A PRACTICALFRAMEWORK FOR A COLLABORATIVELY DEVELOPED, MULTI-PURPOSE 3D MODEL OF THE METROPOLIS

Paul Cote, Harvard University Graduate School of Design
Published in Computers in Urban Planning and Urban Management

University College London May 2005

Many hands and minds are occupied with making three-dimensional representations of places in the city. As internet giants like Google and Microsoft are making urban three dimensional models available for visualizing and referencing information (Musgrove, 2005; Perez, 2005,) we expect that access to 3d urban representations of cities will become common to a broad base of citizens for a diverse set of purposes: from choosing theatre seats, or referencing family snapshots, to protecting public safety. This paper introduces a framework for coordinating and collecting multi-purpose 3D models of buildings and patches of terrain. The framework facilitates the work of independently motivated modelers, and may result in a collective repository of models which can be drawn upon to compose views and models appropriate to the varied needs of a multitude of users.

Our novel approach to organizing an using urban three dimensional models results from a recent convergence of 3D modeling technology with relational database management tools that allows for the coordination, sharing, storage and versatile retrieval of unprecedented quantities of model elements, many of which may be alternate representations of the same place. Our development of a prototype for the Boston Metropolitan area is also bolstered by the recent emergence of a broad scale, publicly available, LIDAR-based broad model of the city that serves as an initial base model for any place in the city (MassGIS 2005). These recent developments are brought together in a simple relational data model that is portable and scaleable.

This paper describes the design of the relational schema at the core of the system that yields specific visualizations and editable models of the city tailored to user needs. We describe how the system provides an armature and incentive for individual modelers across a metropolitan region to coordinate their efforts toward a continually updated and enhanced multi-purpose 3D model. We conclude by discussing how this model management framework will yield new ways of understanding, managing and designing the city.

1 Model Management Schema

The model management system is a repository of model elements (representations of buildings, other fixtures and ground conditions) stored in a simple relational database schema (Codd, 1971). This schema is part of a geographic information system (GIS) (ESRI 1994) which also houses a raster based terrain model and an orthorectified color aerial photograph, also referenced to the same coordinate system. This schema may exist on a large centrally managed collection and distribution system while subsets of the model may be managed as independent repositories located in individual firms and agencies or as editable working models running on the desktops of individual model makers.

At the core of the system is a wholesale model of the region comprised of the elevation model with a cell-size of 5 meters, and an orthophoto of half-meter resolution. Together these raster elements represent the ground condition. Buildings are represented with a table of roofprint features. Each row in the roofprints is constituted of a polygon feature with three attributes for each feature: Unique ID, Ground Elevation and Roof Height. The roofprint elements were derived from a region-wide LIDAR survey. The elevation model and the orthophotograph were collected from a photogrammetric survey. Both of these surveys were conducted by the Massachusetts GIS (MassGIS 2005).

Two-dimensional information terrain and building information in a GIS with elevation and height attributes provide an easy means of visualizing the three dimensional shapes of buildings. Extruded representations of buildings and photographs draped on a raster terrain model aren't capable of representing many aspects of real three dimensional geometry (Hoinkes 1995), yet they are useful for visualizing the form of the city as shown in Figure 1.

Thanks to a recent collaboration between software vendors (ESRI 1994; @LastSoftware 2005) subsets of this basic model may be extracted and reformatted for editing in a 3D modeling package as the basis and context for creating more detailed models of buildings and ground conditions. Elements created in the 3D editing tool may be encapsulated as multipatch (Armit 1971; ESRI 1998) elements that also carry color or image texture information. (Figure 2). These elements may then be inserted into a Multipatch table along with attributes that describe the aspects of the model elements.

The attributes of elements in the Multipatch table are given in Table 1. These attributes are used by the system to select the appropriate detailed models to render an appropriate scene based on user specifications such as: design scenario, historical time period, model purpose, and provenance. These attributes allow for systematic retrieval of model elements from the repository based on level of detail, so that more detailed models are selected near a specified point of view or animation path. Each row in the Multipatch table also contains a binary object containing an editable model of the model element.

The model retrieval process begins with a selection of the most appropriate elements from the Multipatch table, and then completes the urban scene with rough models from the Roofrints table. Selection of the appropriate non-overlapping roofprint elements is accomplished through the Substitutions table, which stores the correspondence of roofprints and multipatch building elements representing the same real world entities. This table simply records the one-to-many relationship between each unique multipatch ID, with the IDs for overlapping roofprints. Figure 3 illustrates a view composed of non-overlapping elements retrieved from the Multipatch and Roofprints table.

2 Scene Retrieval Queries

The model management schema contains elements, many of which may be representations the same physical entity created for distinct purposes, by distinct authorities or for viewing at distinct levels of detail. Selection of appropriate model elements to compose a user-specified scene is accomplished with two relatively simple SQL Create View queries. The first query (Figure 4) creates the Multipatch View, which specifies which elements should be retrieved from the Multipatch table according to the user's specification of the View Date, and the appropriate level of detail, considering each entity's position relative to the scene's Viewpoint (or Viewpath if a scene is being selected for a moving-camera animation.) Our query examples use syntax for spatial SQL operators provided by Egenhofer (1994).

The second query (Figure 5) specifies which roofprint elements are required to fill in for buildings not represented by elements selected in the Multipatch View. This query is completely derived from the Multipatch view and the substitutions table; selecting those roofprints with Unique IDs that are not in a selection of rows from the substitutions table having Multipatch IDs corresponding with those selected in the Multipatch View. Figures 6 and 7 illustrate the result of two scene retrieval queries, the first having a Viewpoint in the plaza itself, the second with the viewpoint more distant from City Hall.

3 Implementation: Distribution and Collection

The database structure we have created is relatively simple; a quality that makes it amenable to a distributed framework for model sharing and collection. This is what largely sets this framework apart from other extant frameworks for building broad-scale urban models. None of the broad-scale efforts to build urban models described in recent reviews (Batty et al, 2000; Shiode, 2001) emphasize a distributed framework for model collection or concerns for long-range maintenance. Yet in a metropolitan area like Boston, which is a patchwork of independent towns and university campuses the best possible model would pull together the best official information from all of these sources. We see the development of web-based tools for delivering urban 3d models (Musgrove, 2005; Perez, 2005) as the killer application that will cause a wide constituency to expect up-to-date models to be assembled from authoritative sources. The fact that our system is based on a simple framework and tools that are accessible and already in widespread use makes us hopeful that our implementation may foster an unprecedented distributed approach to the collection of models that would yield for one user the best looking view of the city, or for another, the most authoritative, and for yet another, a view of the city as it was at a certain point in history. The key to making this work is not so much a technical problem but requires consideration of the social aspects of model sharing and collection.

Our prototype model management schema is implemented inside an Oracle 10g Database using ESRI's Spatial Data Engine (SDE) middleware (ESRI 2004:2). The scene retrieval queries return views into a GIS Viewer, ArcGlobe (ESRI 2004), which is capable of rendering very broad-scale scenes with multipatch models and extruded roofprints registered on a raster terrain model draped with an aerial photograph. The same views may also be rendered in ArcMap, a 2D Viewer that has the capability of extracting the terrain, aerial photograph and building elements into a model that can be edited in a 3D modelling program, Sketchup, (@lastSoftware 2005:2). Sketchup is a 3D modelling tool that facilitates the creation of digital 3D models including texture-mapped surfaces. Sketchup can also import 3d models saved in the most common 3D export formats, Drawing exchange Format (dxf) and 3DStudio (3ds). Using the geometry exported from the GIS, the SketchUp user can alter the representations exported from the GIS, create new elements, or place elements imported from other packages into the correct geographic position and scale. The Sketchup/ArcGIS interoperability extension allows the user to select the true 3D geometry from Sketchup and upload it to the Multipatch Table in a personal geodatabase. The upload process presents a form for filling out the proper attributes, and provides an option to also upload an encapsulated Sketchup format file, which will be available for editing on future downloads from the GIS. Figure 8 provides a diagram of the model extraction and ingest workflow.

Though we rely on specific commercial software products in our implementation, we hasten to point out that there is nothing about this architecture that is proprietary. The Multipatch format is an open standard (ESRI 1998), and there are other promising open formats for model encapsulation, CityGML and Interoperable Function Classes (Kolbe 2004). There are many relational database tools capable of dealing with geographically referenced geometric objects, including a completely open source tool, PostGIS (2005). Since our model schema already maintains versions of the same element in different formats – Multipatch for visualization, and a SketchUp object for editing, it is easy to imagine maintaining these other formats as well, including the KZM format used by Google Earth, so that the schema can support a wide variety of viewing technology.

Our choice of commercial tools fills a need for expediency in our development effort, and also lends practicality to the effort to build a network of contributors. The ESRI/@LastSoftware interoperability tool offers an unprecedented integration of a 3D editing tool with a GIS. Another advantage is the scalability of the system in two directions. The SDE architecture used for maintaining the full model repository accessed through desktop systems, easily lends itself to extracting subsets of the model as desktop oriented Personal Geodatabases with no requirement for changes to the schema. The fact that ArcGIS and SketchUp are already in common use in public agencies, private firms and universities makes them a good fit for a distributed architecture for model sharing and collection.

It is useful to consider the various modes of model sharing and collection in order to anticipate the ways that a metropolitan model might develop. We consider four modes of model sharing: sharing as acquisition of models for individual use, collaboration among specific individuals firms or agencies, and contribution of work to a publicly accessible repository of models.

3.1 Sharing for Model Acquisition

Our early implementation of the system within the Harvard Graduate School of Design (GSD) has shown us that the unprecedented ease of transferring urban 3D data from GIS to a 3D modelling package provides a compelling incentive to use the system as a source of base model data. Our SDE-based server architecture facilitates this sort of one-way sharing by allowing users with the proper privileges and a reasonably fast internet connection to create subsets of the model in Geodatabase or SketchUp form. We expect that this mode of sharing would work equally well for users outside of the GSD wou will be equally motivated to acquire useful base models for their project-oriented work.

3.2 Sharing for Collaboration

Many projects involve the coordination between individuals, firms and public agencies. Individual designers working on alternatives on the same site or neighboring sites, and the reviewers of these efforts benefit from being able to share models and to view their proposals in a common context. Agencies and administrators that maintain models for specific territories often share models with neighboring authorities to compile the best or most authoritative model of their context. Our model system facilitates this sort of sharing by providing a common geographic referencing system and normalized format, even if the individuals are sharing private versions without regard for a centrally accessible resource.

3.3 Sharing to Contribute to a Multipurpose Model Repository

The ultimate potential of our system is in the integration of a large number of models from many sources. Our initial implementation was begun by assimilating many models made available to us by the Boston Redevelopment Authority, which like many public agencies in Massachusetts is expected to share their information with the public. This process has involved a somewhat labor intensive selective import of components of their large CAD models into the Multipatch table through SketchUp. This model of a central model builder doing the work of assimilation has its limitations when scaled up over the multitude of administrative entities maintaining the official models of their territories. Our aspiration is to build a federated system in which the official public models of specific administrative territories are maintained in local instances of our schema and assimilated with a central version either automatically through the use of materialized views – in which the central multipatch table is a composite of several locally maintained instances that is compiled automatically by the database, or by periodic transfer of the data through the mail.

The prospect of changing the internal workflows in public agencies is not a task to be treated lightly. Nevertheless, we believe that the advantages of a model management and collaboration for an agency's internal use may provide sufficient motivation for the adoption of our framework within particular agencies. The ability to share official versions of local 3D models may result as a side-effect of this independent adoption.

Our SDE-Based infrastructure that hosts our schema is well adapted for this sort of sharing

While we have observed that users are self motivated in using a repository of model resources and that the use of this system facilitates and promotes the sharing of coordinated models between users, we should turn our attention to the motivation and mechanisms for contributing models to a centrally available resource.

3.3.1 Element Creation and Enhancement Process

When the Scenario-Making workflow is used to create a scene intended for the purpose of adding or enhancing 3D elements, the Roofprint features are exported as extruded 3D geometric objects, and the selected CAD Data features are exported from the Multipatch table. Incidentally, the appropriate patch of terrain and aerial photo is also clipped and sent to the CAD package as a texture-mapped triangulated mesh.

The model elements and terrain model arrive in the CAD system with their geographic referencing system intact. In this context, the user may alter or replace existing elements or create new ones. When the elements are finished, they may be uploaded as multipatch and CAD Data features back to the database. Upon upload, the user is required to enter those attributes for of the element that are user-specified. Uploading model elements never overwrites existing elements in the database. When an element is uploaded, the substitution table should be automatically updated to add entries for any entities in the Roofprints table that overlap with the new element (except in the case of elements of type Ground.)

In most cases, elements uploaded to the system will appear in the model the next time an appropriate scenario is generated.

3.3.2 Model Management Process

A main design goal for this system is to make all of the model enhancements made by users immediately accessible the next time the scenario retrieval process is invoked. However, there are several situations when the attributes of existing elements must be adjusted in order to make certain query selections logically consistent. If new of date dependent elements are added, the min and max dates will have to be adjusted for newer or older buildings that may already exist in the database. In cases where an element has been added that has attributes identical with elements already in the database, someone with permission to change the existing elements will be able to assign priorities to the Alternative attribute. The idea of ranking elements that are otherwise congruent in other respects would permit queries that allow the construction of scenes composed of the Newest or the Most Official features. The ranking of otherwise congruent features will allow obsolete elements in the database to become deprecated without being deleted.

4 Social and Institutional Considerations

The metropolitan model infrastructure is more than a computer system. The system demonstrates an emergent social dynamic as the collection grows and is enhanced by the independent activities of its users. In order to make the most this effect, we need to understand why different classes of users might be motivated to coordinate and contribute their work. It is likely that some users would be quite happy to download snapshots of pieces of the model and to enhance these and simply keep working with these downloads as static, independent CAD models. In this case, we have at least tricked the user into coordinating his work with the georeferenced metropolitan model framework, and that user could easily share his work with others who have used the same framework. This alone is a big improvement on traditional project-based modelling culture.

There are ways that we could discourage the use of parallel snapshots downloaded from the system. By limiting the number of detailed CAD data elements in the Element Creation/Enhancement process to one, or limiting the spatial scope of model downloads, a user could get just what is necessary for enhancing or creating a single element, but in order to visualize that erlement in a larger context, or to take advantage of future enhancements in the model repository, the easiest way to do this will be to upload his new model elements to the common infrastructure.

We have reason to believe that a large number of detailed models may be added to the system by students who will be compelled to model buildings for school assignments. Furthermore, the official models created by city agencies, such as the Boston Redevelopment Authority, or universities like Harvard and MIT may be added to the system as a means of improving accessibility to public resources, and to reduce the cost of sharing information with the public and the development community.

There is a high likelihood that important contributors to the model infrastructure will have model elements that they don’t want to share with users outside of their firm or agency. One way to address this concern is with a federated architecture, having private Multipatch tables that exist in different locations, which are accessed only by Scenario Retrieval processes launched within that agency. In this case local administrators could deal with the Model Management processes in their own locations. A new flag could be added to these local Multipatch tables designating whether a particular element is public or for in-house use only. Alternatively the distributed system might synthesize its multipatch table as a materialzed view of resources that are distributed over a multitude of remote servers.

Abstract:

This paper introduces a practical framework for building and maintaining detailed, multi-purpose three-dimensional representations of entire metropolitan areas. We describe the convergence of technologies that make such an infrastructure technically feasible now for the first time. We specify a relational schema for storing true 3D models of buildings and ground conditions. The system is capable of retrieving the appropriate elements to compose logically consistent models of the city based on specific historic, contemporary or future design scenarios, with the appropriate levels of detail for a specific point of view or flight path. We show how this schema can for the basis of a distributed system that will serve as an armature and incentive for individual modelers across a metropolitan region to coordinate their efforts toward a continually updated and enhanced 3D model. We conclude by discussing some of the concerns and next steps in developing the technical and social aspects of this framework.

5 Introduction

There are many hands and minds occupied in representing the city. Can we make an infrastructure that facilitates re-use and improvement of a collective three-dimensional urban model? The answer to this question today is different than it was two years ago. Two innovations in the past couple of years make such an infrastructure possible and practical. First, a new data collection technology (LIDAR) has led to the wholesale capture of rough, metropolitan-sized urban models. Second, new technology for encapsulating 3D models makes it practical to share and coordinate large collections of models in a spatially referenced database. This paper discusses the design of a system for building and maintaining a three dimensional model of the metropolitan region by pooling the efforts of many hands.

A three-dimensional representation of the city can be seen as an infrastructure project. A spatial information infrastructure provides a framework upon which a multitude of other information can be referenced (CGER 1993). Many of the things and events in our lives, from GPS-equipped children and cell phones, and even gunshots can be tracked to three dimensional coordinates (Morville 2005), yet most of our representations of places are only two-dimensional (Batty, 2000.) With no three dimensional context that lets us associate a 3D point with a recognizable place, then there is a lot we don't know about where the event actually happened or what other things and events it may be associated with. The need to develop three-dimensional contextual referencing systems for densely populated areas is growing, and we now see a utility beginning to fill this need. Google, now has a 3D viewer that will allow anyone on the internet to intimately inspect a 3D models of selected urban areas and to tag locations. Soon their search engine will soon have an option of indexing and retrieving content based on spatial associations (Perez 2005.) Other internet giants, Microsoft and Yahoo are racing to build their own 3D models (Vise, 2005.) These representations of the city will serve as a referencing framework for knowledge and important means of using, planning and protecting places.

How should detailed model of the city be built and maintained? Though a rough model can be collected wholesale from the air, the detail of these models leaves a lot to be desired, and knowledge, such as the names of buildings or the uses of various parts of a site cannot be detected by aerial scanners. Local jurisdictions, universities, design firms and design students require and routinely create much more detailed models of fragments of the city for various projects. There are various technical and social reasons that these independent model fragments have not been coordinated, but recent technical innovations in data collection, such as airborne and ground based LIDAR are yielding wholesale models of cities which can serve as a framework for facilitating and coordinating more detailed modeling projects. And new data formats for encapsulating and sharing true 3d models are nabliing for the first time to collection and systematic management of collections of 3D models.

We need a way for modelers to pool their efforts: sharing and improving a common representation of the metropolitan area. If Google is going to reference the location in a town library, wouldn't it be in everybody's interest to use the town's official 3d model of that library? It is already possible for individuals to post 3D content on Google Earth, how do we organize a seamless patchwork of the most authoritative, or best-looking models of the places in the city?

Cities change and the urban model should represent more than a single snapshot in time. A good repository of urban models should be able to hold representations of various versions of the same place and to render these according to user preferences. A versatile model should help us to compare urban design projects that aren't yet built, to see the best quality representation of a building when we are close, and a less detailed one when we are far away or on a slow internet connection.) A reference to Bobby Orr's debut performance with the Boston Bruins in 1966 should be referenced to a model of the old Boston Garden, not the Fleet Center.

Our research regards new means of organizing three-dimensional content in a multi-user database. The system facilitates the model-making process for users over a wide area, and at the same time coordinates and gathers together their work, creating a versatile, sustainable model of the city. It is capable of rendering different scenes based on the historic epoch referenced, or to compare unbuilt design scenarios, and handles viewpoint-specific level of detail concerns.

To understand why such a framework has recently become practical, it will be helpful to explore key aspects and issues of the various architectures for creating and managing spatial data, particularly CAD, Relational Database Management Systems and GIS. This review will show how the recent advent several technologies: semi-detailed wholesale metropolitan models based on LIDAR surveys; and tools for encapsulating true 3D CAD models as multipatch objects that may be stored as features within an enterprise-scale GIS, have created an unprecedented capacity to create detailed three dimensional models that are scaleable in detail and scope. This background will allow us to illustrate how our model framework, with its core relational model, serves as a resource for model making, and a versatile repository of models that can be used for visualization and analysis, or further model-building.

6 Review of Modelling Technologies

This review of the primary technologies for building and maintaining metropolitan scale models will permit us to introduce some terms and issues that will be essential for understanding the proposed model storage framework.

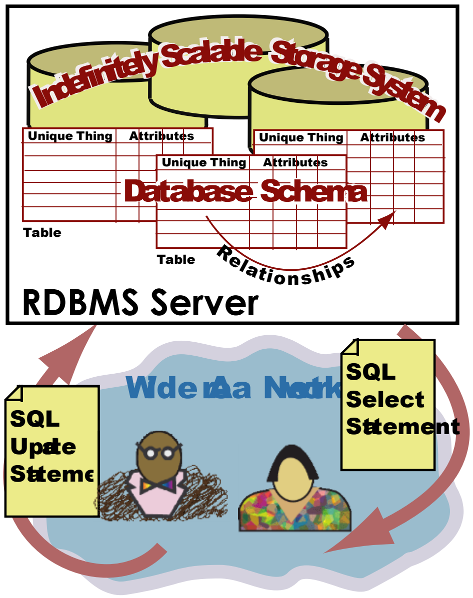
6.1 The Problem of Scalability in 3D Data Models

The chief obstacle to the development of true 3D representations of metropolitan areas lies in the technical problems of scalability inherent in the file-based architectures commonly used for storing 3D models. Though parts of cities are modeled in 3D for specific purposes, these efforts are difficult to assemble in a common framework because the difficulty in managing lots of model elements stored in classic CAD formats, which store geometric elements and their textures as separate files in the file system. The difficulty of managing and sharing CAD models rises in direct proportion to the number of elements involved and their level of detail. The number of formats for storing CAD data is also a problem, especially considering that they change over time, which poses a problem for modeling projects that are intended to be sustained over long periods. Furthermore, popular 3d rendering packages do not deal well with standard geodetic coordinate systems, and project-specific coordinate systems do not lend themselves to simple co-registration. As a result there is much effort spent in duplicating effort by modelers working on independent projects. All of these problems make it difficult to build a large shared model using the data formats designed for desktop-based creation and editing of 3d models.

6.2 Scalable Relational Data Models:

The problem of very large databases has been with us since the beginnings of automated computing. Edgar F. Codd's Relational Model of Data for Large Shared Data Banks provides a formal theory and a technology, based on the mathematical theory of sets, wherein distinct entities are represented as collections of attributes stored as rows in tables (Codd 1970.) Tables are abstractions that are managed by a Relational Database Management System (RDBMS), which handles data storage and access in a systematic way that is transparent to the user regardless of the numbers or even the physical locations of physical storage devices. Systematic relationships among different classes of entities (e.g. buildings and land parcels) can be represented in an organization of

Figure 2



Relational Database Management System

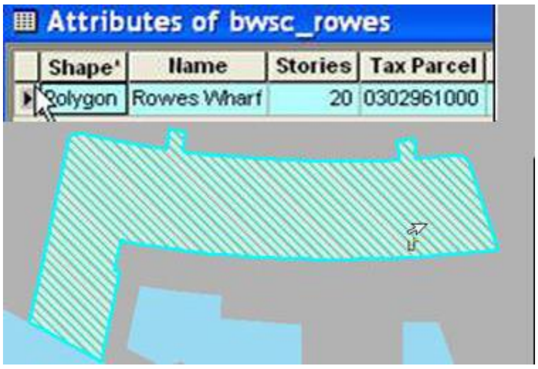
tables known as a *Schema*. In a well designed RDBMS, any information about entities, or sets of entities defined by their relationships, can be extracted using a standard logical toolkit known as SQL (Structured Query Language.) A schema and a set of SQL queries designed for a particular purpose is known as a *data model*. Relational databases are an important facility in the development of data management systems that must scale and have many users and require controlled access to resources across a wide area network. These databases have become a mainstay of all modern large-scale database applications.

In the original conception of RDBMS, the attributes or features that distinguish entities are presumed to be one-dimensional: (numeric) or categorical (character-based).

6.3 Vector GIS – Two Dimensional Features in RDBMS

Shortly after the relational model came into use, various simple extensions were made that allow two-dimensional entities – points in space, linear features, and polygons, to be encapsulated and considered as Features of entities in an RDBMS. Subsequently SQL was extended so that spatial relationships between entities could be represented and analyzed in relational models (Egenhofer 1994). The spatially extended relational database model is often referred to as *Vector GIS* (Geographic Information Systems.) Vector GIS make possible development of digital representations of physical infrastructure. The planimetric base information available for most cities is collected with aerial photogrammetric surveys and includes representations of such entity classes as building footprints, edge of pavement and property parcels. Even though these representations lack the third dimension, the visualization power and analytical capabilities of these systems has led to the development of spatial data infrastructures critical for understanding and managing assets and activities (CGER 1993).

Figure 3

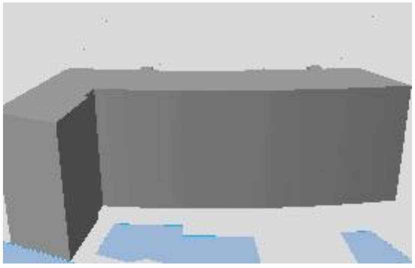


Vector Features in a Database Table

6.4 Visualizing the form of the city in Two and a Half Dimensions

Since entities in a vector GIS system can have one dimensional attributes in addition to two-dimensional features, no additional extensions are needed in order to represent heights of features, which may in turn be represented as extruded envelopes (Hoinkes1995). A common means of doing this is by associating height attributes of tax parcels with the building footprints that sit within each parcel. Though these extruded models add a useful dimension to existing planimetric data, they are sorely lacking as representations of the actual 3D topography of cities. First, because rough quality of the planimetric data takes no account of the actual massing of buildings, and second, because it is very awkward to represent situations such as overhangs and arches with an extruded model (which is why extrusions are known as 2.5 Dimensional, not true 3D models.)

Figure 4

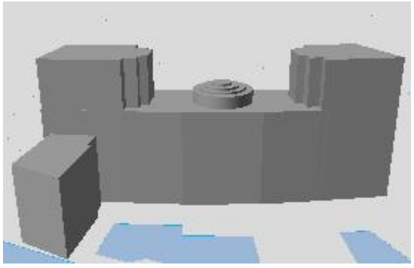


Building footprint extruded to Assessor's story height

6.5 Wholesale Means of City-Capture

A breakthrough in data gathering technology in the late 1990's has created sources of 2.5D data that are much more useful than simple extruded building footprints. Airborne laser scanners collect height information at very fine intervals ~0.5 meters through a process called LIDAR (Light Detection and Ranging). Height data from LIDAR surveys may be automatically transformed into articulated planimetric polygons representing the critical massing platforms of buildings. We will call these more articulated extrusions *Roofprint Models*. Though these models are still only 2.5D they are head and shoulders above extruded footprints when it comes to representing the shape of urban elements and the spaces in between. In 2001, a LIDAR survey of the greater Boston area was flown, and by 2003 it yielded a 2.5D roofprint model of Boston and surrounding cities.

Figure 5:



Roofprints extruded to LIDAR or photogrammetric measurements

Other means of capturing real 3D models of cityscapes, by merging aerial survey data with laser scans and photography collected by specially outfitted trucks have been demonstrated over limited areas (Frueh 2004). The most expansive application of this seems to be a detailed model of the Central Business District of Philadelphia collected by a company called GeoSim Cities at a cost of roughly $150,000 per square kilometer (Eisenberg 2005).

The Boston-Area LIDAR survey roofprints are extremely important as a base for our framework owing to the fact that these models provide an adequate representation of basic massing, that they are complete and seamless across the entire metropolitan area, and, thanks to MassGIS, they are a free public resource for Boston area modelers. Automatically generated wholesale snapshots of urban areas will not suffice as a model infrastructure. Cities change, ultimately people will create alternate versions of places for different purposes and they will want to attach knowledge to these representations. Nevertheless, a rough snapshot is an essential starting place, as a framework on which to build and reference these multiple versions.

6.6 Encapsulation of Complex 3D Data Objects

The documentation for the ESRI Shapefile data format, published in 1998 revealed another extension of the data types that can be stored in their newly open-format for vector-relational tables (ESRI 1998). The **Multipatch** is a means of encapsulating true multifaceted 3D forms with colors and image textures mapped to specific surfaces into a compact data object (Armit 1971). This means of encapsulation is critical to us because it allows us to overcome the difficulties posed by CAD formats in which the relationships between model elements and their textures is dependent on the idiosyncrasies of desktop-based filesystems. This potentially useful extension of the relational model received little attention until recently because there were virtually no tools for creating multipatch features or for transforming existing 3d model assets to multipatch objects.

In 2004 a collaboration between ESRI, the maker of GIS systems, and @LastSoftware, the maker of a simple three dimensional modeling package provided some interoperability extensions to each of their software tools that facilitate interchange of data between the two software programs. The ArcGIS Sketchup extension allows the export of 2.5 and 3D data (including terrain conditions with aerial photography) from a GIS database to SketchUp's 3D modeling environment. SketchUp is used to create true 3D objects with colors and image textures applied to specific faces. SketchUp can also serve to import 3d models from other CAD packages. When these new 3D features are created on a reference framework extracted from GIS, the geospatial coordinates of the objects are maintained. The SketchUp ArcGIS extension allows selected 3d objects from SketchUp to be encapsulated as multipatch objects inserted with user specified attributes into a table in a relational database.

Figure 6

Multipatch object with imbedded texture images.

Organizing model components in a relational database management system offers many advantages over the traditional file-based architectures used in CAD applications. The first is scaleablility: the number of components is virtually unlimited, so the collection can grow in detail and extent as it needs to. The second is distribution and sharing: because RDBMS are designed to provide controlled access over wide-area networks, the facility for collecting and sharing model elements is built-in. Third, the database system provides systematic access to all of the model elements: this accessibility allows model elements to be selected and re-composed for multiple purposes using simple SQL statements, and operations such as transforming every element in the collection (as in a migration to a newer format standard) can be accomplished with relatively little effort.

Though ESRI and @Last deserve credit for being among the first to make accessible this melding of true 3D modeling tools and Relational Databases, and we use their tools in our prototype implementation, we hasten to note that the Multipatch format is an open standard and there is nothing proprietary about the basic idea of storing encapsulated 3D models in a georeferenced RDBMS. Other means of feature encapsulation also show promise. Interoperable Function Classes (IFCs) and CityGML are emerging international standards fop representing entities both in terms of geometric form and semantic relationships (Kolbe 2005.) The sophistication of these data models is admirable, and though Multipatch is primitive by comparison, it has the advantage of being implemented today in a widely used commercial tool. Because of the encapsulation of building representations as a simple data object, a data model constructed for Multipatch objects should be able to accommodate these other types of objects easily as they become practical.

6.7 Summary of the Progress toward a Public Virtual City Infrastructure

There have been many successful projects oriented toward building detailed and expansive 3d models of cities. Good surveys of this activity are given by Batty (2000) and Shiode (2001). To date, none of these projects has aimed at developing a repository of detailed models collected over a broad base of contributors. The focus of all of the major urban simulation and visualization projects reviewed by Batty and Shiode has been on visualization, with little attention having been given to facilitating the process of authoring and the life-cycle of the model as an investment in infrastructure. Our system may be most similar to The Virtual L.A. Project, which has been designed to link multiple small 3d models. Yet the granularity of the models in Virtual LA the models range in size from one to fifteen square miles (Jepson 2005), while the granularity of our schema is intended to store single building entities.

Three developments have altered the environment recently, and make possible the development of a sustainable coordinated three dimensional model that collects the work of a wide base of users:

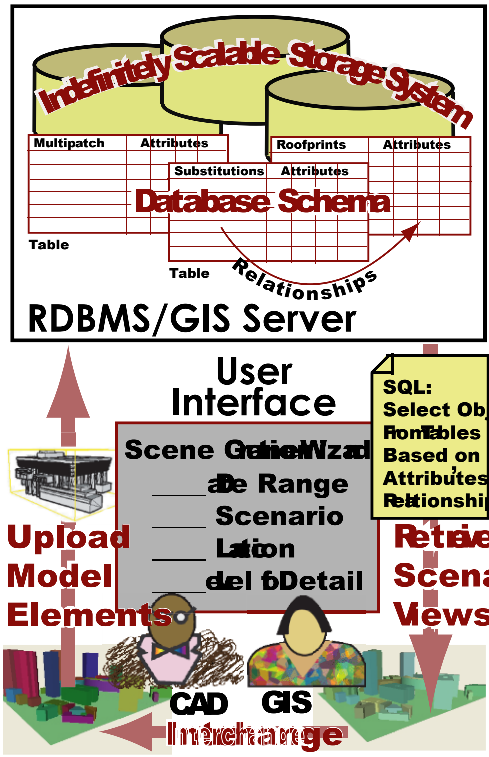
- A simple, inexpensive means of authoring 3d models, encapsulated with textures – provided by @LastSoftware's Sketchup tool.
- A means of organizing encapsulated models in a enterprise-scale relational schema – provided by ESRI's geodatabase model.
- A complete, rough model of the entire Boston metropolitan area that is freely available to use as an exhaustive base model.

We are able to take advantage of these recent developments by building a relatively simple relational schema that will permit a large number of independent modelers to draw upon a repository of ready-made models which they can improve for their own purposes. As a side effect, the work of these modelers is coordinated and yields an ever-growing and improving representation of the city. The model management system is able to retrieve model elements systematically to generate urban scenes that vary in terms of viewpoint-specific level of detail, historical time period, or alternative design proposal.

7 Overview of the Model Management System

The model management system is a repository of model elements (representations of buildings and other fixtures and ground conditions). The schema is relatively simple, composed of only three tables in an enterprise-scale relational database. Users may be located anywhere on the internet. Typical interaction with the system takes the form of three processes: Scenario Retrieval, in which the system builds urban scenes based on scenario specifications and exports them to a rendering program such as ArcGlobe or to a CAD system; Model Creation and Enhancement, where georeferenced model elements are created and uploaded to the system; and Model Management, where models are integrated into the system.

Figure 7



Model Framework Overview

7.1 The Model Management Schema

The schema is composed of three tables: The Roofprints Table, which contains the exhaustive model of the city given by the extruded LIDAR roofprints; The Multipatch Table, which stores detailed, user-created three-dimensional models of urban elements tagged with information attributes; and the Substitutions Table, which stores the relationships between roofprint elements and corresponding multipatch elements that occupy the same location.

7.1.1 Roofprints Table

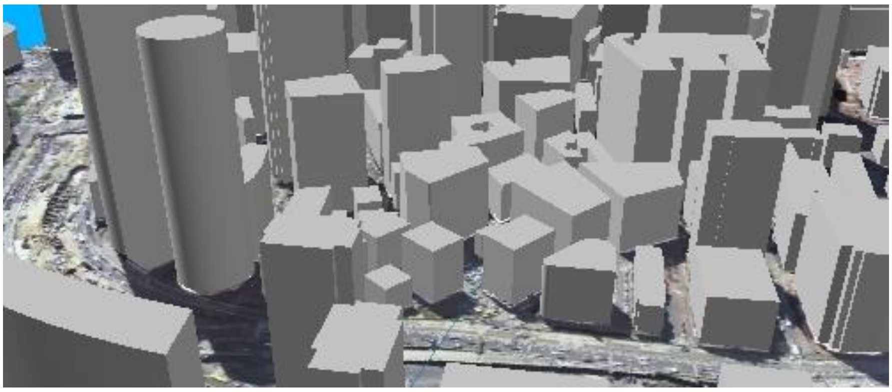
The metropolitan model begins with a complete rough massing model of every building in the area. In the case of our implementation, these are 2D polygons with ground and roof heights that came from the LIDAR-based roofprints given to us by the Massachusetts GIS. Each row in the Roofprints table includes a polygon feature with three attributes:

Roofprint Attributes

Unique ID: a fixed, arbitrary unique number for each polygon
Polygon: contains a 2D planimetric outline of each massing element
Ground Height: The height of the ground at the foot of the building, above mean sea level
Roof Height: The height of the roof of the building, above mean sea level

Updates / Maintenance: The roofprints table is treated as an automatically generated, wholesale snapshot of the city.

Figure 8



The collection begins with a wholesale roofprint model of the region

7.1.2 Multipatch Table

Each row in the Multipatch table represents a scene element (e.g. a building or other fixture, or a patch of articulated terrain or pavement.) Each entity in the system is encoded in two parallel data objects, a Multipatch that is intended for scene rendering purposes, and an editable version, which at this time contains an editable CAD format model of the entity. The Multipatch table may contain multiple representations of the same real-world entity, intended to be viewed at different distances, design scenarios or historical epochs. These characteristics are coded in the attributes for each element in the Multipatch table. Some of the attributes are coded when the entity is uploaded, others in the process of model management.

Multipatch Attributes

The following attributes are entered when a model is uploaded to the system.

Unique ID: a fixed, arbitrary unique number for each multipatch element
Multipatch: the multipatch object intended for scene rendering
CAD Data: an editable version of the scene element
Name: the name of the scene element
Creator: The email address of the person who contributed the model
Contrib_Notes: Notes from the contributor
Type: This field contains one of two values: Building, if the element is a building or other fixture, or Ground, if the element is a patch of articulated ground or pavement e.g. a plaza, bridge or elevated viaduct
Distance: denotes the maximum distance in meters beyond which the details present in the model would not be discernable
Date Min: This is the date that the element was built. This will be left blank if the building is not involved in a historic scenario or if it is proposed
Date Max: This is the date that a building was demolished. This is left blank in the case of buildings that are existing or proposed.
Design Project: For proposed elements that are part of a design scenario, this is a keyword that identifies the project. For elements that actually exist, the design project is 'Built.'
Design Scenario: For projects that have more than one scenario, this field denotes the scenario for which that element should be viewed.

The following attributes are updated by the Model Management administrator during the model management process.

Skyline: This is a Yes/No attribute that reflects whether the element should always be rendered whenever the skyline is in view.
Distance Dependent: This attribute is set to Yes if there are versions of the same element in the multipatch table that are intended to be viewed at different levels of detail.
Date Dependent: this attribute is set to Yes if the element is intended to be viewed for a specific time period.
Alternative: in the case of elements that may have all attributes the same, this number sets the rank. The model with the lowest rank is the one that will be selected in the scene generation process.
Reviewer: The name of the reviewer
Review Date: the date that the model was reviewed
Review Notes: notes from the reviewer

Updates / Maintenance: further information on updates and maintenance of this table are given in the sections on Model Improvement and Management workflow sections, below.

7.1.3 Substitutions Table

This table maintains the correspondence between elements in the Roofprints and Multipatch table. Often there are many roofprints that represent a single building. In the event that a particular multipatch element is selected for a given scene, the corresponding roofprints must be de-selected for that view. This is accomplished by a lookup in the substitutions table.

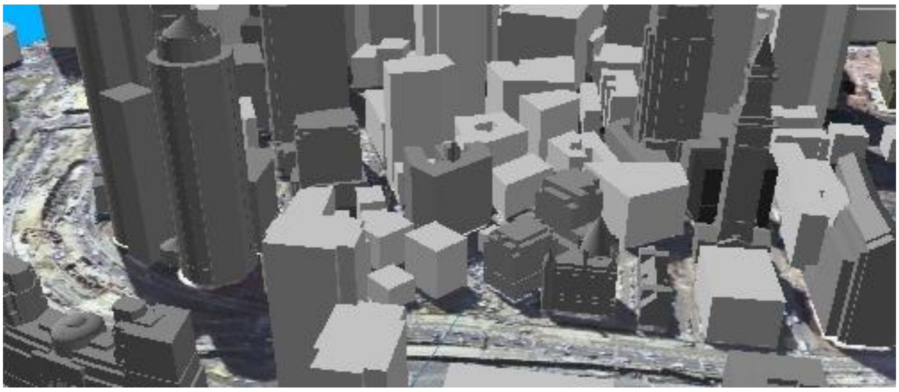
Substitution Attributes:

Roofprint ID: The unique ID of an element from the Roofprints table.

Multipatch ID: The unique ID of an element from the Multipatch table.

Update / Maintenance: This table is updated by selecting the Roofprint elements that are overlapping with a particular Multipatch element; for each roofprint a row is created with its roofprint's Unique ID and the corresponding Multipatch ID. We note that this juxtaposition and substitution is not carried out for multipatch elements of type Ground, which represent plazas or ground conditions which may properly be mutually overlapping with buildings in the roofprint table. This update can be done automatically when models are uploaded into the system.

Figure 9



A substitutions table prevents roofprints from conflicting with detailed models

7.2 Modes of Use

The system has 3 primary modes of use: Scenario Retrieval, Element Creation and Enhancement, and Model Management.

7.2.1 Scenario-Retrieval Process

The primary user interface for the system is a scene generator that takes user specifications for a scenario and selects the appropriate elements from the Multipatch and Roofprints tables for the composition of an urban scene. A scene is comprised of two views (essentially table selections) that are made with two SQL queries: The ***Multipatch View***, which contains the appropriate detailed multipatch elements considering the level of detail desired for buildings positioned near or far relative to the viewpoint, and specifications for the historic epoch of the scene or future design scenario desired for visualization; and the **Roofprint View**, which includes the rough roofprint models corresponding to all of the buildings that aren't selected in the multipatch view for the given scene. Each view is a selection of model elements that may be exported to a renderer for visualization and analytical studies or to a CAD program at the beginning of the Model-Making Workflow.

7.2.2 Generating a Date and Viewpoint Specific Scenario

Below is an example of the queries that select models for a scene based on distance and date dependency. In this case the terms Viewdate and Viewpoint are given to the Scenario Generator. Viewpoint is geometry that indicates the location of the camera in the scene. It is possible to make this detail-specific query work with a camera path instead of a viewpoint -- the query would involve an iteration through the path points.

The scene retrieval process begins with defining the parameters of the scene: where is the scene to be viewed from? What is the historical time period to render? What design scenario should be retrieved from the model repository? These scene parameters will determine a selection of the appropriate elements from the Multipatch table. Based on this selection, the substitution table is used to find the Roofprint objects that should not be retrieved, in order to avoid conflicts of roofprints overlapping with the selected detailed buildings

Multipatch View Query:

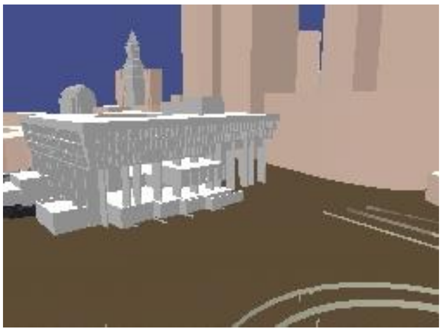
```
CREATE VIEW
'Govt_Ctr_Plaza_Multipatch' AS
SELECT * FROM Multipatch mp
WHERE  Design_Project = 'Built'
AND Skyline = 'Y'
AND Date Dependent = 'N'
OR (Date_Dependent = 'Y'
  AND Date_Min > Viewdate
  AND Date_Max < Viewdate)
AND (Distance_Dependent = 'N'
  OR ( Dist_Depend = 'Y'
      AND
      (DISTANCE(My_Viewpoint,
mp.Geom) <= mp.Distance )))
```

Roofprint View Query:

```
CREATE VIEW
'Govt_Ctr_Plaza_Roofprint'  AS
SELECT * from Roofprints rp
WHERE Unique_ID NOT IN
( SELECT Roofprint_ID FROM
Substitutions Sub
WHERE Sub.Multipatch_ID
IN ( SELECT Unique_ID from
Govy_Ctr_Plaza_Multipatch ))
```

Note that the query that generates the Roofprints view for a scenario is a generic query that simply references the Multipatch view that is generated first.

Figure 10



City Hall Plaza from the Ground.

Figure 11

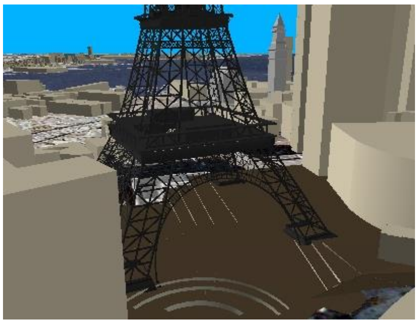
City Hall Plaza from the Air

7.2.3 Generating Scenes for Specific Design Proposals

The queries that select element for a specific design scenario are similar to the queries listed above, except for a couple of things. First, we add an OR clause to select elements from a scenario other than 'Built.' We also note that when a design scenario calls for the elimination of buildings that are currently built, a specific AND NOT clause will have to be added to the view queries, identifying specific Unique IDs for features that should be explicitly eliminated from the view.

Since model views for particular scenarios are merely references to specific elements in the database, the view itself is a very small text file. Therefore there is little trouble saving or sharing thousands of different versions of views, since the actual view elements are stored only once -- in the database. Scenario views saved in SQL form also have the property of making use of the latest state of the on-line model repository each time they are requested.

Figure 12



City Hall Plaza Proposed Scenario

7.2.4 Element Creation and Enhancement Process

When the Scenario-Making workflow is used to create a scene intended for the purpose of adding or enhancing 3D elements, the Roofprint features are exported as extruded 3D geometric objects, and the selected CAD Data features are exported from the Multipatch table. Incidentally, the appropriate patch of terrain and aerial photo is also clipped and sent to the CAD package as a texture-mapped triangulated mesh.

The model elements and terrain model arrive in the CAD system with their geographic referencing system intact. In this context, the user may alter or replace existing elements or create new ones. When the elements are finished, they may be uploaded as multipatch and CAD Data features back to the database. Upon upload, the user is required to enter those attributes for of the element that are user-specified. Uploading model elements never overwrites existing elements in the database. When an element is uploaded, the substitution table should be automatically updated to add entries for any entities in the Roofprints table that overlap with the new element (except in the case of elements of type Ground.)

In most cases, elements uploaded to the system will appear in the model the next time an appropriate scenario is generated.

7.2.5 Model Management Process

A main design goal for this system is to make all of the model enhancements made by users immediately accessible the next time the scenario retrieval process is invoked. However, there are several situations when the attributes of existing elements must be adjusted in order to make certain query selections logically consistent. If new of date dependent elements are added, the min and max dates will have to be adjusted for newer or older buildings that may already exist in the database. In cases where an element has been added that has attributes identical with elements already in the database, someone with permission to change the existing elements will be able to assign priorities to the Alternative attribute. The idea of ranking elements that are otherwise congruent in other respects would permit queries that allow the construction of scenes composed of the Newest or the Most Official features. The ranking of otherwise congruent features will allow obsolete elements in the database to become deprecated without being deleted.

8 Social and Institutional Considerations

The metropolitan model infrastructure is more than a computer system. The system demonstrates an emergent social dynamic as the collection grows and is enhanced by the independent activities of its users. In order to make the most this effect, we need to understand why different classes of users might be motivated to coordinate and contribute their work. It is likely that some users would be quite happy to download snapshots of pieces of the model and to enhance these and simply keep working with these downloads as static, independent CAD models. In this case, we have at least tricked the user into coordinating his work with the georeferenced metropolitan model framework, and that user could easily share his work with others who have used the same framework. This alone is a big improvement on traditional project-based modelling culture.

There are ways that we could discourage the use of parallel snapshots downloaded from the system. By limiting the number of detailed CAD data elements in the Element Creation/Enhancement process to one, or limiting the spatial scope of model downloads, a user could get just what is necessary for enhancing or creating a single element, but in order to visualize that erlement in a larger context, or to take advantage of future enhancements in the model repository, the easiest way to do this will be to upload his new model elements to the common infrastructure.

We have reason to believe that a large number of detailed models may be added to the system by students who will be compelled to model buildings for school assignments. Furthermore, the official models created by city agencies, such as the Boston Redevelopment Authority, or universities like Harvard and MIT may be added to the system as a means of improving accessibility to public resources, and to reduce the cost of sharing information with the public and the development community.

There is a high likelihood that important contributors to the model infrastructure will have model elements that they don’t want to share with users outside of their firm or agency. One way to address this concern is with a federated architecture, having private Multipatch tables that exist in different locations, which are accessed only by Scenario Retrieval processes launched within that agency. In this case local administrators could deal with the Model Management processes in their own locations. A new flag could be added to these local Multipatch tables designating whether a particular element is public or for in-house use only. Alternatively the distributed system might synthesize its multipatch table as a materialzed view of resources that are distributed over a multitude of remote servers.

9 Applications of Multi-Purpose Collaboratively Built Metropolitan Scale 3d Models

Cities are three-dimensional, and cover wide areas that span administrative divisions. As yet, our representations of cities are primarily 2 and 2.5 dimensional, and integrating these representations across administrative boundaries involves difficult processes that must be repeated each time an administrative unit updates their data. Where three-dimensional models do exist, they are limited in spatial scope, yet many of the conditions that we would like to examine in three dimensions involve wide expanses of distance with all of the intervening elements. A means of representing the city that overcomes these issues will portend a new way to understand cities and to manage them more efficiently and effectively.

9.1 Urban Design

Design projects are sometimes built in consideration of views: What sorts of views will be possible from particular parts of the project? How will monumental or potentially unsightly pieces of a project be seen from other places in the city? View studies may in fact involve conditions that are not on the site of the proposal. For example, independent reviewers of a design may be interested in how a design proposal impacts views from one place in the city to another that may be blocked by the proposal in question. These view corridor studies can be very difficult to perform using extant technologies and data. This difficulty has determined that many projects and review processes fail to make careful examinations of the opportunities and problems that may accrue one design scenario or another. By facilitating studies of distant views, very large urban models would undoubtedly lead to the design of more legible and visually appealing cities.

In 2002 a new bridge was built in Boston. The Leonard Zakim Bridge, with its striking cable-stayed design has become a new emblem of the city. Images of the Zakim bridge now festoon everything from taxicab logos to Boston's own marketing materials. This bridge has added new splendour to those places and spaces that now have a view of it. These places have new value (even in unbuilt upper stories) which some people may now want to protect and that that developers may see as market opportunities. Figure 13 shows an expansive view study that considers the view of the tops of the twin spires of Zakim Bridge (in green) as seen from the upper floors of Somerville city hall. Figure 14 shows how the buildings of the proposed Northpoint development in East Cambridge will obscure the view of the bridge. (Thanks to Stephen Hardy for this image.)

Figure 13

View of Bridge from City Hall

Figure 14

View After Northpoint Build-out

9.2 Project Review

The traditional design review process considers one project at a time. When there are proposals that are being considered next to each other, it is technically difficult. It is usually expected that design firms will coordinate their models. A system for model coordination such as we propose here would facilitate the development of models that could easily demonstrate the various permutations of multiple proposals on adjacent sites.

One of the most compelling applications of a shared model infrastructure is illustrated by the recent mistake in which the inability to share a three dimensional representation of ther nearly completed Fleet Center caused a section of Boston's new Central Artery after conflicts were discoveres between the arena and the freeway after construction was already underway; leading to cost overruns of $991,000 and much embarrassment to the Artery Tunnel Authority and the project contractor, Bechtel. Lewis (2002.)

9.3 Public Safety

Some cities are now being outfitted with sensors that will instantaneously pinpoint in three dimensions the locations of gunshots by triangulating on their sound. (Gesslien 2005). This sounds like a very useful way to catch crooks, however considering a gunshot may be coming from an upper story window, much of the value of a three dimensional coordinate for its source may be wasted if our representation of its context is 2D. This is especially true if the precision of the point places it only somewhere in the proximity of the correct building footprint. Visualizing the point on a 3d model showing the height of the nearby buildings would certainly speed the process of guessing the correct source of the shot. A little time saved in this process may make a big difference.

In another Boston example, consider the montlhy passage of the Liquid natural Gas tanker through the Inner Harbor. Substantial resources have been expended in analyzing the risks of an explosion of this tanker, which is in view of buildings located in three or four different cities. One application of a metropolitan model in this regard would be to evaluate the risk to particular buildings in the event of an explosion. In a preventative approach, we may want to investigate what particular windows of publicly accessible buildings could have a shot at the tanker over the course of its journey.

9.4 Administration, Planning and Decision Support

The fact that zoning and land use maps are two dimensional is an accident of history. In reality, land use and zoning are three dimensional, and many urban zoning districts designate different uses for ground floors and upper stories. Understanding existing and proposed buildings in terms of the amount of different uses on different levels, as opposed to simple 2D polygons will lead to much more systematic understandings of those buildings impacts on parking, transportation, evacuation scenarios, etcetera.

9.5 Web-Based Visualization and Referencing

Applications such as Google Earth are beginning to make urban models available on the web and through cell phones. Three dimensional data can be posted and made available through these tools by users. This will have many applications from mere satisfaction of the urge to fly, much better interfaces for choosing seats in theatres or stadiums, to greater civic participation in urban design. Currently the data for these is posted as whole scene files. This architecture presents some inefficiency when there may be relatively small elements that make the important difference between one scene and another. These web-based visualization tools would benefit from something like our scene-retrieval process. And the models themselves, if server based, may be better managed in a framework like the one proposed here. In the meantime, a model system like ours will be a good staging area for generating the scene files that may be posted for web-based visualization.

10 Conclusions

The technical innovation that we have implemented and described here is relatively modest: A schema for organizing a collaborative urban model, capable of rendering scenes appropriate for a given point of view, or specified design scenario or historical epoch. This innovation is made possible by many very substantial technical advances that are converging at this time – Expansive 3d scans of metropolitan areas (LIDAR), a new format for encapsulating and sharing 3d models (Multipatch), and a relatively old means of building an information infrastructure (Relational Database Management Systems and GIS.)

The technical aspect of our proposal is not nearly as interesting as the social context that makes it important. Our growing interest and utility for information that is referenced in three dimensions demands a framework that can be used to create and contextualize references to things where they are. Virtual representations of cities will become important means of understanding and using the real city. What sort of things and events will we be able to reference in these virtual cities? Who will control access to the virtual streets and sidewalks?

We have created a proptotype of the model management system that works in ArcGIS and Sketchup. Our aim over the next year is to use this system to begin to assemble substantial amounts of existing project oriented 3D models from the Central Artery project, and the Boston Redevelopment Authority. Through courses at the Graduate School of Design, we will experiment with and document means of adding representations for other landmarks and objects such as bridges and elevated highways. At the same time, we will document and promote the use of the system as a means of facilitating the process of compiling and creating project oriented models in the local design community, and we will document compelling applications for expansive urban 3D models. We hope that through these efforts, that a community will form around the idea of a cooperatively developed urban model infrastructure for the Boston Metropolitan Area.

Acknowledgements

This project was conceived and funded as a joint effort between the Harvard Graduate School of Design and the Boston Redevelopment Authority. We would like to thank all of the staff at the BRA who have provided data and ideas and continue to support this research. MassGIS and should be commended for sharing their valuable data resources with the public for the betterment of the people of Massachusetts. Many of the detailed models shown in our illustrations were created by the Central Artery Tunnel Authority. The model of the Eiffel Tower is courtesy of the Official Website of the Eiffel Tower. Members of my 2005 seminar, Topographic Modeling in 3D deserve credit for some of the ideas and inspiration. Stephen Hardy produced the images of the views from Somerville City Hall. Julie Dorsey was helpful in reading this paper and offering helpful comments. Stephen Ervin, Assistant Dean for Information Technology at the Harvard Graduate School of Design has been very encouraging and supportive of this research.

References:

@Last Software Inc (2005) *The SketchUp and ArcGIS Interoperability Solution*, http://download.sketchup.com/downloads/markets/SketchUpESRIArcGISUserGuide.pdf

@Last Software Inc (2005:2) Sketchup for Microsoft Windows User Guide http://download.sketchup.com/downloads/sidebars/SU5_PCUserGuide.pdf

Armit, Andrew, (1971) *Multipatch and MultiObject Design Systems*, Proceedings of the Royal Society London, Vol. A 321 p. 235.

Batty, M, Chapman, D, Evans, S, Haklay, M, Kueppers, S, Shiode, N, Smith, A, Torrens, P,Visualizing the City: Communicating Urban Design to Planners and Decision-Makers, Center for Advanced Spatial Analysis, Paper 26, 2000 http://www.casa.ucl.ac.uk/visualcities.pdf

Chan, R, Jepson, W, Friedman, S, "Urban Simulation: An Innovative Tool for Interactive Planning and Consensus Building" in *Proceedings of the 1998 National Planning Conference*. December 1998 http://www.asu.edu/caed/proceedings98/Chan/chan.html

Clinton, W., (1994) "Coordinating Geographic Data Acquisition and Access to the National Spatial Data Infrastructure." Executive Order 12096, Federal Register 59, 17671-4, (Washington, D.C.).

Codd, E. F. (1970) "A Relational Model of Data for Large Shared Data Banks" Communications of the ACM, Vol. 13, No. 6, pp. 377-387

Commission on Geosciences, Environment and Resources (CGER) (1993) *Toward a Coordinated Spatial Data Infrastructure for the Nation,* National Research Council,National Academy Press Washington, D.C. 1993 http://www.nap.edu/books/0309048990/html/R1.html

Egenhofer, Max (1994) "Spatial SQL: A Query and Presentation Language" IEEE Transactions on Knowledge and Data Engineering 6 (1): 86-95.

Eisenberg, Anne (2005) "A 3-D View of the City, Block by Block", New York Times, February 17, 2005

ESRI Inc (1998) *Shapefile Technical Description An ESRI White Paper* p20-24, http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf

ESRI Inc (2004) *What Is ArcGIS*, ESRI Press 2004 http://downloads.esri.com/support/documentation/ao_/698What_is_ArcGIS.pdf

Fueh, C., Sammon, R., Zakhor, A., (2004) "Automated Texture Mapping of 3D City Models With Oblique Aerial Imagery" Second International Symposium on 3D Data Processing, Visualization and Transmission; pp. 396-403 University of California, Berkeley, 2004

GeoSim Cities (2005) Virtual Philadelphia, Website of the Spring Conference on Computer Graphics, Budmerice, Slovakia May 12 - 14, 2005 http://www.sccg.sk/pages/vri/2005/philadelphia/#top

Gesslein, Daniel (2005) *Gunshot Sensors Put Thugs on the Run*, New York Post November 6 2005 http://www.shotspotter.com/NYPost_11_06_05.htm

Hoinkes, R. and Lange, E. 3D for Free: Toolkit Expands Visual Dimensions of GIS, accessible online and in GIS World, July 1995

Jepson, W. (2000) Home Page for the Virtual L.A. Project, School of Architecture, UCLA, Los Angeles, CA. http://www.aud.ucla.edu/~bill/UST.html

Kolbe, T, Gröger, G, Plümer, L, (2005) "CityGML – Interoperable Access to 3D City Models" in Oosterom, Zlatanova, Fende, Editors: *Proceedings of the International Symposium on Geo-Information for Disaster Management*, 2005; Springer Verlag. http://www.citygml.org/docs/Gi4Dm_2005_Kolbe_Groeger.pdf

Lewis, R (2002) "Pike pursuing refund from Bechtel for Big Dig errors" FleetCenter case, 5 others getting new legal scrutiny Boston Globe February 17, 2002

Morville, Peter (2005) *AmbientFindability*: What We Find Changes Who We Become. O'Reily 2005.

Perez, Juan Carlos (2005) Google Earth to be linked with enterprise search tools; IDG News Service 11/09/2005

PostGIS (2005) User Guide http://postgis.refractions.net/

Shiode, Narushige (2001) "*3D urban models: recent developments in the digital modeling of urban environments in three-dimensions"* GeoJournal 52 (3), 263-269.)

Batty M, 2000, ``The new urban geography of the third dimension'' Environment and Planning B:

Planning and Design 27 483 ^ 484

Ferguson C H, 2005, ``What's next for Google'' Technology Review 108(1) 38 ^ 46

Tech Firms Try to Conquer the Globe

Google and Microsoft Compete to Offer Navigable 3-D Maps of the Earth

By Mike Musgrove

Washington Post Staff Writer

Wednesday, June 15, 2005; Page D05

Building Footprints (2-D, from Lidar data) - January 2005

http://www.mass.gov/mgis/lidarbuildingfp2d.htm