselves and the services and the framework (Figure 4).

5. CONCLUSIONS

Web services allow the exposure of an application programming interface over the web to be accessed remotely. Web services are implemented using industry agreed on standards (XML, SOAP, HTTP), thus holding the promise of wider use than competing technologies. By combining relatively simple individual Web services with other distributed services, a more complex service can be provided in a platform independent way. The design of a GIS web service for NDVI estimation using satellite imagery was discussed in this study. Near real time NOAA/AVHRR data provided by the FORTH satellite ground receiving station were used. NDVI calculations were performed by Java-based in house software, which was provided for use by end users through the developed GIS web services' framework.

Future research will be concentrated on providing automatic registration, description and discovery capabilities to the system, developing a pricing system (although one is not planned to be used shortly since the service is supposed to remain free for the scientific community at large) and mainly integrating more environmental related web services based on the already available satellite images. These can include but are limited to: a service for image orthorectification, a service that will provide vector data for the area of interest so as to give the user the ability to combine different web services and services that will enable the calculation of other environmental parameters like surface temperature and perceptible water, aerosol distribution etc.

REFERENCES

Apache Foundation, 2004. http://www.apache.org.

Chrysoulakis, N. & Cartalis, C. 2002. A new algorithm for the detection of plumes caused by industrial accidents, on the basis of NOAA/AVHRR imagery, *International Journal of Remote Sensing*, 24, 3353 – 3367.

Cracknell, A.P. 1997. The Advanced Very High Resolution Radiometer. London: Taylor & Francis.

Dartcom, 2002. SIAMIV software user guide. Eighth edition. Dartcom, Devon, UK.

El Saleous, N.Z., Vermote, E.F., Justice, C.O., Townshend, J.R.G., Tucker , C.J. & Goward, S.N. 2000. Improvements in the global biospheric record from the Advanced Very High Resolution Rariometer (AVHRR), *International Journal of Remote Sensing*, 21, 1251 – 1277.

Goodrum, G., Kidwell K.B. & Winston, W. (eds) 2000. NOAA KLM Users Guide. U. S. Department of Commerce, National Oceanic and Atmospheric Administration, National Climatic Data Center, Suitland, USA.

Hugget, S. & Opie C. 2002. A fresh approach to automatic navigation of AVHRR images, *International Journal of Remote Sensing*, 23, 3783 - 3794.

Kidwell, K. B. (ed) 1998. NOAA Polar Orbiter Data Users Guide. U. S. Department of Commerce, National Oceanic and Atmospheric Administration, National Climatic Data Center, Suitland, USA.

Kotzinos, D. & Prastacos, P. 2001. Java-based applet for performing GIS on he Web, *Systems, Analysis Modeling and Simulation*, 41, 593-606.

Lillesand, M.T. & Kiefer, W.K. 1987. Remote Sensing and Image Interpretation. New York: John Wiley & Sons.

Los, S.O., Justice, C.O. & Tucker, C.J. 1994. A global 1x1 degree NDVI data set for climate studies derived from the GIMMS continental NDVI data, *International Journal of Remote Sensing*, 15, 3493 – 3518.

Pantham, S. 2000. Pure JFC 2D Graphics and Imaging. A Code – Intensive Premium Reference. Sams Publishing, Indianapolis, Indiana, USA.

Saganich, Al. 2001. Java and Web Services Primer, *http://www.onjava.com/lpt/a/1025.*

Sellers P.J., Tucker , C.J., Collatz, G.J., Los, S.O., Justice, C.O., Dazlich, D.A. & Randall, D.A. 1994. A global 1 degree by 1 degree data set for climate studies. Part 2. the generation of global fields of terrestrial biophysical parameters from the NDVI, *Science*, 275, 502 – 509.

SUN Microsystems, 2004. Java 2 Enterprise Edition Web site, http://java.sun.com/j2ee

Uddi.org. 2004. http://www.uddi.org.

W3C. 2004. http://www.w3.org/2000/xp/Group/.

W3C. 2004. http://www.w3.org/Graphics/SVG/Overview.htm8.

Xml.org. 2004. http://www.xml.org.