Relational Data Model for Spatio – Temporal GIS Allowing for Attributes with Independent Temporal Extents

CHGIS Data Model – Phase Two

Merrick Lex Berman
Harvard Yenching Inst
8 Oct 2001

Subsequent to the International Workshop on Historical GIS, held in Shanghai, China, Aug 23-25th, 2001,[1] the draft for the Spatio-Temporal Database Design is now being revised. The new CHGIS Data Model being proposed here, is an attempt to limit the number of records populating the database while keeping track of events that include changes in name, location, administrative status, or relationships in the administrative hierarchy. The ideas presented here are greatly influenced by the work of Martina De Moor (University of Ghent),[2] Lawrence Crissman (Griffith University),[3] and the author's previous CHGIS data model.[4]

The key departure from the earlier data model (footnote 4) is the tracking of attributes in separate tables, each with its own temporal extents. Whereas I had previously envisioned the simplicity of freezing attributes for each historical instance (or state in between two events), I am now attempting to allow for each attribute to have its own temporal instances. The objective is not so much to save tablespace, but to allow for tracking of variable attributes, such as name, feature type, administrative status, and avoid any duplication of records in the database caused by changes in parent or child units. In the phase two model, only a case of changing spatial objects (or "versions" of spatial objects, in Crissman's parlance) will **indirectly** impact parent – child relations.

In the earlier model (footnote 4), a change in any of the attributes, such as name or administrative status of a parent record, would indirectly impact all child records in the database. In the phase two model, similar attribute changes are tracked in separate "Instance Tables," and only in the case where the spatial object, or footprint changes are child records affected. For example, a county boundary change would impact the hierarchical relations of subordinate villages and towns; causing any towns with **new** parent relationships to be updated in the the "Instance Relation Table," which is to say that an affected town would have a new "part of" relationship, and the previous "part of" record must also have an "end date" alteration. Such changes, however, are limited to the immediately subordinate – superior levels in the hierarchy.

The second important difference in the phase two model is to explicitly accommodate the tracking of the concept of "place." Although the earlier model attempted to capture the same transformations over time by focussing on events and attribute changes, the phase two model makes "place" the focal point of the database. Unfortunately, there exist no clearly defined guidelines as to what consitutes a place, and how to determine when a place has been changed into a new place. For now, the working definition is that a unique "place" in the database is the first (but not necessarily earliest) known named instance of a human settlement, historical administrative unit, natural feature, or spot feature at a known location in geographical space and a known point or extent in time.

This idea of place, however, can only be thought of as a level of abstraction separate from the actual objects that we need to keep track of in the database. For example, when we think of San Francisco, we may have a range of notions that come to mind, from the fuzzy area including the entire bay and surrounding cities, to the city itself, or to a special place familiar to us, such as Fisherman's Wharf, Union Square, or South of Market. All of these things are somehow, "San Francisco," and we must first differentiate them to correspond with trackable objects in the database, and then we must also **satisfice**[5] them, or associate an appropriate bounding box to enclose the space that represents each unique place feature. In the model being proposed, we first establish a loose concept of "San Francisco" in the Place Table and assign a Place ID, for example, San Francisco PID1. Then for each "Place Object" that we wish to track, we assign a unique "Place Object ID," as in: San Francisco (the city) POID1, San Fransisco Bay POID2, San Francisco

County POID3. We can then relate all of the Place Objects to a Place, but can also track the attribute histories of each place as unique unrelated objects.

Our Place Objects are tracked in their various transformations and movements, by means of independent "Instance" Tables. The Instance, in this case, is a definable point in time or span of time, and the proposed model separates Placename, Administrative Status, Feature, and Spatial Instances into their own timelines. One of the fundamental tasks of the CHGIS model is to keep track of the unsychronized changes of in these four rough categories. For Placenames, we may track not only the sequence of "official" names that were used in government records, but can populate the table with as many "alternate" names as we choose, each having their own temporal sequence, overlaps, or discontinuities, as needed to fit the historical evidence. Admin Status is treated in a similar fashion, with the Administrative Hierarchy being tracked in the "Instance Relation Table," as in Admin Instance X is "part of" Admin Instance Z.

The Feature Instance Table is proposed as a catch-all for any other Feature Types that the database shall include, and offers considerable flexibility for the user to incorporate their own specialized data into the database. Features such as temples, archaeological sites, dialect regions, fault zones, botanical specimen collection sites, photographs, and so forth can be organized as individual Feature Instances for each occurrence. Also of interest to users will be the Event Table, where CHGIS will include specific historical events that directly affected other Instance records, but where other important events may be entered as a means of performing temporal queries. For instance, a researcher may wish to include the exact dates of major events, such as earthquakes, in the Event Table, and then search for any records that began or ended within a certain time period of the event. Combinations of event records can be used as a matrix for other temporal queries of interest, such as the number and temporal distribution of name change occurrences during a dynastic period, or administrative changes that took place during the overlapping period of a certain number of years before and after a dynastic change, etc.

Quite independent of the Instance tables described above, the Spatial Objects in the database may be used to represent any record for which their Spatial Instance has been related to a particular Place Object. Although Feature Instances may be created that have no concrete s patial analog (such as Zen Buddhism), the base CHGIS datasets will include only records for which we can **at the very least** identify a point location. Also included will be polylines for coastlines, major river systems, and elevation contours. Polygon objects for prefecture (or Fu) level units and above will be included in the base database, and for some areas, county polygons as well. The Spatial Objects need not be kept in separate tables (depending on the software implementation) but they should allow for the tracking of object types on separate timelines. For example, a polygon may be related to a particular Spatial Instance. At the same time, a point object is related to the instance, in relation to the location of the administrative seat for that polygon. But let us say that the location of the seat changes. Not the name, the status, the polygon, or any other aspect of the Spatial Instance, except for the location of the seat. If we allow for tracking of the feature types with separate timelines, we can easily create a new Point Object record with the needed Begin – End dates and associate it to our Spatial Instance.

Derived Spatial Data may be extracted automatically for each spatial object, providing a point or bounding box for each item in the database. This will allow for quite simple discovery of objects in the database and their related attributes, without the use of any GIS mechanism.

Lastly, the data model provides for links to Historical Sources, Commentaries, Spatial Data Sources, Spatial Metadata and Vernacular Script texts needed for any of the information recorded in the tables described above.

Below are descriptions of the core tables for the system, which fall into three general categories: raw data tables, temporal instance tables, and linktables. The overall diagram should be consulted first.

Overall Diagram of Tables

http://www.people.fas.harvard.edu/~chgis/work/design/figs/diagram-phase2.pdf

List of Table Descriptions:

Description of Tables

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Place Table"**>>>>>>>>>>>>>>>>>>

Place Table

Attributes: Place ID
Common Name
Notes

Objective: to uniquely identify a "place" whose attributes, locations, and transformations can be tracked across time.

Definitions:

1) Place -- a unique "place" in the database is the first (but not necessarily earliest) known named instance of a human settlement, historical administrative unit, natural feature, or spot feature at a known location in geographical space and a known point or extent in time. (For example, we may know of the contemporary "place" Sandia Pueblo, New Mexico, and give it a Place ID, with the common name "Sandia" and a Place Object ID telling us that it is Sandia as a Spot Feature. Then in the Placename Instance Table, the associated Feature Type for "Pueblo" would be used. If we know the alternate native name, "Napeya," this info is recorded in a separate Placename Instance record with appropriate temporal extents. Similarly, when the Sandians were ousted in the late 16^{th} century, they resettled in the village of Payupki, in the Hopi Lands of Arizona. Here, we could record the "place" Sandia Pueblo, now actually identified as a specific Place Object, as having moved to a new location (with a new Spatial Instance record). In the mid-18^{th} century, when Sandia Pueblo was rebuilt and resettled at the original location back in New Mexico, this would be reflected in another Spatial Instance record. In this way we can track various other attributes, each with their own unique temporal histories, such as Administrative Status and Feature Types, and relate them back to our concept of "place" using the Place Object ID as a key.

2) Common Name -- a name used to characterize a place, for the convenience of the database users. The actual names, alternate names, alternate language names, and all their variants are tracked in separate Name Table, Alternate Name Table, and Alternate Language Name Table.

Notes:

1) The real difficulty we face with our loose conception of a "place" is that we are actually referring to many different unique objects in the database for each "place." Think of the "place" as a conceptual space in which a number of independent objects share a common toponym, even though they have unrelated histories. Take for example "Mississippi" as a place. In fact, Mississippi River has a completely separate history from the delta, the state, and any number of other objects with the same name. In order to keep track of these, "place objects," we link them with a "place type," then allow for completely independent attribute histories to be tracked for each separate place object.

Relations:

One-to-many: Place Object Table

>>>>>>>>>>>>>>>>>>>>>>>>Begin “Place Object Table”>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Place Object Table

Attributes: Place ID
Place Object ID
Place Type ID

Objective: a relational table to show the one to many links between places and place objects, and to identify each place object with a place type.

Definitions:

1) Place Object -- a unique object with geographic and temporal extents, each attribute of which can be tracked with independent histories of change found in Instance Tables. By creating unique place objects, we can follow all changes related to any particular attribute of a place object without affecting the other contemporaneous attributes.

2) Place Type -- an attribute class for diffentiating Place Objects (for a complete list see the Place Type Table description below). For example, a mountain and a temple could have the same Place ID and name, but would be first categorized by their Place Types (T for Hypsographic feature, and S for Spot feature), and then assigned unique Place Object IDs. In fact, there might be more than one Temple with the same Place ID, the same name, and the same Place Type, but since we assign it a unique Place Object ID, we can still keep track of both temples and their own unique histories recorded in the Instance Tables.

Relations:

Many-to-one: Place Table
Many-to-one: Place Type Table
Many-to-many: Event Linktable
One-to-many: Placename Instance Table
One-to-many: Admin Instance Table
One-to-many: Feature Instance Table
One-to-many: Spatial Data Linktable

Relationship notes:

Event Linktable -- Naturally there can be many events linked to any particular place object, but at the same time, a single event might impact many different place objects.

Spatial Data Linktable -- In the case of spatial data, we can record a sequence of spatial data changes for a single place object, but to conserve table space, we can also allow for any particular spatial object to be used as a representation of multiple place objects.

>>>>>>>>>>>>>>>>>>>>>>>Begin “Place Type Table”>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Place Type Table

Attributes: Place Type ID
Place Type
Description
Name ID

Objective: Place Types are used to differentiate the actual objects being tracked in the database within the looser concept of "place." For example, Emeishan can be assigned a Place ID, but in order to keep track of Emeishan the natural feature separately from Emeishan the administrative division, and Emeishan the temple complex, we assign the Place Types: T, A, S, where T = Hypsographic Terrain Feature, A = Admininistrative Boundary, S = Spot Feature. In this example, the place types are drawn from an authoritative source, NIMA GNS Class Codes,[6] but they can also be created for other special purposes, if needed. Further sub-classification of Place Types is accomplished by assigning Administrative Units and Feature Types in the Administrative Instance Table and Feature Instance Table. For example, if we wished to keep track of Emeishan Temple, and Emeishan City Hall, both of which are Spot Features, both the temple and the building would be given unique Place Object IDs. They would share the same Place Type, "S," but in the Feature Instance Table, each would be assigned the appropriate feature ID, such as "Buddhist Temple" and "government building". This method allows for the rough sorting of all Place Objects according the following types:

```
A     Administrative Boundary Features
H     Hydrographic Features
L     Area Features
P     Populated Place Features
R     Road / Railroad Features
S     Spot Features
T     Hypsographic Features
U     Undersea Features
V     Vegetation Features
```

before querying any of the spatio-temporal records, which could help in greatly narrowing the search parameters.

See also the discussion under Feature Type Table.

Definitions:

1) Name ID uniquely identifies all names used in the database. See complete description for Name Table.

One-to-many: Place Object Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Event Linktable"**>>>>>>>>>>>>>>>>>

Event Linktable

Attributes: Place Object ID
Event ID

Objective: a relational table to show the many to many links between events and place objects.

Definitions:

1) Because every place object in the database is being tracked across time, each one will have many events linked to it. Likewise, a single event could impact many different objects at the same time. This table allows, for example, a flood event to be recorded which impacts six different counties. Although the event is described in general terms, this table allows for spatial and temporal searches for particular places to discover related events that impact them.

Relations:

Many-to-many: Place Object Table
Many-to-many: Event Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Event Table"**>>>>>>>>>>>>>>>>>>

Event Table

Attributes: Event ID
Begin Event
End Event
Next Event
Event Description
Historical Source ID
Commentary ID

Objective: to record events that affect place objects in the database and/or their attributes

Definitions:

1) Begin – End – Next Event are dates, used according to the dating guidelines set out by Man Zhimin, Fudan University.[7] The End Date is meant to be the end of the event itself, not the end of the temporal "instance," or "state," created by the event. The end of the instance span is recorded in "Next Event," which matches the Begin Date of the next subsequent Event record that impacts the same place. This idea originates in the ESRI White Paper, "An Area Event System for Forest Management History,"[8] and differntiates between the temporal extent of the actual event and the temporal extent of the instance created by theat event. For example, a fire in a town might start on a certain day and burn for many days or weeks…this period would be captured by the Begin – End dates. Then the burnt down area may remain in the same state for many months or years before the next event, rebuilding the burnt down area, begins, which would be the Next Event date, as well as the Begin date of the new Event Record

Notes:

1) Events can occur which do not impact the attributes being tracked in the Instance tables. This allows for the addition of event data that can be related to place objects and temporal spans without having to alter any of the other tables in the database. For example, the CHGIS database will not contain records about the building of temples and monasteries, and yet these can easily be added to the Event Table and associated with a particular place using the Place ID. A corresponding new Event Type, “construction, place of worship” should be added to the Event Type Table.

2) Each event may impact other tables, causing new records to be added. A name change event, would require creation of a new record in the Name Instance Table. Each of these impacts are recorded in the Event Instance Linktable.

Relations:

Many-to-many: Event Linktable
Many-to-many: Event Type Linktable
One-to-many: Event Instance Linktable

>>>>>>>>>>>>>>>>>>>>>>>Begin “Event Type Linktable”>>>>>>>>>>>>>>>>>

Event Type Linktable

Attributes: Event ID
Event Type ID

Objective: a relational table to show the many to many links between events and the event types used to define them. One Event can have many related Event Types.

Definitions:

2) If a place had a name change event, the Event ID is linked to the Event Type ID for Name Change. When events involve simultaneous changes in multiple attributes, for example, administrative status and location, two records would be added to the Event Type Linktable, one for the admin status change, one for the location change. In this way users can keep track of which attributes and their tables were impacted by any event.

Relations:

Many-to-many: Event Table
Many-to-many: Event Type Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Event Type Table"**>>>>>>>>>>>>>>>>>

Event Type Table

Attributes: Event Type ID
Event Type
Impacted Attribute
Notes

Objective: records event types used to define events, and in particular which attributes are impacted by an event.

Definitions:

1) Event Types are needed to keep track of both the type of Event being recorded, as well as to know which other attributes and their tables are being impacted. For example, let's say that Name Change, was Event Type ID = 1. The affected attribute table would be Name Table.

2) Impacted Attributes are used to indicate which tables have been affected by an event. Each Event Type should be assigned as uniquely affecting one attribute table. If an event impacts many attribute tables at once, the Event Type Linktable is used to show this one-to-many relationship. For example if Event Type ID 1 = Name Change, and Event Type ID 2 = Admin Status Change; and if a certain event X impacted both Name and Admin Status, then two rows should appear in the Event Type Linktable, as in:

Event	Event Type ID
X	1
X	2

Here we are only keeping track of the types of event change that occur. The actual relations between any particular event and all the instances being impacted are tracked in the Event Instance Linktable.

Relations:

One-to-many: Event Type Linktable

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Event Instance Linktable"**>>>>>>>>>>>>>>>>>

Event Instance Linktable

Attributes: Event ID
Instance ID

Objective: a relational table to show the one to many links between events and all instances impacted by the event, regardless of whether they are Placename Instances, Admin Instances, Feature Instances, or Spatial Instances.

Definitions:

1) Any given event may consequently impact multiple attributes. For example, in a change involving both administrative status and location, four records would be added to the Event Type Linktable,

two for the admin status change, two for the location change. In the two admin status records, the Event ID is related to both the existing impacted Admin Instance ID, AND the Admin Instance ID that was created due to this event. In the same fashion, the Event Instance Linktable relates to the Placename Instance Table, Feature Instance Table, and Spatial Instance Table.

Relations:

Many-to-one: Event Table
Many-to-many: Placename Instance Table
Many-to-many: Admin Instance Table
Many-to-many: Feature Instance Table
Many-to-many: Spatial Instance Table

>>>>>>>>>>>>>>>>>>>>>>>Begin "Placename Instance Linktable">>>>>>>>>>>>>>>>>

Placename Instance Linktable

Attributes: Placename Instance ID
Place Object ID
Name ID
Feature Type ID
Admin Unit (Y – N)
Begin Date
End Date

Objective: a relational table to show the one to many links between place objects and placename instances.

Definitions:

1) Placename Instance represents the temporal span during which one placename remained unchanged in reference to a particular place object.

2) Name ID is used to uniquely identify the part of the toponym that we think of as a "name," as opposed to "identifier." For example, in Delaware River, we think of "Delaware" as the name, and "River" as the identifier. These distinctions are referred to as "placename" (with a Name ID) and "feature type" (with a Feature Type ID) in this data model. (In Chinese these correspond roughly to "tong ming" and "zhuan ming.") Note: the Feature Type ID is always used when available in Chinese, because the Feature Type is an inseparable part of the placename. In other words, we must always say "Delaware River," to mean this place object.

3) Admin Unit is used ONLY when it is an inseparable part of the placename. In Chinese, the general rule is that county level units and above usually have the Admin Unit Name in their spoken and written forms. However, for all placenames consisting of only one Chinese character, the admin units are absolutely necessary and MUST be used as an inseparable part of the placename. Here the case of needed amin unit as part of the placename is indicated with the Y – N flag.[9]

Relations:

Many-to-one: Place Obect Table
Many-to-many: Event Instance Table
Many-to_one: Placename Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Placename Table"**>>>>>>>>>>>>>>>>>

Placename Table

Attributes: Name ID
Common Name

Objective: an authority table for all placenames that appear in the database, with the idea that names even if used in many different instances or for unrelated place objects, need not have more than one entry as long as they are identical.

Definitions:

1) Common Names are to be entered in Pinyin Romanization whenever possible, and are used purely for convenience of the database user, as a way of associating a name with a Name ID. The actual vernacular names in an unlimited number of native encodings, as well as the transliterations into any number of romanization systems are linked to this table using the Language Linktable.

Relations:

Many-to-many: Placename Instance Table
Many-to-many: Language Linktable

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Language Linktable"**>>>>>>>>>>>>>>>>>

Language Linktable

Attributes: Name ID
Language Table
Charset Encoding

Objective: a relational table allowing for any Unique Name ID record to have multiple representations in an unlimited number of separate language tables.

Definitions:

1) Language Table is the name of a table in a vernacular script. For example, "CHT_name" is being used for placenames written in Traditional Chinese Characters, using BIG5 encoding. Any number of encodings can be used…." CHT2_name" may be used for UTF-8 Traditional characters, and so forth.

2) Charset Encoding, as mentioned above, specifies which character set encoding has been used to enter the vernacular script placenames.

Relations:

Many-to-many: Placename Table
Many-to-many: Language Tables (various)

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Language Tables"**>>>>>>>>>>>>>>>>>

Language Tables

Attributes: Name ID, Admin Unit ID, Feature Type ID
Vernacular Name

Objective: an authority table containing the vernacular script names of all names in the database as needed, with the idea that where the identical name is used by many different place objects or instances, it need only be entered into the database once.

Definitions:

1) Language Tables are a number of different tables in a vernacular scripts, saved in character set encodings as described in the Language Linktable. Language Tables, include not only tables for recording placenames, as in the previous example, "CHT_name," but will also be created as separate tables for storing Administrative Unit Names, Feature Names, Source Citations, and Commentary Texts.

Relations:

Many-to-many: Language Linktable
Many-to-many: Admin Unit Linktable
Many-to-many: Feature Type Linktable
One-to-many: Source Linktable
One-to-many: Commentary Linktable

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Transliteration Linktable"**>>>>>>>>>>>>>>>>>

Transliteration Linktable

Attributes: Language Table
Name ID, Admin Unit ID, Feature Type ID
Transliteration ID

Objective: a relational table allowing for any unique vernacular name recorded in one of the Language Tables to have multiple representations in an unlimited number of separate language tables.

Definitions:

1) Language Table is the name of a table in a vernacular script, which identifies where the related Name ID, Amind Unit ID, or Feature ID will be found.

2) Transliteration ID links to the actual transliterated name, always saved as plain ASCII text.

Relations:

Many-to-many: Transliteration Table
Many-to-many: Language Tables (various)

>>>>>>>>>>>>>>>>>>>>>>>**Begin “Transliteration Table”**>>>>>>>>>>>>>>>>>

Transliteration Table

Attributes: Transliteration ID
Transliteration System
Transliteration

Objective: an authoritative list of all transliterated names in the database.

Definitions:

1) Transliteration System, such as Pinyin, Wade-Giles, Yale, Hepburn, Extended-Wylie, etc. Preferably this will be used as an ID number and the definitions of the systems will be kept in a separate Transliteration System Table (to be expanded as needed).

2) Transliteration is the actual text saved as plain ASCII text. Since certain symbols, such as the umlaut, caret, and the aspiration mark can be problematic, we will standardize usage as follows:

   Umlaut, as in “green” lü, will be written with the lower case letter “v”, as in: lv
   Caret, as in Tibetan Pinyin for “virtue” dê, will be dropped, as in: de
   Aspiration mark, often indicated with an apostrophe ‘, will take the accent mark (lower case of the tilde ~ key), as in `

   Note: the reason for the latter is that the single apostrophe is a very common character in scripting languages and will interfere with query processing.

Relations:

Many-to-many: Transliteration Linktable

>>>>>>>>>>>>>>>>>>>>>>>**Begin “Admin Instance Table”**>>>>>>>>>>>>>>>>>

Admin Instance Table

Attributes: Admin Instance ID
Place Object ID
Admin Unit ID
Begin Date
End Date

Objective: a relational table to show the one to many links between place objects and administrative status instances. Admin Instance represents the temporal span during which one administrative status of an administrative division, as well as its relationship to parent units in the database, remained unchanged in reference to a particular place object.

Definitions:

1) Admin Unit, described below, refers to the Adminstrative Division or Status that was officially designated by government authority. Admin Unit does not necessarily represent a definable area of administrative authority, although it is often associated with territorial claims.

Notes:

1) An Admin Instance can be affected not only directly, as when the status of the administrative unit was changed by government authority; but also indirectly, in the case of changes in the area controlled by a parent unit in the administrative hierarchy.

   Let's consider a few examples. If village X is part of Prefecture A, from the year 1700 to 1850, and if village X is promoted to County X in the year 1790, we need to make sure that the Admin Instance Record for village X is updated to have the correct End Date of 1790. An Event should be added to the Event Table, with the corresponding Event Type. Then a new record should be added to the Admin Instance Table, showing that County X was created in 1790. Now let us suppose that in 1830, Prefecture A lost some of it's territory to Prefecture B, including the area occupied by County X. In this case, a spatial query should be done to determine ALL the child units impacted by the movement of territory into Prefecture B. For those records, the Instance RelationTable records showing them to be "part of" Prefecture A need to be updated with the correct End Date, 1820. However, if the only immediately subordinate unit affected was County X, then no other smaller records need be changed, as long as they are STILL part of County X! The Event Table needs to be updated, and new Spatial Data records should be created in the Spatial Data Linktable for the new polygon objects representing the smaller Prefecture A and larger Prefecture B. Then the impacted child units discovered as having moved to Prefecture B should have new instances in the Instance Relation Table, showing that they are part of Prefecture B, Beginning 1830. Following these steps avoids having to impact any other records in the database, except those on the immediately subordinate hierarchical level that were impacted by a change in spatial object.

Relations:

Many-to-one: Place Obect Table
Many-to-many: Instance Relation Table
Many-to-many: Event Instance Linktable
Many-to-one: Admin Unit Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Instance Relation Table"**>>>>>>>>>>>>>>>>>

Instance Relation Table

Attributes:
- Instance ID
- Relation Type
- Related Instance ID
- Begin Date
- Begin Date Rule
- End Date
- End Date Rule
- Relationship Details

Objective: a relational table used to define relationships between instances. For example, we can keep track of which administrative units were immediately superior to other administrative units for a particular time span, using a "part of" relationship in the Relation Type field. The administrative hierarchy of relationships can be derived by stepping up or down the chain of parent or child units.

Definitions:

1) See also the explanation in Administrative Instance Table, Note 1 above.

   The value in the "Part of" field is the Admin Instance ID of the immediately superior unit. For example if Prefecture X (which was part of Province Z from 1800 to 1900) had subordinate Counties A, B, and C, from 1820 to 1850, the table would look something like the following:

AID	part of	Begin	End
X	Z	1800	1900
A	X	1820	1850
B	X	1820	1850
C	X	1820	1850

   Now let's suppose evidence is found that the territory of Prefecture X (containing County B) actually was transferred to Prefecture M (also part of Province Z) in 1840. The impact of the new evidence can be captured in several changes, as follows:

AID	part of	Begin	End
M	Z	1800	1900
X	Z	1800	1900
A	X	1820	1850
B	X	1820	1840
B	M	1840	1850
C	X	1820	1850

2) In addition to "part of," many other relationships can be tracked in the Instance Relation Table. For Example, "capital of," can be used to indicate which Admin Instance Record served as Capital of another Admin Instance Record, allowing for the function of "serving as capital" to be tracked on an independent timeline from either administrative unit. Similarly, we can keep track of temples (as Feature Instances) associated with certain monasteries, or, in fact, by creating a Feature Instance for a particular religious sect, we could keep track of all the temples and monasteries associated with that sect. Similarly, a feature instance may be created for a particular architect, and a set of structures associated with that person using a "designed by" relationship. There is no reason why the list cannot be expanded as needed, or can adopt such relational elements from other standards, such as the ADL content standard.

3) Temporal Sequence may also be tracked in this table, using a "preceded by" relation type.

Notes:

1) The most difficult event to capture, in terms of changes in the admin hierarchy, are those caused by changes in the area, or spatial object, related to the jurisdiction of any given administrative division. The method of handling this is briefly explained in the Instance Relation table in Administrative Instance Table, Note 1 above.

2) Special attention needs to be taken when updating records to make sure that all impacted Child units have their End Dates changed and new rows added to indicate the new instance of their rank in the administrative hierarchy.

Relations:

Many-to-many: Admin Instance Table
Many-to-many: Admin Name Linktable

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Admin Unit Table"**>>>>>>>>>>>>>>>>>

Admin Unit Table

Attributes: Admin Unit ID
Common Name

Objective: an authority table for all administrative units that appear in the database, with the idea that these units even if used in many different instances or for unrelated place objects, need not have more than one entry as long as they are identical.

Definitions:

3) Common Names are to be entered in Pinyin Romanization whenever possible, and are used purely for convenience of the database user, as a way of associating an administrative unit with an Admin Unit ID. The actual vernacular names in an unlimited number of native encodings, as well as the transliterations into any numb er of romanization systems are linked to this table using the Admin Name Linktable.

Relations:

Many-to-many: Admin Instance Table
Many-to-many: Admin Name Linktable

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Admin Name Linktable"**>>>>>>>>>>>>>>>>>

Admin Name Linktable

Attributes: Admin Unit ID
Language Table
Charset Encoding

Objective: a relational table allowing for any Admin Unit record to have multiple representations in an unlimited number of separate language tables.

Definitions:

1) Language Table is the name of a table in a vernacular script. For example, "CHT_admin" is being used for administrative unit names written in Traditional Chinese Characters, using BIG5 encoding. Any number of encodings can be used…." CHT2_admin" may be used for UTF-8 Traditional characters, and so forth.

2) Charset Encoding, as mentioned above, specifies which character set encoding has been used to enter the vernacular script placenames.

Relations:

Many-to-many: Admin Unit Table
Many-to-many: Language Tables (various)

>>>>>>>>>>>>>>>>>>>>>>>>Begin "Feature Instance Table">>>>>>>>>>>>>>>>>>

Feature Instance Table

Attributes: Feature Instance ID
Place Object ID
Feature Type ID
Begin Date
End Date

Objective: a relational table to show the one to many links between place objects and feature type instances. Feature Type Instance represents the temporal span during which a feature type remained unchanged in reference to a particular place object.

Definitions:

1) Feature Types are attribute values that may be assigned to any Place Object for a known span of time, which is referred to as a "Feature Instance." The Feature Instances may overlap both temporally and spatially. Feature Types do NOT include Placenames or Administrative Units, which are treated separately (as described above), but may include types found in an authority list such as the Feature Names found in the GNS Feature Designation (endnote 5), or the Alexandria Digital Library Feature Type Thesaurus.[10] However, it should be noted that the CHGIS project requires the creation of a substantial number of new terms for specific features found in the Chinese historical record, such as those recorded by Thomas Hahn.[11]

   The list of unique Feature Types used in CHGIS datasets, can be linked to all of the mentioned sources by means of the Other Classification System Table. This also will allow for searches based on those feature types derived from other systems, even if they have been named differently in the CHGIS usage.

Notes:

1) The Feature Instances provide considerable flexibility for the user in joining their own data to CHGIS base datasets. Let us imagine that a particular Place Object ID has been assigned to a Monastery. At the same time, the researcher knows that the Monastery served as a taxable unit at certain times, and these could easily be associated with the Monastery by creating rows in the Feature Instance Table and Event Table. Similarly the Monastery may have been a martial arts academy, or had a famous garden, or was on a parcel where paleontologists discovered dinosaur eggs ….all of these characteristics could be associated with the place object by creating Feature Instances, along with their appropriate Feature Types, Events, and Event Types. Interestingly, an Event could be recorded for the actual discovery date of a dinosaur egg, while the Feature Instance could record the time range during which the egg is believed to date from. (In this sense, one can think of the Event Table as a way of recording both Transaction Time and Valid Time as needed, while the Feature Instance would only record the Valid Time.)

2) Each Feature Instance should be associated with only one Feature Type. Multiple Feature Types that are related to the same place object for the same temporal span are treated as overlapping

attributes, not joined. For example a Feature Instance could be created for a certain climatic condition, and another created for a land use that had the same temporal span. This may be due to the sampling information and not to the independent variables of climate and land use. Whenever two feature types can be distinguished from one another, they should be. This can also easily be done after the fact, by simply changing the value of Feature Type in the Feature Instance Table.

Relations:

Many-to-one: Place Obect Table
Many-to-one: Feature Type Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin “Feature Type Table”**>>>>>>>>>>>>>>>>>>

Feature Type Table

Attributes: Feature Type ID
Common Name

Objective: an authority table for all Feature Types that appear in the database, with the idea that these types even if used in many different instances or for unrelated place objects, need not have more than one entry as long as they are identical.

Definitions:

1) See definition of Feature Type under “Feature Instance Table” above.

2) Common Names are to be entered in Pinyin Romanization whenever possible, and are used purely for convenience of the database user, as a way of associating an administrative unit with an Admin Unit ID.   The actual vernacular names in an unlimited number of native encodings, as well as the transliterations into any number of romanization systems are linked to this table using the Admin Name Linktable.

Relations:

Many-to-many: Feature Instance Table
Many-to-many: Feature Name Linktable
One-to-many: Other Classification System Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin “Feature Name Linktable”**>>>>>>>>>>>>>>>>>>

Feature Name Linktable

Attributes: Feature Type ID
Language Table
Charset Encoding

Objective: a relational table allowing for any Feature Type record to have multiple representations in an unlimited number of separate language tables.

Relations:

Many-to-many: Feature Type Table
Many-to-many: Language Tables (various)

>>>>>>>>>>>>>>>>>>>>>>>>Begin "Spatial Instance Table">>>>>>>>>>>>>>>>>>

Spatial Instance Table

Attributes:
- Spatial Instance ID
- Place Object ID
- Spatial Object ID
- Spatial Data Type
- Begin Date
- End Date

Objective: a relational table to show the one to many links between place objects and spatial instances. Spatial Instance represents the temporal span during which one spatial object remained unchanged in reference to a particular place object.

Definitions:

1) Spatial Instance is fixed on the temporal span for which a spatial object (associated with a particular place object) remained unchanged.   Spatial Instances do not contain the actual spatial object data, and can be represented by multiple Spatial Objects.

2) Spatial Object records contain actual spatial object data in GIS files. The Spatial Object records may be kept in separate object type files (as shown in this model for points, polylines, and polygons), or in a single geodatabase, depending on the software used in the implementation. One factor in keeping separate object type tables is to allow the user to select the spatial object types BEFORE actually going to the GIS files. The user would do this by first identifying the spatial instances of interest, then choosing to load the related spatial objects from Point, Polyline, or Polygon Tables.

3) Multiple Spatial Objects for each Spatial Instance are allowed by treating them as Alternate Spatial Data Versions, regardless of spatial object type. For example, multiple unique point objects could be related to a single Spatial Instance, such as label points, administrative office locations, or centroids of related polygons, as needed. Likewise, when we have mutliple polyline and polygon objects for the same Spatial Instance, these alternative spatial object versions are stored in the same spatial object tables with their own unique IDs, then linked to the default spatial objects using a Spatial Data Versions Table. Each object (and version) should have source information and explanations of variations recorded in the Data Source Table.

Notes:

1) FGDC Metadata is not stored in the CHGIS database. However, some basic spatial metadata is kept in the Data Source Table, and the Metadata Linktable will point to the location where the actual full FGDC Metadata records are stored.

2) Events that impact Spatial Objects are tracked in the Event Instance Linktable. In the case of a change in spatial object that impacts other Instances, for example, when a change in area impacts the Child records of an administrative division, a new row should be created in the Event table and links to the impacted Instances recorded in the Event Instance Linktable, including a link back to the Spatial Instance record.

Relations:

Many-to-one: Place Obect Table
Many-to-many: Event Instance Linktable
One-to-many: Point Table
One-to-many: Polyline Table
One-to-many: Polygon Table
One-to-many: Raster Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Point Table"**>>>>>>>>>>>>>>>>>

Point Table

Attributes: Spatial Instance ID
Spatial Object ID
Point Object Info
Point Object Type

Objective: spatial data file containing point data in GIS.

Definitions:

1) Multiple spatial objects can exist for any particular Spatial Instance, although one object will be considered to be the Default Object. For example, a label point can be created separately from a point showing the location of an administrative office, main crossroads, or other significant points related to the Spatial Instance. In the CHGIS database, the location of the administrative office of an administrative division is used as the Default Point Object, while all the others can be stored in the same table with their own Unique IDs, and linked together via the Spatial Data Versions Table.

2) "Alternate Versions," mentioned above, are all unique objects in the spatial data tables derived from conflicting sources. For example, one map source may indicate a certain boundary, while another source shows quite a different boundary. One of these boundaries is chosen for the CHGIS database to be the Default Object, in other words, the first object that would be represented for any given spatial instance. The other spatial objects depicting "alternate views" of different boundaries are stored in the same polyline or polygon table, given unique Spatial Object IDs, and linked to the default objects using the Spatial Data Versions Table. The notes provided in the Data Source Table must provide full details on the scale and projection of the source data, along other salient information that explains why the conflicting versions exist.

Relations:

Many-to-many: Spatial Instance Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Polyline Table"**>>>>>>>>>>>>>>>>>

Polyline Table

Attributes: Spatial Instance ID
Spatial Object ID
Polyline Object Info

Objective: spatial data file containing Polyline data in GIS.

See discussion under "Point Table."

Relations:

Many-to-many: Spatial Instance Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Polygon Table"**>>>>>>>>>>>>>>>>>

Polygon Table

Attributes: Spatial Instance ID
Spatial Object ID
Polygon Object Info

Objective: spatial data file containing Polygon data in GIS.

See discussion under "Point Table."

Relations:

Many-to-many: Spatial Instance Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Raster Table"**>>>>>>>>>>>>>>>>>

Raster Table

Attributes: Spatial Instance ID
Spatial Object ID
Polygon Object Info

Objective: spatial data file containing raster data (georegistered images, grids, DEM, etc).

Relations:

Many-to-many: Spatial Instance Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Spatial Data Versions Table"**>>>>>>>>>>>>>>>>>

Spatial Data Versions Table

Attributes: Spatial Object ID
Alternate Spatial Object ID
Begin Date
End Date

Objective: a relational table to show the many to many links between spatial objects and alternative versions of the same spatial objects, based on conflicting sources of data or interpretations of data.

Definitions:

1) "Alternate Versions," are all unique objects in the spatial data tables derived from conflicting sources, or from conflicting interpretations of sources. For example, one map source may indicate a certain boundary, while another source shows quite a different boundary. One of these boundaries is chosen for the CHGIS database to be the Default Object, in other words, the first object that would be represented for any given spatial instance. The other spatial objects depicting "alternate views" of different boundaries are stored in the same polyline or polygon table, given unique Spatial Object IDs, and linked to the default objects using the Spatial Data Versions Table. The notes provided in the Data Source Table must provide full details on the scale and projection of the source data, along other salient information that explains why the conflicting versions exist.

Relations:

One-to-many: Point Table
One-to-many: Polyline Table
One-to-many: Polygon Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Derived Spatial Data Table"**>>>>>>>>>>>>>>>>>

Derived Spatial Data Table

Attributes: Spatial Object ID
Min X Coordinate
Min Y Coordinate
Max X Coordinate
Max Y Coordinate
Spatial Data Type

Objective: a table of derived location data which can be used for simplified discovery of spatial objects in the database.

Definitions:

1) X – Y coordinates are to be used in the standard decimal degree format, and using the same datum, projection, central meridian as the CHGIS official dataset release (as described in the metadata.)

The Min – X Min – Y is used as the default for Point Objects, while the bounding box described using all four X – Y fields is used for Polyline, Polygon and Raster Objects.

2) Spatial Data Type indicates whether the Object is found in the Point, Polyline, Polygon or Raster Tables.

Relations:

Many-to-many: Point Table
Many-to-many: Polyline Table
Many-to-many: Polygon Table
Many-to-many: Raster Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Spatial Data Source Linktable"**>>>>>>>>>>>>>>>>>>

Spatial Data Source Linktable

Attributes: Spatial Object ID
Data Souce ID

Objective: a relational table showing the many to many links from Spatial Objects to the Data Sources used to digitize them.

Definitions:

3) Data Sources may be in the form of digital datasets, printed maps, or textual references. A full citation of the source is to be found in the Data Source Table.

Relations:

Many-to-many: Spatial Instance Table
Many-to-many: Spatial Data Versions Table

>>>>>>>>>>>>>>>>>>>>>>>>**Begin "Data Source Table"**>>>>>>>>>>>>>>>>>>

Data Source Table

Attributes: Data Source ID
Data Citation
Data Notes

Objective: a table containing complete bibliographic citations of Data Sources used to digitize all of the spatial objects in the database.

Definitions:

1) Data Citation will include a number of fields in the form of a standard bibliographic citation, with some additional geographic attributes. A recommended model for the latter fields are found in the ECAI Metadata Standard.[12]

2) Data Notes will include specific information about how the data source was interpreted in order to complete the digitization of the related spatial object. This is especially important to differentiate alternative Versions of spatial objects.

3) Complete FGDC Metadata will be linked to, when applicable, using the Metadata Linktable.

Relations:

Many-to-many: Spatial Instance Table
Many-to-many: Spatial Data Versions Table
Many-to-many: Metadata Linktable

>>>>>>>>>>>>>>>>>>>>>>>**Begin "Metadata Linktable"**>>>>>>>>>>>>>>>>>

Metadata Linktable

Attributes: Data Source ID
Metadata URL

Objective: a table containing links to complete FGDC Metadata for Data Sources used to digitize spatial objects in the database.

Notes:

1) Not all of the Data Sources will be GIS datasets. However, for all Data Sources that are GIS datasets, the fulle FGDC Metadata will be extracted and posted on the CHGIS website for easy retreival. This table will provide links to the metadata.

Relations:

Many-to-many: Metadata Linktable

>>>>>>>>>>>>>>>>>>>>>>**Begin "Source Linktable"**>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Source Linktable

Attributes: Source ID
Event ID, Placename Instance ID, Admin Instance ID, Feature Instance ID, Spatial Instance ID

Notes:

1) Sources refer to textual or printed map sources used in the compilation of records in the database. These sources are different from Data Sources in the sense that they are not used specifically for the digitization of spatial objects. However, Sources can indeed be related to Spatial Instance records, as they can to any other temporal Instance records. The idea is for any particular temporal instance in the database, (whether it be Event, Placename, Admin Status, or other Feature

instance,) to have links to all the historical text and map sources used for gathering the information being recorded.

Relations:

Many-to-many: Event Table
Many-to-many: Placename Instance Table
Many-to-many: Admin Instance Table
Many-to-many: Feature InstanceTable
Many-to-many: Spatial Instance Table

>>>>>>>>>>>>>>>>>>>>>>**Begin “Source Table”**>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Source Table

Attributes:	Source ID
Source Citation

Definitions:

Source Citation will include fields for a standard bibliographic citation, conformant to (but not necessarily completely fulfilling) concise MARC format.

Relations:

Many-to-many: Source Linktable

>>>>>>>>>>>>>>>>>>>>>>**Begin “Commentary Linktable”**>>>>>>>>>>>>>>>>>>>>>>>>>>>>

Commentary Linktable

Attributes:	Commentary ID
Event ID, Placename Instance ID, Admin Instance ID, Feature Instance ID, Spatial Instance ID

Notes:

1) Commentaries refer to textual descriptions and interpretations of sources that were used in compilation of records in the database. In addition to actual quotes from sources, source citations, and historical commentaries on the sources, the researchers’ comments (and their names) will be provided. Commentaries, like Sources, can be related to any temporal instance records in the database, as needed.

Relations:

Many-to-many: Event Table
Many-to-many: Placename Instance Table
Many-to-many: Admin Instance Table
Many-to-many: Feature InstanceTable
Many-to-many: Spatial Instance Table

>>>>>>>>>>>>>>>>>>>>>>Begin "Commentary Table">>>>>>>>>>>>>>>>>>>>>>>>>>>>

Commentary Table

Attributes: Commentary ID
Commentary Text

Notes:

See description in Commentary Linktable.

Relations:
Many-to-one: Place Table
Many-to-one: Event Table

>>>>>>>>>>>>>>>>>>>>>>Begin "Other Classification System Table">>>>>>>>>>>>>>>>>>>>

Other Classification System Table

Attributes: Feature Type ID
Other Classification ID
Other Classification Name
Other System Name – Version
Other System URL

Objective: an relational table linking Feature Types that appear in the database with corresponding classifications used in other systems, such as the ADL standard (footnote 9).

Definitions:

1) See also "Feature Type Table" above.

2) Other Classification ID, for example, the "numerical equivalents for feature type terms" in ADL Standard.[13] Where ADL "caves" ID = 30 can be related to the correspoding CHGIS Feature Type ID. This will allow for feature type searches to be passed from one system to another.

Relations:
Many-to-one: Feature Type Table

---

[1] Intl Workshop on Historical GIS, Shanghai, 2001.
http://www.people.fas.harvard.edu/~chgis/meetings/shanghai.html

[2] "Historical GIS for 200 Years of Belgian Territorial Structures," Intl Workshop on Historical GIS, Shanghai, 2001. http://www.people.fas.harvard.edu/~chgis/meetings/papers/demoor_shanghai.pdf

[3] "Solution to the Parent-Child Multiples Problem in the Spatio-Temporal Database Design for the CHGIS Project," Intl Workshop on Historical GIS, Shanghai, 2001 http://www.people.fas.harvard.edu/~chgis/meetings/papers/crissman_paper.doc

[4] "China Historical GIS Database Model," Intl Workshop on Historical GIS, Shanghai, 2001 http://www.people.fas.harvard.edu/~chgis/meetings/papers/dbase_0909/

[5] See Linda Hill's discussion of the concept of "place" and "satisficing," "Core Elements of Digital Gazetteer's: Placenames, Categories, and Footprints." In J. Borbinha & T. Baker (Eds.), Research and Advanced Technology for Digital Libraries : Proceedings of the 4th European Conference, ECDL 2000 Lisbon, Portugal, September 18-20, 2000 (pp. 280-290). Berlin: Springer. Draft online: http://alexandria.sdc.ucsb.edu/~lhill/paper_drafts/ECDL2000_paperdraft7.pdf

[6] "GNS Feature Designation Cross Reference" http://164.214.2.59/gns/html/fd_cross_ref.html

[7] The levels of temporal uncertainty are indicated in the "Begin Date Rule" "End Date Rule" fields, according to the ideas set forth in Man Zhimin's "Zoujin shuzihua – Zhongguo lishi dili xinxi xitong (CHGIS) 1820 nian shuju jianjie" [Going Digital – A brief introduction to the China Historical GIS 1820 database]. CHGIS Database Meeting, Harvard University, Cambridge, August 9th, 2001. See page 7 for temporal uncertainty guidelines. http://www.people.fas.harvard.edu/~chgis/meetings/papers/man_chgis_demo.doc

[8] "An Area Event System for Forest Management History," ESRI, April 2001. http://fas.harvard.edu/~chgis/work/docs/papers/forestwp.pdf

[9] A full explanation of this usage is found in "Toponyms and Feature Classifications for the China Historical GIS," presented at the Pacific Neighborhood Consortium and Electronic Cultural Atlas Initiative joint meeting, City University of Hong Kong, January, 2001. http://www.dbr.nu/data/pubs/papers/lex_gazteer_pnc01.pdf

[10] Alexandria Digital Library Gazetteer Feature Type Thesaurus. http://alexandria.sdc.ucsb.edu/gazetteer//FeatureTypes/index.htm

[11] List of Spatial Elements in Chinese Mountain Gazetteers, compiled by Thomas Hahn. http://www.library.cornell.edu/wason/gazetteers/

[12] ECAI Metadata Standard, Version 1.1 (Mar 1999) http://www.ecai.org/documentation/ecai_metadata_standard.html

[13] ADL Feature Type Terms. http://alexandria.sdc.ucsb.edu/~lhill/adlgaz/FTTermNumbers.htm