CAA 2009 Williamsburg

Models for Complex Spatio-temporal Relationships and their Implementation using Open Source Components

Riyaz Fazal

Recording Heritage Network, Berkeley, California, USA.

Abstract

Spatial and temporal aspects of an object, site, area, region, event, age, era or epoch have always been important in historical or archaeological documentation or description. The visualization of spatial dimensionality is well modeled by geospatial systems. Timelines are used for visualizing serial progression along the temporal dimension. It is often the linkage between the temporal and spatial dimensions that is more illuminating. Functionality for visualization of spatio-temporal relationships is not available in standard geospatial information systems (GIS). A few systems that model and present spatio-temporal relationships exist currently. TimeMapTM, KML the standards-based format for the data displayed on Google Earth, has been extended so that spatial objects may have temporal attributes and Timeline, an open source tool visualization of complex, nested timelines, are briefly reviewed.

The purpose of this paper is to describe the data and relationship model that enable substantially complex associations between temporal and spatial entities. Modeling temporal aspects as attributes that are subordinate to spatial objects is restrictive. The described model makes possible temporal entities with spatial attributes or spatial entities with temporal attributes. This is achieved by use of a meta relationship scheme, enabling definition of relationships that are appropriate to the subject being described.

The second purpose is to describe the use of open source tools for implementing the data model, the data input interface, and applications for visualization. The functionality and general availability of these tools will encourage further advancement of the model and development of visualization applications.

***Key words**: Spatio-temporal, Data Model, Open Source Software*

1 Introduction

Description and documentation of historical, archaeological and heritage entities have complex spatio-temporal attributes. Furthermore, these attributes are interrelated.

Most models in current use have a simple one-to-one relationship between spatial and temporal entities. A flexible scheme that can model complex spatio-temporal relationships is necessary. Such a scheme is described.

The data model is independent of any application to visualize or view the spatio-temporal relationships; it is intended to be of general purpose and for use in multiple independent applications.

The use of open source frameworks for building applications based on the data model is also described. These tools make the development of web-based applications that are rich and complex, are freely available, and in many cases reduce cost and time of development.

2 Review of Existing Tools

TimeMap

TimeMapTM[1] is a system that records and delivers time-stamped spatial data. It is made up of a Windows-based data and map preparation application and a Java-based application to deliver

[1] Ian Johnson, "Putting Time on a Map," GeoInformatics, July/August, 2004.

the visualization of the dynamic spatio-temporal relationships.

TimeMap can be embedded in web pages and provides an editor for creating the and importing data. It was developed before the explosion of Geoweb tools and API. Geoweb tools make possible the complex visualization of spatial entities with other data entities. Geoweb (or Geospatial Web) is term used to describe the merging of geographical information with other information on the Web, and the tools that enable this merging.

KML and Google Earth

KML, the standards-based format that is used for data displayed on Google Earth has been extended so that spatial objects may have temporal attributes[2]. For data containing the time attributes (time-span or timestamp) the Google Earth application displays a time slider control for traversing the data through the timeline. The temporal attribute is a property of the primary geospatial object.

SIMILE Timeline

Timeline is an open source tool developed by the SIMILE project for the dynamic visualization of complex timelines[3].

Timeline does not have any spatial attributes. Visualizations that integrate such complex timelines with spatial objects would provide enhanced tools for describing historical and archaeological subjects.

3 Data Preparation and Visualization

The systems architectural basis for the model is separation of the functions of data preparation and data visualization (see fig. 1). Preparation is the function of the data content provider, usually a content expert. Visualization is functionality targeted to a larger population that consists of both professional and non-professionals.

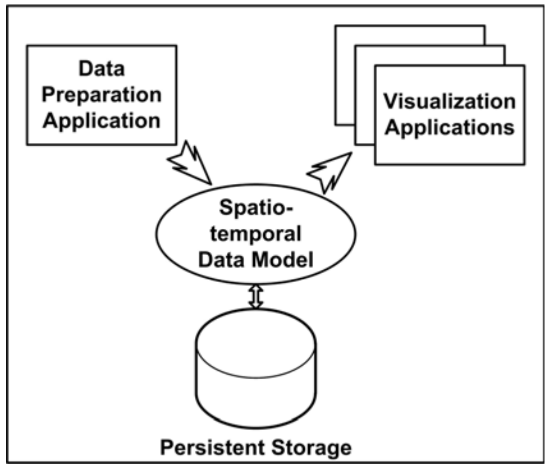

Figure 1. Architectural Basis for Data Model.

This architecture has several benefits. With the modular structure, functionality is separated into well-defined components that are easier to develop, maintain, disseminate and use. In addition, the well-defined data structure makes it possible for multiple, independent projects to develop visualization applications.

4 Model for Spatio-temporal data

The most important aspect of the data model is the representation of complex relationships between entities. The Object-Role Modeling method is useful for representing entities and relationships of an information system at a conceptual level.[4]

Object-Role Modeling must not be confused with Object Relational Mapping that is a technique for mapping and converting data between software object types and relational database management systems.

Object-Role Modeling (ORM) provides a simplified way to represent the application domain by using natural language and diagrams. There is

[2] Google, "Time and Animation," KML Developer's Guide, http://code.google.com/apis/kml/documentation/time.html.

[3] Massachussetts Institute of Technology, "Timeline, Web Widget for Visualizing Temporal Data," SIMILE, http://www.simile-widgets.org/timeline/.

[4] Terry Halpin, 1998, "Object-Role Modeling (ORM/NIAM), In Handbook on Architectures of Information Systems," ed. P. Bernus, K. Mertins & G. Schmidt, (Berlin: Springer-Verlag, 1998).

well-defined notation for creating ORM diagrams and the notational elements used here are summarized in figure 2.

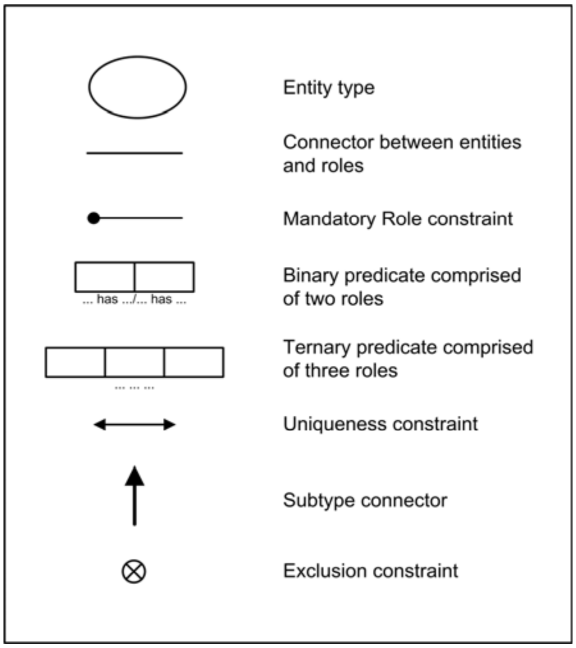

Figure 2. Object-Relational Model Notation used in the Spatio-Temporal Model

The model has three primary entities or objects: subject, temporal and spatial.

The subject entity is the topic being studied or represented. A sample list includes:

- Empires (e.g., Mughal empire, Aztec empire),
- Reigns (e.g., reign of Babur, reign of Ahuitzotl),
- Monumental sites (e.g., Gur-e Amir complex—tomb of Timur, Uxmal complex—Maya city),
- Buildings (portal to the Gur-e Amir, Pyramid of the Magician at Uxmal),
- Journeys (e.g., journeys of ibn Battuta, journey of Lewis and Clark),
- Lives of People (e.g., Chingiz Khan, Moctezuma II).

In figure 3, the subject entity has associations with temporal entities, spatial entities and other subject entities. Temporal entities have associations with spatial entities. Subject entities also include other information such as Core Data as defined by by the European Union,[5] and subject specific attributes such as architectural elements for monumental sites and buildings.

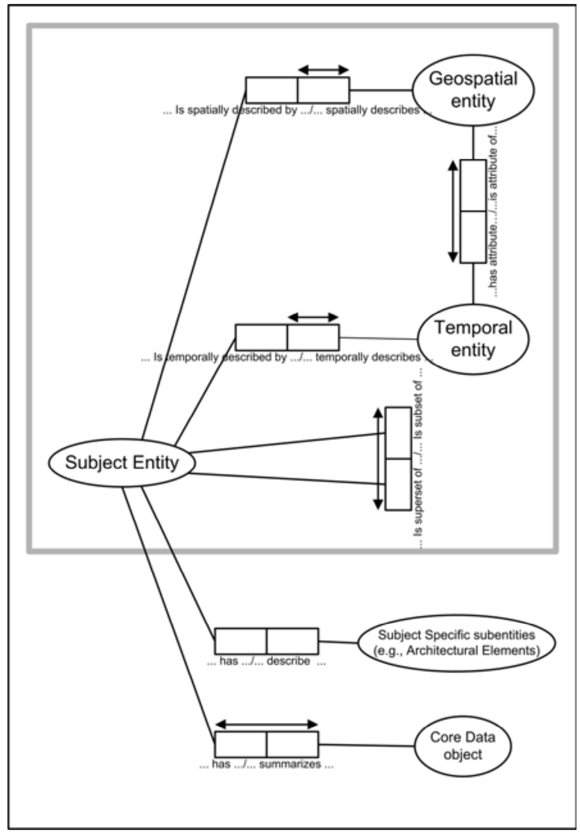

Figure 3. ORM diagram showing Relationships of the Subject Entity

The focus is on spatio-temporal modeling and the focus will be on the entities and associations in the gray rectangles in figure 3.

The temporal entity defines the mark in time of a subject entity. The time attribute can be either:

- specific date in time (e.g., October 12, 1492), or
- date period or span (e.g., 1526-1531).

5 Council of Europe, Committee of Ministers, "Recommendation No R (95) of the Committee of Ministers to Member States on Coordinating Documentation Methods and Systems Related to Historic Buildings and Monuments of Architectural Heritage," 1995.

A subject entity can be associated with multiple temporal entities (e.g., Mughal emperor Humayun's reign was from 1531 to 1539 and then 1555 to 1556). However, a temporal entity is associated with only one subject entity. The temporal entity has additional attributes of calendar used to define the date, resolution of date (year, month, day) and quality of data (accurate, approximate, very approximate). Figure 4 shows the temporal entity associations in an ORM diagram.

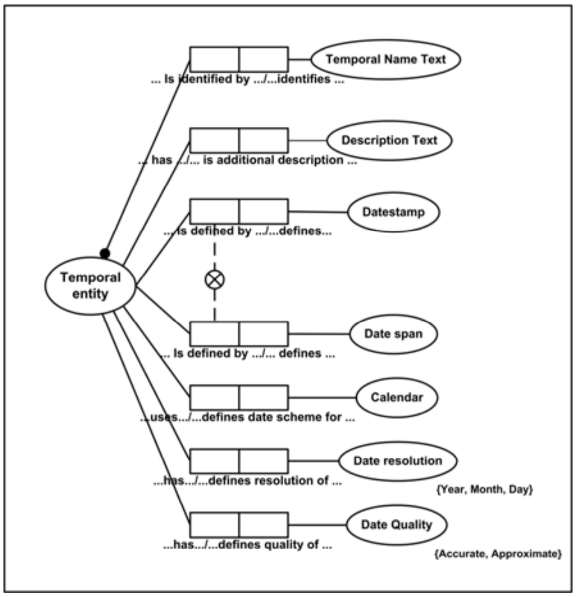

Figure 4. ORM Diagram for Temporal entity.

The spatial entity defines the geographic aspect of a subject entity. The spatial entity can represent areas (polygon), paths (linestring) and points.

The spatial entity can be an attribute of both the subject entity and the temporal entity. For example, during the reign of the Mughal emperor Akbar, Fatehpur Sikri was the capital city (subject entity) from 1570 to 1585 (temporal entity) and is located at 27.09466˚N 77.66278˚E (spatial entity). Figure 5 shows the ORM diagram for the spatial entity.

A subject entity may also be associated with another subject entity. It may be a subset, superset or be connected to (or intersects with) another subject entity. Akbar's reign, for example, is a superset of the Faterhpur Sikri capital city entity. The reign of Sher Shah Suri (subject entity) is connected with the reign of Humayun (subject entity)—he displaced the Mughal emperor from his kingdom for a period of fifteen years before Humayun was able defeat the Suri dynasty and return from exile—but the temporal attributes of these subject entities do not overlap.

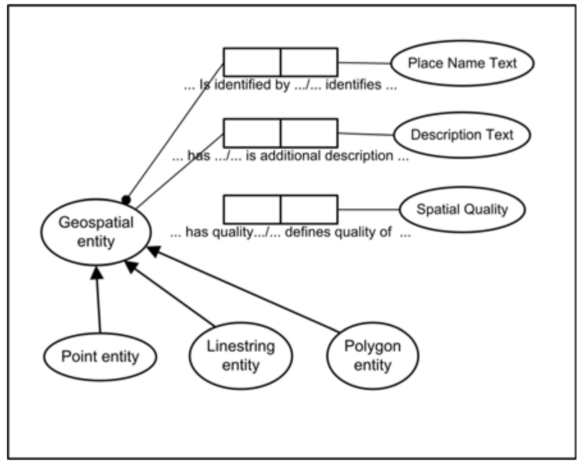

Figure 5. ORM Diagram for Spatial entity.

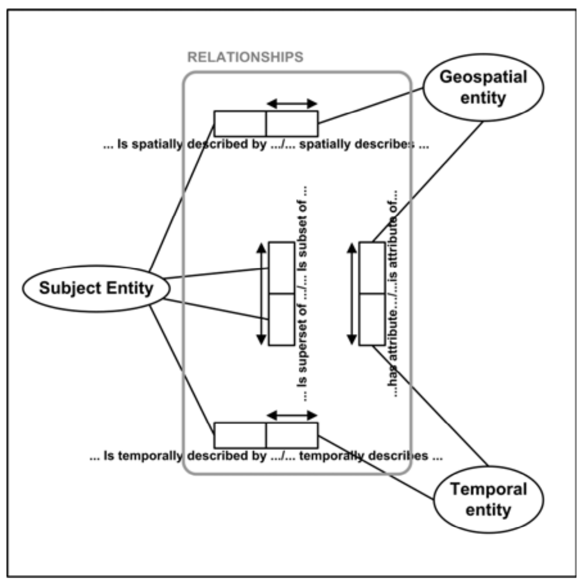

Figure 6. ORM Diagram for Spatio-temporal Associations.

Spatial entity ORM diagram is shown in figure 5. The base entity is a basic geospatial object that is inherited by subtypes Point, Linestring and

Polygon. The spatial entity also has an attribute expressing the quality of the data (accurate or approximate).

Figure 6 show just the three primary entities of the spatio-temporal model, and their relationships.

It shows most of the associations between these major entities (inside the gray rectangle, see fig. 6), illustrating the complexity of defining relationships on a case-by-case basis.

To overcome this complexity a rationalized model was defined using Relationship and Meta Relationship entities (see fig. 7).

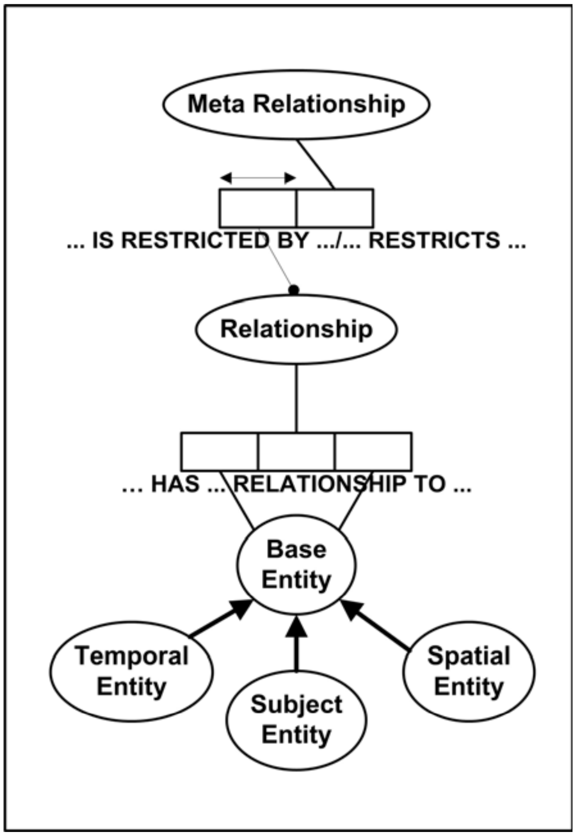

Figure 7. Rationalized Spatio-temporal Data Model

First the primary entities (subject, temporal, spatial) are defined as subtypes of a base entity. It is now possible to define associations between any two base entities using a Relationship entity. This flexibility could allow uncontrolled and illogical associations but for the presence of the Meta Relationship entity.

The Meta Relationship entity defines the allowable relationships between primary entity types. Table 1 shows the allowable relationships. Note that the spatial entity is an attribute of the temporal entity. Converse relationships are also listed as both relationships are defined and created in the dataset. The listed relationships support the necessary associations for historical and archaeological studies.

	Entity type	Allowed Relationship	Entity type	Converse
1.	subject	is subset of	subject	2.
2.	subject	is superset of	subject	1.
3.	subject	intersects with	subject	3.
4.	temporal	temporally defines	subject	5
5.	subject	is temporally defined by	temporal	4.
6.	spatial	spatially defines	subject	7.
7.	subject	is spatially defined by	spatial	6.
8.	temporal	is subset of	temporal	9.
9.	temporal	is superset of	temporal	8.
10.	temporal	has spatial attribute	spatial	11.
11.	spatial	is a spatial attribute of	temporal	10.

Table 1. Allowed relationships for historical and archaeological studies.

When relationships are created the software logic performs consistency checks with the allowed relationships. Inconsistent relationships are disallowed.

Another advantage of the Meta Relationships entity is that allowed relationships can be modified to the requirements of the application domain. Any meta change must be logically consistent.

5 Model Implementation and Open Source Software

One goal of the data model is to have independent applications to prepare and to present or visualize the data.

To create the application to prepare the data Django, a high-level Python Web framework was used.[6]

Django uses Python's object-oriented capabilities to define data models. It converts the object model to an appropriate database schema.

Django works with MySQL and PostgreSQL relational database management systems. As spatial components are an integral part of the model, PostgreSQL with PostGIS extension was used.

PostgreSQL[7] is a powerful full-functional enterprise-class system that is SQL compliant and supports a comprehensive list of data types. PostgreSQL is open-source software.

PostGIS[8] is the spatial extension to PostgreSQL to provide support for geospatial objects. It is compliant with Open GIS Consortium (OGC) standards for geometries and functions. It makes possible complex queries and sophisticated spatial operations on these objects. It can import and export most GIS data types.

GeoDjango[9] is an add-on to Django that supports spatial fields and enables spatial queries. GeoDjango works with PostgreSQL/PostGIS and requires additional libraries GEOS and PROJ.4. All these are open-source components with proven records of functionality and performance.

From the spatio-temporal object model defined in Django, the framework creates a database schema. The PostgreSQL database implementation created from the data model is shown as an Entity Relationship Diagram (ERD). To aid clarity, the ERD (see fig. 8) shows only tables related to the primary entities, important fields and relationships.

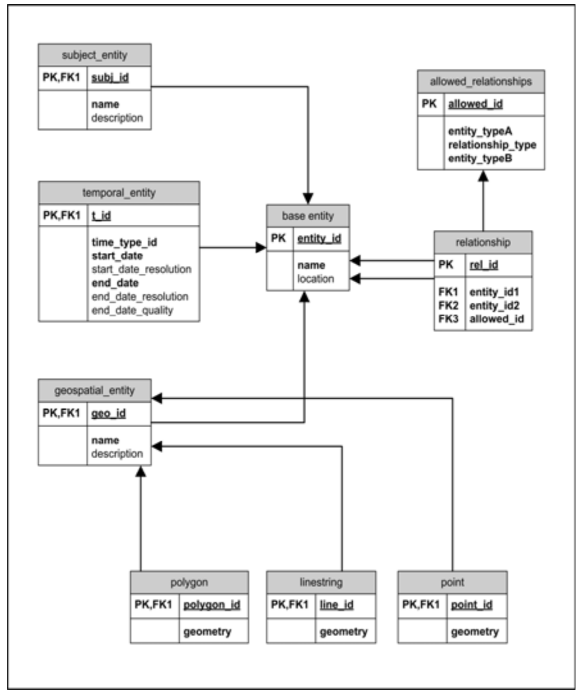

Figure 8. Entity Relationship Diagram for Relational tables.

The Django framework also creates a Web administrative interface that can be used for preparing the data. See figures 9 and 10 for sample interface pages.

Figure 9. Administrative Interface for Allowed Associations.

---

[6] Django Software Foundation, "Django Documentation," http://docs.djangoproject.com/en/dev/

[7] PostgreSQL Global Development Group, "PostgreSQL 8.3.3 Documentation," 2008. http://www.postgresql.org/files/documentation/pdf/8.3/postgresql-8.3-US.pdf

[8] Paul Ramsey, ed., PostGIS Manual. http://postgis.refractions.net/docs/postgis.pdf.

[9] Justin Bronin, "GeoDjango v1.0 Documentation," http://geodjango.org/docs/index.html

For the demonstration implementation of the data model the calendar option was left out. All dates use the Gregorian calendar.

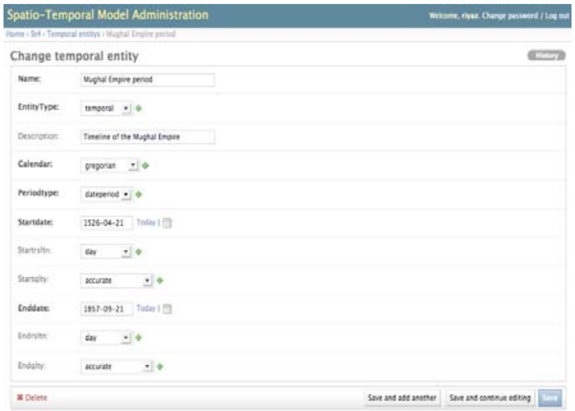

Figure 10. Administrative Interface for Temporal Objects.

With the add-on GeoDjango module the administration interface provides a graphical editor for creating geospatial objects (see fig. 11). For complex boundaries it is recommended that a good GIS system be used.

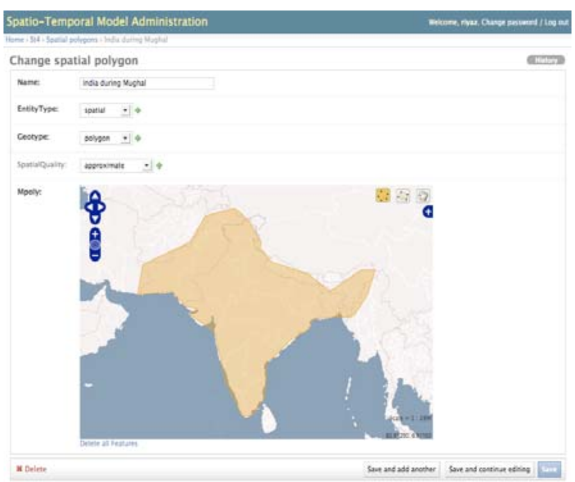

Figure 11. Administrative Interface for Spatial Objects

PostGIS and GEOS provide several utilities to import objects in common geospatial data formats including ESRI Shapefiles, Geography Markup Language (GML) and Well-known Text (WKT).

6 Use Cases

The systems architecture for the model and storage is designed to work with visualization applications on diverse platforms. The applications must be able to access the data in persistent storage. The current formats for persistent storage are PostgreSQL, and extracted datasets in JSON (Javascript Object Notation) and XML.

Use cases are shown here to illustrate the flexibility of the model. They are not meant to be exhaustive.

Export to *Timeline*
With Django and Python the timeline data can be exported to the JSON format support by SIMILE Timeline. The Django framework provides templates in several formats that make possible the output of data in XML, JSON and HTML.

Hierarchical Timelines
With allowed relationships between temporal entities, it is possible to create hierarchical timelines.

For example, the following hierarchy is easily retrieved. Level 2 entities, the reigns of khans, are subordinate to the level 1 entity, the Mongol Empire.

Level 1
Mongol Empire (1206 – 1294)

> ***Level 2***
> Reign of Chingiz Khan (1206 – 1227)
> Reign of Ögedei Khan (1229 – 1241)
> Reign of Güyük Khan (1246 – 1248)
> Reign of Möngke Khan (1251 – 1259)
> Reign of Khubilai Khan (1260 – 1294

The relationship between temporal and spatial entities can be utilized to display spatial entities in hierarchical format as well.

Relationship tracing
With all relationships defined in a single table it is to extract all the associations for an entity.

For example, it enables extraction all associated subject, temporal and spatial entities for a particular subject entity such as the Mughal Empire. The subset subject entities include all the reigns of the Mughal emperors (Babur, Humayun, Akbar, etc.). The associated temporal entities for each subset subject are accessed and the geospatial entities associated with the temporal entities. This extraction can down as many levels as desired to the extent of the data content.

Animation
The complex relationship model enables development of sophisticated animation that reveal for a subject, its temporal and spatial attributes in a dynamic display. The functionality of rich interface applications (open-source) on the Web makes it possible to distribute these applications to a wide audience.

7 Next Steps

The data model and architecture is a flexible foundation for defining and visualizing complex spatio-temporal relationships. The meta relationship entity allows definition of allowable relationships that are relevant to specific data content domains.

Further development includes hosting a public Web based data preparation interface and data access API for the archaeological community. This will run the data model through a variety of content domains and test its flexibility and robustness.

Reference implementations of the visualization applications described in the Use Cases above will also be offered as code samples from which domain specific application may be developed.

Bibliography

Bronin, Justin "GeoDjango v1.0 Documentation," http://geodjango.org/docs/index.html

Council of Europe, Committee of Ministers. "Recommendation No R (95) of the Committee of Ministers to Member States on Coordinating Documentation Methods and Systems Related to Historic Buildings and Monuments of Architectural Heritage," 1995.

Django Software Foundation, "Django Documentation," http://docs.djangoproject.com/en/dev/ (accessed May 21, 2009)

Google. "Time and Animation," KML Developer's Guide, http://code.google.com/apis/kml/documentation/time.html (accessed May 21, 2009).

Halpin, Terry. 1998, "Object-Role Modeling (ORM/NIAM), In Handbook on Architectures of Information Systems," edited by P. Bernus, K. Mertins and G. Schmidt, Berlin: Springer-Verlag, 1998.

Johnson, Ian. "Putting Time on a Map," *GeoInformatics*, July/August, 2004.

Massachusetts Institute of Technology. "Timeline, Web Widget for Visualizing Temporal Data," SIMILE, http://www.simile-widgets.org/timeline/ (accessed May 21, 200)

PostgreSQL Global Development Group. "PostgreSQL 8.3.3 Documentation," http://www.postgresql.org/files/documentation/pdf/8.3/postgresql-8.3-US.pdf (accessed May 22, 2009)

Ramsey, Paul, ed. "PostGIS Manual," http://postgis.refractions.net/docs/postgis.pdf (accessed May 22, 2009)