Prototype Implementations for Multi-typed 3D Urban Features Visualization on Mobile 3D and Web 3D Environments

Kiwon LEE, KOREA

Key words: GML, Mobile 3D, OpenGL|ES, Web 3D, X3D

SUMMARY

This study aims at urban 3D features modeling/rendering system architecture design and its prototype implementations on 3D mobile and web 3D application. Mobile 3D is a cutting-edge technology. However, platforms and devices for mobile 3D GIS technology manipulating geographic data and information provide some limitations to design and implement. Despite, emerging standard mobile 3D graphic rendering stimulates mobile 3D GIS and 3D cartographic services on hand-held devices. Core functions of mobile 3D GIS by feature-based approach was designed as some modules: data model and design of spatial features, editing and manipulating of 3D urban features, generating of geometrically complex type features, supporting of both database and file system, handling of attributes for 3D features, texture mapping of complex types of 3D objects and digital elevation model. With these functions, an integrated urban 3D modeling and rendering system was implemented using standard mobile 3D graphic API. As well as this work for mobile 3D design and implementation, multiple types of geographic features encoded by GML (Geography Markup Language) for the XML-databases building are taken into account of practical applicability for 3D urban modeling approaches. For this process, it was carried out to implement GML-processing software with the main functions: supporting and customizing map style sheet, attribute editing, manipulating geo-base features, web publishing, importing/exporting ESRI-shape file and data transferring to X3D for 3D complex features. It is thought that functions handling X3D from GML or XML database can be used as major components within web-based geospatial data services supporting complex types of transportation application data model. These implementations such as mobile 3D and web 3D can be easily extended to practical needs for data communication and sharing of multiple and complex typed urban features on an integrated architecture.

Prototype Implementations for Multi-typed 3D Urban Features Visualization on Mobile 3D and Web 3D Environments

Kiwon LEE, KOREA

1. INTRODUCTION

Recently, 3D urban modeling is regarded as one of the important applications based on the geo-spatial information. Moreover, GIS as one of main streams in IT meets new technological trends. Regarding these issues, two aspects for 3D are considered in this study: mobile 3D and web 3D.

Related to the urban applications of GIS, some researches in the mid 2000s have given useful token for 3D urban modeling with aerial images and satellite imagery, in the aspect of 3D visualization (Suveg and Vosselman, 2004; Takase *et al*., 2004; Varshosaz, 2004, Zhou *et al*., 2004). While Kolbe *et al*.(2005) presented CityGML schema for 3D urban modeling, which used for XML-DBMS building for urban applications. While, Peng (2006) studied in-situ 3D modeling with a virtual city.

As for mobile 3D technology, it is based on 3D CG (Computer Graphics), and it covers with general CG pipeline and pixel pipeline from 3D geometric modeling in 3D model coordinate system to 2D rendering on the projected plane. Ervin and Hasbrouck (2001) overviewed and categorized 3D computer graphic techniques for 3D landscape modeling. As for graphic pipeline engine for 3D visulization, several de-facto standard APIs (Application Programming Interface) can be used: OpenGL, Direct3D, Java3D, and so forth, and OpenGL API is adopted a prototype implementation in this study, mainly due to directly linked with OpenGL|ES (Embedded System) API for mobile 3D application.

Mobile 3D is not a main component of GIS application yet. But there are some cases of mobile 3D. Rakkolainen and Vainio (2001) and Brachtl *et al*.(2001) developed a mobile browser for 3D simple features and a PDA based 3D navigation system, respectively. Lipman (2004) implemented 3D construction visualization model on VRML (Virtual Reality Modeling Language) browser. Sanna *et al*. (2004) proposed a basic architecture of complex 3D model on PDA. Lee and Kim (2006) developed a 3D mobile authoring and visualization system for complex type features using OpenGL|ES API. Nurminen (2006) developed a 3D mobile city map named m-LOMA.

Nevertheless, mobile devices are widely used in the GIS application domain, and practical needs regarding mobile GIS by wireless communication and various sensor systems are abruptly increasing. However, most mobile GIS-based applications are currently used to manipulation and processing for 2D geo-based feature. In order to handle large volume of spatial data and perform complex operation and 3D graphic process, these devices prevent the developers from proceeding conventional design and implementation processes, due to severe blocks to resolve: limited CPU and memory, absence or limited performance of graphic accelerators, small size of display panel, and so on.

As for web 3D technology for geo-spatial information processing, two main concepts are considered: GML (Geography Markup Language) and X3D (Extensible 3D Graphics).

GML is a markup language that is used to encode both spatial and non-spatial geographic information. GML is used to express geographic information in a manner that can be readily shared on the internet. One of the advantages of using GML is that it enables one to leverage the whole world of XML technologies. In particular, GML builds on XML (eXtensible Markup Language), XML Schema, XLink, and XPointer. GML data can also be easily mixed with non-spatial data (Peng and Tsou, 2004).
The development of real-time communication of 3D data across all applications and network applications has evolved to the X3D standard. X3D is a royalty-free open standards file format and run-time architecture to represent and communicate 3D scenes and objects using XML. It is an ISO ratified standard that provides a system for the storage, retrieval and playback of real time graphics content embedded in applications. Currently, interests on VRML/X3D application have been increased in the various domains.
Honjo and Lim (2001) developed landscape visulization system using VRML, and Yan (2006) developed an integrated visulization system based on web 2D/3D, as SVG (Scalable Vector Graphics) and X3D/VRML. Further, Hetherington *et al*.(2006) applied X3D to information rich virual modeling. Nadalutti *et al*.(2006) developed a mobile X3D browering system, being implemented with OpenGL|ES.
The main themes dealt with this study are several standard technologies on mobile 3D graphics, GML and X3D. As shown in the previous works, these technologies are separate technology items, but they are synergic in an integrated system or application. Therefore, a prototype for integrated applications using these is tentatively implemented in this study, and an application strategy for mobile 3D and web 3D in urban modeling purpose is presented.

2. APPLIED STANDARD TECHNOLOGIES

2.1 Mobile Graphic API: OPENGL|ES

OPENGL|ES which stands for OPENGL for Embedded Systems, released by the Khronos Group (Astle and Durnil, 2004), is a low-level, lightweight API for advanced embedded graphics using well-defined subset profiles of full OpenGL API.
As 3D graphic pipeline and pixel pipeline processes (Knaus, 2003), OpenGL|ES provides functions for primitives and vertex arrays for 3D geometric modeling, cooridinate functions, color and lighting functions, and buffering and pixel operations. As well, texture processing functions in OpenGL|ES API are enough to implement a certain actual images such as digital photo, aerial images, and satellite images.

2.2 GML (Geography Markup Language)

Geography Markup Language (GML) terms an XML encoding for the transport and storage of geographic information, including both spatial and non-spatial properties of geographic features. It is the key information technology behind the geo-spatial Internet (Ron *et al*., 2004). Using GML, we can deliver geographic information as distinct features, and then control how they are displayed in a web browser. Features, which describe real world entities, are the fundamental objects used in GML. GML features can be concrete and tangible, or abstract and conceptual. As well, GML features are described in terms of their properties,

which can include spatial (geometric or topological), temporal or other non-spatiotemporal descriptions of the feature. Rather, GML concrete features must be defined in GML application schemas, which are created by users such as database administrators (Peng and Tsou, 2003; Ron *et al*., 2004).
GML provides a set of core schema components (e.g. features, geometry, topology, temporal, etc) together with a simple semantic model between objects and properties that is similar to E-R (Entity-Relationship) diagrams or the class and property model of RDF (Resource Description Framework). Using the GML model and its schema components, users can describe the geographic types, whether concrete or conceptual, which are used within their application domain. The set of objects is created in the form of one or more GML application schemas, which is XML schemas that make use of the GML schema components, and which comply with the GML semantic model and syntactic rules.
XML schema is regarded as the implementation encoding of GML. Therefore, GML is composed of two kinds of document: schema document of defining structure and schema instance document. The practical core concept of GML is feature, geometry and xlinks. Feature describes properties (name, type, value). FeatureCollection is one of feature types for complex typed real objects. Geometry is used for describing geometry properties of feature. It supports both 2D and 3D geometry component, such as point, polyline, polygon, multipoint, mulitpolyline, solid and etc. While xLinks is for reference relationship between objects.

2.3 X3D (Extensible 3D Graphics)

X3D is to define various interactive web-based 3D content including web 3D graphics which can be integrated with multimedia across a variety of hardware platforms, extending the earlier VRML(Virtual Reality Modeling Language). It is regarded as a universal interchange format for integrated 3D graphics and multimedia, since it is represented by the XML. Conceptually, every X3D application contains graphic objects which can be dynamically updated through a variety of means based on the delivery context and the user's own design.
The basic unit of X3D runtime environment is the scene-graph, which is directed, acyclic graphs containing the objects in the 3D world, in addition to relationships among the objects. These relationships consist in the transformation hierarchy on spatial relationships and the behavior hierarchy on fields and event flows in the 3D space. In these hierarchies, nodes within the scene graph contain descriptive fields and one or more child nodes to produce the desired hierarchy of objects in the scene. X3D supports 3D functions such as polygonal geometry, parametric geometry, hierarchical transformations, lighting, materials, multi-pass/multi-stage texture mapping, pixel and vertex shading, and hardware acceleration, in addition to text, 2D vector graphics, and 2D/3D compositing.

Fig. 1 shows relationships between XML (GML) and X3D with XSLT (eXtensible Stylesheet Language Transformations), which controls graphic attributes in visualization process by transforming an XML document into other formats.

Fig. 2 represents the main purposes and relationship of three graphic APIs.

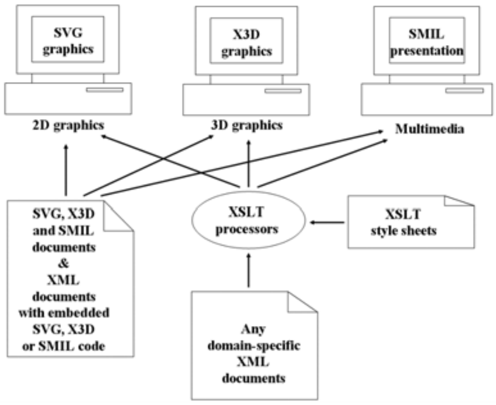

Fig. 1. XML(GML), X3D, and XSLT in web environment (Geroimenko and Chen, 2005).

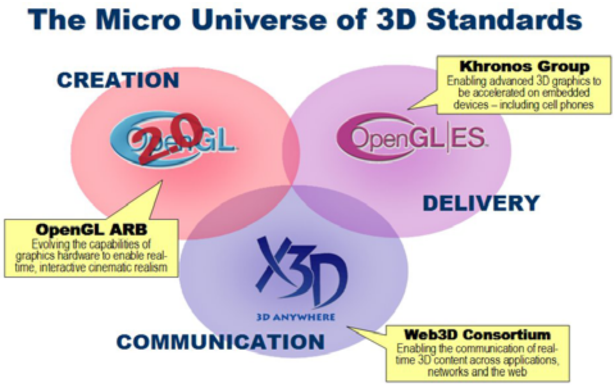

Fig. 2. 3D Standards for graphic application developers: OpenGL-OpenGL|ES-X3D, excerpted from SIGGRAPH materials.

3. IMPLEMENTED RESULTS

3.1 3D Data Model and System Design

One of major interests in this study is 3D structure of the urban environment. A variety of urban features, like large and tall commercial buildings, small buildings in residential areas, parks with trees and fences, roads and railroads, can be considered for this approach. For generalization, these features were grouped into several categories: terrain, building, transportation, vegetation, and user-defined ones.
In OPENGL-based 3D graphic application, because this API does not provide its own data structure, a simple 3D data structure, 3D vector model with database attributes, was designed and applied for mobile devices. Urban building can be modeled by the composite types of 3D primitives such as simple cube, triangle, square, pentagon, hexagon or other ones, similar to general 3D modeling tools in other 3D graphic applications. As one of important urban composing features, transportation feature such as road, traffic sign board, or other

transportation utilities can also be modeled and edited. In the case of road, polygon primitives in the 3D coordinate system are used. In terrain modeling, DEM and TIN data can be used in data modeling, and TIN model is used in the rendering stage due to its storage space efficiency. This modeling process was carried out in GML scheme and visualization was performed X3D and XSLT.

This mobile 3D visualization system was implemented using standard mobile 3D graphic API, OPENGL|ES by Kronos group, for mobile platform using MS EVC 4.0 MFC. This system also tested in PDA with specifications as follows: Window Mobile 2003 SE, Intel Strong ARM 312 MHz, 320 by 240 65K display, and NAND memory of 64MB and SDRAM 64 MB without hardware accelerator.

Fig. 3 shows a framework model schema in GML for urban modeling. City class is composed of basic sub classes such as road, building, facilities and elevation surface sub-class. The relation between city class and each sub class is inclusion, and the algebraic relation is one to many because one city has many buildings, road structures, facilities on elevation surface. Linking with GML, City class, as compound type including sub-classes is derived from FeatureCollection type of GML, and each sub-class is derived from Feature type of GML. City object, which is derived from FeatureCollection of GML, has elements of gml:description for its explanation, gml:name for its name, gml:boundedBy for its range and cityMember for containing structure objects. The structured object of City, as derived from feature of GML has its attribute and geometry property. For 3D geometric characteristics of these objects, gml:solid as GML geometric component is used, and it has multi-facet by 3D coordinates. While, surface elevation uses gml:GridCoverage element with gml:domainSet, gml:rangeSet, gml:coverageFunction, to encode DEM data set into GML.

For building component, we produced ASCII file of GML document with actual geo-based coordinates, which it was encoded using solid element, as 3D component of GML schemas and it added to building layer of urban city model. Especially, for surface elevation component, gridcoverage was used to encode satellite images into GML.

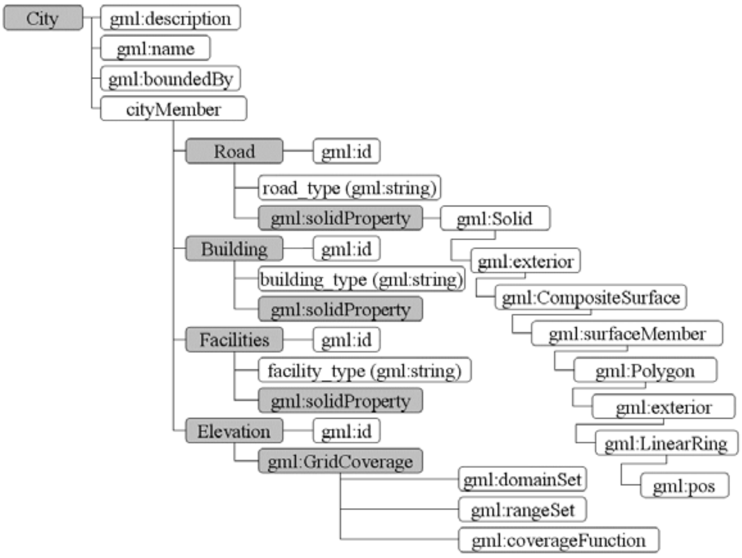

Fig. 3. GML application schema of urban modeling.

Fig. 4 shows a part of GML instance document in this framework model. Because GML document can be written in a text editor, manual GML typing is a time consumptive work. Therefore, implementation for semi-automatic generation of GML instance document was carried out using DOM(Document Object Model, W3C Standard) API which is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. ElevationGrid and Indexedfaceset element, as X3D geometry component are used to represent terrain and building. The one is Grid object including height. Range and height values of it are assigned by Gridcoverage of source data (GML instance). The other is object that can create 3D geometry using coordinates of each surface of target. Coordinate values of it are assigned by solid of source data (GML instance). Fig. 5 shows an example of XLST processing for web 3D visulization in X3D.

The development and operation environment of this implementation is as follows: Visual C++ 6.0 MFC and MSXML 4.0 SDK as development environment of the main modules and user interface, MSXML 4.0 as parser.

```
- <gml:featureMember>
  - <tta:TRN_RoadNetwork_RoadSegment>
      <tta:UFID>S1</tta:UFID>
      <tta:RNM>RNS</tta:RNM>
      <tta:STJ>J1</tta:STJ>
      <tta:EDJ>J4</tta:EDJ>
      <tta:ORT>001</tta:ORT>
      <tta:RNO>RNS1</tta:RNO>
      <tta:RCT>005</tta:RCT>
      <tta:UTL>999</tta:UTL>
      <tta:UTN>etc</tta:UTN>
    - <gml:centerLineOf>
      - <gml:LineString>
          <gml:coordinates>192290.785549,447191.784494
            192461.502357,447322.745334</gml:coordinates>
        </gml:LineString>
      </gml:centerLineOf>
    </tta:TRN_RoadNetwork_RoadSegment>
  </gml:featureMember>
- <gml:featureMember>
  - <tta:TRN_RoadNetwork_RoadSegment>
      <tta:UFID>S2</tta:UFID>
      <tta:RNM>RNS</tta:RNM>
      <tta:STJ>J4</tta:STJ>
      <tta:EDJ>J5</tta:EDJ>
```

(A)

```
- <TRN_StructuresMember>
  - <TRN_Structures_SubwayEntr>
      <SSN>a</SSN>
      <SNM>b</SNM>
      <SNO>c</SNO>
    - <gml:solidProperty>
      - <gml:Solid>
        - <gml:exterior>
          - <gml:Surface>
            - <gml:patches>
              - <gml:PolygonPatch>
                - <gml:exterior>
                  - <gml:LinearRing>
                      <gml:coordinates>0 0 0,1 0 0,1 1 0,0 1 0,0 0
                        0</gml:coordinates>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:PolygonPatch>
              - <gml:PolygonPatch>
                - <gml:exterior>
                  - <gml:LinearRing>
                      <gml:coordinates>1 0 0,1 0 1,1 1 1,1 1 0,1 0
                        0</gml:coordinates>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:PolygonPatch>
```

(B)

Fig. 4. GML encoding cases: (A) a part of the rail road in 2D, (B) a part of the subway structure in 3D.

3.2 Implemented Results and Application Strategy

Fig. 6 represents several cases in mobile 3D on PDA: (A) and (B) show browsing results of texture mapped 3D scene using Pocket Cortona browser and Mobi3D by Nadalutti *et al.*(2006), respectively. But (C) ~ (E) show actual 3D geo-spatial feature authoring and an integrated 3D scene composed of multiple 3D features, terrain, and satellite imagery. Fig. 6 (C) ~ (E) are implemented results in this study.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:gml="http://www.opengis.net/gml" xmlns:tta="http://www.hansung.ac.kr/Transportation">
  <xsl:output method="xml" encoding="UTF-8" media-type="model/x3d+xml" indent="yes"
  cdata-section-elements="Script"
  doctype-system="http://www.web3D.org/TaskGroups/x3d/translation/x3d-compact.dtd"/>
  <xsl:template match="/">
    <X3D profile="Immersive">
      <Scene>
        <Group DEF="SubwayEntr">
          <xsl:for-each select="tta:Transportation/tta:TransportationMember/tta:TRN_Structure/
           tta:TRN_StructureMember/tta:TRN_Structures/tta:TRN_StructuresMember/tta:TRN_Structures_SubwayEntr/
           gml:solidProperty/gml:Solid/gml:exterior/gml:Surface/gml:patches/gml:PolygonPatch">
            <xsl:variable name="coord" select="gml:exterior/gml:LinearRing/gml:coordinates"/>
            <Shape>
              <Appearance>
                <Material diffuseColor="1 0 0" emissiveColor="0 1 0" specularColor="1 1 0"/>
                <ImageTexture url='"subway.jpg"'/>
              </Appearance>
              <IndexedFaceSet convex="false" solid="false" coordIndex="0, 1, 2, 3, -1">
                <Coordinate point="{$coord}"/>
              </IndexedFaceSet>
            </Shape>
          </xsl:for-each>
        </Group>
      </Scene>
    </X3D>
  </xsl:template>
</xsl:stylesheet>
```

Fig. 5. An example of XLST from GML to X3D with 3D properties for visulization.

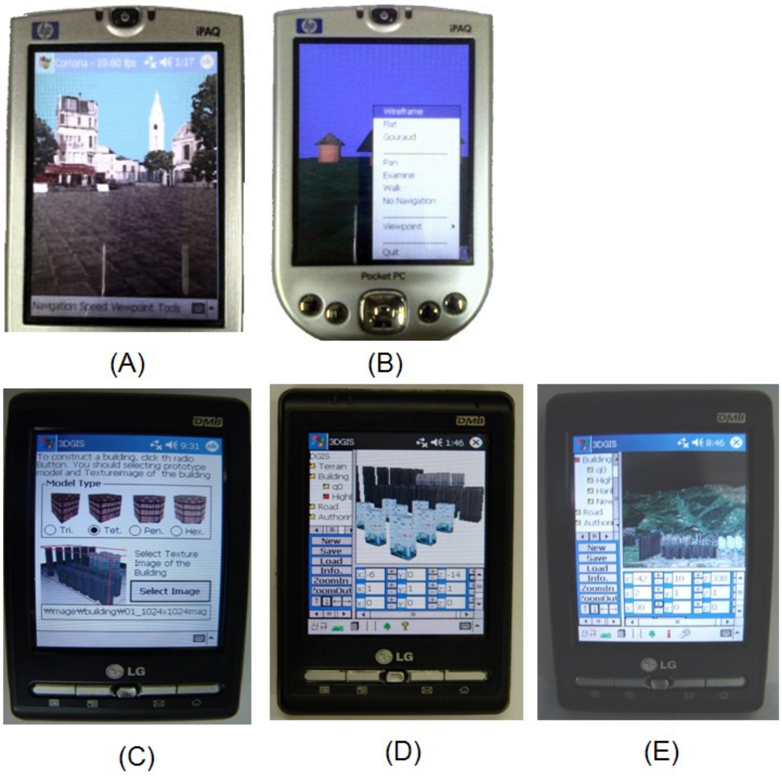

Fig. 6. Mobile 3D on PDA: (A) Texture mapped 3D scene using Pocket Cortona browser, (B) Mobi3D by Nadalutti *et al*.(2006), (C) 3D geo-spatial feature authoring, (D) User interface for 3D geo-based features modeling, (E) An integrated 3D scene composed of multiple 3D features, terrain, and satellite imagery.

Fig. 7 shows an implemented result of mobile 3D (A) and web 3D (B) with respect to 3D terrain model which is generated from DEM and satellite imagery. Especially, user interface for GML encoding and X3D conversion is represented in (B). This result is one case according to the developement and application strategies proposed in this study (Fig. 8).
This approach is composed of GML database building in server side and three clients (Fig. 8): web client in web browser plugged in X3D viewer such as Octago Player (http://www.octaga.com/), mobile client or general user in mobile browser such as Pocket Cortona (Fig. 6(A)) or Mobi3D (Fig. 6(B)), and mobile 3D urban modeling system in this study (Fig. 6(C) ~(E) and Fig. 7(A). In this strategy, several 3D urban applications are possible: ① web 3D visulization for 3D urban application through GML database-XLST-X3D, ② mobile 3D visulaization for 3D urban application through GML database-XLST-X3D, ③ mobile 3D application by GML database model to direct OpenGL or OpenGL|ES programming, ④ mobile 3D application for database sharing with web environment by GML database-XLST-X3D-OpenGL|ES, and ⑤ mobile 3D urban feature modeling or authoring system for database builder, mobile operator, and client. These five cases are not needed any other commercialized tools, and GML database also linked with legacy database and other GIS data structure/file formats and image file formats.

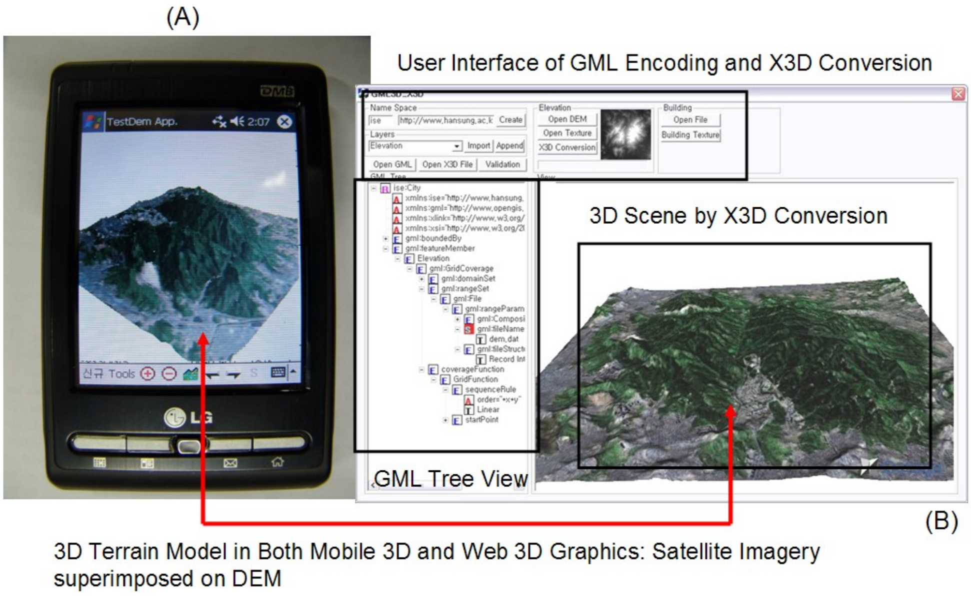

Fig. 7. Terrain visualization in Mobile 3D by OpenGL|ES API and Web 3D by X3D: DEM with satellite imagery.

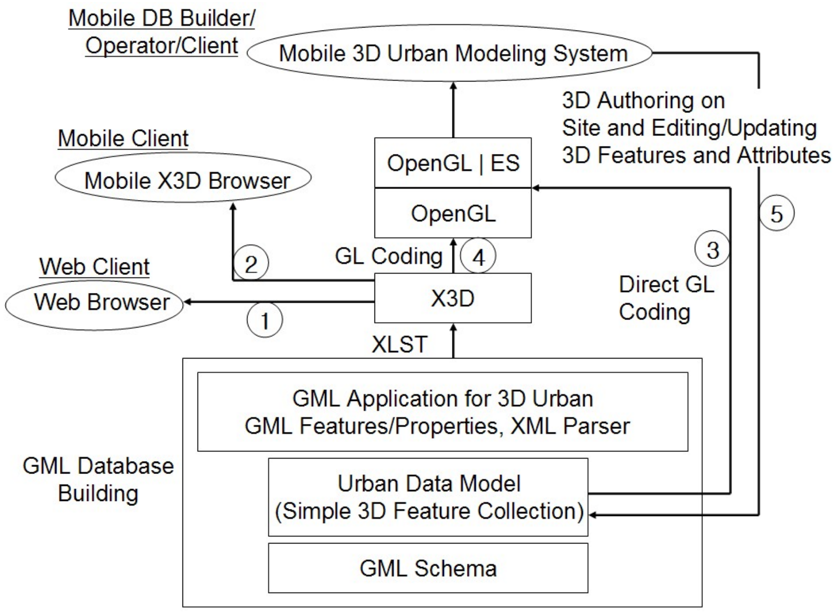

Fig. 8. Application strategy of mobile 3D and Web 3D for 3D urban visualization application.

4. CONCLUDING REMARKS

In this study, three standard technologies such as GML, X3D, and OpenGL|ES are used for the purpose of 3D urban visualization application. A prototype for 3D urban visualization in both web 3D and mobile 3D environments was implemented for more advanced and pratical applications without commercialized tools. However, 3D urban data model in this work is just a simple framework model, but it can be extended to more detailed data model, dependent upon the application level handling more realistic complex-typed 3D features. In the case of web 3D, X3D graphical processing and scheme is an essential component of smart graphics which could be structured and semantic graphics. Especially, mobile 3D graphics is the beginning stage for GIS development and application, and the proposed approach may be regarded as a guide for further researches in this subject.

REFERENCES

Astle, D. and D. Durnil, 2004. OPENGL|ES Game Development, Premier Press, 293p.

Brachtl, M., J. Slajs, and P. Slavik, 2001, PDA based navigation system for a 3D environment, Computers and Graphics, 25: 627-634.

Ervin, S. M. and H. H. Hasbrouck, 2001. Landscape Modeling: Digital Techniques for Landscape Visualiztion, McGraw-Hill, 289p.

Hetherington, R., B. Farrimond, S. Presland, 2006. Information rich temporal virtual models using X3D, Computers and Graphics, 30: 287-298.

Honjo, T. and E.-M. Lim, 2001. Visualization of landscape by VRML system, Landscape and Urban Planning, 55: 175-183.

Knaus, C. (ed), 2003. OpenGL ES 1.0 Reference Manual Version 1.0, Silicon Graphics, Inc., 226p.

Lee, K. and S.-Y. Kim, 2006. Development of Mobile 3D Urban Landscape Authoring and Rendering System, Korean Journal of Remote Sensing, 22(3): 1-8.

Lipman, R. R., 2004. Mobile 3D visualization for steel structures, Automation in Construction, 13: 119-125.

Nadalutti, D., L. Chittaro, and F. Buttussi, 2006. Rendering of X3D Content on Mobile Devices with OpenGL ES, Proceedings of Web3D 2006: 19-26.

Nurminen, A., 2006. m-LOMA – a Mobile 3D City Map, Proceedings of Web3D 2006: 7-18.

Geroimenko, V. and C. Chen, 2005. Visualizing Information Using SVG and X3D: XML-based Technologies for the XML-based Web, Springer, 298p.

Kolbe, T.H., G. Groger, and L. Plumer, 2005. CityGML-Interoperable Access to 3D City Models, Proceedings on The First International Workshop on the Next Generation 3D City Models.

Peng, C., 2006. In-situ 3D concept design with a virtual city, Design Studies, 27: 439-455.

Peng, Z.-R. and M.-H. Tsou, 2003. Internet GIS: Distributed Geographic Information Systems for the Internet and Wireless Networks, Wiley, 679p.

Rakkolainen, I. and T. Vainio, 2001. A 3D City Info for Mobile Users. Computers and Graphics, 25: 619-625.

Ron L., Burggraf, D. S., Trninic, M. and Rae, L., 2004. Geography Markup Language: Foundation for the Geo-Web, Wiley, 388p.

Sanna, A., C. Zunino, and F. Lamberti, 2004. A distributed architecture for searching, retrieving and visualizing complex 3D models on Personal Digital Assistants, Int. Jour. Human-Computer Studies, 60: 701-716.

Suveg, I. and G. Vosselman, 2004. Reconstruction of 3D building models from aerial images and maps, ISPRS Journal of Photogrammetry & Remote Sensing, 58: 202-224.

Takase, Y., N. Sho, A. Sone, and K. Shimiya, 2004. Automatic generation of 3D city models and related applications, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, XXXIV-5/W10.

Varshosaz, M., 2004. True realistic 3D models of buildings in urban areas, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS, XXXIV-

5/W10.
Yan, W., 2006. Integrating Web 2D and 3D Technologies for Architectural Visualization: Applcation of SVG and X3D/VRML in Environmental Behavior Simulation, Proceedings on Web3D, ACM: 37-45.
Zhou, G., C. Song, J. Simmers, and P. Cheng, 2004. Urban 3D GIS from LiDAR and digital aerial images, Computers & Geosciences, 30: 345-353.

BIOGRAPHICAL NOTES

Academic experience: B.S.(1987), M.S(1989), Ph.D.(1995), Seoul National University
Current position: Associate Professor, Dept. of Information System Engineering, Hansung University, Seoul, Korea
Practical experience: Digital mapping system design and development, 3D computer graphics, Satellite image processing
Activities: Executive director, Korean Remote Sensing Society, 2005 –
Member, ASPRS, Member, Korean Geo-spatial Information System Society

CONTACTS

Prof. **Kiwon LEE**
Hansung University
Dept. of Information System Engineering, Hansung University
389 Samsun-dong, Sungbuk-gu
Seoul, 136-792
KOREA (ROK)
Tel. +82 2 760 4254
Fax +82 2 760 4347
Email: kilee@hansung.ac.kr