Adapting the Town of Brookline's GIS Infrastructure for Managing Three Dimensional Models

Prepared for
Feng Yang
Director of GIS
Town of Brookline

By Paul Cote
Lecturer in Urban Planning and Design
Harvard Graduate School of Design

July 31 2007

Table of Contents

1. Executive Summary 3
2. Integrating Brookline's Existing Building Information 3
2.1 A Framework for Building Identification 4
2.2 Extruded Building Massing Parts 5
2.3 Three-Dimensional Building Skin Parts 5
2.4 Design Schemes 6
2.5 Scheme Parts Table. 7
2.6 Query Interface for Rendering Views of the Town 7
3. Cost, Benefit and Risk Assessment 8
3.1 Building Identifiers Table Recommendations 8
3.2 Building Massing Parts Table 9
3.3 Building Skin Parts Table 9
4. Summary Recommendations & References 11
Annex A Technical Implementation of the Model Management System.... A-1
A.1 Elements of the Model Management System A-1
A.2 Table Schemas A-3
A.3 Scene Retrieval Queries: A-6
A.4 Portrayal of Views A-8
A.5 Use of the Building Model Management System to Export Site Models to Sketchup A-8
A.6 Populating the Model Management System A-9
Annex B Extensions to the Building Model Management System B-1
B.1 Adding an Addresses Table B-1
B.2 Create and Maintain a Town Wide Ground Cover Layer B-1
B.3 Terrain Model Management System B-1
B.4 Extension to Interior Spaces Model B-2
Annex C Contents of Data CD C-1

1. Executive Summary

The issue addressed by this study is how the Town of Brookline should manage information about buildings. The challenge is to propose adaptations of the town's GIS infrastructure in order that information about buildings, including digital three dimensional models, may be collected and provide a means of understanding the form of places in the past or in proposed future schemes. While it may be possible to create models of places for specific prposes, the challenge is to organize model elements into a single repository that is referenced by project specific models The temporal aspect of this problem is not only that the historic models for the city should be preserved to provide historic and future views, but also that the system should ensure the continued development of the repository itself through expected changes in technology and data formats.

The study begins with a summary of the current activities and resources related to creating and collecting information about Brookline buildings, including three dimensional models. Simple modifications to the Brookline GIS system are proposed that will facilitate the management of building information. The discussion of the future development and maintenance of the model repository includes a description of the tools and formats that are emerging and will need to be considered as the Brookline model management system evolves.

Annex A to this study describe the technical implementation and procedures for adding data to the system. Annex B discusses future extensions to this database schema. Annex C describes the contents of the data and demo CD that is provided withthis project, including a description of the tools that were developed for creating and populating the Brookline Building Model Management tables. The CD provided with this report contains the template database schema and the tools for developing the feature data and tables from the town's existing GIS database. The powerpoint presentation included with this document and on the CD provides a summary tour of the concepts covered in this document.

2. Integrating Brookline's Existing Building Information

The Town of Brookline has several layers related to buildings already in its GIS system. Two dimensional building footprints originally collected in 1996 have undergone updates over the years, and comprise a nearly complete database about buildings in the town. Although there is no attribute data associated with the building footprints database, there is a Property Parcel database that is maintained on an on-going basis by the Town Assessor. Though this database is not explicitly about buildings, it does include much information useful for starting a buildings database. The town also has a growing collection of three dimensional models that have been created for buildings or groups of buildings in selected locations. The town also has access to several data layers from

MassGIS, which conducted a laser-scan, yielding elevations at one-meter postings for the entire town, which can be used to add height information to buildings.

2.1 A Framework for Building Identification

Since the town does not have an existing building numbering system or a database of buildings, we begin with a basic framework for identifying buildings. This would be comprised of a database table that includes a unique ID string, owner contact information, construction and demolition dates, etc. The full schema for the buildings table is given in Annex A. A first-stab at populating this table is accomplished using the town's Assessor Parcels Database. No doubt this initial database is incomplete and incorrect in many cases (noteably where there is more than one building per parcel) but it is intended as a starting place that can be articulated as specific buildings receive attention in data development or maintenance operations. The system for uniquely identifying buildings begins with the unique parcel ID for the parcel that overlays the center of each building footprint. An additional string is appended to the parcel ID to differentiate buildings where there is more than one par parcel.

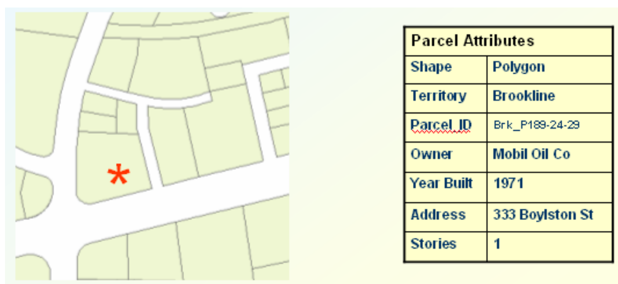


Figure 1: Parcels

Ideally, the rows in the Building Identifiers table have a one-to-one relationship with the buildings in the town. It will be useful to be clear about how buildings are discriminate in other parts of the Town government. This definition may come from the inspectional services department. It may be expected that the Brookline GIS may eventually further discriminate buildings into sub-objects, such as separately owned condominiums and further, into interior spaces. Prospects and considerations for these extensions are discussed in Annex B.

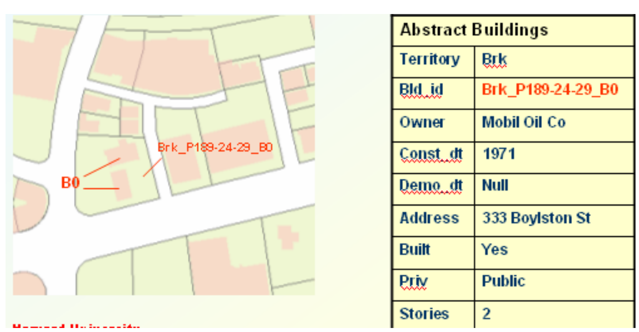


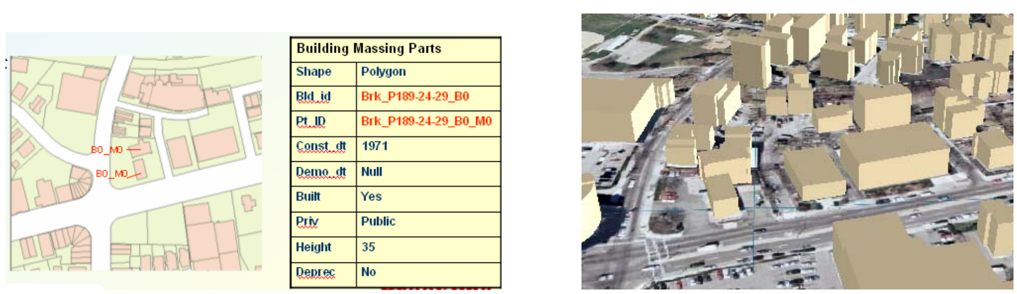
Figure 2: Building Identifiers

For the purposes of making a data model for temporal visualization it is necessary to discriminate separate parts of buildings that were built or demolished at different times, or in the case of the extruded massing model, represent parts of buildings with different heights. The repository may also represent building parts or entire buildings that have been proposed but not built. Each Building Identifier and Building Part entity has Built attribute that establishes whether that entity should be rendered in a date-specific scenario. Buildings or building parts designated as "Unbuilt" would only be rendered in the context of special design schemes. Design schemes can also simulate the removal of building parts that are "Built."

For the purposes of experimenting with design scenarios in-house, it may be desireable to mark specific buildings or building parts that should not be shared with outsiders. This requires that each building or building part is marked with a Privileges attribute that is designated either as Public or Private. Ultimately a finer breakdown of privilege groups and encrypted passwords may be developed. The tables that manage building parts for visualization are described below.

2.2 Extruded Building Massing Parts

The Building Prep tools included on the data CD for this project include a tool that associates each building footprint in the Town's existing Buildings layer with an initial building ID from the building identifiers table described in the previous section. The same tool also uses the MassGIS LIDAR elevation survey to establish an estimated elevation for the foot and the top of the building. There may be more than one extruded massing polygon per building, as would be useful for portraying a high-rise building that is centered on a two-story pediment. This is achieved by considering each polygon in this table a "Massing Part." Massing Prts have their own IDs that append an incremented number to the building ID. Each row in this table also includes a foreign key to the parent row in the Building Identifiers table. Massing Parts have attributes that may over-ride attributes of he parent Abstract Building, such as construction or demolition dates and viewer permissions.

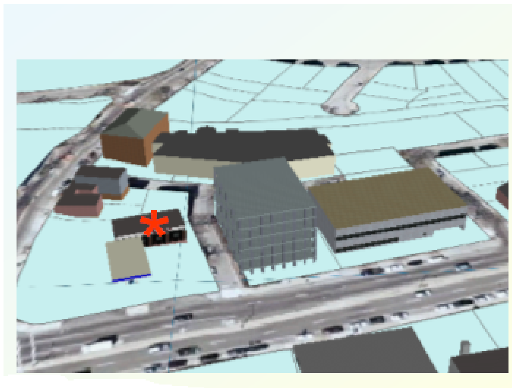


Building Massing Parts	
Shape	Polygon
Bld_id	Brk_P189-24-29_B0
Pt_ID	Brk_P189-24-29_B0_M0
Const_dt	1971
Demo_dt	Null
Built	Yes
Priv	Public
Height	35
Deprec	No

Figure 3: Building Parts

2.3 Three-Dimensional Building Skin Parts

Building skin models are models representing the shells of buildings or building parts. In the database these models are represented as encapsulated data objects. In our ESRI implementation, the skin models are represented as Multipatch objects, which are used for viewing and analytical purposes; and SketchupModels that can be downloaded as into an editable Sketchup model. Each of these data objects is encapsulated and occupies one cell in the table. The town has a collection of skin part models that have been made for specific projects. The Skin Parts table has a unique ID for each Building Skin Model, a foreign key to the Building Identifiers table, and other attributes. The full schema is described in Annex A.



Building Skin Parts	
Shape	Multipatch or other
Model	Sketchup or Collada Blob
Bld_ID	Brk_P189-24-29_B0
Part_ID	Brk_P189-24-29_B0_S0
Owner	Null
Built	1971
Demo	Null
Built	Built
Priv	Public

Figure 4: Building Skin Parts

A single building may have multiple skin parts. This enables us to represent additions to buildings that have been built or removed at different dates without having separate models that duplicate most of the elements of a building. Annex A includes a section describing the procedure for adding building skin parts to the database.

2.4 Design Schemes

The Building Parts and Building Identifiers tables described above are sufficient to retrieve the appropriate building parts to recreate views of the town that existed at a given date. In order to manage building parts and design scenarios that have never been built, requires two more tables to be added to the schema. The first table, Design Schemes, permits schemes to be registered with a Unique ID, a Scheme Name and a Description. A Design Scheme, allows certain building parts to be turned on, and others to be turned off, relative to what would otherwise be rendered for a specific temporal view. For example, to visualize a design scenario, one building part representing an unbuilt ice cream shop might be turned on, and two other building parts that actually exist must be turned off. These part-specific instructions are handled in the Scheme-Parts table. Annex A includes procedures for creating a new design scheme and its associated records in the Scheme Parts Table..

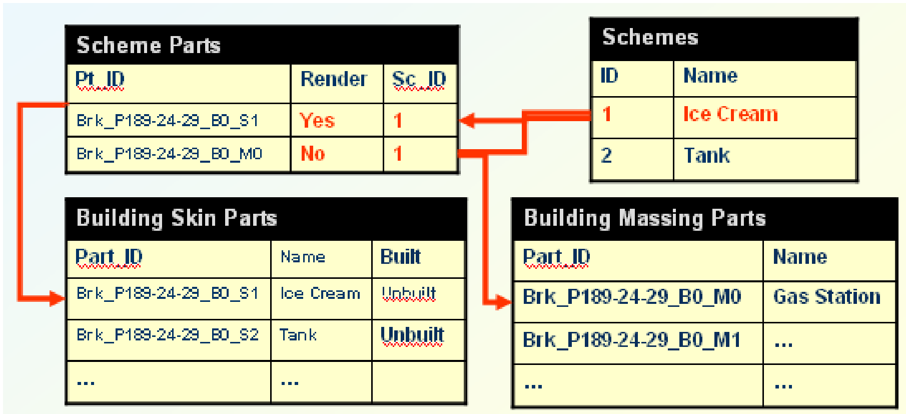


Figure 5: Schemes and Parts

2.5 Scheme Parts Table.

As discussed in the previous section, it is necessary to accommodate exceptions to the rules for selecting building parts to render based on their dates of construction and demolition. The attributes of the scheme parts table identify the Scheme ID, the Building Part ID, and a Yes/No attribute that determines whether the part in question should be turned on or off if the specific scheme-view is invoked. The procedure for constructing the unique IDs of building parts, assures that each part, whether a Massing Part or Skin Part, or a class of part yet to be established, will have a unique ID; this condition allows us to keep a simpler, more stable schema with only one Scheme-Parts table.

2.6 Query Interface for Rendering Views of the Town

The benefit of arranging and maintaining the town's information resources in the tables described above, is that views of the city can be retrieved from the database using a simple interface that asks the user to provide the Date for the view, The Design Scheme desired to be viewed, and whether the View is permitted to the Public or only for In-House use. This query tool will produce a model of the city. By storing all of the three dimensional models in the cities repository in a single table, and retrieving views when needed, the model views will always reflect the latest state of the model repository and eliminate the trouble of maintaining separate models that will undoubtedly become out-of-sync, or redundant. The scene retrieval tool is included on the data CD provided with this study and the explicit text of the queries used is given in Annex A of this report

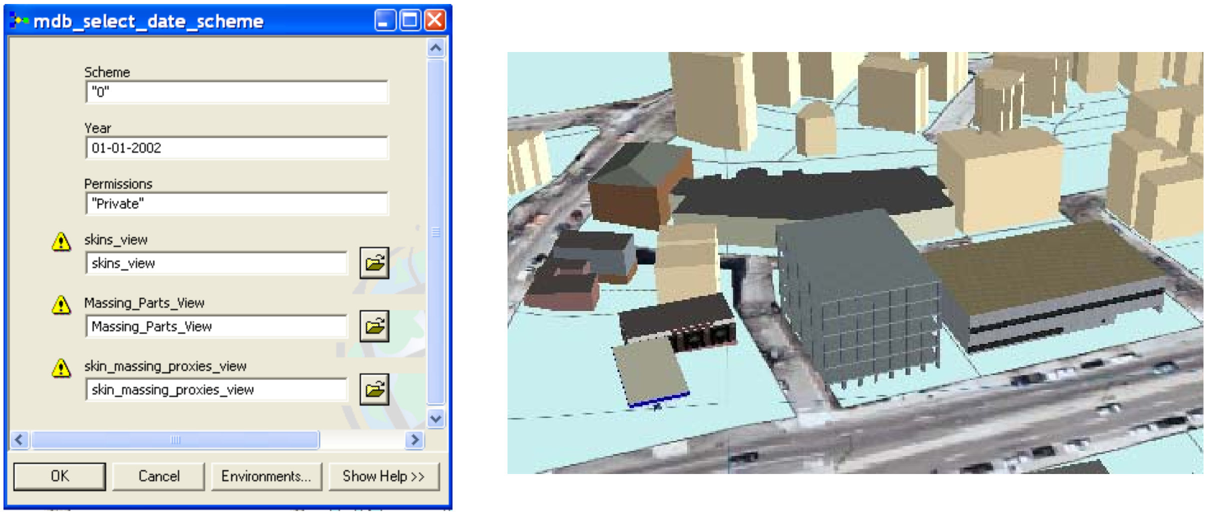


Figure 6: Scene Retrieval Tool

3. Cost, Benefit and Risk Assessment

Any proposal to develop an information system should consider the ongoing costs of maintenance. These costs far outweigh the costs of development over the life-span of the system. These should be weighed against the risks that the investments in time required to develop and maintain the data are meeting some business purpose or other motivation within the town government. The system should also be looked at with respect to the risk that the assets that are developed and managed in the system do not become obsolete or too expensive to adapt in the case that the technical environment or business-needs should change. This section examines the costs and benefits of developing and maintaining each of the substantial pieces of the proposed model management system. Because interoperability and open standards are an important part of the risk equation for developing information resources, we will begin with a brief discussion of trends in this area that will impact proposed system in the next few years.

3.1 Building Identifiers Table Recommendations

The primary business practices that should be considered with respect to the Building Information Management System are the existing and planned business practices in the city that regard Building Identifiers. It is possible that these business process occur in Inspectional Services, Fire or Police departments, and perhaps also by local utility companies. The core of this model is the Building Identifiers table, and in particular the assignment of unique IDs for buildings. Before implementing the system as it is described here, careful consideration should be given to whether the mechanism for generating Building Id numbers proposed here (see Annex A) will suit these other business practices. If possible, it should be determined whether the Building Identifiers table proposed here can either be shared among other applications existing within the town government. The extent to which Building Identifier information is integrated with the processes of permitting and other existing practices will help to assure that the visualization components of the system retain their utility.

3.2 Building Massing Parts Table

Given that the Town is comfortable with maintaining a table for Building Identifiers, the extension of this with an association between Building Identifiers and Massing Polygons, is not that costly an endeavor. It is evident that the town GIS office already maintains the Building Footprints feature class. The current proposal simply creates several new attributes for this table, which to a large degree can be automatically generated. The added benefits of maintaining this table, along with its associations with the Building Identifiers Table should be well offset by the benefits of having quick three dimensional visualizations that can also be exported to 3d modeling applications as a basis for development of more detailed three dimensional models.

3.3 Building Skin Parts Table

The Cost, Benefit and Risk assessment for development of the Building Skin Parts table has a few more things to consider than the Building Identifiers Table and the Building Massing Pats table. First, the skins table is a completely new piece of the town's GIS infrastructure that serves no other purpose than composition of three dimensional models for visualization. In addition, the technology for authoring, exchanging, visualizing and storing three-dimensional models is certain to change in the next few years.

The town has already demonstrated an interest in collecting visualization models for building skins. It was suggested at the outset of this study that the collection of building models would grow both as a result of development of neighborhood-wide models and also by incorporating models made by developers as part of the design review process. The cost of integrating these models with the system as described in this study would be approximately 15 minutes per building, except where substantial editing of a Sketchup model is required. It is likely that this cost is justified given the much more flexible management of these model assets that would be enabled by the new system. From this perspective the building skins table will be worth developing.

Before deciding to build the Skin Parts table solely on the grounds of the short term business case, however, it is worth considering expected changes in the technical environment in the near-to mid-range future. Our first concern with this table is whether it is the appropriate structure to support storage and visualization. An argument might be posed that the representation of building skins should be more articulated than simple encapsulated building parts. For example should the skin of buildings be segregated into walls, wall openings, protrusions, roof surfaces, and so on, as is recommended by the CityGML Level of Detail 3 Building model? If so, would this make our skin parts table obsolete? Our conclusion is that there is in fact a structural reason to hold simplified skin models that are solely for the purpose of visualization. In a CityGML model with more detailed LOD 3 and 4 buildings, it will be desirable to have simplified versions of these for web-based visualization, and either these will be created on the fly as needed, or stored persistently when building models are created. We believe that the Building Skins table will be a useful structure for dealing with visualization models within a city model management system, even though this is not explicitly called for in the CityGML specification.

The next question to be considered is the shelf-life of the two different types of assets that are stored in the Building Skin Parts table. The most critical of these assets is the Multipatch object, which is used by the ESRI geodatabase for exchange between the database and ESRI visualization tools such as ArcGlobe. This multipatch object is also used for simple analytical operations, such as overlay analysis and spatial query. It is likely that Multipatch will continue to be solely an ESRI data format, but until other spatial database creators develop a similar composite 3d model data type, users desiring this capability in a GIS data models should consider themselves hooked on ESRI.

The next question is whether the design for accommodating building skin models is vulnerable due to its attachment to proprietary exchange formats. The means of exchange between Sketchup tools and the ESRI database is of particular concern. The dependence on the proprietary Sketchup format and tools for exchange and storage of the editable building skins is vulnerable in that the capability of moving models from Sketchup to the ESRI database is a process that depends on active cooperation and development between ESRI and Google Sketchup each time either party releases a new software revision. This sort of interoperability architecture is an extreme example of an unsafe long-term investment. A much better approach to this would be an architecture where three dimensional models could be exchanged between authoring clients, storage systems and visualization tools using a stable, independently managed exchange format. There are two exchange standards that are available in mainstream 3d authoring tools today. One is Collada (Khronos Group) which is suited for visualization, and the other is IFC (International Alliance for Interoperability) which is more suited to articulated, semantically rich building models. If we accept the argument that the model management system should have a building skin parts table that serves the purpose of storing encapsulated models optimized for the sole purpose of visualization, then Collada would be a suitable exchange and storage format for these types of data objects. In the event that the town moved toward a finer-grained system that accommodates a finer articulation of building skins or interior components of buildings, then an IFC oriented system should be considered, but even in this case, it is likely that the pre-simplified skin models that might be derived from the IFC model, would end up delivered, if not stored as Collada format objects. In any case, a change in the workflows and formats for exchange of 3d building skin models may threaten the shelf life of the editable model objects stored in the Skin Parts table, the multipatch representations of these models should continue to perform, so loing as ESRI continues to support multipatch.

As it happens, ESRI has announced an intention to support Collada exchange in its next release of ArcGIS, although it is not clear whether this is will be for import as well as export. For the purposes of supporting a more interoperable Building Skins capability, we would hope that ESRI and other GIS vendors plan to support a means of importing Collada files. The system should store the Collada objects as encapsulated models with textures that could be exchanged with a multitude of editing tools. The system should also convert these to its own internal format for visualization and analysis, which, in the case of ESRI products, is Multipatch. In this case, the existing schema would be modifies with a simple addition of a ColladaModel field that would store the encapsulated Collada model.

To conclude this somewhat confusing section on the advisability of developing and maintaining the Building Skins table, we have a short-term and a long-term answer. The idea of having a repository for building skin models makes sense, and the Multipatch format for storage, analysis and visualization of these assets is likely to remain stable so long as ESRI is your GIS vendor. To the extent that new building models are planned to be developed with Sketchup versions 5 or 6, then this table should continue to be developed using the procedures described in annex A of this report. The building Skin Models table is expected to evolve over the next few years to handle Collada format model assets instead of or in addition to Sketrchup format models for interchange. To the extent that the town should plan to make large investments in building skin models, such as a wholesale effort to model all of the buildings in the city, or to develop building models with interior features or more articulated skins, then we would recommend waiting until more stable and open specifications for structure and interchange (Collada and IFC, respectively) become supported in commercial authoring tools and their interchange with data management systems.

4. Summary Recommendations

The following actions are recommended in order of priority

1. Study the suggested schema for building idenitifiers for the Building Model Management System, along with the extensions for the Address Table, and consider how these new information tables might integrate with existing systems in use by Town of Brookline departments. Make changes where necessary to assure that the building identifiers table will be most cost-effective for long-term development and maintenance. Understand the impacts on current systems that may be created by new tasks associated with the update of building information and decide if these will be worthwhile to continue over time. Modify the tools provided with this study, to create the initial building identifiers table from the Tax Parcel information.

2. Assess whatever changes may be necessary to the schema of the building footprints layer to adapt it for the purposes of a Building Massing Parts Layer. Decide whether it will be worthwhile to maintain the additional attributes in this table when building footprints are modified. Modify the tools provided with this study to create the initial state of the Building Massing Parts Table.

3. Understand the costs and benefits of storing the Building Skin Parts table as part of the town's GIS. Import existing sketchup models into the Skin Parts Table following the procedures discussed in Annex A. As new building models are created, have them added to this table on an ongoing basis.

4. Use the Building Model Management Systems to help planners understand the impact of proposed changes

5. Develop a web viewing system that would be useful for first responders or the public for viewing the model.

Questions or comments about this study can be directed to Paul Cote, pbcote@gsd.harvard.edu, 617-496-0546.

References & Further Reading:

Demo Data and Related Presentations:
http://www.gsd.harvard.edu/pbcote/research/brookline
ESRI/PENBAY Building Information Spatial Data Model:
http://www.bisdm.org
Industry Foundation Classes (IFC):
http://www.iai-international.org
The National BIM Standard
http://www.facilityinformationcouncil.org/bim/
City Geography Markup Language:
http://www.citygml.org/

Annex A Technical Implementation of the Model Management System

This annex describes the implementation of the model management system. Section 1 describes the feature datasets, tables and query tools. Section 2 describes procedures for routine use and development of the database. The tables and tools described here are included in the data CD.

A.1 Elements of the Model Management System

The system developed in this study is concerned with management of information about buildings including the attributes, such as when the building was built, when it was demolished, contact information for the building operator, etc. From the beginning, it holds a three dimensional massing model of every building in the town. A building may be made up of various massing parts that may have different heights, construction dates, uses or other attributes. Any building or part of a building may also be represented with a detailed three dimensional model of the building's skin.

A.1.1 Relational Concepts

The system is designed according to the concepts of relational database management systems. This ensures that all of the elements of a given type are stores in a single table. This is what makes the system scalable. These tables assure that any building model created in the system remains in the system. If multiple versions of a building part exist, then they are differentiated by their construction or demolition dates or by the deprecated version attribute. This design assures that all of the assets in the system are accessible systematically by pre-defined queries written in Structured Query Language (SQL).

A.1.2 Date Queries, Model Views and Detail Swapping

The database is designed so that views of the city may be retrieved from the database by simply entering a date into a form. A view of the database is a query that results in temporary layers that can be used for visualization or export to a three dimensional modeling system, providing context for further development of models. Since a view of the database is based on a query, each time the view is retrieved; it accesses the latest versions of the appropriate model elements. The idea of model views eliminates the potential that different models of parts of the city will be maintained independently and become out-of-sync with other models of the city.

The SQL queries designed to operate on this relational schema will select the buildings that existed at a particular date. Detailed skin models that exist for appropriate buildings are retrieved to a view layer, massing models to fill in the rest of the city are retrieved into another view table. Using a viewer such as ArcGlobe, the detailed models may be set to swap with their less detailed massing models when the detailed models are greater than a certain distance from the viewer.

A.1.3 Design Schemes

The system is designed so that fictitious design schemes can be configured in order to examine scenarios that have never been built. These schemes may include proposed buildings and also proposed demolitions of other buildings. Buildings and building parts have a “Built/Not Built” attribute that is used to designate elements that should not be rendered in views unless they are identified in a specific design scheme.

A.1.4 Viewer Privileges

Particular design schemes or building models may be restricted in terms of what sort of user is allowed to see them. Some building designs in the review process may not be shareable with the public. The Privileges attribute set on building parts and design schemes is set to Public by default, it may also be set to define other groups of users that have permission to view specific building parts or schemes.

A.1.5 Unique Identifiers

The relation of building parts to building information is maintained by unique identifiers for each building. In Brookline, there is not a pre-existing system of identifiers for buildings, so we have adapted the Tax assessor Parcel ID to identify buildings. This is not entirely logical, since there may be more than one building per parcel, so we have added an extension to the Parcel ID to differentiate buildings on a specific parcel. Furthermore, since the model management system may be used to integrate models from other cities, we have added a City prefix to the Parcel ID to assure that these IDs will be unique when data form neighboring territories as added. The various massing and skin model parts tables have an attribute identifying the parent Building ID, and also have a part ID that adds a suffix to the Unique Building ID so that multiple parts of the same building may be differentiated. Therefore, the identifier: Brk_P_218-09-00_B0_M0 Uniquely identifies Massing Part 0, for Building 0 of Parcel 218-09-00, in the Town of Brookline.

Considering the potential round trip of data from this schema through a cityGML, requires that every feature in the schema have a universally unique identifier. The hierarchal method of building these IDs is helpful, since IDs can be manually chosen for a particular building part without the user having knowledge or control over IDs that may be chosen in a different location.

A.2 Table Schemas

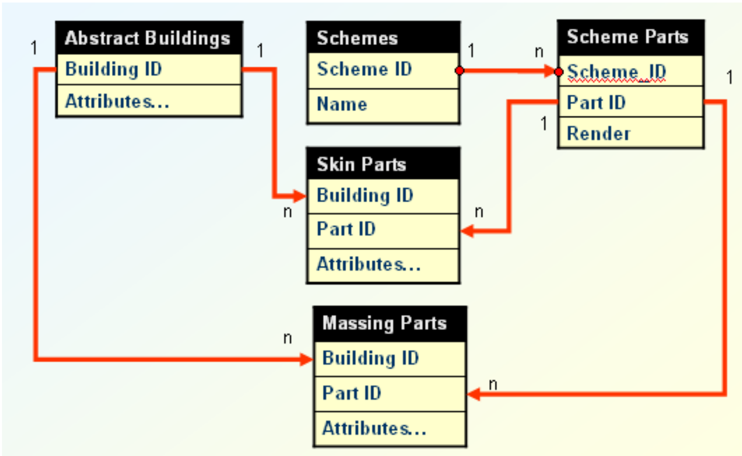


Figure 7: Model Management Schema

A.2.1 Buildings Table Schema

This table carries information about buildings. These attributes were taken from the Brookline Parcels Data Layer.

Column Name	Description
bld_id	Unique Building ID
par_id	Parcel ID
Bld_name	Building Name
Bld_addr	Building Address
Bld-use	Building Use (Massachussets D.O.R. Code
Bld_contct	Building Operator Contact Info
Bld_con_dt	Building Date of Construction
Bld_dem_dt	Building Date of Demolition
Bld_built	Built or Unbuilt (designames proposed buildings)
Build_priv	Viewer Privileges (Public or Other)
Bld_stry_a	Building Stories Above Ground
Bld_st_a_h	Height of Above Ground Stories
Bld_stry_b	Building Below Ground Stories
Bld_st_b_h	Height of Below Ground Stories
Bld_rf_t	Building Roof Type (CityGML Codelist)
Bld_gfa	Building Gross Floor Area
Bld_zn	Building Zone

A.2.2 Massing Parts Table Schema

Massing parts are extruded polygons. Each massing polygon must have a foot elevation, roof elevation and height. Other attributes of building massing parts may over-ride those of the parent building. In Brookline, this table was created from the Town's existing building footprints data layer, elevation and height attributes were calculated from the MassGIS LIDAR Survey.

Column Name	**Description**
Shape	Polygon
Pt_id	Unique Part ID
bld_id	Building ID
pt_name	Part Name
pt_addr	Part Address
Pt_use	Part Use (Massachussets D.O.R. Code)
Pt_contct	Part Operator Contact Info
Pt_con_dt	Part Date of Construction
Pt_dem_dt	Part Date of Demolition
Pt_built	Built or Unbuilt (designates proposed building parts)
Pt_priv	Viewer Privileges (Public or Other)
foot_el	Elevation of Building Foot (above sea level)
roof_el	Elevation of building roof (above sea level)
Height	Height of building part
Pt_stry_a	Part Stories Above Ground
Pt_st_a_h	Height of Above Ground Stories
Pt_stry_b	Part Below Ground Stories
Pt_st_b_h	Height of Below Ground Stories
Pt_rf_t	Part Roof Type (CityGML Codelist)
Pt_gfa	Part Gross Floor Area
Pt_zn	Part Zone
Pt_add_dt	Date of addition to the database
Pt_add_user	User adding part to the database
Pt_Notes	Notes related to building part
Pt_deprec	Part Deprecated (0=current, n = obsolete version)

A.2.3 Building Skin Parts Table Schema

The elements of this table are three-dimensional models. The Multipatch object is an encapsulated data object intended for viewing. The SketchupData object is an editable sketchup model that can be downloaded to sketchup. Many of the building attributes included in massing polygons, are eliminated here, since the heights of these object are explicityly coded in the models, as are the thematic colors.

Column Name	**Description**
Shape	Multipatch
SketchupData	Editable sketchup model uploaded from sketchup
Pt_id	Unique Part ID
bld_id	Building ID
pt_name	Part Name
pt_addr	Part Address
Pt_use	Part Use (Massachusetts D.O.R. Code)
Pt_contct	Part Operator Contact Info
Pt_con_dt	Part Date of Construction
Pt_dem_dt	Part Date of Demolition
Pt_built	Built or Unbuilt (designates proposed building parts)
Pt_priv	Viewer Privileges (Public or Other)
Pt_add_dt	Date of addition to the database
Pt_add_user	User adding part to the database
Pt_Notes	Notes related to building part
Pt_deprec	Part Deprecated (0=current, n = obsolete version)

A.2.4 Schemes Table

The Schemes table registers the names of design schemes that can include additions of unbuilt buildings and demolitions of building parts that exist at the specifies time period for a given view query.

Column Name	**Description**
Sc_id	Unique Scheme ID
Sc_name	Scheme Name
Sc_desc	Scheme Description

A.2.5 Scheme Parts Table

This table registers building parts that should be rendered or not for each scheme. Because of the rules for creating Unique IDS for Massing Parts and Skin Parts, discussed above, assures that any part will have a unique ID, any type of part may be registered in this single table

Column Name	Description
Sc_id	Foreign Key to Schemes Table
Pt_id	Foreign Key to Massing Parts Table
Render	Should Part be rendered in this scheme (Yes/No)

A.3 Scene Retrieval Queries:

Retrieving the appropriate building elements from the Model Management system is achieved with a series of SQL Queries that generate a view of the area of interest

A.3.1 Query Interface

From the user’s perspective, this is accomplished by simply typing in the Date to be reflected by the scene, and optionally, the name of the Scheme to be applied and the Permissions Group that applies to the scene. The Scene Retrieval tool is an interface for a three queries in structured query language (SQL.) The full SQL for these queries is given below.

A.3.2 Query Logic

The logic behind the Scene Retrieval Tool follows these three steps:

1. **Select Building Part Skins:** Buildngs are selected from the Building table based on their dates of construction and demolition. The resulting set of building unique identifiers is used to select the appropriate parts from the Building Part Skin Models table. If the Constructed or Demolished dates of any of the Buildng Skin Parts reflects that that part did not exist during the specified time period, it is deselected. The result of this query become the Building Skins View. This view is not actually a dataset of its own, it is simply a view of selected data from the repository.
2. **Select Building Part Polys:** The same logic described in step 1 is applied to the Building Part Polygon table with the additional step of deselecting Building Polygon Parts that correspond with buildings that found matches in the Building Skin Parts query. The results of this query become the Building Polys View
3. **Select Building Skin Poly Substitutes:** the two queries discussed above select a complete model of the city representing a specific time period and scheme. If the model is very big, and if we have a portrayal system that is capable of substituting low detail models for high based on the distance from the view focus, it will be helpful to have a table of polygon-based massing models to substitute for the

building skin parts when they are far away. The result of this query is the Skin Substitute Polys View.

A.3.3 Query: Select Detailed Building Skin Model View

```
Create View Skin_Parts_View AS
Select * From Skin_Parts
Where
([skin_parts.bld_id] = [buildings.bld_id] )
AND (
  (([buildings.bld_con_dt] IS NULL AND [buildings.bld_dem_dt] is NULL) OR
  ([buildings.bld_con_dt] <= %year%  AND [buildings.bld_dem_dt] > %year% ) OR
  ([buildings.bld_dem_dt] > %year%  AND  [buildings.bld_con_dt] is NULL) OR
  ([buildings.bld_con_dt] <= %year% AND [buildings.bld_dem_dt] is NULL ))
  AND
  (([skin_parts.pt_con_dt] IS NULL AND [skin_parts.pt_dem_dt] is NULL ) or
  ([skin_parts.pt_con_dt] <= %year% AND [skin_parts.pt_dem_dt] > %year%) or
  ([skin_parts.pt_con_dt] <= %year% AND [skin_parts.pt_dem_dt] IS NULL) or
  ([skin_parts.pt_con_dt] IS NULL AND [skin_parts.pt_dem_dt] > %year%))
  AND
  (( [buildings.bld_priv] = %permissions% or [buildings.bld_priv] = "Public") AND
    ( [skin_parts.pt_priv] = %permissions% or [skin_parts.pt_priv] = "Public" ))
  AND
    ([skin_parts.pt_deprec] = 0 )
  AND
   ( [buildings.bld_built] = "Built"  AND
     [skin_parts.pt_built] = "Built"  AND
     [skin_parts.pt_id] NOT IN
        ( select [scheme_skin_parts.pt_id]
          from scheme_skin_parts
          where
           ([scheme_skin_parts.sc_id] = %scheme%
          AND [scheme_skin_parts.render] = "No") ))
    OR
      ( [skin_parts.pt_id] IN
       ( select [scheme_skin_parts.pt_id] from scheme_skin_parts
          where [scheme_skin_parts.sc_id] = %scheme%
           AND [scheme_skin_parts.render] = "Yes"))
    )
```

A.3.4 Select Massing Parts View Query

```
Create View Massing_Parts_View as
Select * From Massing_Parts
Where
( [buildings.bld_id] = [massing_parts.bld_id] )
AND (
  (([buildings.bld_con_dt] IS NULL AND [buildings.bld_dem_dt] is NULL) OR
  ([buildings.bld_con_dt] <= %year%  AND [buildings.bld_dem_dt] > %year% ) OR
  ([buildings.bld_dem_dt] > %year%  AND  [buildings.bld_con_dt] is NULL) OR
  ([buildings.bld_con_dt] <= %year% AND [buildings.bld_dem_dt] is NULL ))
  AND
  (([massing_parts.pt_con_dt] IS NULL AND [massing_parts.pt_dem_dt] is NULL ) or
  ([massing_parts.pt_con_dt] <= %year% AND [massing_parts.pt_dem_dt] > %year%) or
  ([massing_parts.pt_con_dt] <= %year% AND [massing_parts.pt_dem_dt] IS NULL) or
  ([massing_parts.pt_con_dt] IS NULL AND [massing_parts.pt_dem_dt] > %year%))
  AND
    ([massing_parts.pt_deprec] = 0 )
  AND
   (( [buildings.bld_priv] = %permissions% or [buildings.bld_priv] = "Public") AND
    ( [massing_parts.pt_priv] = %permissions% or [massing_parts.pt_priv] = "Public" ))
  AND
   ([massing_parts.bld_id] not in
```

```
      (select [tmp_skin_parts_table.skin_parts_bld_id] from skin_parts_view))
   AND
     ([buildings.bld_built] = "Built" AND
      [massing_parts.pt_built] = "Built" AND
         [massing_parts.pt_id] NOT IN
           (select [scheme_massing_parts.pt_id]
            from scheme_massing_parts where [scheme_massing_parts.sc_id] = %scheme%
            AND [scheme_massing_parts.render] = "No") )
    OR
       ( [massing_parts.pt_id] IN
          ( select [scheme_massing_parts.pt_id] from scheme_massing_parts
            where [scheme_massing_parts.sc_id] = %scheme% AND
[scheme_massing_parts.render] = "Yes"))
  )
```

A.4 Portrayal of Views

A viewer for large complex three dimensional models should be able to portray georeferenced model elements and control their visibility based on the point of view of the observer. This will allow the use of very large and detailed three dimensional models, and overcome some of the scalability problems with traditional three dimensional modeling programs. There are a few viewers on the market with this capability, including ESRI's ArcGlobe, Google Earth, and Multigen Paradigm. Unfortunately, each of these viewers is idiosyncratic with regard to data formats and exchange protocols; nevertheless, the following conceptual capabilities apply across each of them.

The viewer connects directly to the database, by means of the view retrival filters that have been defined with the Scene Retrieval tool. The viewer should be able to download data from the database as needed based on the distance of each feature from the focus of the viewer. The Massing Poly models, and the Skin Substitute Poly models are viewable from the furthest distance (e.g. 4 Kilometers). When the focus of the view becomes near to (e.g. 1 Kilometer) a building that is portrayed with a Building Part Skin Model, the viewer turns off the extruded polygon substitute and turns on the object from the Skins View layer.

A.5 Use of the Building Model Management System to Export Site Models to Sketchup

Creating new skin models for the model management system requires beginning with a georeferenced base model in Sketchup. This model must be exported from the GIS with a terrain model and aerial photograph that will serve as a reference framework for locating the new model in X,Y and Z. This is accomplished with the Sketchup exporter tool. The exporter will export the whatever extruded massing parts and the Skin models that are selected. It will also export a piece of an aerial photo and a TIN model of the terrain if they exist in the current map document. IN order to keep the size of the image and TIN model of manageable size, the following steps are recommended for exporting data from ArcGIS to sketchup.

1. Plan to work on no more than one city block at a time in sketchup

2. Zoom your view in ArcGIS to cover just the block of interest and the a piece of the surrounding blocks, not much more.
3. Turn off all layers except for the aerial photo and whatever other layers you want incorporated into your reference image.
4. Use File->Export Map, to export a Jpeg image of your current view. Open the Export Options and choose “Create World File” and also adjust the pixel resolution so that your dimensions are around 1000 x 1000.
5. Export the jpeg, and open it back into your map.
6. Double-click the MMS_Create TIN tool from the Brookline MMS toolbox.
7. Set your Analysis Extent to be “Same as Display”
8. Choose a location for your output TIN. TINS cannot be created in geodatabases.
9. Click OK to create a new TIN model
10. Now follow the instructions for the sketchup export plugin to see how to export your data to sketchup. We recommend that you simply extrude the Massing polygons to their Roof_Elevation and let the building feet remain at Zero.

In sketchup, you can assign the image to a projected texture and map it to the TIN. It is recommended that modeling be done on the flat image, and then building parts, once they are created, should be moved vertically up to establish their height. Once you have modeled a building part and wish to export it to a new ArcGIS featureclass. Then you can use the ArcGIS Data-Management->General->Append wizard to append your new feature to your Skin_Parts table.

A.6 Populating the Model Management System

Population of the model management system is expected to begin with the migration of a lot of massing polygons that are stored in municipal planimetric databases, or incorporated from the State’s LIDAR survey. Georeferenced 3d Skin models may be have been collected from submissions from the design review process, or taken from on-line repositories such as the Google Sketchup 3d Warehouse. In any case, and no matter what database management system is used, the following principles regarding the assignment of building IDs, and the granularity normalization of building models and parts will be important to understand when populating the model initially and on an on-going basis. Examples illustrating the tools and procedures used to populate the database for the Town of Brookline are provided in Annex 6.

A.6.1 Assign Territory ID

Because the model management system is intended to allow aggregation of information across town boundaries, it will be important to prepend the building unique identifier with a code designating the city or other administrative territory.

A.6.2 Assignment of Building IDs

Unique building identifiers are important for the model management system in order to associate building information, such as building name or building operator contact information with the geometric representations of building parts. Some cities may have unique building identifiers established. Where these exist, the building identifiers are probably already associated with building polygons and other useful information, such as permitting databases, etc, and so it is no doubt propitious to use these as the unique building identifiers in the Model Management System.

In cities that have not established building unique identifiers, it will be useful to develop an identification system based on the property tax assessment system. This has advantages of being exhaustive and, though not perfect, it provides an initial idea of the Owner, Use, and Construction Date of the buildings. The unique parcel identifier can serve as a base upon which additional digits can be appended to discriminate among multiple buildings in the same parcel if needed, as described below. This system provides an easy way to assign building identifiers and building part identifiers that are assured to be unique, even if one is working with an off-line subset of elements from the model.

A.6.3 Building and Building Part IDs

Since there is often more than one building per parcel, each building can have its own number that is appended to the parcel ID. We insert hyphens between the territory ID and the parcel ID and the building number to eliminate possible ambiguous situations that can occur if one of these ids is of variable length. The same system is used to number the building parts in each of the building parts tables. This system of IDs yields a hierarchal ID for each building part from which the territory, parcel, and building can be identified. The building part ID “Brk-11027-0-0” represents the first part of the first building in Brookline parcel 11027.

A.6.4 Lumping and Splitting Strategies

Wholesale assignment of building IDs based on Territory and Parcel will inevitably lead to errors. Luckily the errors are predictable, and their consequences can be either lived with, or easily detected and fixed where they make a difference. Where there is more than one building polygon on a parcel, there is not an automated way to tell if these polygons are parts of a single building or if there are indeed two distinct buildings on the parcel. Therefore, all of the building parts occurring in a single parcel, these model elements are “Lumped” together as a single building. Since individual building massing parts can have their own height information, the impact of this lumping will have no visual impact on views of the model repository may not make any difference. If more information is available that may distinquish building parts as individual buildings, then the building would be “Split”, first by defining the distinct building with a new Unique Building Identifier and row of attributes in the Building Info table, and reassignment of the appropriate Building IDs and Building Part IDs in the Building Parts table.

Another aspect of lumping that may occur in the wholesale assignment of Building IDs from parcels will happen in the common event of a single Building Part that straddles several parcels. This happens because the parcel boundaries are usually not considered when building footprints or skin models are constructed. Again, these errors can typically be lived with until there is particular attention focused on a set of parcels and buildings. When this occurs, the same sort of splitting discussed in the previous paragraph is performed.

A routine for wholesale assignment of building IDs to existing collections of building massing polygons and building skins would begin by deciding which layer has the coarsest level of granularity. One layer or another will typically have a larger-grained idea of what a building is. For example, the MassGIS Massing Polygon layer often lumps many buildings together into one polygon. So we begin with this one.

1. Prepare a property Parcels layer with a Territory ID that identifies the town.
2. Prepare the Massing Parts with a unique Polygon ID
3. Derive label points for each Massing Part
4. Use an overlay function and a join to move a parcel ID and a Territory ID to each Massing Part
5. Generate a new field for building massing part number, which will be populated with all zeros
6. Create a new field for Building Massing Polygon Unique ID that concatenates the territory identifier and the parcel ID and appending a “-0” to form the initial Building Part Identifier .

Step 4 will undoubtedly include some errors, particularly where one building polygon overlays more than one parcel. These can be ignored until a particular building polygon has to be split. At this point all of the building parts for a specific building have the same building part ID (which should be 0) and it should have Null for the construction and demolition dates. These part-specific dates and IDs only need to be updated when specific information about a particular building part becomes important for some model view or other.

Since in our example, the building polygons are generally coarser-grained than the building skin models, a similar approach can be taken with the skins table as follows:

1. aggregate the building part skin models into a single table
2. derive label points for the building skin parts
3. Overlay the skin part label points with the building polygons so that a building ID can be assigned to each skin model.
4. The attributes of the label points, now including the building ID, are now moved to the skin models, and each skin part unique identifier is initialized by appending a “-0” to the Building ID.

A.6.5 Managing Granularity and Normalization

The workflow for adding skin models to the repository will inevitably lead to some Lumping of the building skin models, potentially assigning several models that are clearly separate buildings to a single building ID assigned to the overlapping massing polygon. At first, this may seem like a bad strategy, since a finer association of Skin Parts and Parcels_IDs could be achieved without first lumping to Massing Polys. But the important thing is that there be a distinct relation between each Skin Model and a Massing Model through a common Building ID so that the view creation queries will turn off the appropriate polygon for each skin model.

The critical issue to consider with regard to granularity of building part models is that in order for the View Retrieval queries to perform correctly, there has to be normalization between the Building Unique Identifiers assigned to the Building Part Massing Models and the Building Part Skin Models. For example, if a row house exists as a single polygon in the Massing Parts table, and separate parts in the Skin Parts table, each one of the skin parts should reference the same single building unique identifier that has been associated with the large massing polygon. This way the substitutions will behave appropriately. If one section of the row house is distinguished in terms of its construction or demolition dates, its building name, use, or any other of the building attributes, then the larger element from the Massing Polygon Part table will need to be divided and appropriately assigned to the same distinct building ID as the coinciding Skin Part Model. This splitting of Buildng Massing Polygons will be a situation that needs to be checked whenever new Skin Models are added to the repository.

A.6.6 Creating a Scheme

When there is a need to portray a combination of buildings answering to a specific time-period, but with some variation, such as the portrayal of building parts that are proposed, or the removal of buildings that might perhaps be proposed for modification, this is achieved by defining a new record in the scheme table, to record the name of the scheme and its permissions for viewing. Then the massing or skin model parts that are called for addition are selected, and their building part IDs are recorded in the appropriate scheme part table. This can be accomplished with a simple tool or script.

A.6.7 Deprecated Building Part Models

IN the course of maintaining the model repository, it is inevitable that skin models and massing models will be improved. Rather than calling for older building models to be deleted from the repository when this happens, each element in the model has an integer attribute that can be incremented if the element is not to be rendered as a default object for its applicable date range. These models can be selected for special rendering by including them in a custom scheme. The value for the Deprecated attribute is initialized at zero, which means ‘Use Me.’

Annex B Extensions to the Building Model Management System

There are many reasons to be concerned with the management of building data beyond the singular purpose of creating visualizations of the cities buildings at current time periods. Many aspects of the town's business involve managing information about buildings already, and so the first extensions of the system proposed here, should probably be considered before the initial tables are made, in order to understand how the tables proposed here integrate with existing systems. Next we may be able to anticipate extensions to the town;s building information model that may come in the near to longer-term future. These extensions are discussed below.

B.1 Adding an Addresses Table

In our building ideintifiers table we have one field for the address of the building. This field is initially populated with an address from the Property Parcel's database. In fact, a building may have any number of addresses (e.g. all of the shops in a row of shops in a strip development.) This one-to-many relationship between buildings and addresses should be handled with an additional table. When this is done, the address field in the Building Identifiers Table should be altered to hold a foreign key to the Addresses Table.

There are a couple of schemas that have been developed for addresses, including one by the Federal Geographic Data Committee. A town-wide user needs assessment of existing business practices and resources related to addresses should be conducted before creating this table.

B.2 Create and Maintain a Town Wide Ground Cover Layer

Currently the Model Management system assumes an existing terrain model and orthorectified aerial photos to convey ground conditions. The work of Green Mountain Geographics has shown that while aerial photos are better than nothing, as a means of understanding what is on the ground, in between the buildings, it is much better to have a thematic GIS layer that distinquishes different sorts of pavement and planted areas with patches of consistent color. We would call this a ground cover map. A very worthwhile extension of the town's three dimensional modeling system would be to develop this ground cover map based on its existing planimetric survey information. This planimetric data includes edge of street, and back of sidewalk. A toolset has been provided with the data CD that combines these layers and makes topological inferences to make an initial ground cover layer for the city. This work is exploratory and incomplete at the moment.

B.3 Terrain Model Management System

Terrain is, of course, a very important aspect of understanding the three dimensional context of places. Raster terrain models of various levels of quality are available from the U.S. Geological Survey. provided by MassGIS provides an exceptionally good terrain model for the entire state as a raster with a horizontal resolution of 5 meters, which is good enough for rough visualization and positioning of buildings. This raster is included on the data CD and is used in the 3d visualizations accessed through the Brookline_ArcGlobe.3dd arcglobe document.

Brookline and other communities have much better terrain information that has been gathered by photogrammetric surveys. These surveys typically result in contours with roughly 2ft interval, and information about the edge of pavement. This data can be used to create very detailed terrain models that include details such as curbs and grade separations. These terrain models would take the form of Masspoints and Breaklines that would be used to generate faceted surfaces also known as Triangulated Irregular Network (TIN) models. The system will divide the city into patches so that edge-mapped terrain models can be exported to Sketchup or another 3d editor for elaboration and application of textures, and may be imported back into ArcGIS as multipatch terrain skin models, which can become a new component of the model management system. These terrain models should be articulated and colored based on the Ground Cover layer described in the previous section.

B.4 Extension to Interior Spaces Model

The Building Model Management System described in this study is concerned only with the externally visible parts of buildings. There is a substantial amount of information about how buildings are used and this information varies potentially according to individual spaces within each building. It is not likely that this information will be gathered for every building in the city, but certain buildings may be modeled this way, such as Schools or other public buildings, and perhaps other high-occupancy structures.

The Building Information Management system has been designed to extend or be extended by the interior spaces in a GIS data model has been explored by a Penobscot Bay Media.[1] This model creates GIS objects for interior spaces that can be used in GIS queries and analysis. The space outlines are more or less 2-dimensional floorplans, but because each space outline knows its elevation at top and bottom, it can be used to create extruded models for visualization. The interior space models would become another level of detail alternative to Skin Models or Massing Models, and could be thematically mixed and matched.

We recommend that the Town wait for authoring tools, exchange formats and database import routines to be developed to facilitate the creation and exchange of these data from a CAD program and the GIS data model before integrating interior space models with the GIS.

[1] GIS Model for Interior Spaces; Penobscot Bay Media, 2007; http://www.penbaymedia.com/services.cfm?callname=gis_datamodel

Annex C Contents of Data CD

The CD provided with this report contains the data, tools and arcmap documents necessary to demonstrate all of the features of the Model Management System Schema, including the data and tools necessary for creating an initial version of the MMS schema from the Brookline and MassGIS layers as they exist today. This section gives a brief overview of the contents of that CD.

The top folder on the CD is named Brookline_mms. Within this folder you will find this document, and a powerpoint presentation and two other folders:

Buildings_mms: contains an arcmap document, and an arcglobe document that open up the data and tools that demonstrate the Buildng Mosel Management System. The toolbox, MMS_Views.tbx contains the Scene Retrieval Tool, and the toolbox, MMS_Prep_Buildings includes the tools for creating the model management schema from the base brookline layers. The Geodatabase, MMS_Brookline.mdb contains the working schema template, and the original brookline and MassGIS layers used to creae the schema.

Terrain_mms: contains the ArcMap document and tools for creating the terrain model management schema. This work can be looked at and demonstrated by opening the ArcMap Document included and the tools in the Prep_Brookline_terrain toolbox.