## Data Modeling Demystified NEARC Spring 2013

Brian Hebert, Solutions Architect www.scribekey.com

#### Abstract and Goal

Classic relational database and object-oriented *modeling diagrams, tools, and techniques* are often too *complicated* for GIS users and project managers to fully understand. This presentation will demonstrate a *simplified approach to modeling* geospatial feature classes and tables, using simple *attribute lists* generated through data profiling. Attendees will then be able to *apply this useful technique on their own data projects* for either improving existing data models, or creating new ones.

Goal: Provide you with a detailed but straightforward method you can use with your own projects and data. Additionally help with better understanding of classic modeling tools and techniques.

## Outline

- Review object oriented and relational modeling
- Discuss RDB normalization
- Identify a communication/skills gap
- Describe refactoring
- Walk through an alternative 3 step table based modeling technique
- Review domains
- Recap
- Q & A

## Data Models and Modeling

- Application: Object Oriented (OO) Class Diagrams – Generate C#, Java, C++
- Relational Database: Entity-Relationship (ER) Diagrams – Generate SQL
- Analysis, N-Dimensional Cube: Star, Snowflake Diagrams (generally unfamiliar to GIS practitioners) – Create Cube

## Why Both OO and RDB?



ArcGIS is an Object-Relational System (ORDB). Data and application metadata are stored in the backend geodatabase, loaded into middle tier, written with OO C++ code, and presented through rich GUI presentation applications, ArcMap, ArcCatalog, ArcToolBox, etc. As such, we use both OO and RDB modeling techniques.

OO Databases never really made it. This class 3 tier approach is how OO data models are implemented.

The current trend is to use tools like UML to design OO systems.

## UML OO Diagraming



UML (Unified Modeling Language) diagrams capture type hierarchies. They show **type family trees**, focusing on how more specific sub-classes inherit properties and behavior from more general super-classes.

Becoming familiar with UML and related tools requires significant time and experience.



#### Designing in ArcGIS: CASE Tools or ArcCatalog



**ESRI CASE Tools** 

#### ArcCatalog Tools



Geodatabase Diagram

With CASE or ArcCatalog Tools, the goal is the same, to generate a Geodatabase

www.scribekey.com

#### **RDB Entity-Relationship Diagrams and Tools**



Entity Relationship diagrams and related tools are typically used to create highly normalized, editing/production databases. Normalization protects referential integrity and prevents anomalies that can compromise data quality, break relationships, etc.

The models we typically use in GIS are often highly de-normalized. Denormalized databases are much easier to query and use for maps and reports.

#### De-Normalized Data Example: Navteq



If Navteq Streets data were normalized there would be multiple separate tables for street names, town names, states, zip codes, street types, etc. If this were the case, creating Geocoding or Routing applications would be MUCH more complex SQL Query: De-Normalized and Normalized

SELECT ID, NAME FROM STREETS WHERE NAME='EIm St'

SELECT STREETS.ID, STREET\_NAMES.NAME FROM STREETS INNER JOIN STREET\_NAMES ON STREETS.STREET\_NAME\_ID = STREET\_NAMES.ID AND STREET\_NAMES.NAME='EIm St'

Queries for normalized data are much more complex. Additionally, if data is being used in a read-only mode, there is typically no reason for highly normalized models.

WARNING! In spite of this, many data RDB data modelers start a project with the assumption that the data should be highly normalized.



## An Alternate Technique

- 1) Use database tables to capture/store information about feature classes, tables, attributes, and domain values. This is metadata.
- 2) Arrange lists of attributes and entities in cross reference matrices.
- 3) Use SQL queries to discover your classes, and help you create your entities.

GIS users are typically very familiar with data in tables. There is no immediate need to learn an unfamiliar or complex diagraming technique to do what you need to do.

# This helps involves the people that know the most about the data in the modeling process.

### Scenario

- A university has asked us to integrate existing tabular data describing Students, Faculty, and Staff, with GIS building layer.
- They have asked for a class diagram.
- They also want to create tables in ArcGIS
- They want to determine who is where, when, etc.
- The focus is not GIS per se, but will help us concentrate on important concepts.
- Reduced attribute sets will be used.



#### Attribute Lists

STUDENTS	FACULTY	STAFF
Student_Name	PROF_Name	Name
ST_DOB	DOB	Birthday
Major	Department	Title
GPA	Salary	Supervisor
Student_Year	Tenure	Salary
Dormatory	Highest Degree	Office
Advisor	Office	

From an initial view of the 3 entity attributes we see that they share some common general attributes as well as having some specialized attributes of their own.

## Sidebar: Refactoring

- A great deal of complex OO and RDB data modeling is carried out in data-free environments, assuming a blank slate to start with.
- In reality, this is rarely the case, and we typically have to fit existing data into our models.
- As such, it is typically helpful to use a data-driven approach to modeling, where the focus is on changing existing entities, rather than creating new ones.
- This is related to a technique called REFACTORING.



#### Creating Entity and Attribute Metadata Tables

	nating Constant - secting Constant - market of Lange (Sec arthough Constant) arthough Constant - arthough Con	E lone ⇒ lonetre Blance None La Solo Tag		Latin Latin Millionit <sup>+</sup> + 7 Li	- (日 (本 - 型 - 本) Test terretre	□□□ (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	
<u></u>	Million .						
	CREET + DR NM + 120 HM	- DRIVER	COL ALAS -	DATA OVE - DR	A KIN - KING	+1 (0) 001 +	DIP SAC + 14
	1 st, louis anashri	F0 P	0	00	1		
	2 st. hosia amaint	Shan 2	lage .	Occupation			1
	8 st, kun analm	STATUP ST	RETATION	string	2.9	Exment state Pederal (*formation Processing Standards (*195) code	U.S. Conset Ro
	4 m Jours ( anstern	LOUNTER D	CONTENT.	trong	1.1	Lummit county Vesseni information Proteining Standards (Hiffs) code	105. Certae Bul-
	s rt jours arrains	ANDICODE A	NUCCER	string	H . F	Exment official eads for the landmark for use by federal agrinism for data to	U.S. Comment Res
	and local aready	AACANI	MAR	22.mg	22 Y	Area landmark cleritifier	U.S. Certail Br
	7 at louis praster	PALMANE PA	ALAAN)	10mm	100 V	Enhantemation of expanded heat for prefix qualifier, prefix direction, prefix	U.S. Carries Bu
	Bal, bola arealm	MITTEC M	micc	31114	2.1	WAT/THER leature class tode	U.S. Cerura Bu
	# sh, louis arealm	ALAND A	UAND.	Double	34.9	Convert land area	U.S. Censel Bu
	20 st_Jovis arashm	AVEATER A	WATER	Deuble	24.9	Content water and	6.5. Conset Bur
	II multure president	INTERNAT IN	TAITTE	trong	10.14	Lument instale of the internal point .	U.L. Cernat Ru
	12 m_kuns arcsim	ANTETION IN	arreph.	anne	32.9	Current long rude of the internal point	U.S. Crimit Bu
	13 if loss interator	CO: 11	0	00			
	25 pl Joseph anti-anti-	Diego D	wire .	Geumeter	10		
	15 st, Josis presmater	STATUP II	TATS/P	Tiring.	- 2	Commit state Pederal Information Processing Renderth (FIPS) code	U.S. Cernet Ro.
	16 st_kovis propriater	countries of	OUNTY##	tong		Current ocurry Reserval Information Processing Standards (FIPE) code	U.S. Consei Ru
	17 il Join assentar	ANDICODE A	NSICODE	10.1%	1	Commit official and a for the water body for use by foderal aporties for data	6.5. Conset Ru
	IR et jours arrowater	AVTIMENT AV	CONDO.	trong	22.9	Area hydrography identifier	U.S. Cernatific
	19.9 jours presupter	FULLNAME PL	RINAME	String	180 *	Concatenation of expended text for prefix qualifies, prefix direction, prefix	U.S. Cernet Bu
12	20 m louis ansavatar	MULC M	HPCI .	String	- 5	MAS/TISES feature class mide	115, Center Ro.
	21, pt, you's an any only for	ALTINE A	UND	Double	34	Exminit land area	M.S. Certain Ba
	22 st Josis erestrater	AVAULTER A	WATCH	Dusble	36 V	Current water area	U.S. Cernan Bur
	22 st. Jouis arconater	BATETLAT IN	TRUTT	tong	33, V	Exment Schede of the Internel point	U.S. Certain Ba
	J4 t1_XXX intervator	INTERION IN	INCOME.	string	11.1	Current longitude of the internal peak	U.S. Contait Bu
	25 m_10.04 3800	10 1	0	040			
C	26 st blues 1400	Shape D	taba .	Generatry .			
	S2 st lease bg00	STATIFASE -1	TATI FPOD	tiong .	- 2	Centralia 2000 state Federal Information Processing Mendands (1995) codes	SLS, Cerner Ru
	TRACING INT	Providence in	convision.	Addined		Parson 2006 counts Factor of Information, Bernard to Mancherin (BUDI) code.	11.5 Carnet Run

Step 1 is to capture Attribute Lists for each of the Entities in your model. If the model is based on existing data, this can be done with a data profiling tool. Otherwise, it can be done manually.

Here Attribute information including Entity, Data Type, Length, Uniqueness, and FGDC XML Metadata definitions have been captured with a data profiler in MS Access.

#### Standardize Raw Data Attribute Names

- Mapping original raw attribute names to standardized sets is a key part of refactoring.
- This helps determine the unique list of attributes found across all entities, and is required for this technique to work.

Entity	Attribute_Raw	Attribute_Standard
Student	Student_Name	NAME
Student	ST_DOB	DOB
Student	Major	DEPARTMENT
Student	GPA	GPA
Student	Student_Year	STUDENT_YEAR
Student	Dormatory	BUILDING
Student	Advisor	FACULTY
Faculty	PROF_Name	NAME
Faculty	DOB	DOB
Faculty	Department	DEPARTMENT
Faculty	Salary	SALARY
Faculty	Tenure	TENURE
Faculty	Highest Degree	HIGHEST_DEGREE
Faculty	Office	BUILDING
Staff	Name	NAME
Staff	Birthday	DOB
Staff	Title	TITLE
Staff	Supervisor	SUPERVISOR
Staff	Salary	SALARY
Staff	Office	BUILDING

#### Step 1: Standardized Entity Attribute List

What's in a name?

How we name entities, attribute, and domain values is very important for easy-to-use, self describing data models. One important rule of thumb: An attribute meaning the same thing should have the same name, data type, length, and domain, in all the entities it appears in.

In general, it is best to pick very commonly used and understood words vs. trying to be consistent.

When choosing names and terms it can be helpful to see which ones are the most commonly used by getting word counts with Google. For example, Google returns 105 million hits when searching for 'lon/lat' and 23.8 million when searching for 'lat/lon' indicating 'lon/lat' is a far more common term.

Google	lon/lat	1					
	Web	Images	Maps	Shopping	More 🗸	Search tools	
	About 1	05,000,000 r	esults (0.37	seconds)			

ENTITY	ATTRIBUTE
Student	Name
Student	DOB
Student	Major
Student	GPA
Student	Student_Year
Student	Building
Student	Advisor
Faculty	Name
Faculty	DOB
Faculty	Department
Faculty	Salary
Faculty	Tenure
Faculty	Highest_Degree
Faculty	Building
Staff	Name
Staff	DOB
Staff	Title
Staff	Supervisor
Staff	Salary
Staff	Building

#### Step 2: Create an Attribute Entity Matrix

ID	ATTRIBUTE	STUDENTS	FACULTY	STAFF	GROUP
1	Advisor	1	0	0	100
2	Department	1	1	0	110
3	DOB	1	1	1	111
4	GPA	1	0	0	100
5	Highest Degree	0	1	0	010
6	Name	1	1	1	111
7	Building	1	1	1	111
8	Salary	0	1	1	011
9	Student_Year	1	0	0	100
10	Supervisor	0	0	1	001
11	Tenure	0	1	0	010
12	Title	0	0	1	001

Pivot the initial list of Entities and Attribute, create a unique list of attributes and pivot the Entities to create a matrix. **Note some attributes which actually use the same domain might have different names**, so choose a single name for these, e.g., Major and Department become Department, Dormitory and Building use Building This can be done with SQL and/or other tools.

Use TEXT data types for entity flags and PATTERN UPDATE ATT\_ENT\_MATRIX SET GROUP = STUDENTS & FACULTY & STAFF www.scribekey.com

#### Step 3: Get Distinct Attribute Groups

ATTRIBUTE	GROUP
Supervisor	001
Title	001
Highest Degree	010
Tenure	010
Building	011
Salary	011
Advisor	100
GPA	100
Student_Year	100
Department	110
DOB	111
Name	111

ID	GROUP	GROUP_MEMBERS	CLASS_NAME
1	001	Staff_Info	Staff_Info
2	010	Faculty_Info	Faculty_Info
3	011	Faculty_Staff_Info	Worker
4	100	Student_Info	Student_Info
5	110	Student_Faculty_Info	Education
6	111	Student_Faculty_Staff_Info	Person

Sort Attributes by Group. A supplemental, and sometimes challenging, step is to name the Groups.

Here we have 6 distinct sets of Attributes. Each Entity uses a set of groups. These equate to classes in OO design. Each feature class we create will contain all of the Attribute Groups it uses.

#### **Class Diagram from Attribute Groups**



We see the same information we created using Attribute Groups here as a classic OO type hierarchy. Note that Faculty inherits from both Education and Worker. Instead of using inheritance trees it is sometimes easier to use the concept of interfaces. Note with C# and Java, multiple inheritance is not allowed, as it is with C++. Note also that the classes are divided into 2 types, abstract which will never actually be created, and concrete, which inherit from the abstract classes, but only show specific attributes. Note this diagram does not show relationships between entities.

#### Attribute Sets and Interfaces



Another way to look at the Attribute Groups is as overlapping sets. Note out of a possible 7 groups, we ended up with a total of 6. This is because there were no attributes used only by Students and Staff, shown in the area labeled 6 above. These overlapping areas are equivalent to the class interfaces.

#### The Resulting Tables

Student_Year	Faculty	Staf
DOB	DOB	DOE
Name	Name	Nan
Building	Building	Buil
Department	Department	Sala
Adivsor	Salary	Sup
GPA	Highest_Degree	Title
Student_Year	Tenure	

Staff DOB Name Building Salary Supervisor Title

Simply listing the attributes for each entity provides one of the easiest to understand and useful views.

The tables holding the entity and attribute information in the metadata database can be used by relatively simple application to generate the tables.

# So, why is it so important to know what attribute groups different entities share?

#### SQL Union Views

If you know what attribute groups different entities have in common, you can construct views in relational databases, using the UNION clause, to see different things as the same type of things.

#### SELECT NAME, 'STUDENT' AS TYPE, BUILDING FROM STUDENTS

UNION

SELECT NAME, 'FACULTY' AS TYPE, BUILDING FROM FACULTY

UNION

SELECT NAME, 'STAFF' AS TYPE, BUILDING FROM STAFF

You could then add the UNION view to ArcMap and relate it to the buildings polygon layer, to help see all the people in a single building.

You could do this with any of the attribute sets (classes) the entities have in common, e.g., Worker.

NO.	NAME	STUDENT	BUILDING
1			
2			
3			
NO.	NAME	FACULTY	BUILDING
1			
2			
3			
NO.	NAME	STAFF	BUILDING
1			
2			
3			

#### Union of 3 Different Tables

NO.	NAME	TYPE	BUILDING
1			
2			
3			
4			
5			
6			
7			
8			
9			

1 Resulting View with Same Data

## **Capturing Relationships**

- Describing relationships between entities is an important part of data modeling
- Again, it can be useful to capture this information in metadata tables and matrices.

The Not 1 is a	AL 101	CONTRACTOR AND A	· NAME AND	ATTAC DESCRIPTION OF A	COLUMN TRANSPORT	CONTRACTOR	1
CONTRACTOR OF THE OWNER.	Dissocher Official International	Bandar (11)	Déscrio	Low Party	1.Medifordia Bi	Destablish	-
Second and Property States	Samueling secondly	Tel last manimum	UNKLIN	Take Louis (sec 17 Francis	Galage modificently	Displyhiaty.	1 mg
Interaction The	Terrowity Vial and Conclusion	Thes i makes	1000000	Garland monetary	factor (from the	DerTablero	100
BactinDutyres	Stablet Destrictions	Door Bulley Harr	concess.	Carbon Lines	MatroOlastill	DistiMary	1
Restrations	Service Desiribilitations	Tervise	CRECTO	Testrulianes	Involution/ID	Destations	Time
Adduction (Contraction of	Regulation industry approximation	the date statue	Distant	Angunation Internal Sections	August and Apartm	Citer 10Al arts	1
Adduction to an interest	TErlahen Ukramente	Tatur Davidse (Sullaw	10011709	Angulator tolerafactures	TEAMS-CA.e.MC	Dist 1844 million	1.000
Andulatio	181 Augulatio	International Visition	INACTO	Beginter.	(BUTHAND)	Confuldator	Anna
Regulator	Aughtabers Replaces	Augustatio Mattern	LINEHCOM	Augulator	Regulate Data Against	Dis 19Mary	100.00
Augustation	Statution Propulsion	from Tay	DESCRIPTION	Auguster	Andlastitutit	Distribution	1
(many square)	method Consequences	- includes	10040200	Criming regardier	invition gilleright	Distance.	1 mg
(Figures	diffusion Service	Catholic Pidecter Lobert	10050100	Saluan .	(Physical Based)	Dow Toldary	diam'r
Chainer	Difference Official Parts	Dathook Protection Indon-	UBACTO	Institute .	CPDs alson Chronick	Drie ToMany	1 mg
Places	L'Excert Entries	Lahope Hatshar Indon	URLINE.	Rabibulan Incin	Obstantineerin	Distigation	diam's
(Placese	CRUMME STATISTIC	Cattick President locate	descre-	A/1000	Crossenide earth	2nd lutters	1 mg
(Physical)	Cityphen Official	Cattools Protection Solvers	interime.	durillar -	Onyclosefile	Confidence.	And
OPhiliam	CRIMINE CRIMINE	Dehodic Protection Solare	OBJORN	dow/wine	(Physion/Rear/d)	<b>DraftMara</b>	See
In the second se	Service Servicement	Bei Value	<b>INCOM</b>	the Value Includion	Associate The other	Software	(time)
Laurination	WulterTat, Meter	Aurel Tap	INDUCTOR.	Autor	hundlapOtpetti0	Dist MARKED	1000
Landon and L	TEL WATER	hours dender Summer	10000708	AVRIT .	Williams	Direct Addance	Sera
Inst-Matter	Nuturbetting (Asker	White Setting	Cablements.	Materia	Manufacture operations and	insi Mary	100
and a state of the	Minister, Marial Street, Stree	fame bridge matters	interiors.	Index Value Rosentered	The maximum state of the second	The Public of	100
and a strategiestime.	Registered another products	Augustation Trademon	DEACED.	Infra Salve Proprietors	Registered Sector	Inglident	time
Testudiening .	O'Territure, Septembering	Inst Iver	086010	Crarie	OPlauthyantOkyact00	Destabling	ling
Realisment Real	darigment, lashesmintlys	The Spreet	084408	Out Darmin Income	Gathpland/specifi	dist 1 Mary	ling
Address of Conceptions	Marc Index Party and Art	Description Marc	HALCEN.	1 with	Manuf Report	Dist Maria	time.
subblack? participations	Service, hat also the repeditor	Larris A	166410	1990 V	Stational Install	Distiktory	100
Example State	Testalor Secondroyelline	Tanan Kandra Usalam	CARCER	Engine repairing	Notatio-Kitoking	<b>UnigToMaria</b>	diama di
Report Street St	Sugitation internet contact	Inspirator Stateme	Detcher	Tracker Paperline	AughalizerGalertS	CharloMeter	fungt
Saringmandation	Service and the service of the servi	Surgerbring .	0840384	Trafa .	delfyslangh/stll	D-970Alars	traip
To get	ExtRustreparties Despen	Turned Fire Councilian	20070	Creation .	Equilibrium of Specific	-OraTMilerg	700
Company and a string the set	Intergratively line in	Excendency contains System.	2014/010	destrolation former	Internationay/OkeeN0	<b>DistMary</b>	100
Disergeniny to langeti yelleris	Ingenter Supplier Decomplete	Dati Pressure bettern	THEFTER	Discourse totalities turners	tax mount when the setting	<b>OrielTSMerry</b>	100
Nation Vehicle House Party in the Party of t	1851ation_Asiahcal-Antaparties	Divertifiate (Interi	104413,788	Autor ( a second or	Minute-Interaction	DistAtory	100

Geodatabase Relationships captured by data profiling tool, in MS Access, showing ORIGIN, DESTINATION, and CARDINALITY

	Student	Faculty	Staff	Building
Student		N->1	NA	N->1
Faculty			NA	NA
Staff				N->1
Building				

Easy to construct entity relations matrix

## A Word about Domains

- Explicitly specifying the set of values that can be used for an attribute is an important part of data modeling.
- Even if you are not building a normalized data model, it is still useful to capture domains; this information can be used for both data validation and application GUI construction.
- This information can also be stored in the metadata repository database as well.
- Here are 3 methods for this:
  - 1) Lookup Tables: Valid list of values, e.g., Departments
  - 2) Regular Expressions: allowed patterns, e.g., MM/DD/YYYY
  - 3) SQL Expressions: Flexible assertion, e.g., range, match to another attribute, etc.

#### Domains as Lookup Tables



Even if you are not doing full normalization, it's important to capture domain values in tables. These can then be used to validate data and help build GUI front end pick lists. Note an empty database, with no features or rows, still has domain tables.

### Recap

- Consider a simple, table-centric, systematic and mechanical approach to data modeling.
- You can use a table/matrix approach to modeling your entities (feature classes and tables) as collections of attribute sets, with some attribute sets common to multiple entities. You can capture relationships this way too.
- You can easily capture this information in a metadata database, then use querying and sorting to help you with your task.
- The resulting attribute sets will match a class inheritance and/or an class interface diagram.
- When you know what attribute sets are common to multiple entities, you can use UNION views to query on the common elements.
- Don't automatically start creating a normalized set of entities if your database is to support querying, reports, and map making.
- Even if you aren't normalizing, it's still useful to store domains in your metadata database as lookup tables, regular expressions, and SQL Statements.

#### Thank You – Q&A

Brian Hebert www.scribekey.com