OneGlobe -
Building and Browsing a Transient Digital Earth from Distributed, Heterogeneous Sources

Gunnar Misund, Morten Granlund and Herman Kolås

Østfold University College, Faculty of Computer Science, Halden, Norway
{gunnar.misund, morten.granlund, herman.kolaas}@hiof.no

Abstract. We propose a low-cost and easy-to-maintain approach to generating interactive 3D visualizations of geodata on a global scale. The proposed OneGlobe infrastructure is in short a dynamically choreographed chain of web based geodata services complying to open specifications. The geobrowser is a thin client (VRML viewer), and the computational intensive tasks are distributed to a set of contributing services. We have developed a proof-of-concept implementation, demonstrating the main features of the system on real life data sets delivered from a variety of online geodata providers. We have also studied empirically the effects of server-side content caching and on demand generation of 3D objects from 2D geodata.

1 Introduction

This work is a small contribution in the ongoing quest towards the ultimate representation of our known world. Cartography may be defined as the science, and in many respects the craft and art, of depicting our world, or rater more common, pieces of it, on a planar visual medium. Still, the dream of more faithful representations have lived on, and globes are by many considered the most descriptive and telling representations of Mother Earth.

In 1998 US Vice President Al Gore gave a speech titled "The Digital Earth: Understanding our planet in the 21st Century" [12]. Gore envisioned a "multiresolution, three-dimensional representation of the planet, into which we can embed vast quantities of geo-referenced data." Digital Earth (DE) should be a common ground for both consumers and providers of a wide variety of geodata: "The Digital Earth would be composed of both the 'user interface' - a browsable, 3D version of the planet available at various levels of resolution, a rapidly growing universe of networked geospatial information, and the mechanisms for integrating and displaying information from multiple sources." Further, Gore emphasized the non-bureaucratic aspect of the Digital Earth: "Obviously, no one organization in government, industry or academia could undertake such a project. [...] Like the Web, the Digital Earth would organically evolve over time, as technology improves and the information available expands."

The visionary Digital Earth white paper initiated a set of interesting projects, with participants from academia, vendors, and political bodies, see for instance [10] and [8] for overviews of past and present DE activities. However, many projects, in particular those aiming at working implementations, are as we write, abandoned.

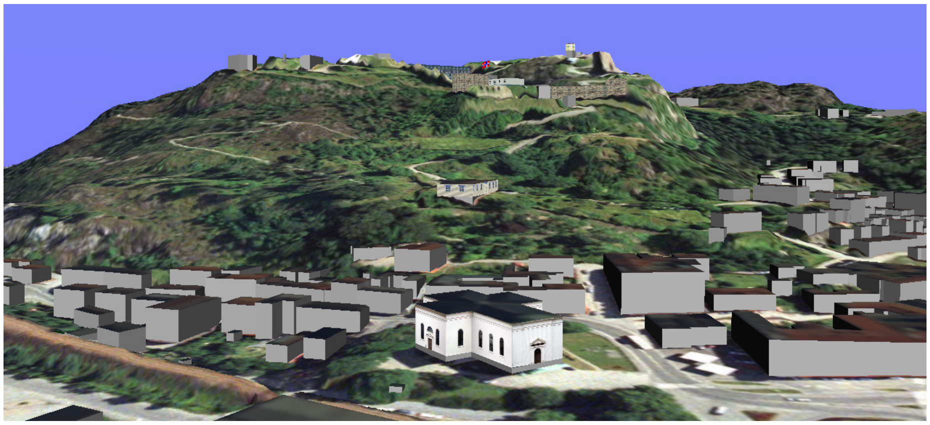

Fig. 1. A digital version of Halden city

In this paper we propose yet another attempt to (partially) realize the Digital Earth, which we will refer to as the OneGlobe framework. A snapshot from the application in shown in Figure 1. In Section 2 we discuss some related approaches, and relevant technologies are identified in Section 3. We then present an outline of our design goals a description of the OneGlobe architecture. A proof-of-concept prototype, along with experiences and empirical results, is presented in Section 5, before we close the paper with some final remarks. The presented research is part of Project OneMap [22][26], a long term effort contributing to the fusion of standard web technologies and geographic content.

2 Related Work

In general, building geobrowsers for accessing massive amounts of geo-referenced and time-stamped data may be viewed as a cross-disciplinary challenge drawing on many fields of research and development, most prominently Geographic Information Science (GIS) and Scientific Visualization (SciVis). From being monolithic and proprietary systems (late 1980's and early 1990's) these two directions have converged and evolved into component based and horizontally integrated frameworks, capable of handling large and distributed data sets [31]. As a backdrop and inspiration for our own work, we present a brief survey of activities, products and projects which explicitly or indirectly are related to geobrowsing and the Digital Earth initiative.

2.1 Open Efforts

According to the open and collaborative nature of Digital Earth, the perhaps most interesting and significant approaches are those that provide tools for free (and preferably, open-source).

TerraVision One of the results from SRI International's DARPA-sponsored Digital Earth project was TerraVision [30]. This ambitious and freely available infrastructure comprised management of massive amounts of distributed and heterogeneous geodata, potentially with global coverage, and a 3D browser for real time navigation. Support was provided for a variety of formats, including the Web Map Service (WMS) (see Section 3). Unfortunately, the project was abandoned in 2002, and the source code is not available. The system was to the authors' knowledge not deployed or tested in real life settings. A noteworthy spin-off from TerraVision was the GeoVRML specification described in Section 3.

Digital Earth Workbench NASA Goddard's Digital Earth Workbench [11] is an open-source interactive system providing retrieval and inspection of huge amounts of distributed Earth related information. The system was primarily developed for internal NASA usage, and is to some extent dependant of expensive special purpose hardware and software. The project is now defunct.

Planet Earth Planet Earth is a combination of a freely accessible web based Earth model [32] and the tools for building and maintaining it [28]. The model is an integration of distributed geodata, in particular terrain models, image textures such as aerial images, and 3D models of buildings and other artifacts.

The project is unique in the sense that it is targeting the collaborative grassroots aspect of DE. Users are encouraged to submit their own data through a browser based application, which then would be seamlessly integrated in the global model.

Planet Earth is modeled with (Geo)VRML (Section 3), and has, as such, quite a few things in common with our OneGlobe framework. The project is in an early state, and provides highly detailed data for Sydney and Perth only. As we write, the project does not appear to be active, and it is a little cumbersome to access their online Earth model.

SINTEF Virtual Globe Virtual Globe is a "client-server application for displaying [...] global scale terrain models" [2]. The models may include 3D VRML representations of for instance buildings. The application is easy to download and run, and retrieves new data from a central server and flushes already preprocessed data, as needed. This prototype offers great performance due to a highly optimized Level-Of-Detail (LOD) triangulated terrain modeling technique [3]. The drawback of this approach is the complexity and computational cost of generating and updating the underlying multi-resolution triangular mesh. Different variants of the Virtual Globe browser are free to download and use, but source code is not available. The server software is not distributed at all.

2.2 Commercial Products

Numerous vendors offer global, 3D terrain visualization applications which in various degree meet the Digital Earth challenge. In the following, we briefly describe some of them.

Keyhole Keyhole [20] was founded in 2001, with the idea of providing a digital 3D model of the entire Earth over the Internet. In October 2004, Google acquired Keyhole to incorporate their technology into the already existing search services. The product is made up of a thick client that downloads image textures from a mapping server. The server provides satellite imagery, ranging from 2-3 months to 2-3 years old. Several of the major US cities are covered with 1-foot to 1-meter precision, but generally the resolution is lower, in particular outside US and in rural areas. There exists little or no public information on design choices or implementation details.

GeoFusion GeoFusion [15] provides software solutions for Digital Earth applications, whereof the most noteworthy is GeoMatrix. GeoMatrix is a software developer toolkit (SDK) for global terrain models, rather than a stand-alone application. With integrated functionality such as level of detail, zooming, and terrain morphing, it allows developers to assemble and expand global models in a high-level manner in a C++ environment. Some of the most popular 3D GIS providers use this SDK in their 3D visualization applications.

ArcGlobe ESRI (Environmental Systems Research Institute) is one of the leading providers of GIS tools and technologies, among them tools for 3D analysis of GIS data. In particular, their ArcGlobe application addresses many of the same areas of use as our OneGlobe project. ArcGlobe is an extension to ESRI's GIS suit, which allows the user to view "out-of-the-box" geospatial data, covering the entire globe, and to add customized information to it. The software supports a range of formats and is highly optimized with heuristics and efficient memory management, which ESRI refers to as pre-loading, indexing, and paging. The application is based on the GeoMatrix SDK from GeoFusion.

2.3 Content Management

The most critical component of a Digital Earth is access to data. Thus, several data infrastructure initiatives, both on national, regional and international levels, have been launched to provide geobrowsers and similar applications with relevant content of sufficient variety and quality. Here we highlight two such international efforts.

Global Spatial Data Infrastructure Association (GSDI) The GSDI Association [14] is "an inclusive organization of organizations, agencies, firms, and individuals from around the world. The purpose of the organization is to promote international cooperation and collaboration in support of local, national and international spatial data infrastructure developments that will allow nations to better address social, economic, and environmental issues of pressing importance."

GSDI frequently arranges international conferences and workshops, and publishes "The SDI Cookbook" [24], from which we quote: "The term Spatial Data Infrastructure (SDI) is often used to denote the relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of and access to spatial data. [...] An SDI must be more than a single data set or database; an SDI hosts geographic data and attributes, sufficient documentation (metadata), a means to discover, visualize, and evaluate the data (catalogues and Web mapping), and some method to provide access to the geographic data. Beyond this are additional services or software to support applications of the data." Clearly, spatial data infrastructures are the very foundation for a Digital Earth.

International Steering Committee for Global Mapping The Global Mapping Project [16] governed by the International Steering Committee for Global Mapping (ISCGM), is an international effort started in 2000 to provide public access to harmonized geodata [17] on a global level. All national mapping agencies are encouraged to contribute data from their respective nations. Four types of raster data are offered (elevation, land cover, land use and vegetation) along with four vector data feature types (drainage systems, transportation, boundaries, populated places). Currently, completed data sets are available from 18 countries, and an additional 130 countries and regions are working on submissions. This corresponds to about 83% of the global land area coverage.

The ISCGM data is not considered to be in the public domain, even though they claim to have a "Worldwide open distribution", but may be used for non-commercial and research purposes. It is possible to browse and view the data with a simple application available from the project homepages. However, further use of the data requires a geographic information system capable of digesting data on the common VPF format.

3 Contributing Technologies

The OneGlobe framework is in essence an assembly of services, rather than a suite of software modules. We rely heavily on open protocols and specifications, and describe the most important ones in this section.

VRML97 Virtual Reality Modeling Language (VRML97)[1] is a file interchange format for visualizations of 3D environments. The visualizations are described in plain text and is, as such, suitable for distribution over the Internet. However, what makes VRML particularly functional, is the fact that it is designed to allow references over the HTTP protocol. This has two important consequences. First of all, it renders the possibility for transparent decentralization of model storage; an imaginary 3D model may retrieve the terrain mesh from the Norwegian Mapping Authority, whereas the textures may be retrieved from NASA. These will be assembled automatically, transparent to the client, who receives the heterogeneous, assembled model as if it were one homogeneous structure. The second consequence is that the references can point to not only files, but to any web service, e.g. a CGI-script or a servlet. As we will demonstrate in Section 5, this makes it possible to integrate content served by various servers that all build different types of 3D data.

Web Services In short, web services enables computers to interact with each other over Internet. The World Wide Web Consortium (W3C) uses the following definition [5]:

> A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Web Services are often used as components in modular network based systems, and are completely independent of operating systems, programming languages or other platform specific features. It is also possible to build service chains, where the result from one service is used to construct a request to another service.

Web Services publish their interfaces describing requests and responses with the Web Services Description Language (WSDL). In order to facilitate discovery and access to Web Services, the Universal Description, Discovery and Integration protocol (UDDI) is used to build registries of services for different purposes in different contexts. UDDI registries are often referred to as ”Yellow Pages” for Web services lookup and discovery.

Open Geospatial Consortium Services Specifications Open Geospatial Consortium (OGC) [25] is ”an international industry consortium of 272 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications. OpenGIS Specifications support interoperable solutions that 'geo-enable' the Web [...] The specifications empower technology developers to make complex spatial information and services.”

Among the OGC specifications, we find a family of Web Services, in particular the Web Map Service (WMS) [7], the Web Feature Service (WFS) [33], the Web Coverage Service (WCS) [9], and a draft specification for the Web 3D Service (W3DS) [29][4]. These services are not ”pure” Web Services, as defined by W3C, but may, however, be encapsulated by a thin translation layer to adhere to the WSDL/UDDI paradigm [21].

The OGC services are specified according to a common template. They all accept a `GetCapabilities` request that returns an XML document describing the provided services in detail. By analyzing the capabilities document, it is possible to construct a valid inquiry, for instance a `GetMap` request returning a digital image representing a map of a given area, in a given resolution and with a given set of map layers. We briefly describe the four services used in the OneGlobe framework.

WMS A Web Map Service is designed to serve maps as digital images. The user typically specify geographic extent, image resolution, a selection of one or more map layers, and optionally, if available, a given style that defines graphical attributes such as color, line thickness and fonts.

WCS A coverage is in this context "digital geospatial information representing space-varying phenomena", a category which in particular includes elevation models. The coverages are served as data with its original semantics (instead of pictures, as in the case of WMS) which can be analyzed, extrapolated and integrated with other data sources.

WFS A geographic feature is an object describing a real world entity, using a set of properties where each property can be thought of as a name, type, value tuple. In most cases, at least one property is described the geometry or location of the entity. A Web Feature Service provides means for querying and retrieving such vector data, for instance coastlines and road systems. The default format of the delivered content is the OGC Geographic Markup Language (GML2) [6], an XML based meta language for describing geographic features. A WFS may also be implemented as a transactional service, enabling users to remotely modify stored content.

W3DS Recently, a service for providing online 3D models have been proposed, as a draft specification. It is full 3D portrayal service, delivering 3D display elements on the VRML97 format. The user specifies an initial view parameters, a bounding box, object layers and rendering options such as light conditions, background image and atmosphere.

4 Architecture and Implementation

We have designed and implemented the OneGlobe infrastructure with two main design goals in mind, ubiquitous access and technological sustainability.

Ubiquitous Access The OneGlobe Digital Earth should be accessible for any interested party, and the content served should be without limitations regarding further use. Hence, we have designed a web service solution providing VRML content. The user only needs a standard web browser and a free of cost plugin, available for all common platforms.

Sustainable Technology The framework should be sustainable, in the sense that it is possible to maintain it over a long span of time independent of shifts in both hardware, software and available content. We have designed a highly modular system, where the components are interacting based on open, well known and widely used standards and specifications. The software involved is relatively simple and easily replaceable. This horizontal integration, as opposed to the case of monolithic systems, allows modules to be interchanged to accommodate shifts in technology and specifications. Such a layered architecture will facilitate interaction with legacy systems, as well as being future-proof. We leverage the family of specifications and standards from the Open Geospatial Consortium (OGC) described in Section 3 as the main foundation of the OneGlobe framework. In addition we will use open-source software where it is possible.

Web servers have a limit on the frequency of hits and the amount of delivered content, before they break down. Hence, a scalable infrastructure has to gracefully handle increasing work load, and also be prepared for rapid growth of content. The proposed framework is scalable, mainly due to the distributed nature, where the point-of-entry service is merely dispatching requests for further treatment on a variety of servers.

Robust systems are fault tolerant in the sense that sub systems may fail without bringing the system down. The OneGlobe infrastructure will provide fall-back mechanisms on many levels, and will gracefully handle broken links in the services chains. It is also robust in the sense that it is easy to replace modules of the system without redesigning the whole system.

OneGlobe may be viewed as a lightweight integration layer, where the responsibility for content management is placed on the core service providers. This makes the system relatively independent of a central authority for development and maintenance.

4.1 Conceptual Model

The proposed OneGlobe infrastructure is in short a dynamically choreographed chain of online geodata services. On the top level, the Digital Earth is deployed as a web service according to the draft OGC Web 3D Service (W3DS) specification (see Section 3). The main idea is to keep the geobrowser as a thin client, and distribute computational intensive tasks to a set of contributing services. In addition, we have strived to make the infrastructure inherently fault tolerant by design.

The conceptual view of the system is a VRML model of the entire Earth provided in a range of levels of detail (LOD), starting with a global view and ending up at street level. In practice, with an approximately doubling of the accuracy for each level, this results in 20 - 25 levels. The levels are organized as a pyramid of tiles, where the root tile is a coarse model covering the entire globe. The initial tile is split into four new tiles, with approximately twice the accuracy of the parent tile, and the new tiles are recursively split until the maximum detail level is reached. The levels of detail, and hence the amount of data in each tile, are adjusted so that any tile may be streamed over a network with modest bandwidth and rendered on a thin client with modest performance.

It is obviously not possible to preprocess and explicitly generate all the VRML files in this hierarchical system[1], so the models have to be generated on demand[2]. Each generated tile will carry four references to the children in the next level, and these references are implemented as requests to a Web 3D Service. A tile is constructed from basically three types of data: 1) Terrain data, 2) Image textures and 3) 3D objects. The needed content will be retrieved using OGC Web Services.

As we see, a request to the OneGlobe service will spawn a chain of cascading geodata services, where each service request may trigger one or more requests to underlying services, until a core service is reached. The core provider will respond with data that will be integrated with content from other services on the way back to the client. To facilitate this rather complex flow of requests and data, we introduce the concept of a Federating Geodata Service.

4.2 Federating Geodata Service

A Federating Geodata Service (FGS) is basically an OGC compliant service, such as the WMS, the WFS and the WCS, with some additional internal features, however hidden from the requesting clients. The main purpose with an FGS is to facilitate service level federation of a set of distributed and autonomous data sources. To enable this, it must implement a Federation Module, which main component is a registry of relevant service providers. In addition, for performance purposes, an FGS has a Content Cache which provides mechanisms for pre-caching, post-caching and flushing of generated content. The general structure of a Federating Geodata Service is illustrated in Figure 2.

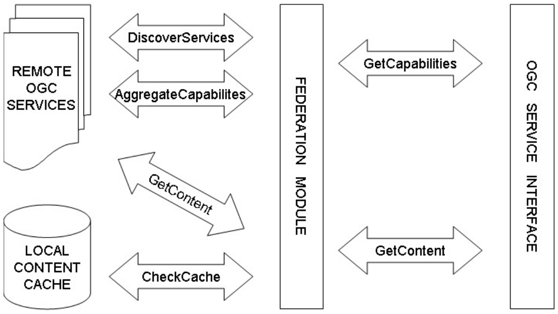

Fig. 2. Federating OGC Service

[1] 25 levels of tiles with an average size of 250 KB would yield approximately 100 Exabyte, or 10^{14} MB.

[2] In the "Future Work" Section of [30] describing the TerraVision system (Section 2), one of three main challenges was generation of data on demand.

Federation Module The Federation Module has two main responsibilities, namely to maintain a registry of OGC service providers and provide integration mechanisms for searching and retrieving heterogeneous content. Dynamic web service applications are depending on ways to discover available services. For this purpose, a range of methods, specifications and tools have been proposed, some of them mentioned in Section 3. In our case, the provider registry could be as simple as a list of OGC services providers, either hard coded, manually updated or dynamically discovered by some web spider software.

By using the well defined OGC services, which again relies on well known and widely adopted formats and specifications, syntactic integration of content from various providers is a relatively trivial task. However, semantic heterogeneity remains a serious challenge. A common way to address this problem is to use ontologies[3], see for instance [18] for details on search and retrieval of geographic information using this approach.

Based on the available ontology, or perhaps a more simple feature catalog, the capabilities documents retrieved from the contributing sources will be analyzed, transformed and aggregated into a single capabilities document, which again the requesting clients will use to formulate their requests. With this approach, the FGS will hide the underlying semantic heterogeneity and enable efficient search and retrieval.

Content Cache Since the underlying service providers may generate their data on demand, efficient caching is crucial for keeping response times at an acceptable level. The cache is a set of tiles that have recently been generated. In response to an incoming retrieval request, the system will first check if the tile is in the cache, otherwise it will forward the inquiry to some of the registered providers. Pre-caching might be implemented by estimating or guessing which tiles the client would request next, the adjacent tiles being good candidates, thus generating and saving tiles for future use. It is also important to provide efficient methods for deleting files that according to some criteria are not likely to be requested in the near future. In case of a total failure of remote data retrieval, the Content Cache also provides a default response based on local and persistent content.

4.3 Service Managers

In the OneGlobe model, we have designed four specialized Federating Geodata Services. The Scene Manager is the top level service entry point, integrating content harvested by the Terrain Manager, Texture Manager and Feature Manager, as illustrated in Figure 3.

Scene Manager (W3DS) The scene manager uses three federating services to retrieve terrain data, image textures and 3D objects like buildings and

[3] An ontology may be defined as an explicit formal specification of a shared conceptualization.

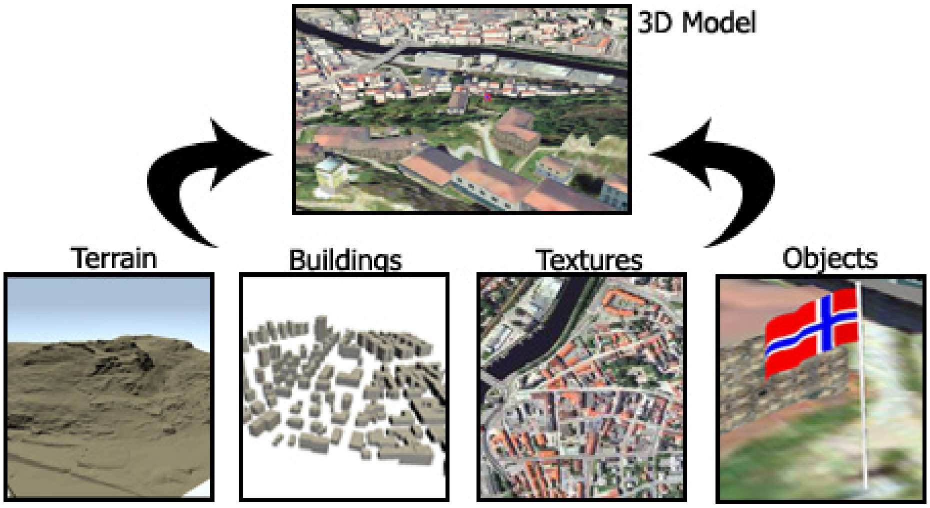

Fig. 3. Scene Manager

other man-made artifacts. In addition it has a Tile Builder module, that converts the elevation model to VRML and drapes the texture over the terrain. The Tile Builder also places the additional VRML objects on their correct locations and incorporate references to the next level of detail formulated as requests to a Scene Manager (itself or another available service).

Terrain Manager (WCS) Elevation data retrieved from different WCS servers may have different grid size and grid location, hence the Terrain Manager has an Interpolation Module which builds a new terrain model from the heterogeneous sources. Special care has to be paid to implement an efficient Content Cache. One approach is to take advantage of the special characteristics of regular grids, for instance by leveraging multi-resolution techniques.

Texture Manager (WMS) Many WMS services are deployed world wide, and they offer a variety of maps, ranging from municipality maps to satellite imagery. The main challenge is to implement a federation module that is able to handle the wide variety of information layers offered by the core services.

Feature Manager (W3DS) The Feature Manager may access two types of OGC services, W3DS and WFS providers. The former delivers 3D VRML objects (or complete scenes), and the latter retrieves 2D map data on the GML vector format. It also contains a 3D Converter module, which transforms 2D geodata into 3D objects, if possible. However, 3D generation from 2D map data is in general not a trivial task.

5 Prototype

Several prototypes have been implemented in an effort to verify the feasibility of the OneGlobe paradigm. First of all, we want to make sure that the principle of cascading geodata services can be applied to real life data on a full range of levels of detail, from global overviews to cityscape scenes. Secondly, we want to evaluate the effect of server side content caching. Lastly, we want to estimate, in an empirically manner, the overhead incurred by on-demand generation of 3D building objects from traditional map data.

In the following sections we describe the proof-of-concept implementations and the associated services and data sets. We also report observations and empirical results.

5.1 Fly-Through

Figure 4 shows seven snapshots from a fly-through within our OneGlobe prototype. The first snapshot shows the initial zoom-out viewpoint, where the entire globe fits the screen. The geometry of the globe consists of a multi-resolution terrain mesh based on the GTOPO30[4] data, projected as a sphere. The increase, or decrease, in level of detail is triggered by the distance between the avatar and the model. The texture, in this case a set of satellite images, is retrieved live through requests to a WMS server from NASA's BlueMarble project. We have included an orbiting model of the ESA Columbus space station, which we are retrieving from the VR department of the Institute for Energy Technology, Halden, Norway (second snapshot). As we zoom in on the model, the relatively coarse globe is seamlessly being replaced by modules with higher resolution - both the geometry and the texture. The third snapshot shows a more detailed version of the globe, along with four billboards representing different types of geo-referenced information, like images, voice recordings and video.

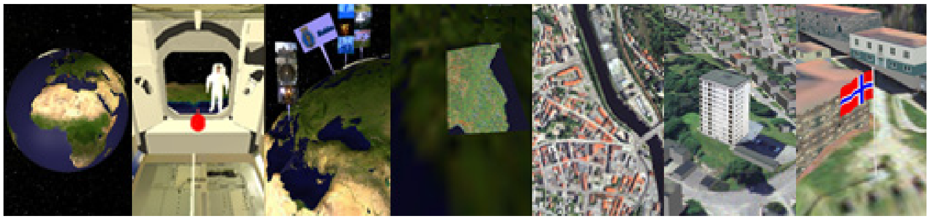

Fig. 4. Prototype snapshots

Snapshot four and five show that it is not only objects that are modular and replaceable, but also the texture. Dependent on which information to portray, we can replace satellite imagery with topographic maps, orthophotos, or any

[4] A global digital elevation model with a grid spacing of approximately 1 kilometer.

other raster image. Finally, the last two snapshots demonstrate the presentation of urban objects, such as buildings, brick walls and even flag poles. All these objects are retrieved from servers offering 3D content as services, most of the objects being generated on-the-fly. The models are available online [27].

5.2 Introducing Heuristics to Improve Performance

Regardless of whether the 3D content provided by a server is converted on-the-fly from an arbitrary data set, retrieved from another server, or fetched as pre-generated data directly from the hard drive, the process of retrieving the data introduces an extra delay to the server-client communication. Therefore, we need an optimization mechanism to compensate, at least partially, for this delay. In this paper, this mechanism is, as a whole, referred to as heuristics[5]. As we have already described in section 2, many of the related projects have incorporated similar optimizations, though often called something else.

The core of the heuristics is a content cache, which functions as an intermediate storage for quick access of 3D content that has high Likelihood of Utilization, or LoU for short (content that is likely to be requested in relatively near future). The heuristics also need a mechanism to estimate the LoU of 3D content, which may depend on several factors such as geospatial distance between the client and the 3D content, the number of times the 3D content has been requested, the time in which the tile was last requested, and the time since the content was first introduced in the content cache. We refer to these factors as measures in this paper. In the test cases described in more detail in section 5.3, the heuristics use a combination of all the four measures mentioned above. This allows for tweaking and tuning regarding how much each measure should influence the resulting value, or LoU.

An important role of the heuristics is to make sure that the cache content has a highest possible LoU at all times. Since the cache has a limited capacity, we need an effective data structure in order to decide which tiles to be inserted in the cache, and what content to be purged. Our prototype realizes this by storing all cache content in a heap-structured priority queue. Further, since the LoU parameters such as the distance between 3D content and the client's current viewpoint, or the time since the content was last requested, are subject to continuous alterations, we need to revalidate the priority queue with given intervals.

The last heuristics element implemented in our prototype, is a background process (daemon) that analyzes all available information from the client and tries to predict what will be requested in the near future. Client data available to the server, also referred to as session data, can be anything from previously requested content, to avatar positions, orientation, or speed that is reported from the client

[5] Using heuristics involves performing qualified guesswork and can be contrasted with using algorithms (which involves using a predefined set of rules or a formula). Another definition describes heuristics as gaining knowledge or some desired result by following a “rule-of-thumb”.

with given intervals. Dependent on how long in advance you predict content, it is increasingly important to give the suggested content special treatment, e.g., a flag, to prevent content that has been created because of its potential future value to be purged based on present relevance.

5.3 The Effect of Applying Server-Side Heuristics

A set of test cases have been deployed in order to get some empiricism on the effect of applying heuristics to a server offering 3D web services. The first case uses a VRML [1] model of Halden [19] which has been converted to a servlet-based service. The thin client sends standard HTTP requests to a servlet. An additional delay can be added to the server-side for each service request, to simulate the alteration in performance as the build delay, i.e., the time the server needs to process and deliver a requested service, increases. The server is equipped with a set of heuristics which try to optimize the server (reduce the build delay) by means described in section 5.2. The test cases show, in general, that the introduction of heuristics may improve the server performance with a factor of three.

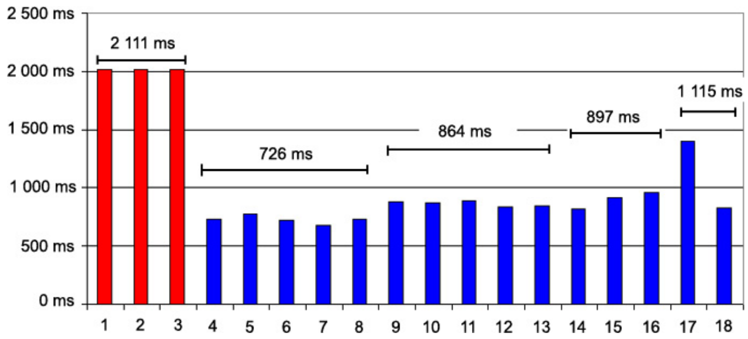

Fig. 5. Response times using various caching heuristics

Figure 5 shows the resulting build delays for 18 different tests, using different heuristics setups and tweaked differently. The three first bars in the figure represent tests without proper heuristics, and therefore, little tweaking could be performed. The average response time was 2,111 seconds. Setup 4 to 8 uses a different heuristics setup, all tweaked differently. The result is an average response time of 726 ms, which is only 34% of the average for the three first setups. The rest of the bars show the resulting build delays for heuristics setups that, apparently, yields results not as good as the one in test four to eight. Still, we see a significantly improvement compared to the three first tests, run without any heuristics. In practice, these results mean that even though the server might need 2 seconds to build a requested tile, the client will effectively experience only 0.67 second lag. Another test case, this time with global dimensions, yields the same results. We have seen that by utilizing a set of server-side heuristics, we have effectively reduced the response time by a factor of three. This means that heuristics can pave the way for, and make possible, feasible implementations of on-the-fly generation of 3D content in VR applications. A more detailed discussion of caching strategies and experiments is found in [13].

5.4 Benchmarking the On-the-Fly Generation Paradigm

Tests also show that on-the-fly generation of buildings, or other man-made structures, is feasible. In the test case described below, covering the center of Halden city in Norway, all buildings are generated on-the-fly from SOSI[6] files. In short, the 3D objects are created by extrusion based on footprint and height extracted from the original files (see [19] for further details).

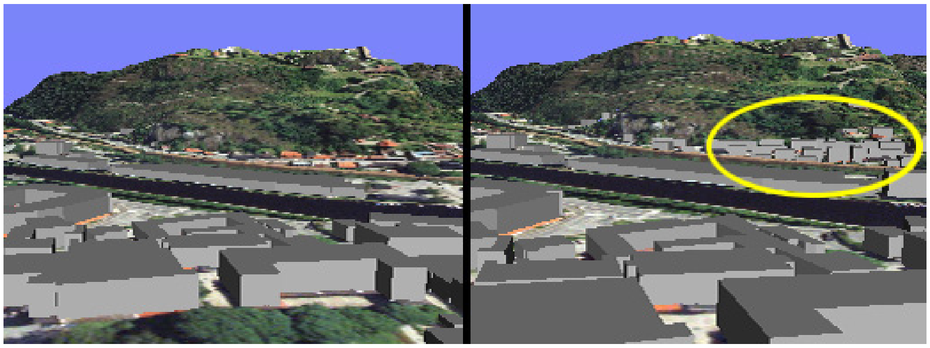

Fig. 6. Screenshots from the prototype, showing houses built on-the-fly

A server accepts HTTP requests with a specified bounding box. The servlet will then build and return the buildings enclosed by this bounding box. These buildings can then be added as an extra layer in any 3D model (in much the same way as with 2D Geographic Information Systems). Figure 6 shows two screenshots from a 3D model of Halden where the buildings are retrieved from this on-the-fly server. Notice the buildings that appear in the far back when zooming in.

Figure 7 shows the delay difference between fetching already generated 3D content from the server (direct file system access) and using the more dynamic on-the-fly generation approach. In this test case, 1000 requests (from a random part of the model) have been sent to a server that offers 3D buildings on the VRML format, whereas half of them is generated on-the-fly and the other half is retrieved as pre-generated files from the server. Further, the requests have been divided into five different coverage sizes, level 0 (covering 1,920,000 square meters) to level 4 (covering 7,500 square meters). The results show the average

[6] SOSI is Norwegian geodata standard.

delay from the client sends the request until the client has successfully received the 3D content. In this test case, introducing on-the-fly generation caused, on average, 17%, 30%, 44%, 65%, and 55% increase in the build delay respectively (whereof the first is for level 0 and the last for level 4). This is without utilizing any form of heuristics (as described in section 5.2). Furthermore, it should be pointed out that the increases presented here are somewhat high, since the client and the server is on the same LAN. When running the same test farther away from the server (increasing the distance[7] between client and server.) the delay caused by data transfer will constitute a greater percentage of the total delay.

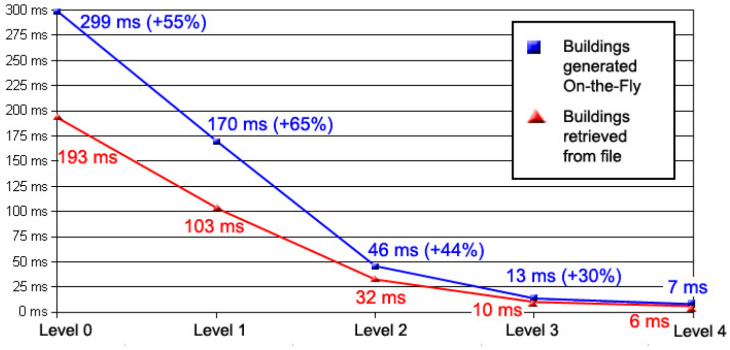

Fig. 7. Overhead incurred by on demand construction of 3D objects

6 Final Remarks

We have proposed a novel approach in the pursuit of a robust and accessible Digital Earth solution, the OneGlobe framework, and developed a proof-of-concept implementation. The test results indicate that using cascading geodata services to collect and integrate data from distributed, heterogeneous sources may work very well in real life applications. Content caching has shown, as expected, to improve server response times, but much research remains in order to fully release the potential of various caching strategies. We have also shown that on-the-fly generation of 3D objects incur a surprisingly small overhead, making this a highly desirable approach.

Regarding the Federating Geodata Services, two aspects are not covered so far in our research:

- We have hard coded the list of services in the Provider Registries, and not implemented dynamic discovery mechanisms. This is by far a trivial task.

[7] Distance as in the sum of physical distance and bandwidth.

- We have not put much efforts in implementing methods for semantic integration. This is also a serious challenge, and we have only scratched the surface by investigating ”lazy integration” of heterogeneous GML data sources [23].

In addition, our method for generation of 3D objects from 2D geodata is quite naïve, and there is clearly need for further research in this area. Regarding the 3D modeling language, there are two obvious candidates for further development:

- The prototype implementation is based on VRML, because X3D viewers are scarce and in beta status. However, our approach would benefit significantly by leveraging the X3D Geospatial profile, and we intend to do so when appropriate tools are available.
- A traditional VRML level of detail hierarchy may only be traversed top-down, facilitating drilling down into more detailed layers. However, zooming out from an entry point on a detailed level is not possible. We are currently working to solve this problem.

There are two aspects of an information infrastructure; tools and content, the one depending heavily on the other. In order to make an infrastructure sustainable, it must be possible to repair, improve and replace the tools as technology advances. This motivates our strong focus on standards and specifications, which in our opinion are more long-lasting than technology. In addition, the distributed nature of the system brakes down the framework in manageable modules, most of them maintained locally by the core content providers. This is also the case with the content itself, which is provided by the parties most interested in generating and updating the information. In our opinion, OneGlobe may come close to Al Gore’s vision: ”Like the Web, the Digital Earth would organically evolve over time, as technology improves and the information available expands.”

References

1. ISO/IEC JTC 1. Vrml 97 - the virtual reality modeling language. International Standard ISO/IEC 14772-1:1997, 1997.
2. Rune Aasgaard. SINTEF Virtual Globe. globe.sintef.no.
3. Rune Aasgaard and Thomas Sevaldrud. Distributed handling of level of detail surfaces with binary triangle trees. In Jan Terje Bjørke and Håvard Tveite, editors, *ScanGIS’2001, Proceedings of the 8th Scandinavian Research Conference on Geographical Information Science*, pages 45–58, Ås, Norway, 2001.
4. Angela Altmaier and Thomas H. Kolbe. Applications and solutions for interoperable 3d geo-visualization. In D. Fritsch, editor, *Proceedings of the Photogrammetric Week 2003 in Stuttgart*. Wichmann Verlag, 2003.
5. World Wide Web Consortium. Web services activity. www.w3.org/2002/ws.
6. S. Cox, A. Cuthbert, R. Lake, and R. Martell. *Geography Markup Language (GML) Implementation Specification*. Open Geospatial Consortium, Inc., 2.1.1 edition, April 2002.
7. Jeff de La Beaujardiere. *Web Map Service (WMS) Implementation Specification*. Open Geospatial Consortium, Inc., 1.3 edition, August 2004.

8. Bonnie DeVarco. Earth as a lens: Global collaboration, geocommunication, and the birth of ecosentience. *PlaNetwork*, jul 2004.
9. John D. Evans. *Web Coverage Service (WCS) Implementation Specification*. Open Geospatial Consortium, Inc., 1.0.0 edition, August 2003.
10. Timothy Foresman. Digital earth: the status and the challenge. In *Proceedings of the Global Mapping Forum*, 2003.
11. NASA Goddard. Digital earth workbench. webserv.gsfc.nasa.gov/DE-Workbench.
12. Al Gore. The Digital Earth: Understanding our planet in the 21st century, 1998.
13. Morten Granlund. Perspective based level of detail management of topographical data. Masters thesis in computer science, Østfold University College, Halden, Norway, 2004.
14. GSDI. Global spatial data infrastructure association. www.gsdi.org.
15. GeoFusion Inc. Geofusion. www.geofusion.com.
16. ISCGM. International steering committee for global mapping. www.iscgm.org/html4.
17. ISCGM. Global map version 1.1 specifications. Technical report, International Steering Committee for Global Mapping (ISCGM), 2000.
18. E. Klien, U. Einspanier, M. Lutz, and S. Hübner. An architecture for ontology-based discovery and retrieval of geographic information. In F. Toppen and P. Prastacos, editors, *7th Conference on Geographic Information Science (AGILE 2004)*, pages 179–188, 2004.
19. Herman Kolås. VRML model of Fredriksten. WWW, oct 2004. Servlet based 3D model.
20. Steven Levy. Making the ultimate map. *Newsweek International*, June 2004.
21. Josh Lieberman, Lou Reich, and Peter Vretanos, editors. *OWS1.2 UDDI Experiment*. Open GIS Consortium, Inc., 2003.
22. Gunnar Misund. One Map. *HØit*, (1), 2002. ISSN 0805-6692 / 0805-7486.
23. Gunnar Misund and Harald Vålerhaugen. Integration of heterogeneous GML sources. In *Online Proceedings of GML Days 2004*, Vancouver, Canada, 2004.
24. Douglas D. Nebert. *Developing Spatial Data Infrastructures: The SDI Cookbook*. Global Spatial Data Infrastructure Association, 2.0 edition, 2004.
25. OGC. Open geospatial consortium. www.opengeospatial.org.
26. Project OneMap. One World - One Map. www.onemap.org. Project website.
27. Project OneMap. Online OneGlobe models. www.ia-stud.hiof.no/ hermanko/onemap/onlineGlobalModels.html, 2005.
28. planet-earth.org. Planet Earth. www.planet-earth.org.
29. Udo Quadt and Thomas H. Kolbe. *Web 3D Service (W3DS)*. Open Geospatial Consortium, Inc., discussion paper edition, February 2005.
30. Martin Reddy, Yvan Leclerc, Lee Iverson, and Nat Bletter. Terravision II: Visualizing massive terrain databases in vrml. *IEEE Computer Graphics and Applications*, 19(2):30–38, 1999.
31. Theresa-Marie Rhyne. Visualizing geospatial data. Topic #1: Overview of integrating geospatial data with visualization methods. SIGGRAPH 2004 Course #30 Notes, 2004.
32. Chris Thorne and Viveka Weiley. Earth's avatar: the web augmented virtual earth. In *SIGGRAPH '03: Proceedings of the SIGGRAPH 2003 conference on Web graphics*, pages 1–1. ACM Press, 2003.
33. Panagiotis A. Vretanos. *Web Feature Service (WFS) Implementation Specification*. Open Geospatial Consortium, Inc., 1.0.0 edition, May 2002.