Introducing 3D GIS for the Mobile Community
Technical Aspects in the Case of TellMaris

Arne Schilling
Fraunhofer Institute for Computer Graphics, Germany
arne.schilling@igd.fhg.de

Volker Coors
Hft University of Applied Sciences, Germany
coors.fbv@fht-stuttgart.de

Martin Giersich
Fraunhofer Institute for Computer Graphics, Germany
martin.giersich@rostock.igd.fhg.de

Rune Aasgaard
SINTEF Applied Mathematics, Norway
rune.aasgaard@math.sintef.no

Abstract

In this paper we discuss some technical aspects of a EC funded project called TellMaris. The task of this project can be condensed to "3D maps on mobile devices". Moreover, these have to be integrated in a distributed system for Location Based Services. In order to achieve this, methods for the storage, transmission, display of 3D geodata and also for the integration of terrain and man-made structures and other objects have been developed. The actual contents are stored and managed in specialized databases that can be accessed over internet protocols. On the client side several applications are responsible for the display and the user interaction. We reveal some details of these components and how they are integrated in TellMaris.

1. Introduction

The objective of TellMaris (IST 2000-28249, www.tellmaris.com) is the development of a tourist information system for mobile clients which is able to display 3D maps. Surveys have shown that 3D maps in some cases can offer a better guidance in unfamiliar environments than conventional 2D maps. In order to achieve this, state-of-the-art technologies in the fields of telecommunication, geometric modeling, 3D databases, and computer graphics are included. The system will be targeted on boat tourists in the Baltic Sea. For this group, relevant information as well as Location Based Services (LBS) will be integrated. According to [10] a relative large number of boat tourists has already a Laptop computer or PC on board. Together with GPS and network access the conditions are given to provide them with tailor made services. From the user's point of view, two applications can be distinguished.

On board, the TellMarisOnBoard application running on a laptop offers services like weather forecasts, hotel reservations or navigation guidance in ports. It can be used to retrieve harbor information, coastline visualizations, etc. It is based on Java3D and combines a high resolution textured terrain model with 3D GIS data.

The TellMarisGuide is developed for PDAs or cell phones with graphics display and assists the users when they disembark and explore the city. It can be used to request information about sights or restaurants, find the closest facilities, or find out where to get specific boating equipment.

2. Motivation

Numerous providers are presently developing so-called Location Based Services offering a great number of basic services for mobile applications. These solutions often base upon existing GIS tools and complement missing functionality. As middleware XML based webservices, as described in the W3C's WSDL specification ([16]) become more and more important for the communication between these services. In order to grant the interoperability of the different solutions the

Open GIS Consortium started the OpenLS^{TM} initiative. The main advantage of such a distributed system is, that its components can be developed separately and new services from other providers can be integrated very easily.

Three main areas can be identified:

- Mobile client services as an interface to the functionality of the location services, for the map design, and as localization service
- Location services representing the proper basic functionality of the LBS applications, typically routing and geo-coding
- Warehouse services as database of the location services

Normally, two-dimensional maps are rendered dynamically and presented to the user. These are, however, not optimal for large scale views or route descriptions in a mobile application. Although the display of maps for LBS has not yet come beyond prototypical solutions, we are already in the development of 3D maps. In general, maps on mobile devices are seen as a great potential for the mobile market. The step into the third dimension is of additional value, as it enables realistic representations for the near surroundings. It can also be of help when navigating in an unknown city, as it supports much better visual dominant landmarks like towers or churches. For these reasons the TellMaris approach is to combine a detailed 3D map with 2D overview to provide the user with the advantages of both paradigms.

3. 3D Maps for Location Based Services

In recent years, the rapid development of 3D graphics hardware has encouraged the GIS community to overcome the conceptional limitations of traditional 2D maps and immerse into 3D space. New technologies like automated laser scanning ([4]) are now at hand that will produce high quality city models. Even the large pool of available 2D geodata can be exploited using techniques that derive the third dimension with the help of other sources ([15]).

Compared to Desktop Computers the Capabilities of mobile devices like PDAs or cell phones with color display are very limited and the development of 3D graphics on such devices is just at the beginning. Especially geodatasets require lots of resources. For 3D maps for LBS several limitations have to taken into account.

1. The rendering capabilities depend on the amount of memory and the chipset. Currently, small 3D graphic engines available for PDAs can render small city models with approximately 5 fps. But it is expected that the rendering capability will grow very fast in the future as it did on PCs and Laptops in the past ([8]).
2. The currently available mobile network standard GPRS allows a transfer rate of 57,4 kBit/s. With UMTS the bandwidth will increase significantly and transfer rates up to 384 kBit/s will be possible. However, the high data volume of 3D maps will be critical even in high bandwidth wireless networks.
3. The small screen size and the restricted interaction possibilities require new metaphors for exploring and navigating in 3D maps. The Graphical User Interface has to concentrate on the most relevant information allowing for these technical restrictions.

4. Transmission Optimization for 3D content

As the limited bandwidth of current mobile networks is one of the main bottlenecks in the visualization pipeline, intelligent compression methods have to be introduced that reduce the amount of the transmitted data to a minimum. We are using a compression technique called Delphi Compression that is suitable for triangular meshes.

It has been developed by Coors and Rossignac ([6]) and is based on the Edgebreaker algorithm ([14], [13]). Edgebreaker is a state machine to traverse any triangle mesh and compress its connectivity. The succession of case types produced by this traversal are encoded as symbols from the set {C,L,E,R,S}, called the “clers string”.

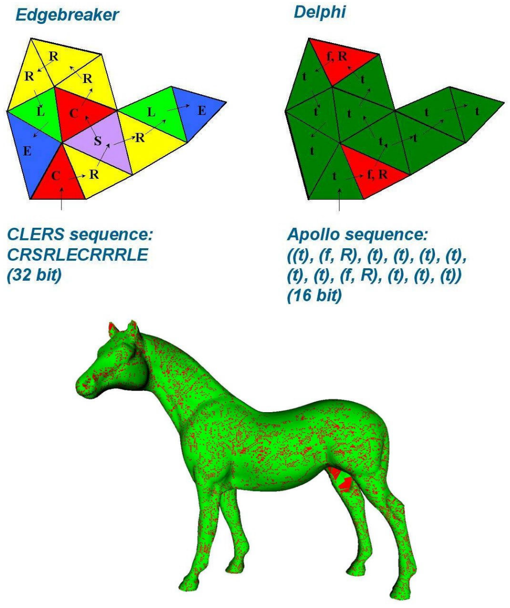


Figure 1. Delphi compression encodes the connectivity of any triangle mesh to a sequence of CLERS symbols, drawn here in different colors. Experimental results have proved a compression rate up to 0.2 bit per triangle for regular meshes.

This clers string is sufficient to describe the complete connectivity. These situations and the associated clers symbols are shown in figure 1. The arrow indicates the direction to the next triangle. Previously visited triangles are not shown. At each state the state machine moves from a triangle Y to an adjacent triangle X. It marks all visited triangles and their bounding vertices.

Left and Right denote the other two triangles that are incident upon X. Let v be the vertex common to X, Left, and Right. If v has not yet been visited, then neither have Left and Right. This is case C. If v has been visited, we distinguish four other cases, which correspond to four situations where one, both, or neither of the Left and Right triangles have been visited. Note that in the S case, Edgebreaker moves to the right, using a recursive call, and then to the left. The clers string is encoded bit wise, i.e. the symbols from the set {C,L,E,R,S} are translated to a bit code (C=0, L=110, E=111, R=101, S=100). As most of the descriptors are Cs, an average of 2 bits per triangle can be achieved.

In Delphi, the tip-vertex of the next triangle in the mesh traversal is predicted. Delphi estimates the triangle connectivity by snapping the tip-vertex to the nearest boundary vertex, if one lies sufficiently close. Based on this estimation, one of the five CLERS symbols is predicted for that triangle. If the guess is correct, only a confirmation bit needs to be transmitted. Because up to 97% of Dephi's guesses are correct, connectivity information is often compressed to a fraction of a bit per triangle. Experimental results lead to a compression rate up to 0.5 Bit per triangle for urban models.

5. Storage and Provision of 3D Geodata

Unlike other approaches we consider 3D maps not only as terrain models with additional artificial textures taken from paper maps. It contains a digital elevation model which can be enriched with satellite images or aerial photos as well as identifiable objects from buildings and other man-made structures. As the requirements and the query model of both elements are different, they are handled in separate components. In TellMaris we have implemented a terrain server and a feature server. Both can be accessed independently, but offer a common interface. When a map is built up, spatial queries are sent to the terrain and the feature server. The results are combined on the client side. Figure 2 shows a combined model from our test area Toensberg.

Figure 2. 3D Model from Toensberg with textured terrain and additional building objects of the city. In the background lower resolution parts are visible (view from the TellMarisOnBoard application).

5.1 Terrain Handling

To get reasonable performance from small mobile clients with narrow communication channels we must use adapted datasets to keep the volume of transferred and rendered data down.

On the client side the system represents the terrain with a Continuous Level of Detail surface model. The surface has quality metrics that quantifies the accuracy of the current surface expansion as an elevation approximation error. To get an adequate image the visible parts of the terrain surface is expanded to a level of detail where the approximation error estimates projected into the screen space is below a certain threshold. This results in a high data density near the viewpoint, decreasing with increasing distance. For areas outside of the view the terrain surface is retained at a very coarse base resolution, where it is practically possible to have overview information for the complete mapped area (possibly the whole earth).

When the user moves the viewpoint interactively around in the model the surface model is continuously evaluated and refined/coarsened accordingly to maintain a surface model using the minimum amount of data for fulfilling the screen space approximation error requirement. Data are requested from the server on an "as-needed" basis, and data that has gone out of scope is retained for a while if the user should turn back, but is eventually deleted from the client using a "least-recently-used" algorithm.

To keep the data structures simple but flexible the system uses a hierarchical triangulation over a virtual regular grid structure. The triangulation uses a nested set of right-isosceles triangles, where a triangle is refined by splitting in two at the midpoint of the hypotenuse (figure 3). To maintain a continuous triangulation it is required that if the triangle has a neighbor along the hypotenuse it should also be split in the same way. This structure is similar to the ROAM algorithm [9] or Lindstrom triangulations [11]. An earlier version of this system was also presented in [2].

As the triangulation is built over a regular grid the positions of the vertices are easily computed from the index positions. In a small-scale case the transform is trivial. To extend this scheme to a world covering system a finite near-conformal cylindrical projection has been developed [1].

The vertex indexes are used when querying the server. For each vertex index the server returns an elevation / elevation approximation error pair. Where the terrain is sufficiently flat no elevation value is stored, and the server returns an approximation error of zero and an elevation of "magic number *interpolate*" to indicate that this elevation should be interpolated. On the server this structure is implemented as a large sparce array, taking advantage of the fact that large areas of the world are fairly smooth.

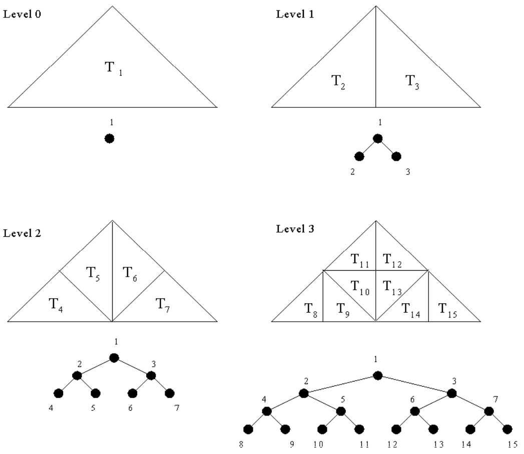


Figure 3. Hierarchical triangulation and successive terrain refinement

The terrain model is covered by texture images. The textures are organized in a quad tree structure so that the appropriate resolution can be selected depending on the distance to the viewpoint. The image pyramid uses the same coordinate system and split pattern as the terrain model which assures that each texture tile is completely and uniquely covered by a set of triangles. This also simplifies the computation of texture coordinates.

The images are stored and compressed individually. Several newer image formats like MrSID, JPEG2000 and ECW also support the storage of several layers in an integrated image pyramid. Initial experiments using JPEG2000 are very promising, since high compression rates with few visual artifacts can be achieved, but the performance of the then available example-implementations of encoders/decoders was not up to production quality. The available textures range from very high resolution images from our test area Toensberg in Norway to natural satellite images covering the whole world.

5.2 Feature Server

In TellMaris we are using a modified version of the Urban Data Model (UDM) format introduced by [7] in order to store geo-objects in a relational database system like Oracle 9i. The UDM model has been extended in order to store more realistic models such as textured buildings and complex hierarchical structures that occur when these models are created with modeling tools such as StudioMax. It is intended to be used as an internal object-oriented data format within the feature server and supports the most important capabilities of the involved input and output formats. It is able to handle 2D geodata (Geometry_2d) as well as textured 3D models (Geometry_3d) that can be loaded e.g. from VRML models. In general, a Feature consists of a geometrical and visual description of this object and a list of properties which can store additional attributes like address, name, age, etc. as name-value pairs. The texture mapping is done similarly as in VRML or Java3D. Figure 4 shows a UML class diagram with the relationships between the elements.

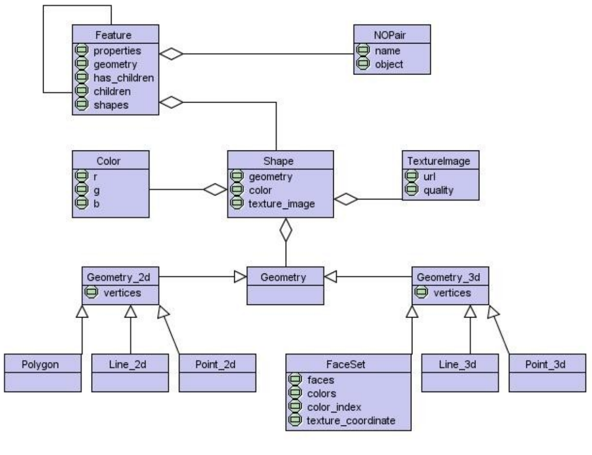


Figure 4. Internal feature format for storing 3D objects used in the Feature Server. UML class diagram.

In order to store geodata into a database, structures equivalent to the above-presented model have been established. Otherwise all necessary geodata hava to be stored in the memory on the server, which can be problematic when the amount of geodata increases and the memory is limited. For large datasets a database should be preferred that is accessed dynamically from a specialized client, respectively server. These classes are mapped to a relational table schema in the database, which can't be described here in detail. Most tables are very similar to the corresponding classes.

The feature server handles incoming requests and translates them to SQL queries according to the table definitions. For the transmission over the internet the object oriented model is converted to XML containing equivalent tags (marshalling) that can be embedded in HTTP messages. For the conversion to the original objects an Unmarshaller is provided to the client. In contrast to the transmission of VRML or X3D code, this is a more flexible mechanism for the dynamic retrieval and display of geodata.

6. Client Integration in a Distributed System

The previously presented components are developed independently and deployed on spatially disjoined servers. To embed these in the TellMaris system they offer XML/SOAP based webservice interfaces. Sharing this standardized protocol, we are able to develop services and clients platform independently and also to access the pool of already published webservices listed in UDDI registries. Figure 5 depicts the overall architecture.

On the client side we have developed two applications for laptop PCs and PDAs respectively mobile phones. Both have network access and thus also to the feature-, terrain- and tourist-database. The transmitted data is adjusted according to the hardware performance and network bandwidth.

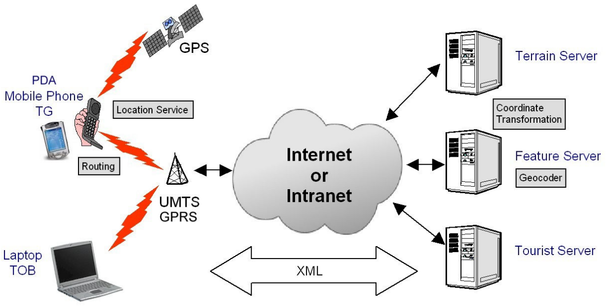


Figure 5. The TellMaris system consists of several services and components on the client and server side.

6.1 TellMarisOnBoard Application

The TellMarisOnBoard (TOB) application supports the boat tourist directly during his trip on the sea. The system runs on a laptop and provides 3D services for navigation and current harbor information. Basic services such as weather forecasts are also regarded as important for boat tourists. The interactive 3D visualization helps to explore the areas where the boat tourist is planning to go and also supports the orientation when he is entering a harbor. The virtual viewpoint can either be moved to the current boat position presenting views from the boat perspective, or it can be moved freely through the scene.

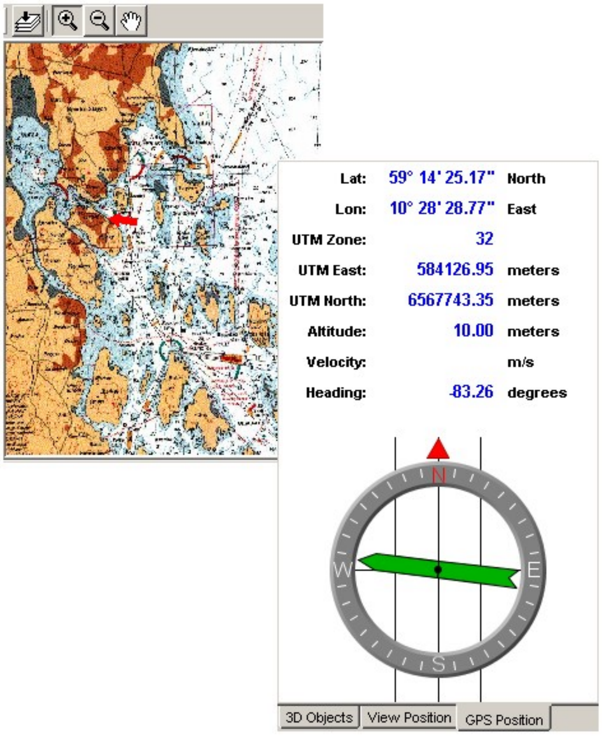

Lat: 59° 14' 25.17" North
Lon: 10° 28' 28.77" East
UTM Zone: 32
UTM East: 584126.95 meters
UTM North: 6567743.35 meters
Altitude: 10.00 meters
Velocity: m/s
Heading: -83.26 degrees

Figure 6. Navigational support within the TOB. The current position is shown on sea charts (arrow) providing the skipper with relevant information. On the right side parameters from GPS receiver and compass are displayed.

The Graphical User Interface (GUI) presents also sea charts providing skippers with important information like water depths and shipping routes (figure 6). As the TOB is used on board, technical devices such as compass or GPS receiver, which are mostly already present, can be connected to complete the system. The 3D view has been realized using Java3D. The content servers are accessed separately and the integration is done within the TOB.

The experiences gained with the laptop solution have been very useful for the development of the

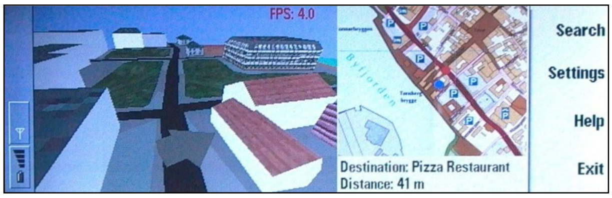
TellMarisGuide on mobile computers, which we describe in the next chapter.

6.2 TellMarisGuide Application

The TellMarisGuide (TG), developed for PDAs and smart phones, assists the boat tourists when they disembark and explore the larger scale harbor surroundings. It can be used to receive tourist information for instance about sights or hotels, it finds the closest restaurant or other facilities of interest. From user-selected positions or, if connected to a GPS receiver, from the current position the the tourists can be guided dynamically to targets of interest. According to the TellMaris approach the TG provides 3D and 2D visualization supporting the skippers when they stroll around in the city or simply routes them to the points of interest.

The 3D graphics of the iPAQ PDA solution is based on the Pocket Cortona Viewer from Parallelgraphics, tailored for mobile computers. It can be used as ActiveX controls and thus integrated in the client software. These ActiveX controls are embedded as API via JNI. So we are able to update the 3D content dynamically according to the current user position and the available data on the server.

The 2D view is an instance of the Scalable Vector Graphics (SVG) Java-API for mobile devices developed by Fraunhofer IGD-R ([3]). The TG for now runs with a fixed data set but on short track there will be also a connector to the specified TellMaris web services interface.



Destination: Pizza Restaurant
Distance: 41 m

Figure 7. Screenshot from the Nokia Communicator display showing the TellMarisGuide, which assists tourists when they walk around in the city.

The solution for mobile phones runs on the Symbian Operating System Nokia 9210 and can thus be used on devices that are already very widespread. For the Nokia Communicator Series a special 3D API called NokiaGL has been developed. Figure 7 shows the User Interface containing a 2D map and the 3D scene. The high resolution color display makes a very detailed and vivid representation possible. In order to find a selected destination, an arrow can be shown that points to the direction where to go.

The user has the choice between 3 different view modes. He can either immerse into the scene and walk around with a pedestrian perspective, he can look from above, which is similar to the 2D map, or he can choose a viewpoint in-between looking slightly downwards, as showed in figure 7. The last alternative seems to be the most useful, as it gives an overview of the near surroundings, which the pedestrian can normally not see from his perspective.

7. Outlook

In order to evaluate the usability of the first prototype, a pilot study of the TG has been carried out last summer in Toensberg, Norway. Although the feedback was in general positive, interesting input concerning cognitive aspects could be received ([8]). Concerning the orientation is was apparently difficult to switch between the 3D and the 2D map, when they were displayed at the same time.

Further developments will improve the user interaction and the system performance. Although the destination direction can be presented, routing is not yet supported so that the user has to find out himself which paths he has to follow. Database systems for 3D GIS are still in the development, but the achieved results in this project have shown the feasibility to integrate these into interactive systems. The 3D database is currently being extended to store arbitrary complex 3D models and also to include temporal structures.

We are now preparing the second evaluation event this summer in Turku, Finnland, where we will present the complete TellMaris system.

8. References

[1] Aasgaard, R, “Projecting a Regular Grid onto a Sphere or Ellipsoid”, *Int. Symposion on Spatial Data Handling*, Ottawa 2002.

[2] Aasgaard, R. and T. Sevaldrud, “Distributed Handling of Level of Detail Surfaces with Binary Triangle Trees”, *ScanGIS*, 2001.

[3] Bieber, G. and V. Leck, “Scalable Vector Graphics for SaiMotion”, *Computer Graphik topics*, 03/2003.

[4] Brenner, C. and N. Haala, „*Automated Reconstruction of 3D City Models*”, Abdelguerfi, M. (ed), 3D Synthetic Environment Reconstruction, Kluwer Academic Publishers, 2001, pp. 75-101.

[5] Brenner, C., N. Haala and D. Fritsch, „Towards fully automated 3D city model generation”, *Automatic*

Extraction of Man-Made Objects from Aerial and Space Images III, 2001.

[6] Coors, V. and J. Rossignac, “Guess Connectivity: Delphi Encoding in Edgebreaker”, Technical Report, Georgia Institute of Technology, Atlanta, USA, 2002

[7] Coors, V., “Feature-preserving Simplification in Web-based 3D-GIS”, *First International Symposium on Smart Graphics*, New York, USA, 2001

[8] Coors, V., O. Gjesdal, J.R. Sulebak, and K. Lakso, “3D Maps for Boat Tourists”, *ENTER International Conference on Information Technology and Travel & Tourism*, Helsinki, Finnland, 29.-31.01.2003.

[9] Duchaineau, M., Wolinsky, M., Sigeti, D.E., Miller, M.C., Aldrich, C. and Mineev-Weinstein, M.B.: “ROAMing Terrain: Real-Time Optimally Adapting Meshes”, *Proc. Visualization '97*, 1997.

[10] Gjesdal, O. “Market research in the boat tourism segment”, *ENTER Conference* 2002.

[11] Lindstrom, P., Koller, D., Hodges, L.F., Ribarsky, W.,Faust, N. and Turner, G.: Real-Time, Continous Level of Detail Rendering of Height Fields, Computer Graphics (Proc. SIGGRAPH '96), 1996

[12] Pilouk, M., “Integrated Modeling for 3D GIS”, PhD thesis, ITC, 1996

[13] Rossignac, J. et al, “3D Compression Made Simple: Edgebreaker on a Corner-Table”, *Shape Modeling International Conference*, Gemoa, Italy, May 2001.

[14] Rossignac, J., “Edgebreaker: Connectivity compression for triangle meshes”, *IEEE Transactions on Visualization and Computer Graphics*, 5(1), Jan-Mar 1999, pp. 47-61.

[15] Schilling, A. and A. Zipf, “Generation of VRML City Models for Focus Based Tour Animations. Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources”, *Web3D Conference*, Saint Malo, France, 9-12.03.2003.

[16] W3C, “Web Services Description Language (WSDL) 1.1”, W3C Note 15 March 2001. http://www.w3.org/TR/wsdl