Technological University Dublin

ARROW@TU Dublin

Books/Book chapters

Digital Media Centre

2015-05-19

Design and Development of Personal GeoServices for Universities

James Carswell
Technological University of Dublin, jcarswell@tudublin.ie

Andrea Ballatore
University of California

Thoa Pham
National University of Ireland, Maynooth

Junjun Yin
University of Illinois

Linh TruongHong
University College Dublin

Follow this and additional works at: https://arrow.tudublin.ie/dmcbk

Part of the Computer Engineering Commons

Recommended Citation

Design and Development of Personal GeoServices for Universities

James Carswell[1], Andrea Ballatore[2], Thoa Pham[3], Junjun Yin[4], Linh Truong-Hong[5]

[1]Digital Media Centre, Dublin Institute of Technology (DIT), Ireland
jcarswell@dit.ie

[2]Center for Spatial Studies University of California, Santa Barbara, USA
aballatore@spatial.ucsb.edu

[3]Department of Computer Science, National University of Ireland, Maynooth (NUIM)
aballatore@spatial.ucsb.edu

[4]Department of Geography and Geographic Information Science, University of Illinois at Urbana-Champaign (UIUC), USA
thoa.phamthi@gmail.com

[5]Department of Civil Engineering, University College Dublin (UCD), Ireland
linh.truonghong@gmail.com

Abstract. Personal GeoServices are emerging as an interaction paradigm linking users to information rich environments like a university campus or to Big Data sources like the Internet of Things by delivering spatially intelligent web-services. OpenStreetMap (OSM) constitutes a valuable source of spatial base-data that can be extracted, integrated, and utilized with such heterogeneous data sources for free. In this paper, we present a Personal GeoServices application built on OSM spatial data and university-specific business data for staff, faculty, and students. While generic products such as Google Maps and Google Earth enable basic forms of spatial exploration, the domain of a university campus presents specific business information needs, such as "What classes are scheduled in that room over there?" and "How can I get to Prof. Murray's office from here?". Within the framework of the StratAG project (www.StratAG.ie), an eCampus Demonstrator was developed for the National University of Ireland Maynooth (NUIM) to assist university users in exploring and analysing their surroundings within a detailed data environment. This work describes this system in detail, discussing the usage of OSM vector data, and providing insights for developers of spatial information systems for universities.

1   Introduction

As today's spatially aware users are becoming more sophisticated and inter-
ested in retrieving more personalized information[1], they will increasingly re-
quire more detailed two dimensional (2D) and three dimensional (3D) city-
scapes linked to relevant non-spatial attribute data. In this work, Personal
GeoServices are developed that employ mobile devices (spatially aware
smartphones) to provide contextual search and visualization utilities over 2D
OpenStreetMap (OSM) building footprint data and detailed 3D building mod-
el data. At smaller map scales, Google Maps/Earth with satellite/street views
can assist users searching for general information at specific locations. Users
can search a street address on the map, then explore the location in street view
mode, or find how to reach a certain address or location given options like
pedestrian or driving constraints. However, generic query tools available in
these familiar products are usually limited to keyword-based search. At larger
local scales, where detailed 2D and 3D geometries and associated business
data are needed, there is a recognized lack of advanced spatial search func-
tionality and linked attribute information available in these products for do-
main-specific, task-oriented, and personalized visual exploration of an area
[18].

  For instance, the following types of questions (queries/searches) cannot be
answered when interacting with Google Maps/Earth on a typical university
campus: "What classes are scheduled in that room over there?"; "Whose of-
fice window is that up there?"; "What computer labs can I actually see around
me from this location on campus?"; "What are their opening times?". In order
to answer these types of task specific queries, Location Based Services (LBS)
need the ability to search and link spatial map data together with non-spatial
business data.

  Spatial data includes detailed topology and geometry of objects while busi-
ness data can describe the attributes or semantic aspect of related objects in
some business domain. Conventional business data is often produced and
managed by traditional enterprise information systems, often ignoring the
spatial dimension altogether. However, it can often be indirectly or virtually
associated to spatial data via its location given by a generic address, room
number, building name, or sometimes even geographic (lat/long) coordi-
nates [18]. Linking spatial data and business data together in one application
can help to fulfil more task specific needs. In particular, decision-making ap-
plications need access to detailed local-scale data typically found in museums,
hospitals, shopping malls, retail/office park settings, or a university campus.
Business data specific to a university campus may be in the form of class

---

[1]   http://googleblog.blogspot.ie/2013/01/mapping-creates-jobs-and-drives-
global.html

schedules for a specific classroom, lists of equipment installed in a lab, office hours or contact details for a lecturer, today's special meal deal in the cafeteria, etc.

Typically, 2D vector "footprint" data provides just boundary geometry representation of physical objects (e.g. buildings, roads, rivers, etc.) in the horizontal plane. However, in our eCampus application, we link spatial and non-spatial attribute details in the vertical dimension as well for more advanced 3D information search operations. For instance, 3D model data of a building can include detailed digital representations of physical and functional characteristics for its different floors, rooms, windows and doors, where all objects are potentially available for interrogation.

In this paper, we describe our prototype eCampus information system in which the ideas above are implemented. Within the framework of the StratAG project (www.StratAG.ie), the eCampus Demonstrator was developed for the National University of Ireland Maynooth (NUIM), in collaboration with Dublin Institute of Technology (DIT), and University College Dublin (UCD). This Personal GeoServices application aims to assist users in exploring and analyzing their surroundings to answer more task specific user queries within a detailed data environment. The Demonstrator addresses two types of users: public users (e.g. visitors) and local users (e.g. students, faculty, and staff). Access privileges and query levels depend on user type. For example, visitors are presented with a campus map for general information querying about campus buildings and rooms. Visitors are also provided with general campus news, events, and utilities for navigation to various buildings or rooms, or any other locations on campus using different routing options. In addition to these general functions, staff and students are able to overlay on the map their individual class schedules together with personalized news feeds and events tailored to their academic and social interests.

Each project partner in the StratAG cluster is responsible for developing different functionality in the eCampus Demonstrator, such as; utilities for 2D/3D directional and visibility-based querying (DIT), path navigation assistance (NUIM), personalizing news and events according to user interests (UCD), or for detailed mapping and modelling of the campus infrastructure itself (NUIM/DIT). RESTful web-services [1, 3, 5] were chosen as the deployment technology for these distributed components due to its simplicity when applied to the geospatial domain [7]. Regarding 3D maps, there are several commercial and free mapping products available that allow users to incorporate 3D building models for such visualization and interaction type applications. Of these, Google Earth (GE) was selected for displaying the 3D building models in this work because it is both free and increasingly familiar to web and wireless GeoService users, although currently GE does have voluminous data processing limitations that must be addressed to accommodate real-time display.

In the remainder of the paper, we first discuss some related work before introducing our approach to integrating 2D OSM and 3D model data within the application. The 2D/3D building information is then imported into a spatial database and converted to GE readable format in the case of 3D map display. Then we present the system architecture of the eCampus Demonstrator based on a Resource Oriented Architecture (ROA) model. Some unique search functionalities of eCampus are described in detail together with its graphical user interfaces. Finally we draw conclusions and give some possible direction for future work.

2   Related eCampus Applications

Assisting people with exploring an area, such as a university campus, with a mapping interface is very useful, and is not a new idea in itself. Some existing projects provide this type of GeoService, as listed below.

Kent State University Campus Maps [14] is a web-based application providing an interactive 2D map with detailed information and images of each building in the campus. It also highlights specific locations on the campus such as computer labs, parking, sculpture walk, and residence halls. The Get Direction functionality allows finding pedestrian/driving/cycling routes between two locations, e.g. using building names. However, this application provides very basic query/search functionality overall, have no 3D maps, or any form of personalization.

The Interactive Map developed in University of California; Berkeley [15] provides an isometric campus map with clickable buildings to provide detailed information and an image of each building. There is also a list of building names placed outside the map window from where users can choose a name, and then the corresponding building object on the map is highlighted. This application has no search facility, no routing service, and provides limited spatial support.

The University College Dublin Mobile services [16], released natively on Android and iOS, has the following functionalities: Campus maps, details about places, and tours; Campus directory; Access to Library; Access to e-learning facility (Blackboard); Schedule of general events (lectures, concerts, etc.); Campus news; Image search on university archives; and Emergency numbers. The application has two main versions, one for staff and one for students, with different permissions. Although this application provides many useful services, particularly in relation to campus events, its spatial support is still very limited. It has only a simple, non-interactive 2D map, with no routing functionality.

In [9], the early stages of a web-based campus information system was developed for the University of Karabuk, Turkey, allowing users to explore the

university campus in 3D. It provides information at the building level and points of interests, but room level details have not been fully incorporated, and the implementation does not provide utilities to further query the area beyond its physical, spatial nature.

The Youngstown State University developed interactive 2D and 3D campus maps [30] which allows the retrieval of building information when clicking on a building or to download a KMZ file (zipped KML archive file) of an area to interact with in Google Earth. However, the attribute information provided is also limited to building level only.

A clear trend is evident in the above GeoService applications in that more advanced functionality for exploring non-spatial business data of sub-objects like rooms, windows and doors to retrieve the content, schedule, or purpose of a specific room in a building is still obviously lacking. This is largely due to insufficient levels of geometric granularity of building models available in today's online mapping platforms like Google Earth, and importantly a subsequent lack of any spatially linked business data.

3   OSM and Volunteered Geographical Information (VGI)

The availability of detailed geographic data is critical for delivering comprehensive GeoService applications. Most geospatial data in Europe is collected and controlled by either national mapping agencies (e.g. OSi in Ireland) or private companies, such as Google Maps, Yahoo! Maps, and Bing Maps. However, data coverage over many areas can still be of considerably poor quality and extent – especially in less populated areas.

Some initial research to address this shortcoming looked at imaging and georeferencing public displays of "You Are Here" type maps to fill the coverage gap for local navigation purposes [28]. However, a rapid growth in volunteered geographic information (VGI) has also started to fill this gap with OSM being a successful example of this. In relation to VGI, this "citizen-assensor" paradigm contributes to OSM by creating, assembling and disseminating geospatial features including streets, highways, buildings, etc. and gradually this collective geospatial information shows surprising coverage all over the world [29, 31]. Another vitally important feature of using OSM data is that "you are free to copy, distribute, transmit and adapt our maps and data, as long as you credit OpenStreetMap and its contributors"[2], which affords users to build value-added applications on top of it.

Considering this increasing trend of free VGI sourced data, our project is built upon base geospatial data from OSM covering the National University of Ireland, Maynooth (NUIM) and its surrounding areas. A screenshot of the

---

[2]   http://www.openstreetmap.org

OSM map coverage for this area at the time of project implementation is shown in Figure 1(a), while the corresponding map coverage over the same time/area from Google Maps is shown in Figure 1(b). The detailed OSM data is mainly created by students from NUIM and is freely available for inclusion in all value-added projects, a clear example of how OSM grows research and business opportunities through volunteers contributing data [13].
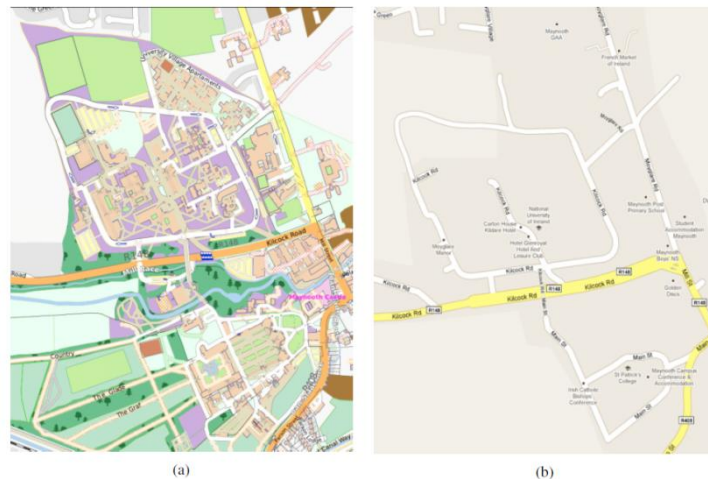

(a)   (b)

Fig. 1. (a) 2012 OSM map coverage of NUIM, Ireland; (b) Map coverage of the same place and time from Google Maps.

In this project, 2D footprints of NUIM campus buildings were downloaded directly from the OSM map interface (www.openstreetmap.org) and/or from its data repository (e.g. Planet OSM[3] and Metro Extracts[4]). For NUIM campus data, which covers a relatively small area, the entire campus can be exported by drawing a bounding box over the campus region to specify the area of interest. The downloaded OSM data contains several layers of geometry types (features), such as polygons (e.g. building footprints), polylines (e.g. roads), and points (e.g. points-of-interest). In our case, a layer of building polygons was extracted and inserted into an Oracle Spatial database via the Feature Manipulation Engine (FME) Workbench utility [22], which is a toolset that converts and transfers data between different data formats. The inserted polygon data in Oracle Spatial is shown in Figure 2.

---

[3]   http://planet.openstreetmap.org/
[4]   http://metro.teczno.com/

Fig. 2. 2D footprints (polygons) of NUIM buildings inserted into Oracle Spatial.

4   Preparing 3D Models for Google Earth Integration

Google Earth (GE) allows developers to upload 3D building models for direct visualization in the GE environment or for inclusion in a webpage using their application programming interface (API). When retrieving data for visualization, a client (desktop or mobile) queries for data either from temporary cache memory or from GE databases. However, due to realtime display requirements and other issues, GE can only process very simple 3D block models if display speed is a priority. Therefore, in order to include our detailed 3D campus building models in GE, we first needed to develop a modelling workflow to transform raw LiDAR point cloud data to Google Earth KML format using various mapping tools like CloudWorx [21] and the FME Workbench utility. The KML format supports both solid and polygon data types, and both are needed to address real-time GE display issues as discussed below.

Raw data used to construct detailed building models for 3D city maps can be obtained from various sources through a wide range of techniques. With recent developments in photogrammetry and remote sensing, building models can be automatically reconstructed given the geometric resolution of satellite imagery and Light Detection and Ranging (LiDAR) point cloud data [19]. The geometric and semantic properties of 3D models are typically stored in five consecutive levels-of-detail (LoD), in which LoD0 defines a coarse regional

scale model while LoD4 denotes architectural building models with detailed walls, roof structures, balconies, interior structures, and detailed vegetation and transportation objects [27]. An advantage of the LoD approach is the coherent modelling of semantics and geometric/topological properties together at each level, where geometric objects get assigned to semantic objects. In order to meet our eCampus objectives, 3D building models have to be at least to LoD3 level.

To achieve this LoD, a modelling workflow (Fig. 3) was developed to reconstruct building models suitable for real-time GE display from LiDAR point clouds. LiDAR point clouds of campus buildings were first acquired using a Leica ScanStation C10 controlled by Cyclone-3D Point Cloud Processing Software installed on a laptop linked to the scanner. Subsequently, the point clouds were registered and geo-referenced within the Cyclone environment [20] and the 3D building models manually created using AutoDesk and a CloudWorx plug-in [21]. Leica CloudWorx for AutoCad offers many manipulation and editing tools to assist users to trace or auto fit lines, arcs and polylines to 3D point cloud data. Finally, by employing FME Workbench, a 3D building model's underlying CAD geometry is transformed to KML format to allow for online display by web-based mapping applications in Google Earth. A more detailed description of each step in this modelling workflow can be found in [17].
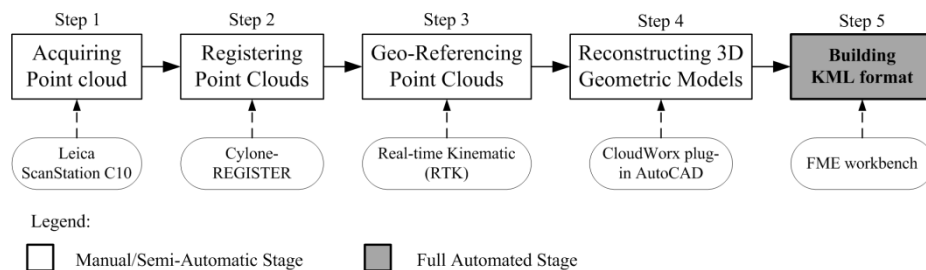

Fig. 3. LiDAR workflow for 3D BIM modeling.

This modelling methodology was applied to the north campus of the NUIM study area. This area was selected because it contains a mixture of simple and complex buildings with various architectural styles (e.g. historic and modern buildings), which can be most problematic when reconstructing 3D building models. The tallest building is 20 m. Due to GE constraints with how it processes (accesses/displays) 3D solids, the AutoCAD DWG format solid models created in Step 4 (above) have some limitations after converting to 3D solid KML format concerning access to their object/sub-object attributes (such as building/room name).

In Google Earth, linked attribute information cannot be accessed by simply clicking (pointing/selecting) just anywhere on the solid building shape. To

overcome this limitation, the solid building shape must be converted to 3D polygon data, which can be accessed in GE by clicking anywhere inside the polygon. However, by transforming all 3D solids to KML polygons, a map scene can contain a huge number of polygons because numerous polygons are required to represent all the doors/windows, which can make the KML file so large that it exceeds the limit that GE supports for real-time display. Therefore, the KML 3D polygon file has to be thinned to show important detail of attribute-bearing objects only. So, an overlay layer of empty 3D polygons around each window/door was added and the DWG solids of the doors/windows were transformed to KML solids for visualisation purposes, otherwise users would only see holes in a wall rather than doors or windows.

In summary, 3D solids and 3D polygons together represent the campus building models in KML format with their associated rooms linked to any available metadata information. While polygons are applied to all objects, 3D solids are also used for visualizing geometries of window/door frame details. This allows us to assign a different appearance to each building sub-component by filling with either a color or texture and importantly the ability to click anywhere inside a polygon shape (e.g. window/door/wall) to select/query the object directly.

5   eCampus Architecture

The eCampus Demonstrator is a browser-based application based on 2D OSM data and 3D GE data and is accessible to both desktop and mobile devices. It aims to help users explore in more detail the campus by providing those detailed maps and utilities for both 2D and 3D querying and visualisation. Different search functionality is provided so that users can ask questions by interacting with the map itself. For instance, they can ask: "What is that building over there?" by pointing at it with their mobile device; "What is the class schedule of this room?" by clicking on its window in the mobile/desktop display or by choosing a room id from a list; "What can I actually see around me?" when standing at a particular location on campus; they can also ask to visualise a route together with directional images (i.e. containing superimposed arrows pointing the way) between two buildings/rooms or any location by choosing a building/room from a list or by clicking locations directly on the map. Query results are visualised on the 2D/3D map overlaid with further business data where available. The application architecture includes 3 layers: interface layer, web-services, and database layer (Fig. 4).
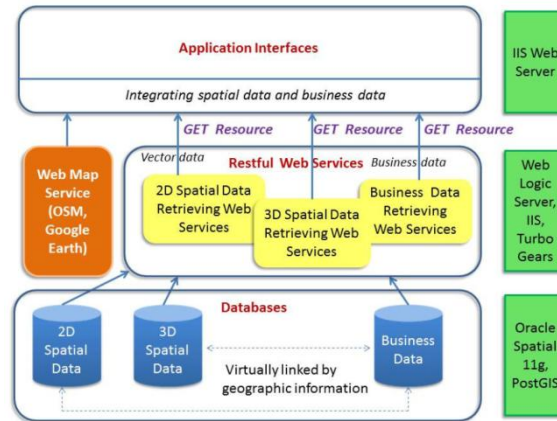
Fig. 4. Three Tier eCampus Demonstrator System Architecture.

5.1 Database layer

The 2D building footprint and detailed 3D models of NUIM campus that serve the spatial data web-services are physically hosted at DIT in Oracle Spatial 11g databases. Other data related to pathways, roads and building images are hosted at NUIM in PostGIS databases. Many 2D campus footprints were first downloaded from OSM and then uploaded to the Oracle Spatial DBMS where geometry data is stored in a single column data type of SDO_GEOMETRY to define the geometry type (e.g. points, lines, polygons, solids, etc.), the dimension, and an array of x, y (and z for 3D) coordinates comprising points or vertices of campus objects.

In Section 4, the workflow for 3D BIM modelling and export to KML format for displaying in GE was discussed. In fact, 3D building detail is also needed and stored in Oracle Spatial for visibility calculations and other advanced search operations. In this case, the FME workbench utility was also used to export the 3D models to the spatial database – a similar process as when exporting to KML. After reaching Step 4 (3D BIM in AutoCAD) in the modelling workflow, there are two groups of layers stored in the AutoCAD database: (i) stored solid components involving exterior walls, window/door frames, roofs, and balconies and (ii) stored polygons of window/door extents. These two layers are then exported to Oracle Spatial. As mentioned previously, the reason for this type of data management is because of "clicking" restrictions inherent to Google Earth regarding pointing/selecting solid building objects/sub-object attributes directly from the map.

5.2 Web-services layer

At the logical level, the spatial data retrieval web-services and business data retrieval webservices are installed in this layer.

Spatial data retrieval web-services.
Spatial data retrieval web-services include routing navigation, image retrieval for directional visualisation of routes, and 2D and 3D visibility based directional querying; 2D Isovist, point-to-select, and field-of-view queries; plus 3D Isovist, point-to-select, and frustum spatial searches. These web-services were developed by various project partners on different platforms, but have the same deployment methods in the form of RESTful web-services [1, 3]. More specifically, an IIS (Internet Information Services) server is appointed to host the web-services, and query requests are constructed using standard HTTP calls containing a valid URL filled with the required query parameters.

The routing navigation web-service needs query parameters such as transport mode (e.g. pedestrian, driving, wheelchair, or directional images), and a starting and destination location in terms of longitude and latitude coordinates taken from clicking on the map. The navigation web-service then returns a list of points (i.e. OSM object vertices along the route) in KML format. The eCampus applications then uses OpenLayers API to read this list and connect the KML points to display the route as a line drawn on the map. Users can also provide a building or room name for the start/end location. The corresponding coordinates of the building/room is retrieved from 2D and 3D spatial databases and passed to the web-service in this case.

The directional images web-service provides a list of images (thumbnails) along the route augmented with superimposed arrows pointing the way. The parameters for this web-service are the same as for normal routing. The web-service returns in JSON format a list of image URLs, their location, and the view angle of each image with respect to the direction (path) needed to follow. Figure 6 shows how the directional image web service result displays on the OSM basemap.

Fig. 5. The directional image web-service result overlaid on the OSM base map. Clicking on each thumbnail opens a larger image showing an arrow pointing the direction to travel.

2D and 3D visibility-based directional query web-services correspond to different search options in relation to different spatial data types in the database. These web-services are divided into two sub-groups, one applied to 2D OSM building footprints and the other applied to the 3D GE (KML) building models. 2D Isovist view, 2D Field-of-View, 2D/3D Line-of-Sight (Point-to-Select), 3D Frustum and 3D Threat Dome are the different types of 2D and 3D spatial queries available. More detail on these spatial search algorithms and the web-services developed for each can be found in [11, 12].
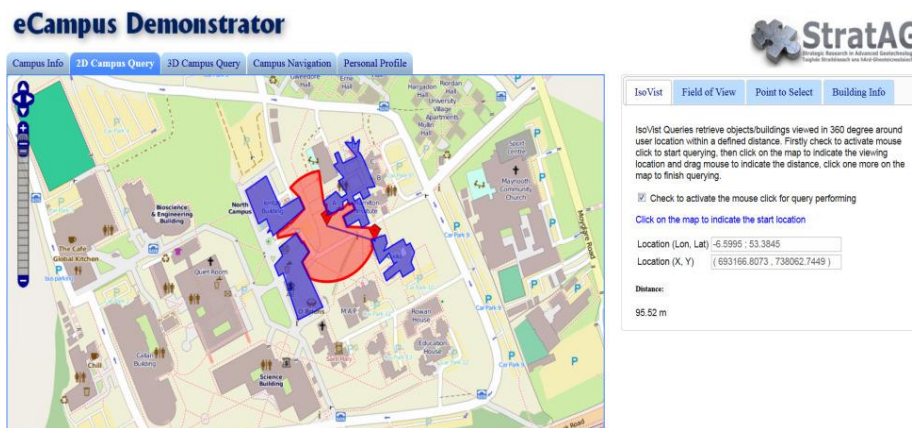

Fig. 6. The 2D Isovist web service result overlaid on the OSM basemap. Only objects that a user can actually see out to a specified distance (e.g. 100m) in 2D get returned by the query.

The business data retrieval web-services.
Useful business data attributes for student/staff users mainly relates to classroom schedules and facilities. RESTful web-services were developed to re-

trieve this information by querying an external NUIM database containing university business data associated to room numbers. The web service returns the classroom schedule in html format.

5.3 Interface layer

The interface layer consists of html pages displayed to users (client side) in a standard web browser. At this level, spatial data returned from 2D queries is visualized on 2D OSM maps as additional layers using OpenLayers API. For 3D queries, the results are returned in JSON format and drawn as placemarks added to a GE view. The integration of spatial data and business data is performed at the client side when attribute information about buildings and associated rooms/objects is found.

6   Advanced GeoService Functionality

The eCampus Demonstrator is developed for both web and wireless devices. However, there are some differences in the look and feel of the interfaces and user interactions between the desktop version and the smartphone version. This is mainly due to additional limitations of the Google Earth API for mobile devices. In addition, on mobile devices information like user location, tilt, and compass readings can be captured automatically, while users need to input them when interacting with the desktop interface. The eCampus Demonstrator GeoService functionality and its organization are depicted in Figure 7.
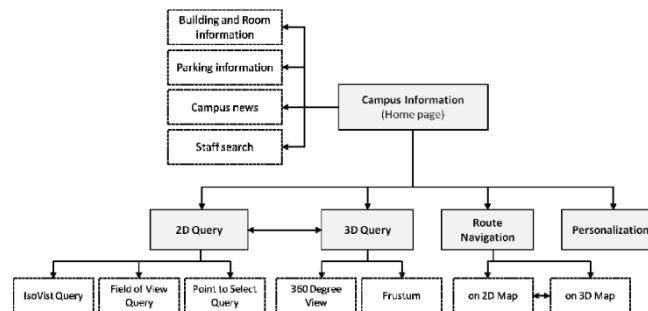

Fig. 7. eCampus search functionality. There are 5 main search functions in the eCampus Demonstrator. Each function provides a further subset of query utilities (dashed boxes).

6.1 Exploring eCampus information

This search functionality is available from the home page of the desktop application (Fig. 10). It shows a 2D OSM basemap of the campus where users

can selectively click on each building (polygon) on the map to explore more detailed information visualised on the right side of the interface. The available business data includes information on rooms within the building, faculties/departments in each building, opening hours of the building, car park information, building images and detailed architectural plans (Figs. 10, 11).
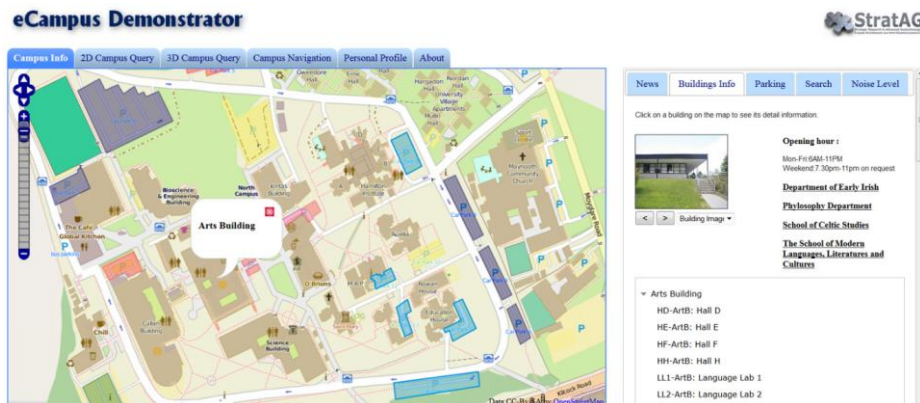

Fig. 8. Selected building information displayed on right side of desktop interface.

Personalised Information.
This functionality provides information specifically targeted to user interests. Campus users manage their interests by adding or removing them from their profile. An auto complete list of keywords is provided to help users in selecting and adding interest keywords. After logging into the system with a student/staff ID, campus information such as school/course calendar and news/events are personalized based on these chosen interests (e.g. History, Sports, Societies, Restaurants, etc.) (Fig. 12).
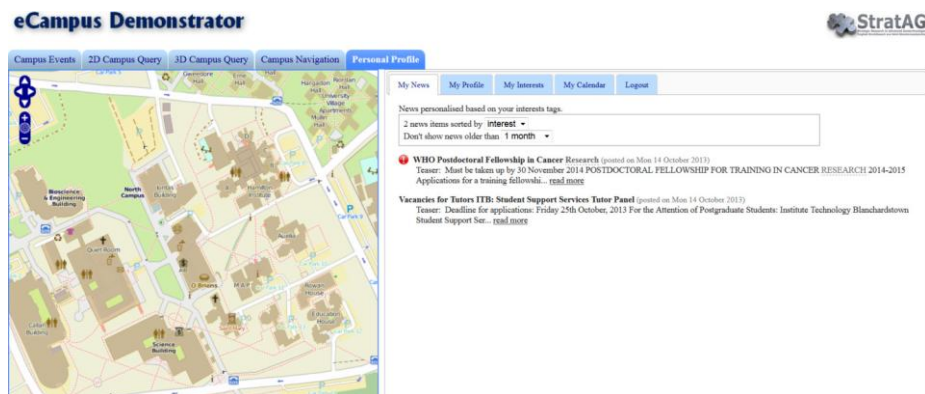

Fig. 9. Personalised news and events overlaid on OSM basemap.

6.2 2D Queries on the desktop

2D search functionality allows us to perform Isovist, Field-of-View and Point-to-Select queries. On the desktop interface, a user first chooses a query type and then clicks a location and drags the mouse to indicate the search radius directly on the OSM map. Depending on the query type, the user may optionally enter a field-of-view angle and search direction. The query result is displayed as blue shaded buildings with the field-of-view (viewshed) shaded in red (Fig. 6). As opposed to a simple range query where all objects out to a certain distance are retrieved, a 2D Isovist is a 360° line-of-sight query around a user's location that is clipped by OSM building footprints. This allows for more task specific queries such as "retrieve all objects around me that I can actually see within 50 meters".

By further constricting the 360° Isovist search, the Field-of-View query allows users to indicate a viewshed angle, direction, and distance from their chosen location. The application then retrieves all visible objects that fall within that angular Field-of-View, in that particular direction, and within that specified distance (Fig. 13).
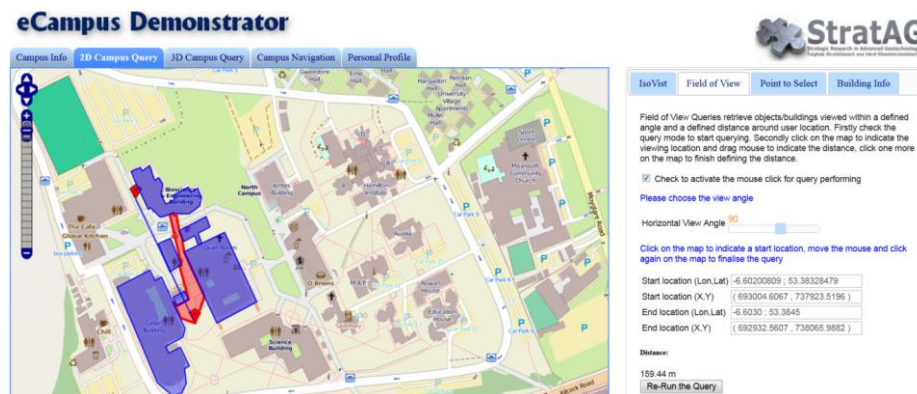

Fig. 10. Results of 2D Field-of-View query looking northwards out to 160m.

The Point-to-Select query intends to mimic a user pointing their smartphone at actual objects in the real-world, but is invoked on the desktop by clicking 2 (or more) locations on the map and retrieving all objects that intersect the line (Fig. 14).

Fig. 11. 2D Point-to-Select query with 4 points indicating the chosen "line-of-sight".

6.3 2D queries on the mobile

Mobile interfaces for exploring the campus on a smartphone are depicted in Figure 15. Building information gets retrieved by simply tapping on any building in the OSM map.
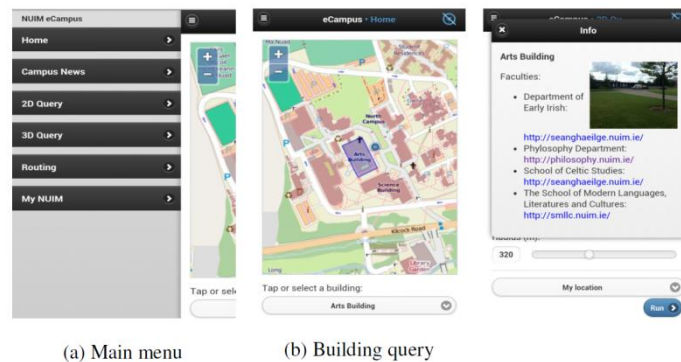

(a) Main menu   (b) Building query

Fig. 12. Mobile eCampus interface on smartphone.

The mobile interface takes into account the current user location from the device GPS, tilt of the device from accelerometer, and azimuth of the pointing direction from the digital compass. The search radius is indicated by dragging the distance slider left or right with the resulting red viewshed changing dynamically to indicate scale. The final viewshed query shape gets clipped by the OSM building footprints before being sent to the spatial database to retrieve any intersected objects (Fig. 16).
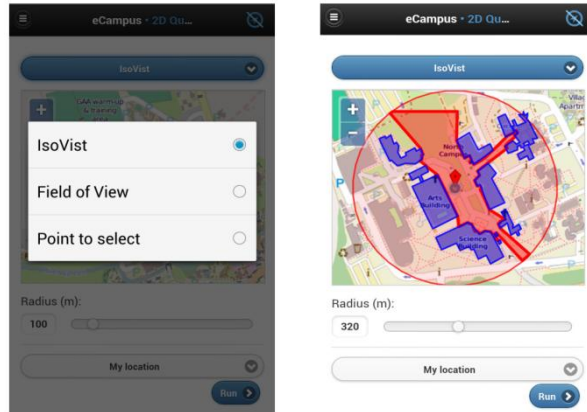
Fig. 13. Mobile 2D Isovist viewshed (in red) set to search out to 320m. Only objects that can actually be seen (in blue) from current location get returned.

Regarding the mobile version of route navigation, users can ask to find a route from their current location (based on their GPS location) to any building/room or between any two campus buildings selected from a list (Fig. 17).

(a) Pedestrian walk   (b) Routing with directional images
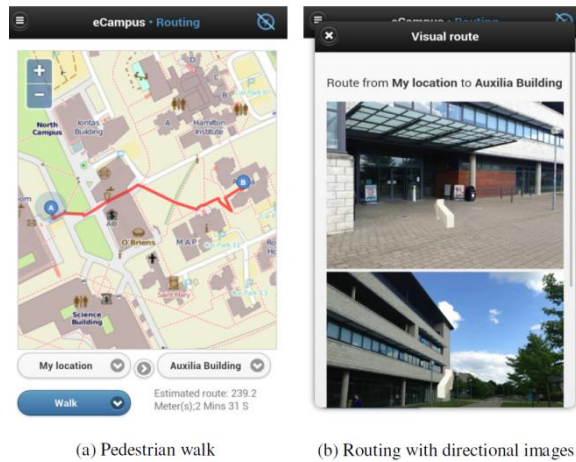
Fig. 14. Mobile route navigation with directional images between selected buildings/rooms.

6.3 3D queries on desktop and mobile

The 3D eCampus search functionality displays NUIM maps in KML format using Google Earth (3D map). At this stage, users can query the campus by clicking directly on any building or door/window (i.e. room). The corresponding building/room information gets overlaid on the 3D map (Fig. 8). The eCampus Demonstrator provides two 3D search options: 360° Isovist query

(Threat Dome) and a directional Frustum query. The 360° Isovist query is a 3D version of the 2D Isovist query, which means it, returns objects that users can actually see around them in all directions horizontal and vertical out to a specified distance (Fig. 18). The final dome shape (i.e. search space) is determined by first calculating all intersections between a sphere centred on the user's location and the 3D building models stored in the spatial database. Next, the intersection points are joined into a 3D polygon shape that is then used as the query "window" into the database to retrieve all visible objects (e.g. rooms/buildings) it contains.

The Frustum view query is the 3D version of 2D Field-of-View query. It constrains the shape of the search space to a location having vertical and horizontal angular visibility (and tilt) in a particular direction, out to a specified distance and clipped to the 3D building models stored in the spatial database. The intersection of the resulting frustum query "window" with other database objects represents what users can actually see in a given 3D direction, constrained, for example, to their actual field-of-view (Fig. 19).
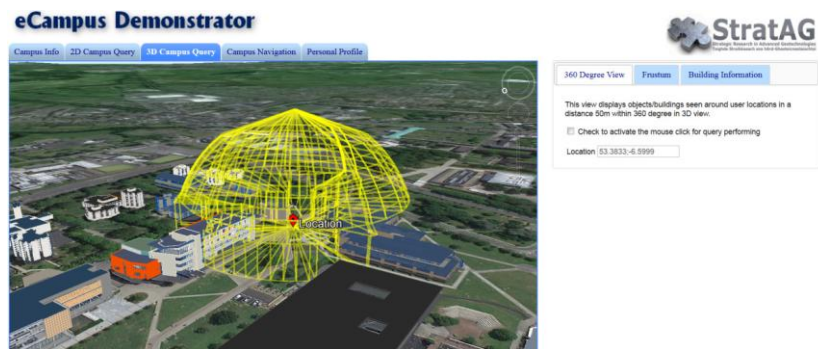

Fig. 15. 3D 360° Isovist (Threat Dome) view query. Only visible objects (e.g. buildings/rooms) that fall within the dome shape get returned by the query.
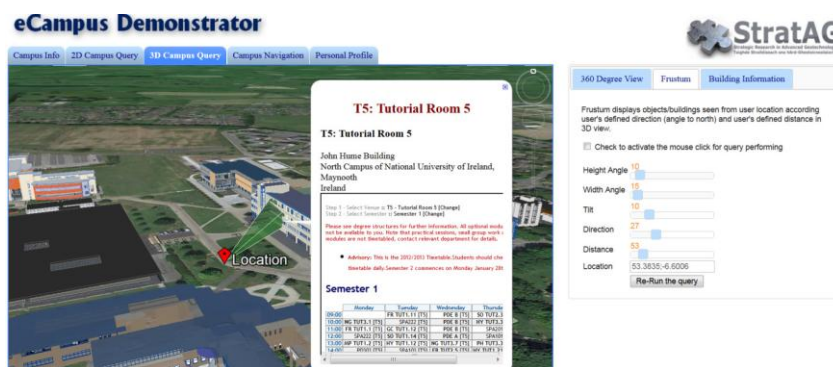

Fig. 16. 3D Frustum query. The final frustum shape is clipped by any building objects in its path before getting utilised as a 3D query window in the spatial database.

Currently, due to API limitations with mobile GE, e.g. in drawing/inserting user defined 3D objects, the mobile 360° Isovist and Frustum query interfaces look the same as in 2D. However, the query results do take into account the true 3D nature of the search space. In this case, the search results provide a list of buildings and rooms intersected by the 3D query but displayed on a 2D OSM map view (Fig. 20) (i.e., the objects intersected by the yellow Threat Dome shape in Figure 18 or by the green frustum shape in Figure 19).
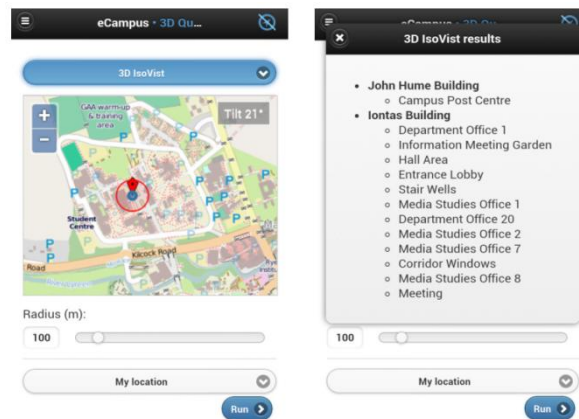

Fig. 17. 3D Isovist search and result. Note query result is shown overlaid in 2D OSM as GE API does not yet allow functionality for user-defined customized 3D objects overlaid on their mobile 3D maps.

7   Discussion

During the development phase of this application, some important limiting factors emerged regarding the technology employed. After first analyzing the capabilities of the chosen technologies for implementation, the following discussion presents some advantages and disadvantages of our approach.

3D modelling and visualisation

When creating 3D virtual cities for such Personal GeoService applications and general mobile spatial interaction in the Internet of Things, the most important task is to generate detailed and geometrically accurate building models. In contrast to traditional methods (e.g. on-site surveying), terrestrial laser scanning is an attractive alternative for collecting building coordinate data in terms of field time and accuracy [23–25]. However, the process of building detailed 3D models from point cloud data is still quite a manual process. In our application, it takes around 6–8 hours for each building. This implies automatic or semiautomatic processes must be developed to reconstruct building models

with LoD3 to reduce post-processing bottlenecks for city-wide 3D modelling workflows used by this type of application.

As the GE platform currently only displays 3D objects above the earth's surface, this restricts visualisation of any underground detail below ground elevation level. To display and interact with entire building models above and below ground, an alternate web mapping platform should be considered. Also, mobile device browsers do not currently (2013) support the Google Earth API, therefore there are limitations when visualising 3D models and other customised vector objects on smartphones.

Integration level of spatial data and business data

As mentioned previously, the integration of spatial data and business data is performed at the client side, i.e. at the visualisation level. We considered three options to provide spatial data and related business data to users as shown in Figure 21.
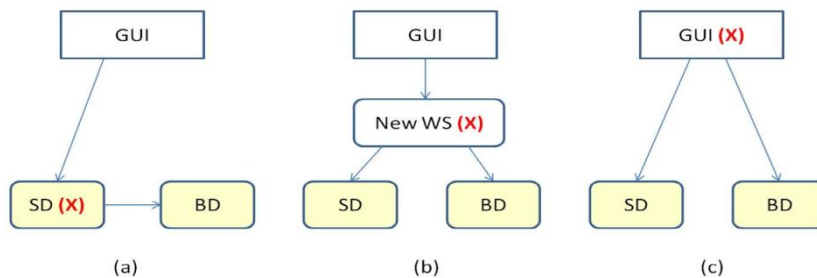

Fig. 18. Different spatial data and business data integration approaches. The red (**X**) represents the integration point. The arrow describes the calling direction. WS: Web-Service; SD: spatial data WS; BD: business data WS; GUI: Graphical User Interface. (a) Early-integration: In this approach, retrieving business data is performed from inside the spatial data retrieving WS. The final results sent back to GUI include spatial data and business data; (b) Aggregated web-service: A new web-service is developed to compose the results returned by the spatial data web-service and the business data web- service; (c) Integration at the visualisation level: The results of the spatial data web-service and business data web-service are overlaid at the visualisation level. We have chosen the third approach: integration at the visualisation level, as this approach provides some advantages compared to the other two approaches (Table 1).

OGC services

The Open Geospatial Consortium (OGC) has developed some standards for geospatial processing technologies to enable applications from different commercial vendors to interoperate. However, the locationing services developed within OGC focus mainly on tracking and location-based applications for mobile devices [8]. These services are far from what we require in this

application, where all RESTful web-services for location-dependent directional and visibility space querying have been developed in our own algorithms.

<p align="center">**Table 1.** Spatial data and business data integration options.</p>

(a) Earlyintegration	There is a dependency of SD retrieving WS and a specific BD retrieving WS. That means SD WS cannot be reused for other purposes
(b) Aggregated web-service	SD and BD WS are independent. It depends on the user needs that the aggregated WS (AggWS) will integrate suitably SD and BD WS. Suppose $fA(WS)$ is the cost of analyzing the result of a WS, $fC(WS)$ is the cost of calling a web-service, then the cost of this approach to display the result is: $$fC(AggWS) + fA(AggWS) + fC(SD) + fA(SD) + fC(BD)$$ Note that the results of AggWS needs to be analyzed to draw the geometry shapes and the business data is then added to the feature data of the geometry object. In case the results of AggWS is in KML format, there is no need of analysing AggWS (so no cost), all spatial data and business data can be visualized. However, in that case the visualisation is fixed according to the API provided.
(c) Integration at visualization level	SD and BD WS are independent. It depends on user needs that suitable SD and BD WS are consumed at the visualisation (client side code source). The cost to display the result to the users is: $$fC(SD) + fA(SD) + fC(BD)$$ Visualization of the results is flexible according to the users' needs.

ROA instead of SOA

For the last decade, Service Oriented Architectures (SOA) has been widely used for distributed applications, particularly on the Web. In Geographic Information Systems (GIS), there is no exception here. For instance in [2], a GIS web-service architecture was proposed based on SOA technology. However, while SOA is a proven approach, in some cases it can be overly complicated and processor heavy. For example, when handling a SOAP message, the client (desktop or mobile) needs to send a request with parameters constructed and wrapped in XML format with special headers and other elements. It also has to parse any response from the server in the same effusive XML format [4].

In the case where the client is a mobile device, this approach contains far too much processing overhead in terms of the volume of data, most of it quite unnecessary, that must be sent/received on mobile devices having relatively limited wireless connection speeds and often a data transmission cost [10]. In this respect, JSON is a much lighter data format in terms of processing and

transmitting wirelessly. Furthermore, we agree with the general statement that the REST architecture provides a "scalable and simple deployment of webservices and particularly appealing for Earth and Space Science" [6] as RESTful web-services have been much used in geo-information sharing.

Dependency of 3D query performance and 3D data details.

In our application, users carry out spatial queries from outside of buildings. Therefore only the geometries of exterior structural components of the building (e.g. facades, roofs, windows, doors, balcony, canopy, etc.) associated with room level attribution (e.g. room name and function) were loaded into the database. In this way, it helps to reduce the complexity of the 3D models and thus improve 3D spatial query performance.

8   Conclusions

Providing users with business context data in location dependent queries helps to fulfil more task specific user needs within detailed data environments. In order to meet that objective, there is a need for 3D building modelling to at least LoD3. The workflow employed in this paper was successful in reconstructing geometrically accurate building models with LoD3 detail. However, the procedure is time consuming for larger project areas where numerous building models need to be reconstructed; therefore, automation of this approach is still an open problem. The flexibility, interoperability and heterogeneity of this kind of GeoService application demand suitable software architecture. In particular to this geospatial application, a Resource Oriented Architecture (ROA) was chosen for the implementation.

At time of writing, the eCampus Demonstrator presented in this paper is among the first GeoService applications to explore an area in detail on both desktop and mobile 2D and 3D maps. It provides users with more personalized search utilities, like directional/visibility query functionality, than contemporary eCampus information systems currently allow. However, there are still some significant 3D data processing limitations that need solving by GE if their mapping platform is to be widely adopted for similar detailed data geo-application development in future. In the meantime, alternative platforms to consider could be 3D games engines such as Unity[5]. We also need more testing with student/staff users to get their feedback on overall functionality as well as the general performance of our application. Semantic Web technologies might also be employed to facilitate the integration of heterogeneous data [26]. In summary, this detailed data eCampus implementation can be

---

[5]   http://unity3d.com/unity

considered as a starting point for developers and researchers when developing for similar application domains, such as business parks, hospitals, airports, and shopping centres.

Acknowledgements

9   References

1. Guinard D, Trifa V, Wilde E (2010) A Resource Oriented Architecture for the Web of things, IEEE International Conference on Internet of Things, Tokyo, Dec, 2010.
2. Alameh N (2003) Chaining Geographic Information Web-services, IEEE Internet Computing, September October, 2003.
3. Lucchi R, Millot M, Elfers C (2008) Resource Oriented Architecture and REST, JRC Scientific and Technical Report.
4. Snell J, Tidwell D, Kulchenko P (2001) Programming Web-services with SOAP, O'Reilly Publisher, December, 2001.
5. Fielding RT (2000) Architectural Styles and the Design of Network-based Software Architectures, PhD dissertation, University of California, Irvine, 2000.
6. Mazzetti P, Nativi S, Caron J (2009) RESTful Implementation of geospatial services for Earth and Space Science applications, International Journal of Digital Earth, vol 2, Supplement 1, pp 40–61.
7. Kurtagic H, Birch J, Zeiss G (2009) An Open Architecture for RESTful Geospatial Web-services, FOSS4G Sydney.
8. *OGC* Report- Summary of OGC Web-services, Phase 4, Interoperability Testbed (2007)
9. Kaharaman I, Karas IR, Rahman AA (2011) Developing web-based 3D Campus Information System, ISG & ISPRS.
10. Yin J and Carswell JD (2011) Touch2Query enabled mobile devices: a case study using OpenStreetMap and iPhone. In Web & Wireless Geographical Information System (W2GIS 2011), Springer, Kyoto, Japan, pp 203-218.
11. Carswell JD (2010) 3DQ: Threat Dome Visibility Querying on Mobile Devices. GIM International, vol 24 (8).
12. Carswell J, Gardiner K, Yin J (2010) Mobile Visibility Querying for LBS. Transactions in GIS, vol 14 (6), pp 791–809.
13. Jacob R, Zheng J, Ciepłuch B, Mooney P and Winstanley AC (2009) Campus Guidance System for International Conferences Based on OpenStreetMap. In: Proceeding W2GIS '09, Maynooth, Ireland, Springer-Verlag, pp 187-198.
14. Kent State University campus maps, available at : http://www.kent.edu/campuses/maps/-map.cfm (accessed on May 2014)
15. University of California, Berkeley Interactive Map, available at: http://www.-berkeley.edu/map/3dmap/3dmap.shtml (accessed on May 2014)
16. University College Dublin Mobile services, available at: http://www.ucd.ie/itservices/-itsupport/mobileservices (accessed on May 2014)

17. Truong-Hong L, Pham Thi TT, Yin J, Carswell J (2013) Detailed 3D Building Models for Google Earth Integration, ICCSA 2013, Springer-Verlag, Ho Chi Minh city.
18. Pham Thi TT, Truong-Hong L, Yin J, Carswell J (2013) Exploring Spatial Business Data: A ROA Based eCampus Application, W2GIS 2013, SpringerıVerlag, LNCS 7820, pp 164-179.
19. Haala N, Kada M (2010) An Update on Automatic 3D Building Reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing 65, pp 570-580.
20. Leica Geosystems AG: Leica Cyclone-3D Point Cloud Processing Software (2011)
21. Leica Geosystems AG: Leica CloudWorx for AutoCAD (2011)
22. Safe software: FME desktop (2013)
23. Truong-Hong L (2011) Automatic Generation of Solid Models of Building Façades from LiDAR Data for Computational Modelling. School of Architecture, Landscape and Civil Engineering, vol PhD. University College Dublin.
24. Truong-Hong L, Laefer DF, Hinks T, Carr H (2012) Flying Voxel Method with Delaunay Triangulation Criterion for Façade/Feature Detection for Computation. ASCE Journal of Computing in Civil Engineering 26, pp 691–707.
25. Truong-Hong L, Laefer DF, Hinks T, Carr H (2012) Combining an Angle Criterion with Voxelization and the Flying Voxel Method in Reconstructing Building Models from LiDAR Data. Computer-Aided Civil and Infrastructure Engineering DOI: 10.1111/j, pp 1467-8667.2012.00761.x.
26. Ballatore A, Wilson DC & Bertolotto M (2013) Survey of Volunteered Open Geo-Knowledge Bases in the Semantic Web; Quality Issues in the Management of Web Information, ISRL 50, pp 93–120, Springer.
27. Gröger G, Kolbe TH, Nagel C, and Hafele KH (2014) OpenGIS City Geography Markup Language (CityGML) Encoding Standard (OGC 12-019). Version 2.0.0. OGC 12-019. Open Geospatial Consortium. URL: http://www.opengeospatial.org/standards/citygml. (accessed on May 2014)
28. Schöning J, Krüger A, Cheverst K, Rohs M, Löchtefeld M and Taher F (2009) PhotoMap: using spontaneously taken images of public maps for pedestrian navigation tasks on mobile devices. ACM, Bonn, Germany.
29. Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal, 69, pp 211–221.
30. YSU 3D campus map. Available at: http://www.ysu.edu/campusmap/ (accessed on May 2014)
31. Haklay M and Weber P (2008) OpenStreetMap – User Generated Street Map, IEEE Pervasive Computing. October-December 2008, pp 12–18.