SOLUTION TO THE ‘PARENT-CHILD MULTIPLES’ PROBLEM IN THE SPATIO-TEMPORAL DATABASE DESIGN FOR THE CHGIS PROJECT

When hierarchical administrative codes are assigned to uniquely identify administrative units in row version RDBMS tables, if a higher-level unit changes then new row versions of all lower-level units need to be created. That would quickly lead to hugely unmanagable tables as higher level changes accumulate through time and code changes proliferate at all lower levels. A workable solution to this problem follows. It primarily requires a change to the CHGIS database design to incorporate ‘Place Codes’ instead of Administrative Codes entailing new Place Change Tables instead of the old Administrative Change Tables described in Version 01210 of the CHGIS DRAFT DATABASE DESIGN AND GEOCODING SYSTEM. However, that document will be revised in other ways as well to reflect the evolution of the design.

PRINCIPLES

The basic features of the solution are as follows (terms defined below are italicised):

1) The CHGIS Spatio-Temporal Database Design will be implemented with spatial objects in ArcGIS 8, ArcView, or MapInfo, etc., obviating the need for ‘space-time composites’ such as an ‘atomic decomposition model’ which are a mess to implement.

2) *Time* of existence of a place needs to be included and be selectable at all levels in the hierarchy. To find a place, one must always query when (at what *time*) it existed as well as where it was (what position it had) or what it was (what type of place it was or what its name was). In other words, ‘What is/was this place?’ or ‘Where is/was this place?’ cannot be asked without including ‘At some particular time?’ Most SQL queries of the CHGIS spatio-temporal RDBMS will begin with specifying a *time* or *point in time*.

3) Distinctions need to be made between a ‘place’ (an abstraction), the ‘location’ or ‘territory’ thought to be occupied by a place at some time, and the representations of places as GIS spatial objects of various kinds or resolutions, etc, or on paper maps or raster images.

4) The number of ancillary coding changes in parents and children when a new place is established in the database or when an existing place undergoes some change are minimised through the use of ‘Place Codes’.

5) Place codes’ are arbitrary, non-hierarchical alphanumeric strings that are employed to encode the children of parents at the times when the places undergo changes of various kinds. Place Codes will be used for the primary coding identification of entries in the database for such things as administrative units as well as other kinds of features.

   Although they consist of arbitrary alphanumeric strings that are not based on any kind of administrative hierarchy, Place Codes can also optionally at least in part incorporate some temporally or spatially related information in order to be able to identify their general times and locations without having direct reference to their precise dates and locations, or to lookup tables.

   Children (subordinate units, or *subs*) need to ‘know’ who their parent (superordinate unit, or *sup*) was at any *time*, (be coded with their immediate parent’s Place Code), but parents need not ‘know’ (be searchable for) their children, except by querying the relevant sub-level(s) to find which constituent units are assigned to the particular parent at a particular time. Note that changes to a parent (establishing new *manifestations*, or *row versions*) that do not affect a child do not require new *manifestations* for such unaffected children, as they are still the same places after such changes and still have the same parent place.

6) A distinction needs to be made between changes to places (different *manifestations*) that involve altered *positions* or *territory* (*versions*) from non-spatial changes (*instances*) such as different names or statuses, but all such *manifestations* need to be time coded. *Variations* in places based on different sources can be treated as either *versions* or *instances*, as appropriate.

7) When a parent changes its *territory* there is inevitably some change in the *position* or *assignment* of one or more of its children, possibly including new ones. However, there is no need to change any of the *assignments* to the parent or *times* of any of the other children that are not involved in the *position* change, as their *manifestations* do not change, *per se*

8) When a child (or set of children with the same parent) undergoes a change to its *position* (*territory* or *footprint*), it is only necessary to create a new parent *version* if its *location* also changes as a result. If the parent *territory* does not change, then there is no need to record a *version* (spatial) change for the parent as its children are already selectable at lower level(s) by their *times*. An *instance change* could be established with a new parent row version, or a new row version describing the change in children could be inserted into the relevant Administrative Feature table, but this would be redundant and unnecessary.

9) *Instance* changes in parents do not affect their children and therefore do not require any new records for them (except in Administrative Code Tables, if any are maintained). Nor do *instance* changes to children have any effect on their parents, and do not require new *instances* of them.

10) Place codes have *version* suffixes that are employed as the unique identifiers that link to the geocodes of their representations as GIS spatial objects or other depictions. *Instance* suffixes appended to *version* suffixes are optional, as each *instance* of a place is searchable in one table or another by its time period and *instances* do not link to geocodes.

11) When for some good reason a hierarchical administrative code is required, SQL queries can be made at successively higher levels to find their arbitrary adm_code components and concatenating them using the SQL script produced by Miho Berman.

DEFINITIONS

Some definitions will be useful in the following discussion (terms in italics are also defined):

Administrative Codes – Arbitrary, hierarchical alphanumeric codes that uniquely identify *administrative units* in terms of code elements assigned at the various relevant levels of an administrative hierarchy.

Administrative Feature Table – A RDBMS table that has *row versions* for each instance of an *administrative unit*.

Administrative Place Change Table – A Place Change Table containing administrative units, or places that occupy formal positions in hierarchical administrative systems.

Administrative Units– *Places* at various levels in an administrative hierarchy. Administrative hierarchies simplify *sub/sup* relationships, as normally a *sub* belongs to one and only one *sup* in an administrative system.

Assignments – The coded relationship of subordinate places (*subs*) to a superordinate *place*(*s*) (*sups*) for some time period. This term embodies the 'part-of' relationship, or the relation between a child and its temporally defined parent(s). For administrative places, *subs* (subordinate units) are assigned to *sups* (superordinate units) for a particular time period. ' Is_Part_Of' or 'Sup_Plcode' are possible field names to hold *sup place codes*.

Attribute – Something that is known about a *place*, such as its name, its *positiont*, or in the case of an adminstrative unit, its capital, including the *time* of the *manifestation*.

Change – Anything that causes the attributes of a *place* to differ before and after a particular point in time, or which is different in alternative sources and establishes a new *manifestation*. *Changes* require new time-marked *versions* or *instances* of a *place*, depending on whether the *positiont* of the *place* differs due to the *change*. A change to the capital/seat of an administrative unit only requires a new *version* in order to enable its representation by capital/seat locations.

Event – A transitory phenomenon that changes its position over time, such as a typhoon or even an empire when looked at in a broad perspective.

Field Names – The specified ten character designation of the fields in the RDBMS tables. The ten character designation is a requirement of ArcView .dbf files based on DBase 1.0 or some other anachronism or stupidity.

Footprint – The *location* defined by either a lat/long point or an area defined by the lat/long of the lower left and upper right corners of the decimal degree rectangle that bounds a *place*.

Geocodes – Alphanumeric codes that identify spatial objects (point, line, area, and region representations) which are stored in some spatial data base, such as an ArcGIS coverage or ArcView shapefiles and MapInfo file sets, so that they can be selected to represent *places*.

Hybrid Place-Feature Table – A RDBMS table that includes *row versions* for both *versions* and *instances* of a set of *places.* In other words, place changes and feature changes are combined so that a temporal search will produce selections of all *manifestations*. There could be advantages in some implementations to such hybrid tables.

Instance (of a *place*) – This type of *manifestation* of a *place* involves differences in the non-spatial (non-territorial) *attributes* of the same place after some change and before another such change. Examples would be name or status changes. Instances do not need to clutter up Place Change Tables, but can be held in Administrative Feature Tables.

Location – Normally a precise designation in terms of latitude and longitude of sites in terms of points or *territories* (lines, areas, or regions), usually on the earth's surface, but vague or fuzzy locations must be recognised and catered for as well as those that are not thought to be on the earth.

Manifestation (of a *place*)– The existence of a *place* for a *time*. This term is used as a generic for *variations*, *versions*, and *instances* of a place.

MR_Code – The second term in a concatenated Place Code will provide a general indication of where the place is located in terms of China's macroregions and their subregions.

Place – Some object thought to occupy or constitute some space (or *location*) at some *time,* normally on the earth's surface in the historical GIS context. Depending on their nature, some *places* can be parts of more than one *place* at a 'higher' or more encompassing *place*, which would require multiple *row versions* for each such parent, one to identify each 'sup place'. Allternatively, different Place Change Tables could be established if there are many such systematic *variations*.

Place Change Table – A RDBMS table that contains *row versions* for each *version* of a place, ie., each *manifestation* that has a different *territory*.

Place Code – An unique designation for a *place*, which could in principle be its toponym in a set of non-identical toponyms. However, arbitrary alphanumeric codes are preferred for place codes, as names of places can change or vary with sources/transliterations without otherwise changing the nature of the place itself. Although *place codes* can be randomly or sequentially assigned or be otherwise arbitrary, it is also possible and perhaps desirable to load them with useful spatial or temporal mnemonics, at least to some degree.

Point in Time – A specific calendar date, but perhaps an arbitrarily defined one, that can be used to select the *times* when an *instance* or *version* of a *place* existed.

Position (or **Territory**, or **Site**) – a space believed to be occupied or constituted by a place. Can be the same as a *location* or *footprint* if so defined in terms of latitude and longitude.

Row Versions – Multiple cases in a RDB table for *versions* and *instances* of the same *place*. The term 'version' when used in '*row version*' is not always the same as *version* defined below, although different *versions* of a place (but not necessarily instances) do always require separate *row versons in a Place Change Table*.

Site – (See *Position*).

Sub – Subordinate administrative unit assigned to a particular *sup* at some specified time range. If there is no difference in the footprint of a *sub* if its assignment changes, then only different *instances* of the *sup* are created although new *versions* of its prior and latter *sups* will be needed. 'Child' can be used as a synonym.

Sup – Superordinate administrative unit whose place code is in the 'Is_Part_Of' or 'Sup_PlCode' field in all of its *subs*. If a change in a *sub* version or assignment changes the footprint of its *sup*(*s*), new *sup version*(*s*) are also required. 'Parent' can be used as a synonym.

Territory – (See *Position*).

Time – The period defined by 'Begin_Date' and 'End_Date' fields between *changes* when an *version* or *instance* of a place exists. See note 4 below for a discussion of what to do about uncertainties in the dates of *changes*.

Time_Code – The first term in a concatenated *Place Code* will provide a general indication of the time span of a *place* in terms of dynasties and regimes.

Variations (of a *place*) –Manifestations of a *plac*e that differ according to alternative sources of information or different regimes, etc. In the case of administrative units, the variations can be treated as either *versions* or *instances* depending on whether there are differences in the *footprints* of the *variations*. If there are a large number of variations, it could be desirable to establish separate Place Change Tables based on the different sources or for different regimes, etc.

Versions (of a *place*) – Differences in a *place* that involve different *locations* before/after some *time*. In the case of *administrative places*, such *changes* involve gaining or loosing *territory* or a changed *position* of a capital/seat. *Versions* of a place created by such *changes* require new *row versions* to be established in a Place Change Table for all affected places. Assignments of *subs* to different *sups* that do not involve changes to the *sub* territory only require new *sub row versions* in Spatial Change tables, rather than instances in an Administrative Change Table, in order to facilitate searches for all *subs* in a single table. Changes to the location of the capital/seat of an administrative unit requires a new *version* only in order to enable the representation of the place by such locations.

NOTES:

1. When a new *instance* resulting from a change in non-spatial attributes of an administrative unit (*place*) is added at the appropriate level in an Administrative Place Change Table, no *changes* are necessary in either higher or lower level units (*sups* or *subs*)– only a new i*nstance row version* needs to be added at that level with the appropriate *time* (begin and end dates), and any resulting modifications to other before or after *instance times* for the same *place*. See Example A. Alternatively, as Pu Lin points out, non-spatial changes (other than a change in capital/seat) need not be contained in a Place Change Table, but can be recorded in other tables as new row versions, such as the Administrative Feature Table for a name change. The point, with respect to either alternative, is that searches for places and their versions and instances must always be done on **both** time **and** place.

2. If a spatial change in a *sub* or set of *subs* causes a change in territory for one or more *sups*, then new *row versions* of the affected *sups* are also required, but **only** for those affected. This applies at all successive superior levels in a hierarchy. See Example B.

3. If a spatial change in a *sub* (or more likely a pair or more of neighbouring *subs*) does not affect the overall *territory* of its *sup*, **no** new *version* of the sup is required, as the *point in time* of the *change* will allow selection of the various components of the *sup*. Only the constituent units at the lower levels (*subs*) whose *territories* are directly affected by the *change* need new before/after *row versions* that have new begin-end dates reflecting their *changes*. Sibling *subs* not affected by the *change* require no new *versions* or *instances*. See Example C.

4. Berman has proposed an excellent solution for uncertainty concerning the time of a change, which is to use the date of a preceding or subsequent change if the beginning or ending data of a manifestation is unknown. However, his example marks such uncertain dates with a following asterisk which could interfere with SQL queries. Crissman would elaborate on the idea by adding additional fields relating to begin- and end-dates that indicate the nature of the uncertainty, perhaps as in 'firm, best guess, possible, or unknown', etc., and that provide the sources for the status of the dates.

5. If a unique *administrative code* representing all the higher-levels to which a unit belongs is required for some purpose, it can be generated as needed by working up the hierarchy of administrative levels using the 'Is_Part_Of' or 'Sup_Plcode' field and concatenating the *administrative codes* that pertain to manifestations at a particular *point in time* at each level. Administrative Code Tables could thereby be created for as many *points in time* as required.

CODING GUIDELINES:

Administrative Codes:

Basically, since Administrative Codes are not used internally in this new CHGIS Spatio-Temporal Database Design, but are only output from it when someone wants them for some purpose, which of possible alternative formats they employ is therefore moot. A possible scheme for assigning Administrative Codes is described in a companion document, AC.dot.

Geocodes:

Geocodes are employed in this new CHGIS Project STDB Design only to identify and select GIS spatial objects or other representations that represent selected *places* at seletcted *times*. *Place Codes* can be used as *Geocodes* for a specific 'slice in time' such as 1820, particularly if the attribute data tables in a GIS are used as Place Change Tables. However, distinct Geocodes will be necessary if the same spatial objects will be used for different *manifestations* of places, as will be the case in a spatio-temporal database, and will also allow alternative spatial or other objects to be used to represent the same place(s). Geocodes could be arbitrary or even random, and it is desirable that they can be assigned automatically. It is proposed that in the first instance, at least, distinct geocodes be based on the longitude and latitude of spatial objects themselves, as they can then be automatically generated from an ArcGIS coverage, although an operator will need to manually select an identifying point for polypoint, polyline, and region objects.

Point object geocodes will be created by concatenating their long/lat locations to three decimal places with an intervening comma.

Polypoint obect (consisting of multiple points) **geocodes** will be created by similarly concatenating the long/lat location of one arbitrarily selected point in the collection.

Line object geocodes will be created by similarly concatenating the long/lat locations of their mid-points.

Polyline object (multiple line objects, including networks) **geocodes** will be created by similarly concatenating the long/lat location of an arbitrarily selected point on one of the constituent lines

Polygon object geocodes will be created by similarly concatenating the long/lat of their label points in an ArcGIS coverage, or by arbitrarily selecting a point within them in ArcView and MapInfo, and concatenating its long/lat.

Region objects (multiple polygons) **geocodes** will be created by similarly concatenating the long/lat location of an arbitrarily selected point in one of the constituent polygons.

Place Codes: These are central to the operation of the CHGIS STDB Design, as place codes are the primary means of identifying places in Place Change Tables and joining them to their attributes or other features in various associated RDBMS tables, including Spatial Data Tables.

Although Place Codes could be completely arbitrary and therefore be assigned on a completely random basis, the results of such a procedure would be inherently confusing to human beings, as identifying a place by such a code would inevitably involve using a look-up table. Therefore, Crissman is of the opinion that place geocodes should be loaded with certain amount of mnemonic baggage to aid humans who will be establishing Place Chang Tables or using place codes for certain of their purposes. These non-random components of place geocodes will allow people to identify the general types, rough time periods, and general locations of places without interfering with the need for flexibility in adding or deleting places in some sets of places. However, loading Place Codes in this manner is optional.

One possible scheme for incorporating meaningful elements in Place Codes is described in a companion document, PC.doc.

Version – If only one version of a place exists with in the *time* indicated in Time_Code, then .N can be used in the Version field. Alternatively, some other symbol compatible with standard SQL queries could precede *version* codes to set them off from the other components of a place code when their elements are concatenated. Additional *versions* of a place would then be indicated with capital letters from before or after N, beginning with F and T, respectively, following the point or other symbol. If it is necessary to insert an additional version between two already coded ones, such as .N and .T, then .Q could be used, etc. If an additional version were required between two sequential letters, then a lowercase letter beginning with n can be suffixed, etc., ad infinitum

Instance – Instances will be indicated with numerals added behind any version suffix, beginning with 01 for the first of two, nothing being required if there is only one instance of a place. Subsequent instances will proceed numerically up to 99 and then go to 0a, 0b, etc., as necessary. If an additional instance of a place is needed between two already coded, such as 13 and 14, 13.5 can be used, and 13.7 for one between 13.5 and 14, etc., using as many decimal places as necessary, ad infitum.

Concatenation – It will be necessary to concatenate a place code from its components in order to use them for sup coding. In the scheme described in PC.doc, they would look like A2QI1Np01.B03 for the third instance of the second version of Shuntian Sheng containing Beijing in the late Qing (or A2,QI,1Np,01.B03 if commas are used to separate the components, apart from versions and instances). Alternatively, A5QI1Np01.B03 could indicate the third instance of the second version of Shuntian Fu, it having been an Administrative Place at either provincial or prefectural level depending on the source consulted. However, apart from their version and any instance suffixes, the basic Place Codes could be any arbitrary alphanumeric string instead of such examples of loaded Place Codes, as described in PC.doc.

PLACE CHANGE TABLE COMPONENT FIELDS

1. **Place_Code** – A concatenation of fields 2-7.
2. **P_Type** – [OPTIONAL] A set of usually two alphanumerics denoting what kind of place the code pertains to. If an administrative place is coded as A, then the numeral for its level is in the second place.
3. **P_Time** – [OPTIONAL] The Dynasty/Regime code, or combinations as discussed in PC.doc.
4. **P_Mregion** – [OPTIONAL] The general location in terms of macroregion/region and sector, as described in PC.doc.
5. **P_Number** – A randomly assigned alphanumeric string. In the scheme discussed in PC.doc, only two digits are required.
6. **P_Version** – Version letter(s) preceded by a '.' or other demarcating symbol, as discussed above
7. **P_Instance** – Instance numerals, as discussed above.
8. **Begin_Time** – Numerals for year, month, and day, insofar as possible, as discussed in the CHGIS STDB Design document.
9. **End_Time** – Numerals for year, month, and day, insofar as possible, as discussed in the CHGIS STDB Design document.
10. **Sup_P_Code** – The Place Code for superordinate places. If a place is part of two or more immediately higher level places, two or more row versions are required for their Place Codes, unless there is a separate Place Change Table for a dual hierarchy of places.
11. . **Change_Typ** – [OPTIONAL] An single word indication of what kind of change initiates the *manifestation*.

12. **Geo_Code** –A longitude/latitude derived arbitrary code for linking to spatial objects in a spatial database. Optional if there is a separate Place to Geocode Table.

SOME POSSIBLE ADDITONAL TABLES LINKED WITH PLACE CODES:

Place Continuity Table – Contains Place Codes and other information on predecessor and successor places, including components of *versions*.

Place Feature Table – Contains information such as toponyms, as well as capital seats for Administrative places.

Alternative Names Table – Can include different languages and orthographies as well as different spellings, etc.

Place to Geocode Table – This table would be particularly useful if different kinds of objects could be selected to represent places.

Administrative Change Table – Records changes to administrative units, and could be used to generate Administrative Codes.

POSSIBLE TABLES LINKED WITH GEOCODES:

Spatial Data Tables – At least one such is required to indentify GIS representations of a selection from a Place Change Table. An ArcGIS coverage or sets of ArcView shape files or MapInfo tab files can serve as Spatial Data Tables. Different Spatial Data Tables that utilise the same Geocodes can contain alternative spatial objects at various resolutions or derived from different base map data, etc., or they can have their own sets of geocodes derived from their objects that link to the same set of Place Codes incorporating Version suffixes.\

Maps or Raster Image Indexes – Geocodes as well as Place Codes can identify other kinds of place representations or displays of places.