Modeling and Visualizing Historical GIS Data

Merrick Lex Berman
China Historical GIS, Center for Geographic Analysis
Spatial Analysis Project, Fairbank Center for Chinese Studies
Harvard University

April 2009

Abstract:

Historical GIS is quite a loose term. It can be applied to almost any data with spatial representations that change over time. Here we will greatly narrow the scope of inquiry and look at the specific problem of how to depict changes in administrative geography over the long course of history, and how to model the data in a way that enables us to visualize changing spatial patterns. Two scenarios will be examined for visualization: one, to show the administrative hierarchy from the center of government to first-level and second-level subordinate units; and another which shows single a path of movement across time. In both examples, the base dataset will be the China Historical GIS, and visualization software will be GoogleEarth.

Defining Historical GIS Objects

Many methods have been proposed and implemented for Historical GIS, including the use of Key Dates, Datestamps, Spatio-temporal Composites, Event-Based GIS, and Multivariable Time Cubes, to name a few. Let's take a brief look at these in order to better define Historical GIS objects and how we can manage them. More extensive coverage of these methods is found in a recent volume of Cambridge Studies in Historical Geography by Ian Gregory and Paul Ell. **[Historical GIS Tech]**

Key Dates, sometimes referred to as Time Slices, are collections of spatial objects grouped according to specific valid dates. For each layer of Time Slice data, all of the spatial objects must be represented according to their state at the specified time. This approach is especially useful if most of the objects have some degree of changes that occur in between the Time Slices and if the number of Time Slices are limited.

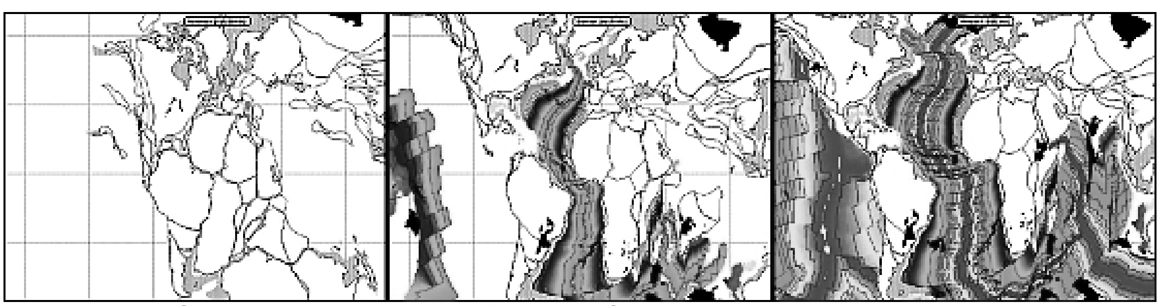

Figure 1: Time Slices showing the breakup of Pangea (Scotese **[Paleomap]***).*

However, if Time Slices must be created for a large number of key dates, the redundancy of creating unique spatial objects for each layer will be a problem, as will record-keeping to track all the objects from one Time Slice to the next. More importantly, if the object changes occur at unrelated times (meaning the objects are asynchronous), then choosing which points of time to depict them in a frozen state becomes hard to justify. A typical scenario involves the creation of Time Slice layers in GIS at regular intervals in order to mesh them with data (such as census) published at those intervals. This results in "fudging" the spatial data in order to represent the state of affairs at particular moments in time. This sort of guesswork applies equally to other methods, of course, but becomes more apparent when sources for the historical geographic features are obviously dated before or after the Time Slice in question, and have been jiggered to fit the key date.

In situations where it is impractical to pin down geographic objects according to specific key dates, models that capture a continuous series of spatial changes can be used, such as Datestamps or Spatio-Temporal Composites. In the case of Datestamps, changes that occur in geographic objects are recorded by inserting new spatial objects in GIS with date attributes. A typical use case would be a boundary change that occurs, resulting in changes of area between two or more adjacent objects. The segment of the original boundary is saved, and the segment showing the new boundary is created, each with it's own valid Datestamp attributes. In this way a query for a particular date will find only the arc segments that existed at that time.

Consider the following example, in which we are going to track three adjacent historical objects during a boundary change by means of Datestamped arc segments. On the left are the objects representing historical counties as they existed at Time 1. On the right are the minimum arc segments used to define the historical boundaries, assuming that a unique segment is required when it forms the boundary between any two objects.

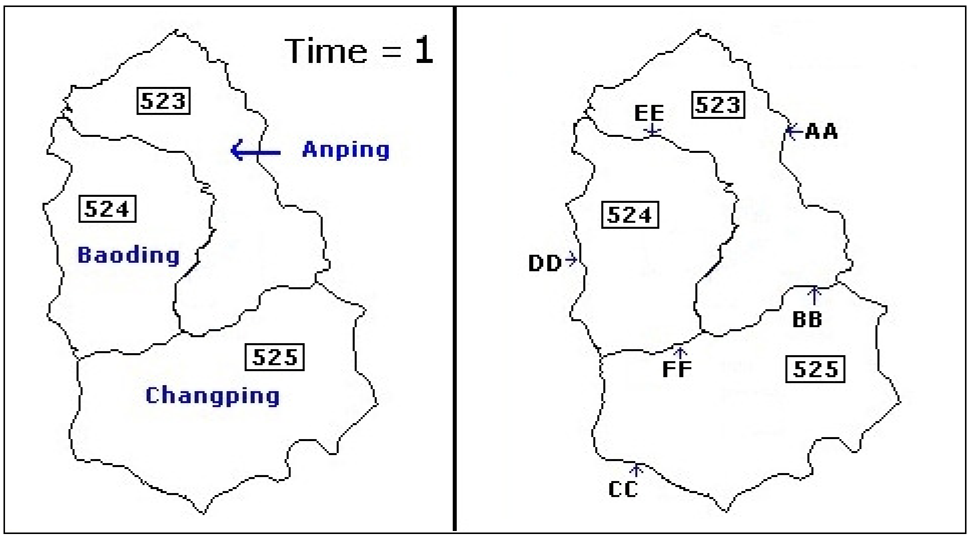

Figure 2: Example of Datestamping arcs to form historical spatial objects

Note that the boundary between objects 523 and 524 requires only one arc segment (EE), while the boundary that runs along the top edge of object 525 must be divided into two arcs (BB and FF) because it serves as the boundary for two different neighbors. The logic behind this is that we must be able to define an object boundary with the minimum number of correct arc segments, therefore, if the arc made up of both BB and FF was a single arc segment (for the sake of argument called BF), we could not correctly define either object 523 or 524, since a large part of the arc BF would be left dangling beyond the edge of either object.

The tables needed to keep track of our original state are shown in Figure 3, where the **arc segment table** is assumed to be the actual attribute table of the arcs in GIS, and where artificial begin and end times are given to make sure that they contain the objects that we are defining. Relational tables include an **historical object table**, in which the objects themselves are given temporal extents and associated with the arcs needed to establish their correct boundaries, and also an **object attribute table** where information related to each object (such as placenames, feature types, etc) can be stored.

arc	begin	end
AA	0	9
BB	0	9
CC	0	9
DD	0	9
EE	0	9
FF	0	9

Arc Segment Table

begin	end	object	arc
1	5	523	AA
1	5	523	BB
1	5	523	EE
1	5	524	DD
1	5	524	EE
1	5	524	FF

Historical Object Table

object	name	type
523	Anping	County
524	Baoding	County
525	Changping	County

Object Attribute Table

Figure 3: Datestamp Relational Tables

Now let us introduce a boundary change. Assume that the historical geographic object identified at Time 1 as Baoding (object 524) expands at Time 4, gaining part of the territory of it's neighbor Anping (object 523). In this scenario, a new arc must be created (arc segment GG) to define the changed boundary, while others must be truncated and renamed as new, smaller segments, as shown in Figure 4.

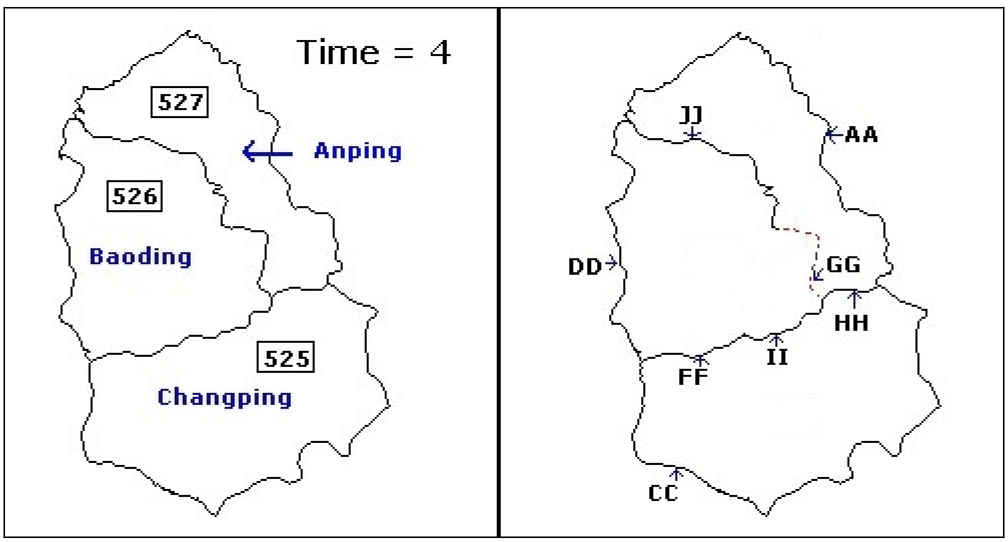

Figure 4: Datestamping Boundary Changes

For example, the segment previously known as BB now only extends as far as the point where it joins the new boundary GG; since the shorter segment is not the same as BB, it must be given a new identity (HH), while the other half of BB which previously extended as far as FF must also be given a new identity (II). The same rule applies the changed boundary EE, which now becomes both GG and JJ. Of the original arc segments, only AA, CC, DD, FF remain unchanged.

All of the correct arc segments which describe our historical objects must, of course, be updated in the associated tables. Note that the original arc segments that had *artificial end times*, must now be corrected with their *actual end times* in order to mesh with the historical objects they define, while new arc segments must be given their correct *begin times*. Another consequence of the change is that two of the original historical objects (523 and 524) have changed, and therefore must be given *actual end times*, and replaced with new unique objects (526 and 527). The changes are reflected in the tables shown in Figure 5.

arc	begin	end
AA	0	9
BB	0	4
CC	0	9
DD	0	9
EE	0	4
FF	0	9
GG	4	9
HH	4	9
II	4	9
JJ	4	9

Arc Segment Table

begin	end	object	arc
1	4	523	AA
1	4	523	BB
1	4	523	EE
1	4	524	DD
1	4	524	EE
1	4	524	FF
4	9	526	DD
4	9	526	FF
4	9	526	GG
4	9	526	II
4	9	526	JJ
4	9	527	AA
4	9	527	GG
4	9	527	HH
4	9	527	JJ

Historical Object Table

object	name	type
523	Anping	County
524	Baoding	County
525	Changping	County
526	Anping	County
527	Baoding	County

Object Attribute Table

Figure 5: Relational Tables for Datestamped Boundary Change

The complexity introduced by a single boundary change demonstrates how laborious the process would be if the study area includes hundreds (or thousands) of objects. When you extend the temporal coverage over decades or centuries, the task becomes mind-boggling. Therefore, it should come as no surprise to realize that the construction of parish level boundary changes in Great Britain over a period of two hundred years took a team of GIS experts more than seven years to complete. **[GBHGIS]**

The preceding example of Datestamping arc segments did not touch upon the subject of topology, which adds yet another level of difficulty to the task. In order for correct topology to be constructed from the constituent arc segments into valid areal units as polygons, elaborate validation routines would need to be created. Alternatively, the storage of arc segments could be abandoned in favor of storing Least Common Geometry polygons, and combining them into historical geographic objects using the Spatio-Temporal Composite method (STC). This does not solve the problem of dealing with topology validation, but rather forces it into the compilation process.

Using the STC method to examine the same boundary change described above, we would first establish the objects as polygons, and then split off the territory from one of the polygons when a boundary change occurs, as shown in Figure 6.

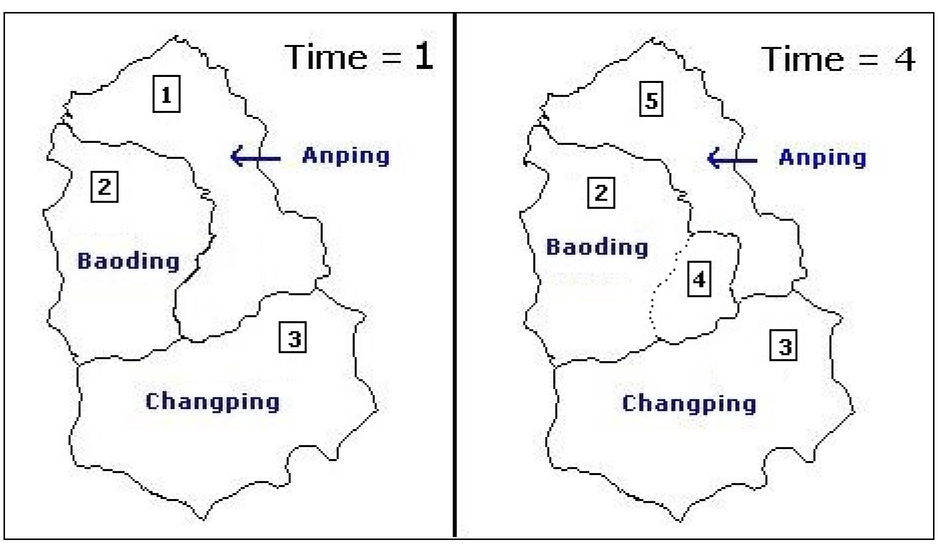

Figure 6: Spatio-Temporal Composite Method of tracking boundary changes

Note that the ID numbers in the STC example refer to the Least Common Geometry [LCG] polygons, not to the historical geographic objects that they represent. In the STC method, unique IDs must be created for all LCG polygons, and this impacts the stored relationships between polygon IDs and the areas we want to represent as historical objects.

As we can see in Figure 6, on the left hand side are polygons 1, 2, and 3, which represent historical geographic objects that existed at Time 1. However, when the change occurs, polygon 1 must be broken up into its LCG components which are assigned new unique IDs 4 and 5. This change then requires modifications to the definitions in the historical objects. Anping County can no longer be defined with polygon 1 at Time 1, but must be redefined as composed of polygons 4 and 5 at time 1, and composed of polygon 5 alone at time 4. By contrast, Baoding can still be defined with polygon 2 at time 1, but after the boundary change Baoding is composed of polygons 2 and 4. Tracking change for the STC method is reflected in the tables shown in Figure 7.

Before LCG Change:

pgn
1
2
3

LCG Table

object	begin	end	pgn
523	0	9	1
524	0	9	2
525	0	9	3

Historical Object Table

object	name	type
523	Anping	County
524	Baoding	County
525	Changping	County

Object Attribute Table

After LCG Change:

pgn
1
2
3
4
5

LCG Table

object	begin	end	pgn
523	0	4	4
523	0	4	5
524	0	4	2
525	0	9	3
523	4	9	5
524	4	9	2
524	4	9	4

Historical Object Table

object	name	type
523	Anping	County
524	Baoding	County
525	Changping	County

Object Attribute Table

Figure 7: Spatio-temporal Composite Tracking Tables

Notice that there is some economy to be found in tracking polygons rather than arc segments, but that very painstaking validations are needed to make sure that the creation of new LCG polygons are properly related to their correct Historical Objects. Before the change was introduced, polygon 1 was able to represent historical object 523 (*ie*, Anping County). However, when it became necessary to split polygon 1 into polygons 4 and 5, the row previously held in the **historical object table** for polygon 1 had to be split into mutliple rows and updated with *correct ending times*.

The origins of the STC date back to Gail Langran's now legendary work, *Time in Geographic Information Systems*, published in 1992. **[TGIS]** In the subsequent decade and a half, many implementations of the STC have been completed, and many strides have been made in the functional logic of how temporal and spatial changes interact **[Claramunt],** but very little has emerged in the way of practical applications that integrate change over time from the major software vendors in the field. The sort of editing and cross-referencing described above remains very much a customized, hand-crafted process, and the basic logic is often modified for particular purposes to such an extent that no generic STC solution has emerged.

Furthermore, one of the major drawbacks in both developing and maintaining Datestamped and STC GIS systems is spatial fragmentation. That is to say, whether the implementation is based on arc segments or on LCG polygons, each time a change is introduced, the fundemental set of GIS objects are fragmented into smaller and smaller pieces. Repetitive boundary changes are reduced to tiny slivers, and the care needed to keep track of which slivers add up into which historical geographic objects is enormous. This leads us to consider whether or not spatial fragmentation can be minimized or even dispensed with, either by storing each instance of historical geographic objects as unique spatial objects in the same data layers, or considering a more radical approach: dispensing with polygons in favor of networks!

Spatial Fragmentation vs. Temporal Fragmentation

As a case study, let us consider the main objective of the China Historical GIS Project, which is to develop a base GIS and database for all known administrative units from the founding of the first Chinese Empire (Qin, 221 BCE) to the end of the last Dynasty (Qing, 1911 CE). In addition to representing each of these historical administrative units (from the Imperial Capital down to the County) as points, the project aimed to show the Province and District level boundaries for the same period of time. In the course of more than 8 years of work, some 50,000 historical geographic places have been added to the database, and in the process of creating those, some collateral information about an additional 150,000 named places have also crept into the works.

Taking a cue from the main developers of the Great Britain Historical GIS **[GBHGIS]**, we decided at the outset not to implement either a Datestamping or STC method for tracking the spatial objects. Instead we opted for a building block approach, in which each unique instance of historical administrative units are stored in the database, and each of them has a direct relationship to a spatial object in one of many GIS layers, which are broken out into thematic divisions. The problem of spatial fragmentation is avoided because historical units are not represented by multiple spatial objects. However, since the other objective of the project is to show the adminstrative hierarchy and its changes over time, the database must be able to show superior or subordinate records for any selected unit, as well as preceding or subsequent changes that the selected unit went through. It turns out that tracking spatial units which change asynchronously, which split and merge repeatedly, which change their names and alter their administrative status at any given moment, is also problematic due to temporal fragmentation.

If we think back to the tables shown in Figures 5 and 7 above, we will recall that spatial changes require increasing amounts of record keeping as the fundamental spatial objects become more numerous, and in some cases become smaller and smaller fragments of arcs and LCG polygons. Now imagine a parallel problem which is to capture the relationships between superior and subordinate objects. The logic for mapping relationships between LCG polygons and historical objects in the STC model is quite similar to keeping track of splits and merges of parent objects and their subordinates. The main difference is that in the STC model, the LCG polygons are stable, and need only to be combined to represent any particular historical geographic object. In the STC model, when boundary changes occur new LCG polygons are created and their relationships to the historical objects are redefined in the tracking tables. The problem with mapping relationships between subordinate and parent adminstrative units is that neither parent nor child units are completely stable. Take for example the hypothetical relationships shown in figure 8.

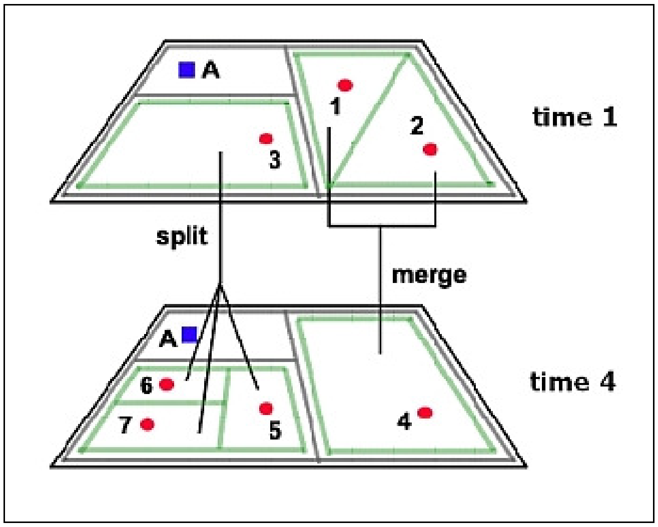

Figure 8: Parent – Child Relationships Over Time

If we consider the state at Time 1, the subordinate unit 3 is part of the parent unit A. The administrative seats are sybolized as a dot for 3, and a square for A. Now suppose that the area of unit 3 is split into three parts at Time 3, which have unique IDs of 5, 6 and 7, but that the name of unit 3 at Time 1 remained the same as the smaller unit 5 at Time 3 (Maple County). These relationships can be managed in two tables, a Main Table for the historical instances, and a PartOf Table for relationships to superior units.

object	*name*	*begin*	*end*
A	Some District	0	9
1	Oak County	0	3
2	Fir County	0	3
3	Maple County	0	3
4	Tree County	3	9
5	Maple County	3	9
6	Mill City	3	9
7	Park County	3	9

Main Table

child	*child_nm*	*begin*	*end*	*parent_nm*	*parent*
1	Oak County	0	3	Some District	A
2	Fir County	0	3	Some District	A
3	Maple County	0	3	Some District	A
4	Tree County	3	9	Some District	A
5	Maple County	3	9	Some District	A
6	Mill City	3	9	Some District	A
7	Park County	3	9	Some District	A

PartOf Table

Figure 9: Parent – Child Relationships (Administrative Hierachy axis)

It can be argued that the entity known as “Maple County” continues to exist throughout, and that it need not be broken up into two discrete historical instances (objects 3 and 5). However, in the case of CHGIS, during the construction of the database we did not know **beforehand** how many instances of change might occur, nor did we know when they would occur or whether they would be changes of spatial footprint, of placename, or of administrative unit status. The only practical solution for CHGIS was to create a unique **historical instance** for each adminstrative unit and to link them together into a chain of events. In this way, the database was agnostic as to whether the change involved splits, merges, placenames, or unit types, and could be navigated along the axes of *administrative hierarchy* or *temporal sequence* as needed.

Creation of unique historical instances, as applied to our example, results in two records for the historical “place” known as Maple County, which had a larger area of jurisdiction at Time 1, and was reduced in size at Time 3. The fact that the “place” which we think of as a single entity -- Maple County -- is represented by multiple instances over time, is an example of temporal fragmentation. In this case, the change was triggered by a spatial footprint change, but it might just as well have been caused by a change in placename or administrative unit status. Maple County could have been incorporated as Maple City, or it could have been renamed Walnut County, for that matter.

Another thing to keep in mind is that every change of a Parent unit that results in the assignment of a new historical instance (and thus a new unique identifier), requires editing of the PartOf Table. For example, if the parent unit described above as Object A (Some District), changed its name at Time 2 (to Another District) it would need a new identifier, Object B, and all rows in the PartOf Table with Object A as the parent would have to first be duplicated and have their parent IDs changed to B and their begin times changed to Time 2; then the original rows for Object A would need to have their end times changed to Time 2. This would then handle the passing of jurisdiction for any and all subordinate units from the instance of parent A (Some District) to parent B (Another District). Similar steps need to be taken in the event that the object IDs exist as children of other instances, (but **only** if the temporal extent of the child objects are not *completely contained* by the temporal extent of their parents).

According to the CHGIS model, editing operations in the PartOf Table are always limited to one step up or down the hierarchy, and no other levels need to be altered (*regardless of how many intereations in the hierarchy there are*)! This is quite an advantage. In the STC method described above, each time that a boundary change occurs, new LCG polygons are created, and all the historical objects that overlap the original polygon being replaced must be renumbered accordingly! This is problematic for many reasons. First, the unique identifiers of tiny polygon slivers have no semantic content to assist the editor in knowing where they belong. Even if the operation can be highly automated through GIS routines, there is little chance that errors will be detected by a *normal* human being. Second, the LCG polygons themselves have no meaning outside of the specific STC implementation and cannot be casually discovered or re-used. In the CHGIS model, each unique instance of an historical object is associated with a unique spatial object, and the spatial object carries with it redundant information about its placename, administrative unit type, and valid dates. The spatial data is also broken into thematic layers: by province, district, county, town, etc, which allows for the information to be opened in GIS without any connection to the database and still be understandable to the user, who could label all the objects by placename, for example. Finally, in the CHGIS model, when the process of editing changes in the Main Table or PartOf Table takes place, the placenames are always stored in both places, so that the human editor can check visually to see if any errors occurred.

As you can see, the table that stores the hierarchical relationships requires careful attention when new instances are introduced to the database, and the rows in the table are subject to temporal fragmentation. Interestingly, the table that stores temporal sequence relationships does not need any begin or end date values, and can function simply by showing which unit or units preceded any other unit. Merges are handled just as easily as splits, and no spatial operations are needed because the only information the table must convey is exactly what object preceded any other object. For example, the temporal sequence table for the example shown in Figure 8 could be as simple as the table in Figure 10.

object	object_name	precededBy	previous_name
4	Tree County	1	Oak County
4	Tree County	2	Fir County
5	Maple County	3	Maple County
6	Mill City	3	Maple County
7	Park County	3	Maple County

Temporal Sequence Table

Figure 10: Temporal Sequence Table (temporal sequence axis)

Knowing that object 4 (Tree County) was preceded by two objects 1 and 2 (Oak and Fir Counties) is enough. If we really need to know more about the boundaries and areas of these objects we can do spatial operations in GIS once we have identified the correct spatial objects. Of course, spatial operations on changing areal units over time is one of the perennial nightmares of GIS, the so-called Modifiable Areal Unit Problem, or MAUP. There are certainly ways in which aggregations of known areal units can be sliced and diced down to LCG polygons, so that statistical information associated with them will not become entirely distorted as they change over time. **[Henderson]** But MAUP solutions are notoriously complex and computationally intensive, even for relatively limited areas and narrow ranges of time. What if we avoid polygons altogether?

A Radical Departure: Hierarchical Networks For Historical GIS

What, *avoid polygons*, have I gone mad? How can we draw our wonderful heat maps of fictitious population densities, as they spread equally across the improbable landscapes of deserts, jagged mountain ranges and the surfaces of lakes? Or, well, at least if we cut out lakes and steep slopes we *still* need our historical boundaries, right? Otherwise how can we depict the unlimited appetite of the human race for claiming territory? Or might there be a reasonable argument made that, in fact, over the long course of human history, the territorial expansion, contraction, and interaction of different nation states and peoples is not really something that is best portrayed by neatly drawn boundaries in the first place? Indeed, as I have argued previously, even if it is delightful to assume that we can collect and interpret all the known facts about towns, counties, and parcels of land going back into the ancient past, it is next to impossible to actually digitize boundaries around them in GIS owing not only to the spotty and inconsistent textual sources but to a total lack of real cartographic evidence if we trespass into pre-Modern times. Let us not even mention the unbelievable cost in man-hours of trying to figure out where those boundaries were in order to digitize them in the first place! **[Boundaries or Networks]**

So, are we going to continue to assume that the paradigm of modern cartography which so neatly carves up our planet into human domains, and particularizes them down to the cadastral level of parcels and driveways and curbstones, is perfectly suited to mapping the ancient past? Is that reasonable? Perhaps the title of Gillray's famous cartoon best sums up the situation for polygon-based historical GIS: *Plum Pudding in Danger!* **[Gillray]**

Figure 11: Plum Pudding in Danger, or State Epicures taking un Petit Souper

Though we may laugh at the Imperialists of the past, as they sought to carve up the planet like gluttons attacking a plump roast, we might also pause to consider that the demarcation of and imposition of territorial claims by human societies extending into the remote past is equally absurd. And if we draw a large boundary encompassing all of the lands from Kamchatka to Vietnam to Poland, and simply call it "the Mongol Empire," does it have any meaning in actual point of fact?

What I propose is to map what we do know: namely, the relationships between parent and subordinate units found in historical texts. The spatial objects used to represent these units are points, and the locations of these points, fortunately, are almost always known to us; either from the archeological record or from continuous occupation to the present day. The symbolic representation of the network model, which reveals the general spatial organization of an administrative system is, at any rate, a valid depiction of the known evidence.

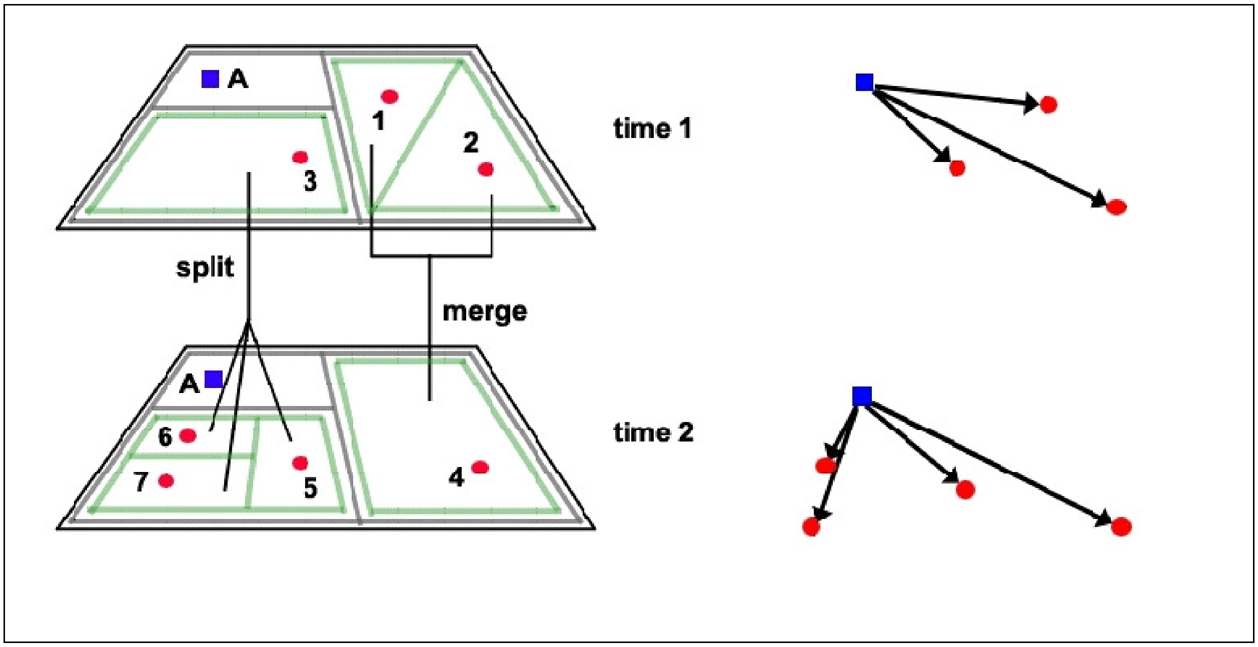

Figure 12: Hierarchical Network Model For Historical GIS

The iteration of relationships in the hierarchical network, as they change over time, also reveals the spatial extent of sub-networks and the places where those overlap or where gaps exist. This enables us to visulalize spatial patterns based on what we do know, and to investigate the areas of interest that are revealed. Doesn't this also make more sense than investing a huge amount of resources up front in order to draw precise boundaries that we know are incorrect, and which exist merely as placeholders to be swept aside at the first gust of disproof? Although we may not be accustomed to visualizing spatial patterns in networks rather than neatly drawn areas that cover every inch of the land, we are compensated for our first impressions by the fact that the relationships shown in the network model are not imposing fantastic and unsupported assumptions.

The CHGIS data model, described above, not only records the parent to child relationships between historical administrative units, but also gives them explicit begin and end dates. Therefore it is quite simple to query the CHGIS database and find out which units were subordinate to any particular parent, and also the valid dates for those parent to child relationships. By looping through these results to obtain the latitude and longitude coordinates of both parent and child, it is possible to serialize the results as line features directly into KML format for viewing in GoogleEarth. **[KML Network]**

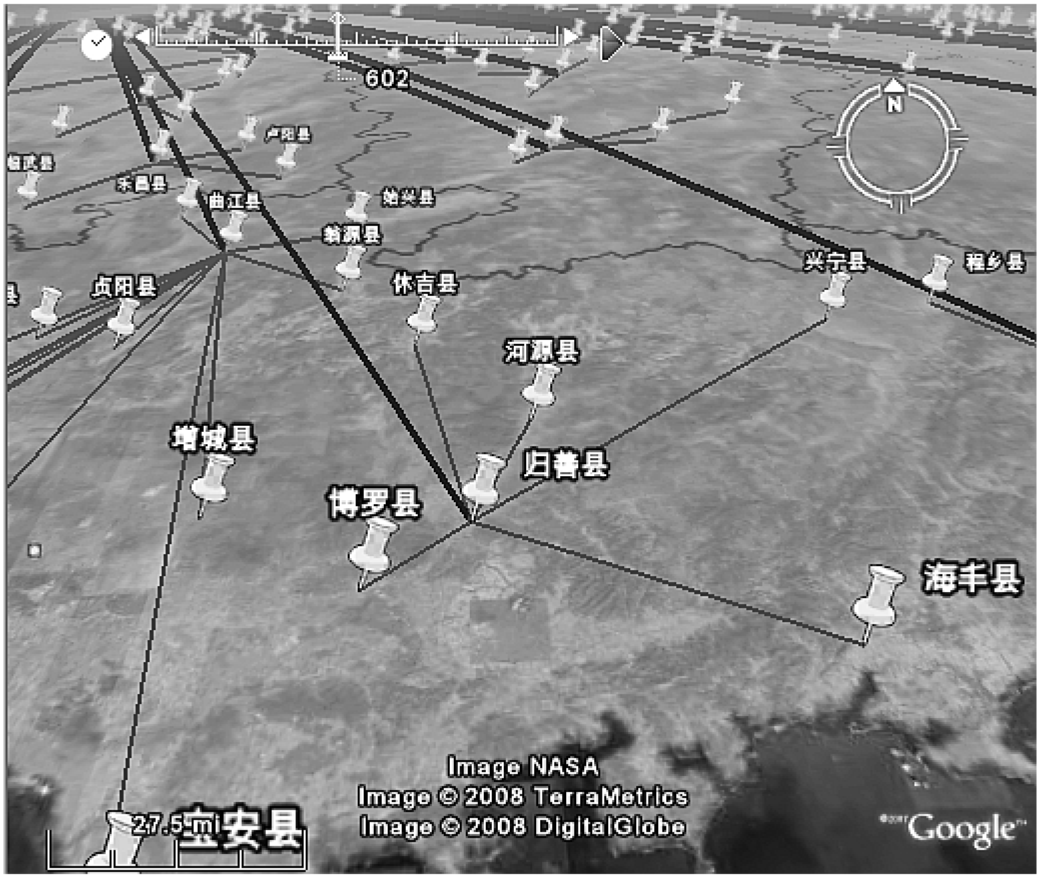

Figure 13: Spatio-Temporal Network Model in GoogleEarth

In GoogleEarth, the changing administrative network can be browsed by sliding the time bar control, or turning on the animation tool. For this presentation, the first level and second level administrative networks were visualized over the course of 1,000 years based on the existing time series data in CHGIS Version 4 **[CHGIS V4],** and are posted online as Time Enabled KML examples. **[teKML]**

Now we can browse through hundreds, or thousands of years of Historical GIS objects, and see the general spatial patterns of changing administrative systems without having to delineate each boundary change. Interestingly, the temporal browsing functionality is only possible using a free software application, GoogleEarth, which provides a schema for spatio-temporal objects, but cannot be done with any of the major commercial GIS packages. (Okay, IDRISI can even model space-time-cubes, but only for raster data…) **[IDRISI]** The commercial vendors just don't ***get it!*** They don't seem to realize that historical data may not always be slanted towards the *day – minute – second* time values needed for tracking airport traffic and weather systems. For some of us, the scope of Historical GIS may extend to the very beginning of human history, or even paleontology. Hopefully, as more researchers define GIS objects that have both spatial footprints and temporal ranges, the software vendors will provide us with better tools. For the moment, we can at least take advantage of GoogleEarth, push it to its limits and discover its weaknesses.

Visualizing Moving Objects in Historical GIS

The network model we have discussed is based on actual dates of Historical Objects. In other words, a relationship between District A and County Z, that existed in the Tang Dynasty from the year 800 to 850, was processed into a KML Placemark feature with a begin time value of 800 and an end time value of 850. However, in order to use GoogleEarth as a tool for visualizing historical events, it may be necessary to use artificially calculated time values, in order to view those events at a reasonable speed.

One of the drawbacks of GoogleEarth, at the moment, is that the time bar feature tends to animate events much too quickly. Even when adjusted to its slowest settings, the progress of the time selector whips from left to right along the timeline and causes geographic objects flash on the screen faster than the eye can detect them. This is not so much of an issue with our hierarchical network model, where we want to accurately reflect the real time values of the Historical Objects being depicted. There are other cases, though, such as animating a sequence of events for which we have accurate locations but not accurate times, when we need to come up with a workaround.

For this presentation, we experimented with a biographical dataset that shows various events in the life of a famous Buddhist monk, Ouyi Zhixu, who lived in the late Ming Dynasty. **[Ouyi]** The events in Ouyi's life could be roughly broken down into years, and put into a logical sequence, but they could not be dated with any precision. In the end, we had only the numbers in the sequence of events, and when these were used as proxy years in GoogleEarth, the animation was much too fast for the human eye to detect. In order to stretch out the time on screen for each event (consisting of an arc showing movement from one location to another), several algorythms were tested to create artificially large spans of time in between each event. The resulting animation was adequate for watching the stages of movement in the monk's life, but if viewing the arcs of movement alone, they did not present an adequate visualization of his life as a whole.

Based on earlier experiments using TimeMap, **[TimeMap]** it became obvious that visualizing movement of a single agent over time involves not only animation of the *paths of movement*, but the retention of the *cumulative trail of movement*, which shows the entire *cloud of activity* that occurs throughout the process. In addition, it is helpful to animate the *nodes of presence*, that is to say, point locations where the agent is paused before moving on to another location. Finally, it is equally useful to highlight the *target locations*, meaning the points that the agent is moving towards before actually arriving there. These tasks were all accomplished in TimeMap, and we attempted to reproduce them in GoogleMap with limited success.

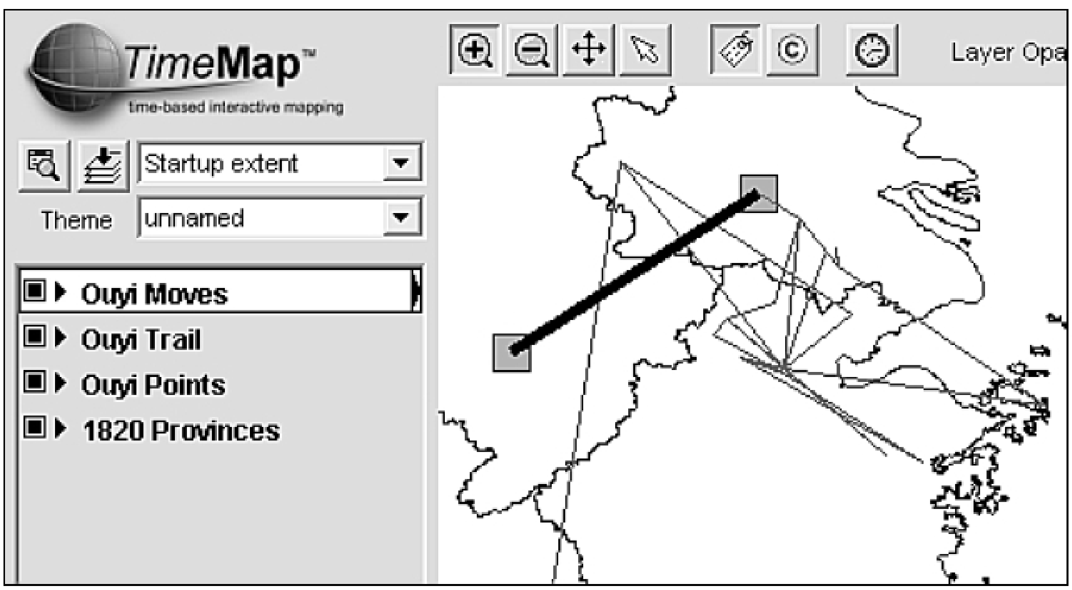

Figure 14: Animation with sequence of events, paths of movement and cumulative trail

These experiments with Historical GIS data demonstrate the inherent complexity in dealing with objects that vary asynchronously in multiple dimensions, including spatial footprint variations, name changes, and attribute changes. Spatial changes may involve enlargements, reductions, displacements, merges, splits, and host of other logical processes. **[Identity Based Change]** The matrix of name changes cannot be measured on any rational scale since the nature of toponym and spelling changes are completely idiosyncratic. Though some attributes, such as administrative feature types, may have logical relationships that could be modeled in an ontology, the characteristics of administrative divisions and how they function in different administrative systems also changes over time. Indeed, the only dimension of data that we are tracking which is consistent throughout the study period is time itself.

Time, although it can suffer from mismatches of definition and formatting when modeled in a database, is nonetheless linear. If we choose an appropriate time standard (for example, ISO 8601, or Julian Day Numbers), we can attach not only attestations about historical periods (such as Chinese Reign Periods and cyclical dates) **[DDBC Time Authority]**, but also pull all of the multivariate instances of change for Historical GIS into a single navigable thread. In this sense, it is the **Timeline** that proves to be the anchor for modeling and visualizing spatio-temporal objects, not **Space**.

It is a mistake to establish spatial objects first and then paste on attributes that vary asynchronously as an afterthought. For example, if spatial data is primary how would we choose an appropriate scale and projection for an Historical GIS of the United Kingdom? Should it be suitable for the British Isles, or for India, or for the Falklands Islands? Instead, let us focus on the attested *dates of existence* of geographic objects, and their attested *dates of relationships*; then we can link the instances of geographic objects in the database with as many spatial representations as we like. **[Pleiades]**

Even so, the time standard for the instances must be consistent! That is something we actually can accomplish by adhering to the standards mentioned above, and to be realistic: historical calendars are no longer subject to change. Of course, some dates may need to be adjusted if new evidence comes along, but if we first synchronize historical calendar timelines with Julian Day Numbers, we can make use of those timelines as proxies for dating Historical GIS objects. If each GIS object is stored – at a minimum – with a single point in coordinates expressed as decimal degrees, we can visualize them today, using the technique described above. Of course, those historical instances in the database can also be linked (whenever possible) to other geometries (such as polylines, polygons, and regions), allowing for spatial searches and spatial analysis operations. If only we can abstain from thinking that all space must be carved up into neatly packaged territories, and avoid the creation of boundaries as the prerequisite for Historical GIS our progress will be faster and provide a more flexible foundation for developing of polygon-based representations later on. Investing enormous sums and man-hours in defining obscure and un-evidenced boundaries, is problematic from my point of view, and should only be undertaken *after* we have built a reliable a skeletal framework of historical objects to serve as a foundation and knowledge organization system.

[Boundaries or Networks] Berman, Merrick Lex. *Boundaries or Networks in Historical GIS: Concepts of Measuring Space and Administrative Geography in Chinese History*. In *Historical Geography*, Vol33, *Special Issue: Emerging Trends in Historical GIS*. Baton Rouge, 2005. http://www.fas.harvard.edu/~chgis/work/docs/papers/HG33berman.pdf

[CHGIS V4] *China Historical GIS*. http://www.fas.harvard.edu/~chgis/

[Claramunt] Research on Spatio Temporal Logic and Processes. http://christophe.claramunt.free.fr/index.php?option=com_content&task=view&id=5&Itemid=6#Time

[DDBC Time Authority] Dharma Drum Buddhist College. *Time Authority Database*. http://authority.ddbc.edu.tw/docs/open_content/notes.php

[GBHGIS] *Great Britain Historical GIS*. http://www.port.ac.uk/research/gbhgis/

[Gillray] Gillray, James. *The Plum-pudding in danger, or State Epicures taking un Petit Souper.* 1805. (Thanks to my colleague, Ben Lewis, who hung this poster on the wall of this office!)

[Henderson] Henderson, Mark and Merrick Lex Berman. *Methods for Space-Time Analysis: Examples from CHGIS*. ESRI Users Conference Paper, 2003. http://proceedings.esri.com/library/userconf/proc03/p0833.pdf

[Historical GIS Tech] Gregory, Ian and Paul Ell. *Historical GIS: Technologies, Methodologies, and Scholarship*. Cambridge: Cambridge Univ Press, 2008.

[Identity Based Change] Hornsby, Kathleen and Max Egenhofer. *Identity-based change: a foundation for spatio-temporal knowledge representation.* International Journal of GIS 14(3): 207-224.

[IDRISI] *IDRISI, Taiga, Earth Trends Modeler.* http://www.clarklabs.org/ See also Ron Eastman's http://www.earthsystemtrends.org/

[KML Network] Berman, Merrick Lex. *Spatio Temporal Network Model for GoogleEarth.* http://gist.fas.harvard.edu/chgis/?p=19

[Ouyi] Berman, Merrick Lex (based on research by Beverly Foulks). *Modeling Spatio-Temporal Networks with CHGIS*. 2nd International Workshop on Monies, Markets, and Finance in China and East Asia, 1600-1900, Ruhr-University Bochum, Oct 2007. http://www.fas.harvard.edu/~chgis/work/docs/papers/Berman_ModelingNetworks_Bochum07.pdf

[Paleomap] Scotese, Christoper. *Paleomap Project*. http://www.scotese.com/

[Pleiades] *Pleiades Project*. http://www.atlantides.org/trac/pleiades/

[teKML] Hays, Bill and Merrick Lex Berman. *Time Enabled KML*. https://cga-download.hmdc.harvard.edu/publish_web/Geo_Tools/teKML/

[TGIS] Langran, Gail. *Time in Geographic Information Systems*. London: Taylor and Francis, 1992.

[TimeMap] Johnson, Ian and Artem Osmakov. *TimeMap Software*. http://www.timemap.net/

[Worboys] Worboys, Mike. http://www.spatial.maine.edu/~worboys/downloads.htm

[Yuan] Yuan, May. http://ags.ou.edu/~myuan/