ENTITY-RELATIONSHIP MODELING OF SPATIAL DATA FOR GEOGRAPHIC INFORMATION SYSTEMS

Hugh W. Calkins
Department of Geography and
National Center for Geographic Information and Analysis,
State University of New York at Buffalo
Amherst, NY 14261-0023

Abstract
This article presents an extension to the basic Entity-Relationship (E-R) database modeling approach for use in designing geographic databases. This extension handles the standard spatial objects found in geographic information systems, multiple representations of the spatial objects, temporal representations, and the traditional coordinate and topological attributes associated with spatial objects and relationships. Spatial operations are also included in this extended E-R modeling approach to represent the instances where relationships between spatial data objects are only implicit in the database but are made explicit through spatial operations. The extended E-R modeling methodology maps directly into a detailed database design and a set of GIS function.

1. Introduction

GIS design (planning, design and implementing a geographic information system) consists of several activities: feasibility analysis, requirements determination, conceptual and detailed database design, and hardware and software selection. Over the past several years, GIS analysts have been interested in, and have used, system design techniques adopted from software engineering, including the software life-cycle model which defines the above mentioned system design tasks (Boehm, 1981, 36). Specific GIS life-cycle models have been developed for GIS by Calkins (1982) and Tomlinson (1994). Figure 1 is an example of a typical life-cycle model.

GIS design models, which describe the implementation procedures for the life-cycle model, outline the basic steps of the design process at a fairly abstract level (Calkins, 1972; Calkins, 1982). Figure 2 is a current version of previous design models. While useful as basic guides for GIS designers, the present models are deficient in that they describe only high-level activities. Additionally, these guides usually do not describe any methods for completing the indicated design tasks. One aspect of these models that has not been given sufficient additional definition is the conceptual and detailed design procedures for geographic databases. [see Figures 1 and 2: shaded steps] An extensive literature on general database design techniques exists (Elmasri and Navathe, 1994; Teorey and Fry, 1982; Ullman, 1988), however this body of knowledge has yet to be adapted for use in designing geographic databases.

GIS design methodologies currently in use do not treat the database design problem in detail. Often, data of interest are simply listed in tabular form on the assumption that using a commercial GIS format obviates the need for any further effort toward database design. More enlightened GIS design approaches link the data needed in the GIS to applications and GIS procedures. These approaches, however, do not offer any assistance in either the conceptual or logical design of the GIS database. Errors in database design can still occur and can be very costly. More attention needs to be paid to geographic database design. Specific tools reflecting the special characteristics of spatial data need to be developed to support the database design portion of the GIS design process. The building of the database for a GIS is frequently the largest single cost item, consuming as much as 80% of the total GIS project cost (Dickinson and Calkins, 1989). In this environment, it is well worth developing enhanced tools for supporting the GIS database design activities.

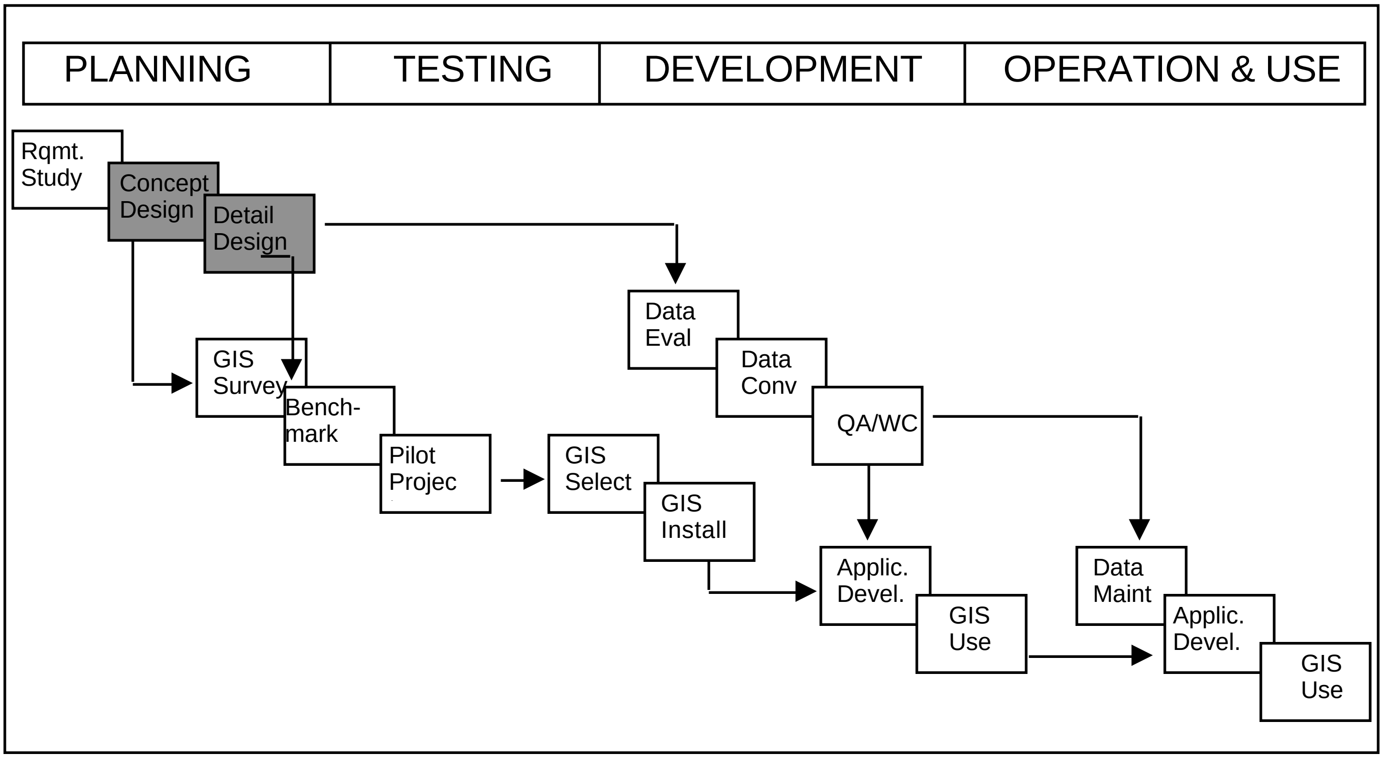


Figure 1 - GIS Life-Cycle

Of the many specific database design techniques developed, the entity-relationship (E-R) modeling technique developed by Chen (Chen, 1976), has gained popularity and been extremely effective over a wide range of application areas. This paper presents a proposed extension to the basic E-R modeling technique specifically for describing geographic data for use in the process of designing GIS database.

2. Conceptual Modeling and Data Base Design

Database design is the information system planning activity where the contents of the intended database are identified and described. Data base design is usually divided into three major activities (Elmasri and Navathe, 1994, 41):

(1) Conceptual data modeling: identify data content and describe data at an abstract, or conceptual, level;
(2) Logical database design: translation of the conceptual database design into the data model of a specific software system; and
(3) Physical design: representation of the data model in the schema of the software.

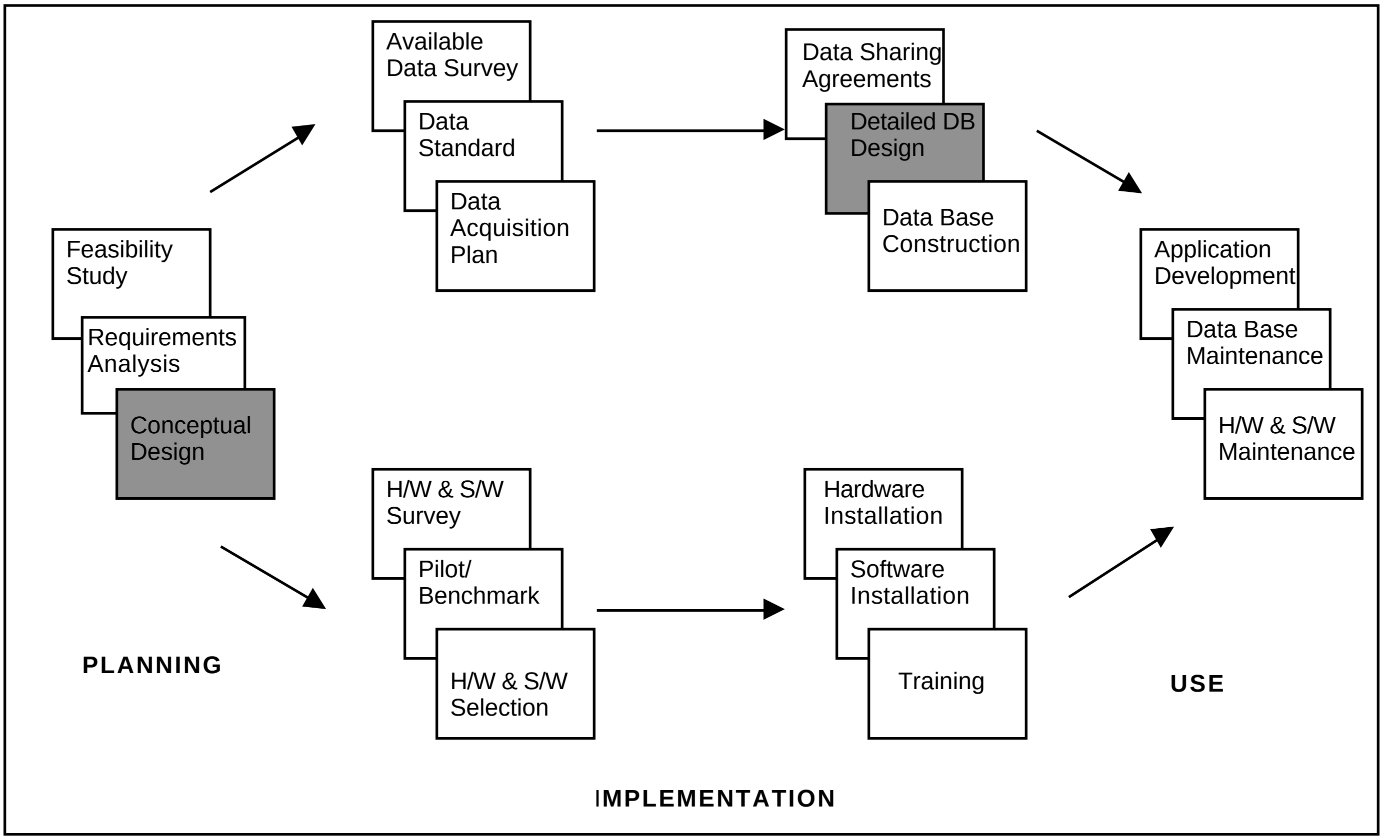


Figure 2 - GIS Design Model

For most GIS implementations, a commercial GIS software package is used, often in conjunction with a commercial database system. In these instances, the basic structure of the logical schema (e.g., a relational data schema) and the entire physical schema are already predetermined. The task of the GIS designer is to prepare a conceptual schema that properly describes the entire GIS database and is suitable for translation into the logical schema of the proposed GIS and database software.

To achieve a good database design, and thereby the desired GIS, the conceptual data model must be complete, i.e., contain all data needed to meet the system’s objectives, and must be directly translatable into the logical and physical database schema. Data identification and description includes defining the objects (entities), the relationships between the objects and the attributes of objects (or relationships) that will be represented in the database. The data description activity also includes assembly of information about the data objects, i.e., metadata (definition, data type, valid values, data quality, etc.).

The purpose of the conceptual data modeling process is to prepare an unambiguous and rigorous description of the data to be included in the database in a form that: 1) is understandable by the proposed users of the database or system; and 2) is sufficiently structured for a programmer or analyst to design the data files and implement data processing routines to operate on the data. The emphasis is on: 1) communication between the user and the programmer/analyst; and 2) review and verification of the data model and database design by both user and analyst. In the past, in a typical GIS design activity, users/analysts have tended to describe their data needs in general, somewhat vague, terms, such as a simple list. The programmer/analyst needs precise information about the data to set-up the database and necessary computer processes. Descriptions of data and algorithms using normal language (such as English) are not usually adequate for implementing a system. Thus, tools allowing greater precision in description are needed for the task of conceptual data modeling and database design. Such tools provide a means to identify and describe the intended database in terms that facilitate two critical activities: user verification and detailed database design.

Conceptual modeling is the representation of the functional application requirements and information system components at a abstract level, i.e., a description of what is to be included into an information system rather than how the intended information system will work. “The quality of a conceptual schema (model) and ultimately that of the information system depends largely on the ability of a developer to extract and understand knowledge about the modeled domain” (Loucopoulos and Zicari (1992). Further, “a conceptual schema of a concrete system is called a conceptualization and represents the basic specification mechanism during requirements analysis. A symbolic representation of a conceptual system is called a representation; for example, the Entity-Relationship Diagram is a representation of the Entity-Relationship Model” (Loucopoulos and Zicari, 1992, 5). The characteristics of a conceptual schema have been set forth by several authors and summarized in Loucopoulos and Zicari (1992, 6). A set of requirements for a conceptual schema have been proposed (Liskov and Zillis, 1977; Balzer and Goldman, 1979, Yeh, 1982; van Griethuysen et al., 1982; Borgida, 1985; Mylopoulos, 1986) and are shown in Table 1.

Table 1: Characteristics of Data Modeling Tools

Implementation Independence	No implementation specific Aspects; no external user representations
Abstraction	General requirement specification
Formality	Formal syntax for description; Easily understood; suitable for analysis
Constructibility	Easy communication between user and analyst; accommodate larger set of facts; handle complexity in problem domain; permit natural decomposition
Ease of Analysis	Criteria for testing completeness, consistency, And lack of ambiguity (require external Specification for testing which normally does Not exist (Olive, 1983)
Traceability	Ability to cross-reference elements of Specification to corresponding elements of Design/implementation
Executability	Validation of specification; ability to simulate Specification against relevant facts

A tool to support conceptual data modeling should have the above characteristics in addition to the ability to ensure completeness of description and the ability to support one-tone mapping of the conceptual model into the logical database design. For an extended discussion of conceptual modeling and information system design, the reader is referred to Loucopoulos and Zicari, 1992).

3. Tools for conceptual database modeling

Various tools to support the conceptual database modeling activity have been developed. One widely accepted and used technique is the entity-relationship modeling technique developed by Chen (Chen, 1976). The entity-relationship (E-R) technique has been applied to many disciplines and has been revised and extended by many researchers to meet a variety of specialized needs. The E-R technique is a graphical method of representing objects (or entities) of a database, all important relationships between the entities, and all attributes of either entities or relationships which must be captured in the database. The set of rules controls the definition of entities, relationships, and attributes and the manner in which they are portrayed in diagrammatic form. All items are names and additional information is appended to the diagram indicating the nature of each relationship, i.e. the cardinality of each relationship (one-to-one, one-to-many, or many-many).

3.1 Basic Entity-Relationship Modeling

The basic entity-relationship modeling approach is based on describing data in terms of the three parts noted above (Chen 1976):

- Entities
- Relationships between entitles
- Attributes of entitles or relationships

Each component has a graphic symbol and there exists a set of rules for building a graph (i.e., an E-R model) of a database using the three basic symbols. Entities are represented as ***rectangles,*** relationships as ***diamonds*** and attribute**s** as ***ellipses***.

The normal relationships included in a E-R model are basically those of:

1. Belonging to;
2. Set and subset relationships;
3. Parent-child relationships; and
4. Component parts of an object.

The implementation rules for identifying entities, relationships, and attributes include an English language sentence structure analogy where the nouns of a descriptive sentence identify entities, verbs identify relationships, and adjectives identify attributes. These rules have been defined by Chen (1983) as follows:

> Rule 1: A common noun (such as person, chair), in English corresponds to an entity type on an E-R diagram.
>
> Rule 2: A transitive verb in English corresponds to a relationship type in an E-R diagram.

Rule 3: An adjective in English corresponds to an attribute of an entity in an E-R diagram.

English statement: A person may own a car and may belong to a political party.

Analysis: "person," "car," and "political party" are nouns and therefore correspond to entity types. … "own" and "belong to" are transitive verbs (or verb phases) and therefore define relationships.

3.2 Example of simple E-R diagram

Figure 3 shows a simple E-R diagram of three typical GIS objects – land parcels, buildings, and building occupants, each with a few selected attributes. Normal E-R symbology is used in Table 3 (see next page).

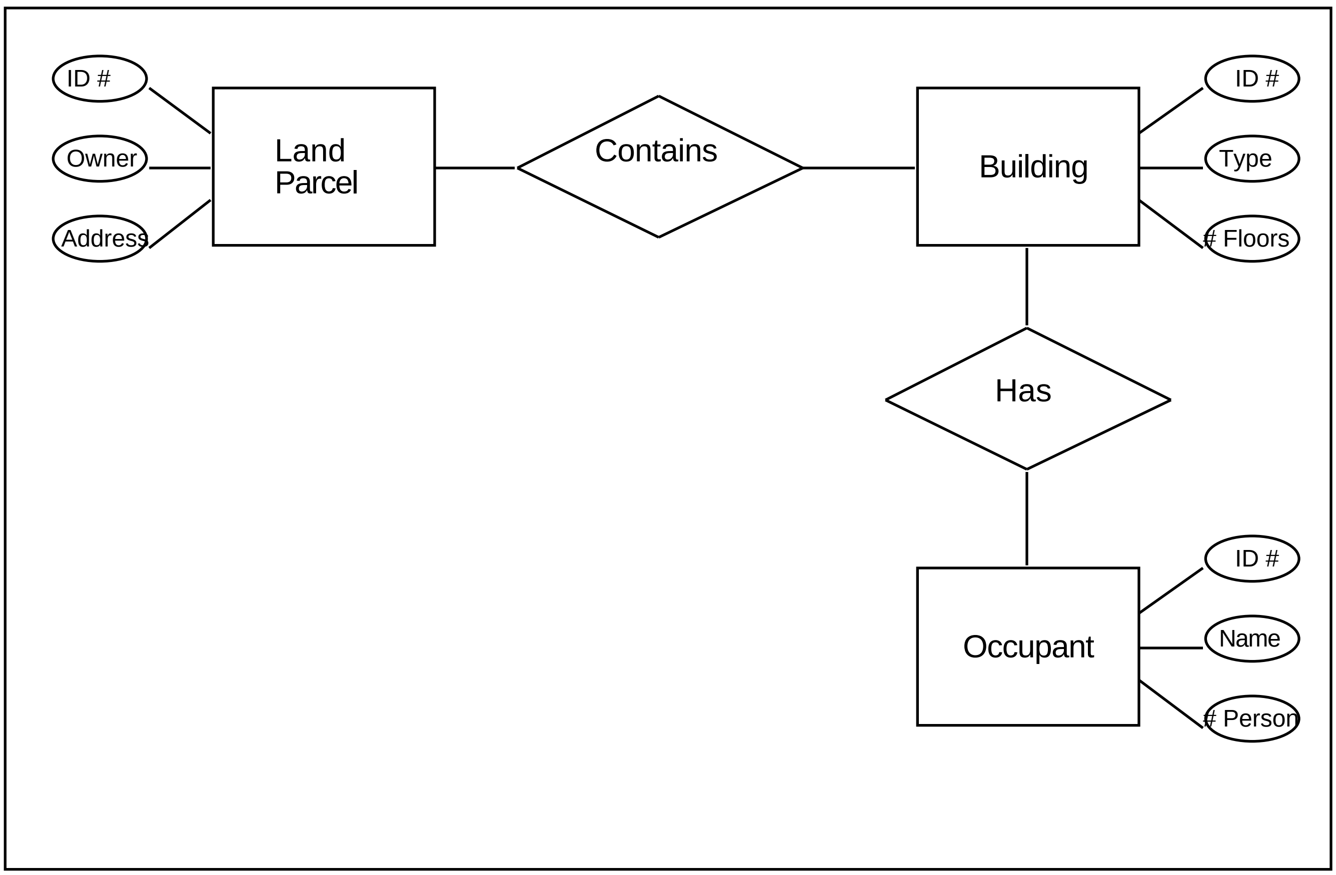


Figure 3 - Example E-R Diagram

Spatial Relationships

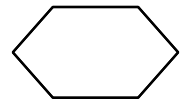
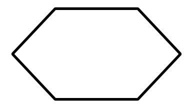
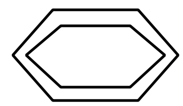
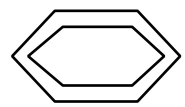
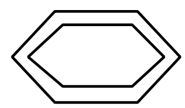
Spatial Relationship	Descriptive Verbs	Common GIS Implementation	E-R Model Symbol
Connectivity	Connect, link	Topology	
Contiguity	Adjacent, abutt	Topology	
Containment	Contained, containing, within	X, Y coord. operation	
Proximity	Closest, nearest	X,Y coord. operation	
Coincidence	Coincident, Coterminous	X,Y coord. operation	

Table 3. Expanded E-R Model Symbology

The process of constructing an E-R diagram uncovers many inconsistencies or contradictions in the definition of entities, relationships, and attributes. Many of these are resolved as the initial E-R diagram is constructed while others are resolved by performing a series of transformations on the diagram after its initial construction. For a discussion of E-R transformations see Jajodia and Ng (1983). The final E-R diagram should be totally free from definitional inconsistencies and contradictions. If properly constructed, an E-R diagram can be directly converted to the logical and physical database schema of the relational, hierarchical or network type database for implementation.

4. Geographic Data Models

The data models in most contemporary GISs are still based on the cartographic (or spatial) data object view. Other data models have begun to evolve, but are still very limited. Current and potential geographic data models include:

- The cartographic data model: points, lines and polygons (topologically encoded) with one, or only a few, attached attributes, such as a land use layer represented as polygons with associated land used code.
- Extended attribute geographic data mode: geometric objects as above but with many attributes, such as census tract data sets;
- Conceptual object/spatial data model: explicit recognition of user defined objects, zero or more associated spatial objects, and sets of attributes for reach defined object (example: user objects of land parcel, building, and occupant, each having its own set of attributes but with different associated spatial objects: polygon for land parcel, footprint for building, and no spatial object of occupant).
- Conceptual objects/complex spatial objects: multiple objects and multiple associated spatial objects (example: a street network with street segments having spatial representations of both line and polygon type and street intersections having spatial representations of both point and polygon type).

Current GIS are based on the cartographic and extended attribute data models. Data modeling tools supporting GIS design will need to accommodate all the above defined data modeling cases. The remainder of this paper describes how an extension to the E-R modeling technique can provide the necessary modeling tool for GIS database design.

5. Representation of spatial relationships

The E-R methodology rules, particularly the sentence verb rule, can be applied to the identification of the spatial relationships found in a geographic information system. Table 2 includes the common set of spatial relationships of interest in a GIS.

Table 2. Spatial Relationships

Spatial Relationship	Descriptive Verbs
Containment	Contained/containing; within
Connectivity	Connect; link
Contiguity	Adjacent; abutting
Coincidence	Coincident; coterminous
Proximity	Closest, nearest

In a geographic information system, spatial relationships are implemented into one of three ways:

- topological encoding;
- x,y coordinates and accompanying spatial operation; and
- the one of the traditional database relationships previously identified

Connectivity and contiguity are implemented through topology: the link-node structure for connectivity through networks and the arc-polygon structure for contiguity. Containment and proximity are implemented through x,y coordinates and related spatial operations: containment is determined using the point-, line-, and polygon-on-polygon overlay spatial operation and proximity is determined by calculating the coordinate distance between two or more x,y coordinate locations. The spatial relationship of coincidence may be complete coincidence or partial coincidence. The polygon-on-polygon overlay operation in ARC/INFO™ calculates partial coincident of polygons in two different coverages. The System 9™ Geographic Information System recognizes coincident features through a "shared primitive" concept (the geometry of a point or line is stored only once and then referenced by all features sharing that piece of geometry). Future versions of commercial GISs will likely implement coincident features through either the "belonging to" database relationship or through x,y coordinates and related spatial operations, whichever is more efficient within the particular GIS.

6. Modeling geographic data with E-R techniques

Previous use of E-R modeling techniques for geographic database design have been reported by Calkins and Marble (1985), Bedard and Paquette (1988), Wang and Newkirk (1988), and Armstrong and Densham (1990). Calkins and Marble (1985) demonstrated the use of the basic E-R) methodology in the design of a cartographic database where there was no special provision for the geographic entities of points, lines, or polygons in the integrated E-R diagram. However, Calkins and Marble (1985, 118) did demonstrate that the E-R diagram of the cartographic database could be transformed into an E-R representation based solely on the spatial entities. Bedard and Paquette (1988) and Wang and Newkirk (1988) include the spatial entities in their E-R diagrams by sing the "ISA" relationship where each entity is defined normally and also as a spatial entity. Armstrong and Densham (1990, 16) defined an extended network representation for spatial decision support systems (SDSS).

All of the above efforts are limited and insufficient for GIS database design in that the identification of the spatial entity corresponding to the real world object being modeled is either missing, presented as a separate or additional entity, or is otherwise redundantly represented in the E-R diagram. The resulting E-R diagrams are less easily understood, are more complex than need be, and are less easily manipulated (or transformed) to remove errors or inconsistencies. Additionally, and most importantly, there is no direct mapping (1:1) to a logical database schema, a primary criterion for a useful database design tool. Thus, these E-R extensions do not adequately meet the communications or operability goals for a database design tool.

Additionally, and more importantly, neither the basic E-R modeling technique nor any of the extensions developed represent the full extent of the spatial relationships that exist in a geographic information system and its associated database. Her it is necessary to recognize a significant difference between a GIS and other types of information systems, particularly business oriented information systems. In a typical business application, all relationships between and among data objects can be explicitly represented in the database and the associated software is usually oriented to query and report generation functions. In a GIS, the set of relationships between and among data objects is represented, in part, explicitly in the database and, in part, is implemented through various software functions. For example, there is a spatial relationship between two map layers of the same area or location (spatial coincidence). However, the GIS database usually does not explicitly represent this relationship. The relationship is derived through the use of the topological overlay spatial operation, which is a software function. It is the latter case (a data relationship that will be implemented through software) that neither the basic E-R modeling methodology, nor any proposed extension, can adequately describe.

Further, additional demands on the data modeling tool will come from the continuing evolution of GISs. It is now considered necessary for such a system to accommodate multiple spatial representations of a single entity and multiple temporal representations of a single entity. Finally, an adequate database design tool for geographic databases needs a structure to represent the database in compact form, as a geographic database usually has a large number of entities and relationships that yield large and complex diagrams. To meet the objective of communication and understandability the graphic tool must be as compact as possible.

7. Modeling a geographic database

Modeling a geographic database using the E-R approach requires an expanded or extended concept for:

- Entity identification and definition; and
- Relationship types and alternate representational forms for spatial relationships.

There are three considerations in the identification and definition of entities in a geographic database:

7.1 Correct identification and definition of entities

Entities in a geographic database are defined as either discrete objects (e.g., a building, a bridge, a household, a business, etc.) or as an abstract object defined in terms of the space it occupies (e.g., a land parcel, a timber stand, a wetland, a soil type, a contour, etc.). In each of these cases we are dealing with entities in the sense of "things" which will have attributes and which will have spatial relationships between themselves. These "things" can be thought of as "regular" entities.

7.2 Defining a corresponding spatial entity for each "regular" entity

A corresponding spatial entity will be one of the spatial data types normally handled in a GIS, e.g., a point, line, area, volumetric unit, etc. The important distinction here is that we have a single entity, its spatial representation and a set of attributes; we do not have two separate objects (Figure 4 illustrates this concept). A limited and simple set of spatial entitles may be used, or alternatively, depending on the anticipated complexity of the implemented geographic information system, an expanded set of spatial entities may be appropriate. The corresponding spatial entity for the regular entity may be implied in the definition of the regular entity, such as abstract entities like a wetland where the spatial entity would normally be a polygon, or a contour where the spatial entity would be a line. Other regular entitles may have a less obvious corresponding spatial entity. Depending on the GIS requirements, the cartographic display needs, the implicit map scale of the database and other factors, an entity may be reasonably represented by one of several corresponding spatial entities. For example, a city in a small-scale database could have a point as its corresponding spatial entity, while the same city would have a polygon as its corresponding spatial entity in a large-scale geographic database.

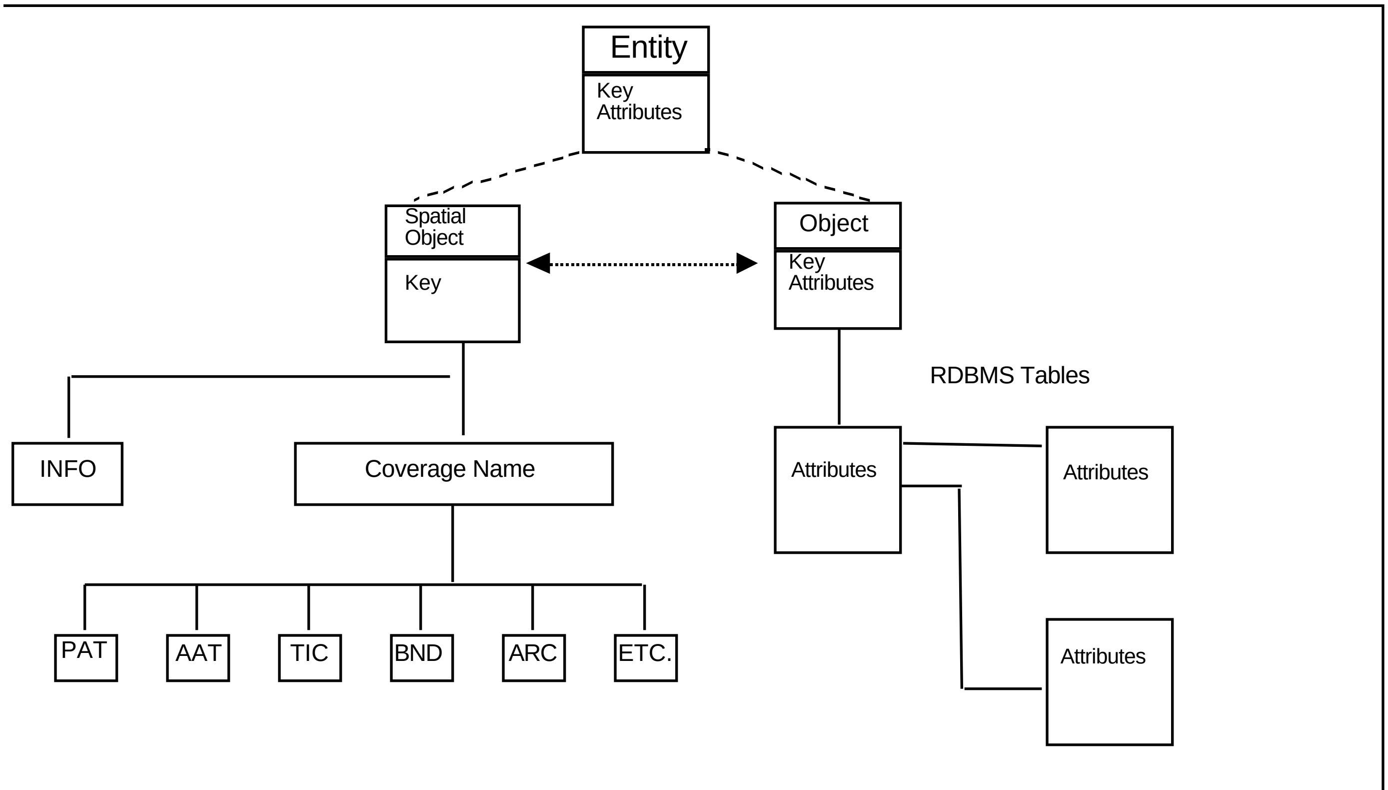


Figure 4 - GIS Representation of Object and Associated Spatial Object

7.3 Recognize multiple instances of geographic entities, both multiple spatial instances and multiple temporal instances

Multipurpose (or corporate) geographic databases may need to accommodate multiple corresponding spatial entities for some of the regular entities included in the GIS. For example, the representation of an urban street system may require that each street segment (the length of street between two intersecting streets) be held in the GIS as both a single-line street network to support address geocoding, network based transportation modeling, etc., and as a double-line (or polygon) street segment for cartographic display, or to be able to locate other entities within the street segment (such as a water line), etc. In each of these instances the "regular" entity is the street segment, although each instance may have a different set of attributes and different corresponding spatial entities. Also, there may be a need to explicitly recognize multiple temporal instances of regular entities. The simple case of multiple temporal instances will be where the corresponding spatial entity remains the same, however, future GISs will, in all likelihood, have to deal with multiple temporal instances where the corresponding spatial entity changes over time.

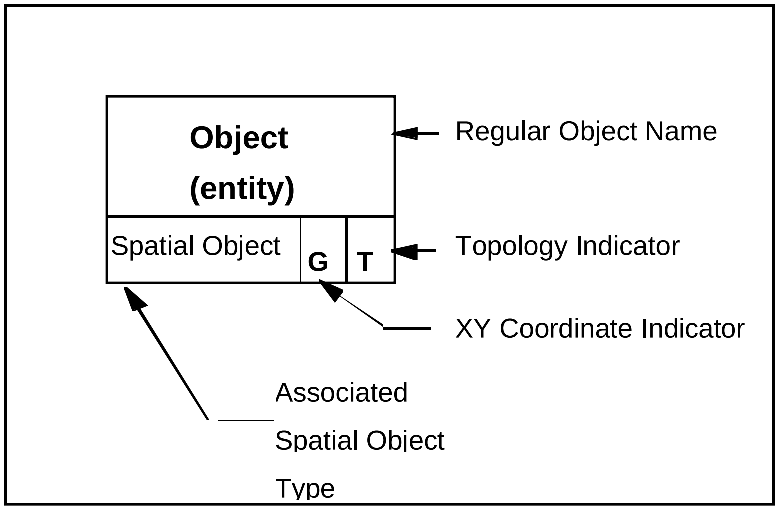


Figure 5 - Entity Symbol for Spatial Objects

7.4 Modeling spatial relationships

In a manner similar to the expansion of the concept of entities, the concept of relationship also needs to be expanded. In addition to the regular, or normal, types of relationship that can exist between entities, a set of spatial relationships need to be defined and included in the E-R model.

Modeling a geographic database using the E-R framework requires additional notation to represent the five spatial relationships and the manner in which they will be implemented. This is accomplished in the E-R model extension for spatial data in two ways: 1) a change in the entity symbol providing for inclusion of information about the corresponding spatial object and associated spatial codes (Figure 5); and 2) defining additional symbols for representing spatial relationships (Figure 6). An expanded E-R symbology set suitable for describing geographic database is shown in Figure 7.

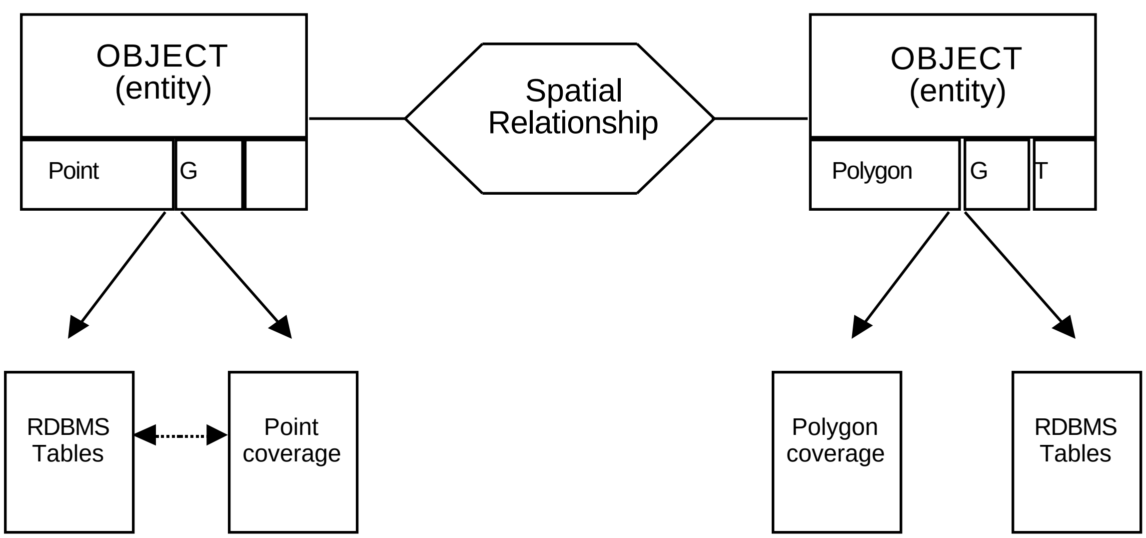


Figure 6 - Diagramming a Spatial Relationship

Three symbols are defined to represent entities: entity (simple); entity (multiple spatial representations); and entity (multiple time periods). The internal structure of the entity symbol contains the name of the entity and additional information indicating the corresponding spatial entity (point, line or polygon), a code indicating topology, and a code indicating encoding of the spatial entity by coordinates (Figure 5). The coordinate code is, at the present time, redundant in that all contemporary GISs represent spatial entities with x,y coordinates. However, it is possible that future geographic databases may include spatial entities where coordinates are not needed. Similarly, topological encoding is normally of only one type and can, for the present, be indicated by a simple code. However, different spatial topologies have been defined and may require different implementations in a GIS (Armstrong and Densham, 1990). In the future, the topology code may be expanded to represent a specific topologic structure particular to a GIS application.

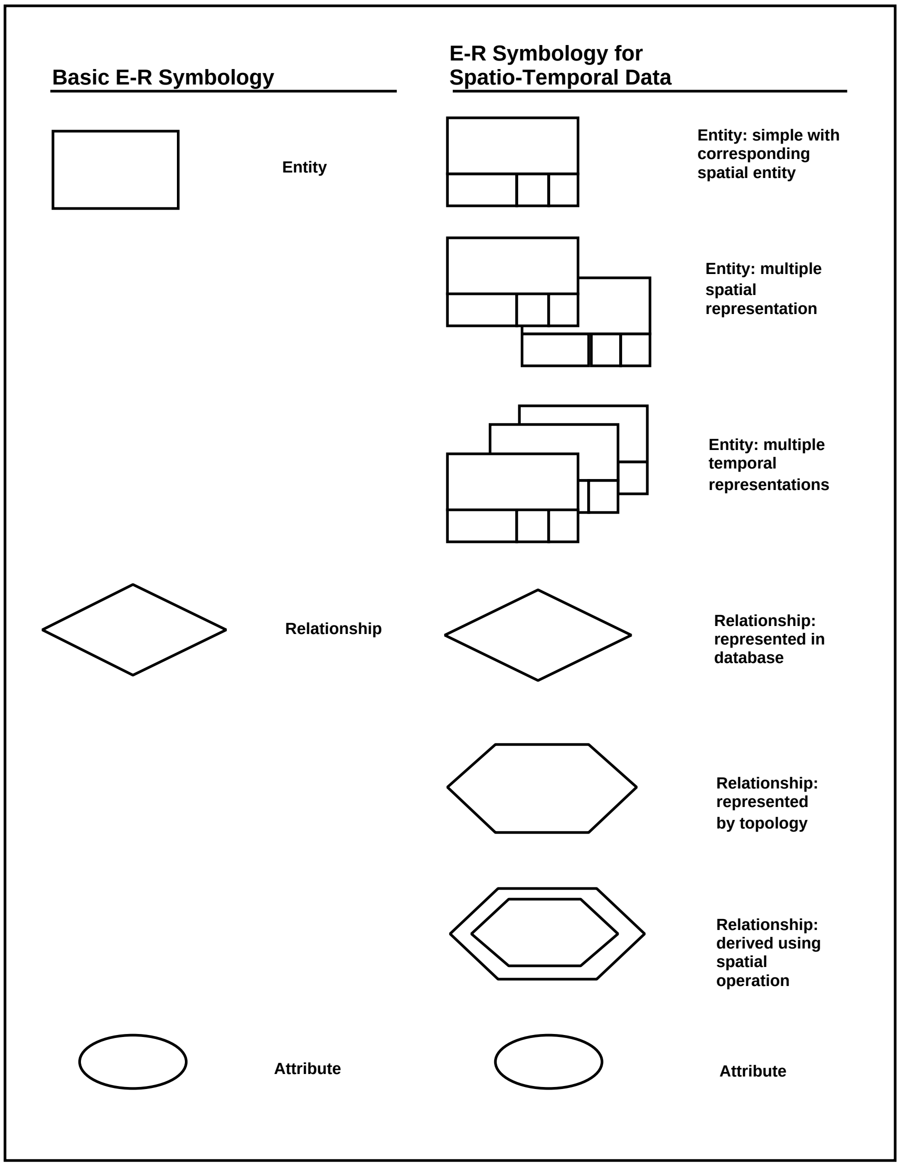


Figure 7 - Extended Entity - Relationship Symbology for Designing GIS Databases, Source: Calkins, 1996

The spatial relationships are defined by three relationship symbols (Figure 7). The traditional diamond symbol can be used for normal database relationships. An elongated hexagon and a double elongated hexagon, are defined to represent spatial relationships. The elongated hexagon represents spatial relationships defined through topology (connectivity and contiguity) and the double elongated hexagon represents spatial relationships defined through x,y coordinates and related spatial operations (coincidence, containment and proximity). The appropriate "verbs" to include in the hexagonal symbols are the descriptors of the spatial relationships (as shown in Figure 3). The spatial operation will be implicitly defined by the relationship symbol (double hexagon), the spatial entity and the topology code. For example, a spatial relationship named "coincident" between entities named "wetlands" and "soils," both of which carry topologic codes and x,y coordinates, indicates the spatial operation of topological overlay. If this does not sufficiently define the spatial operation needed, the name of the spatial operation can be used to describe the relationship, such as shortest path, point-in-polygon, radial search, etc.

8. Conceptual Model of a Spatial Database

An example of a spatial database model using the proposed expanded E-R technique is presented as Figure 8. The data for this model is typical of local government GISs. Table 4 lists the data in the model (major category, entity, attributes and associated spatial object). For readability, attributes are not shown in Figure 8: however, in working with such a database model, attributes should be included in the diagram for evaluation purposes. Also, sets of entities can be identified in such a diagram, as is shown in Figure 8 by the dashed rectangle used to group all entities making up the "street system."

9. Mapping the Conceptual Database Design to Detailed Logical/Physical Specifications

The example E-R diagram shown in Figure 8 will be used to verify with the expected users the data content of the GIS and, by additional reference to the GIS needs analysis, the required spatial operations. Once verified by the users, the E-R representation can be mapped into a detailed database design as follows (Figure 9):

1) Each entity and its attributes map into:
   (a) One or more relational tables with appropriate primary and secondary keys (this assumes the desired level of normalization has been obtained);
   (b) The corresponding spatial entity for the "regular" entity. As most commercial GISs rely on fixed structures for the representation of geometric coordinates and topology, this step is simply reduced to ensuring that each corresponding spatial entity can be handled by the selected GIS package;

2) Each relationship into:
   (a) Regular relationships (diamond) executed by the relational database system's normal query structure. Again, appropriate keys and normalization are required for this mapping.
   (b) Spatial relationships implemented through spatial operations in the GIS. The functionality of each spatial relationship needs to be described, and if not a standard operation of the selected GIS, specifications for the indicated operation need to be written.

Table 4. Data List

Category	Entity (attributes)	Spatial Object
Geographic index (to support geocoding)	**Street_segment** (name, address_range)	Line
	Street_intersection (street_names)	Line
Land records	**Parcel** (subdivision_block_lot#, owner_name, owner_address, situs_address, area, depth, front_footage, assessed_value, last_sale_date, last_sale_price, (owner_name, owner_address, assessed_value as of previous January 1st)	Polygon
	Building (building_ID, date_built building_material, building_assessed_value	Footprint
	Occupancy (occupant_name, occupant_address, occupancy_type_code)	None
Street system	**Street_segment** (name, type width, length, pavement_type)	Polygon
	Street_intersection (length, width, traffic_flow_conditions, intersecting_streets	Polygon
Water system	**Water_main** (type, size, material installation_date)	Line
	Valve (type, installation_date)	Node
	Hydrant (type, installation_date, pressure last_pressure_test_date)	Node
	Service (name, address, type, invalid_indicator)	None
Natural features	**Soil** (soil_code, area)	Polygon
	Wetland (wetland_code, area)	Polygon
	Floodplain (flood_code, area)	Polygon
Administrative areas	**Traffic_zone** (zone_ID#, area)	Polygon
	Census_tract (tract#, population)	Polygon
	Census_block (block#, population	Polygon
	Zoning (zoning_code, area)	Polygon

CONCLUSION

This paper has presented a fairly simple, yet useful extension to the basic Entity-Relationship data modeling technique that allows for modeling current geographic databases. The extension can accommodate the common geographic data models presently in use. GIS applications are growing very rapidly and with this growth will come additional demands on geographic data modeling tools. Specifically, temporal representation of geographic data and events with multiple geographic locations cannot be handled in the modeling structure presented here. These, and possibly other, geographic data types will require additional extensions to geographic data modeling procedures.

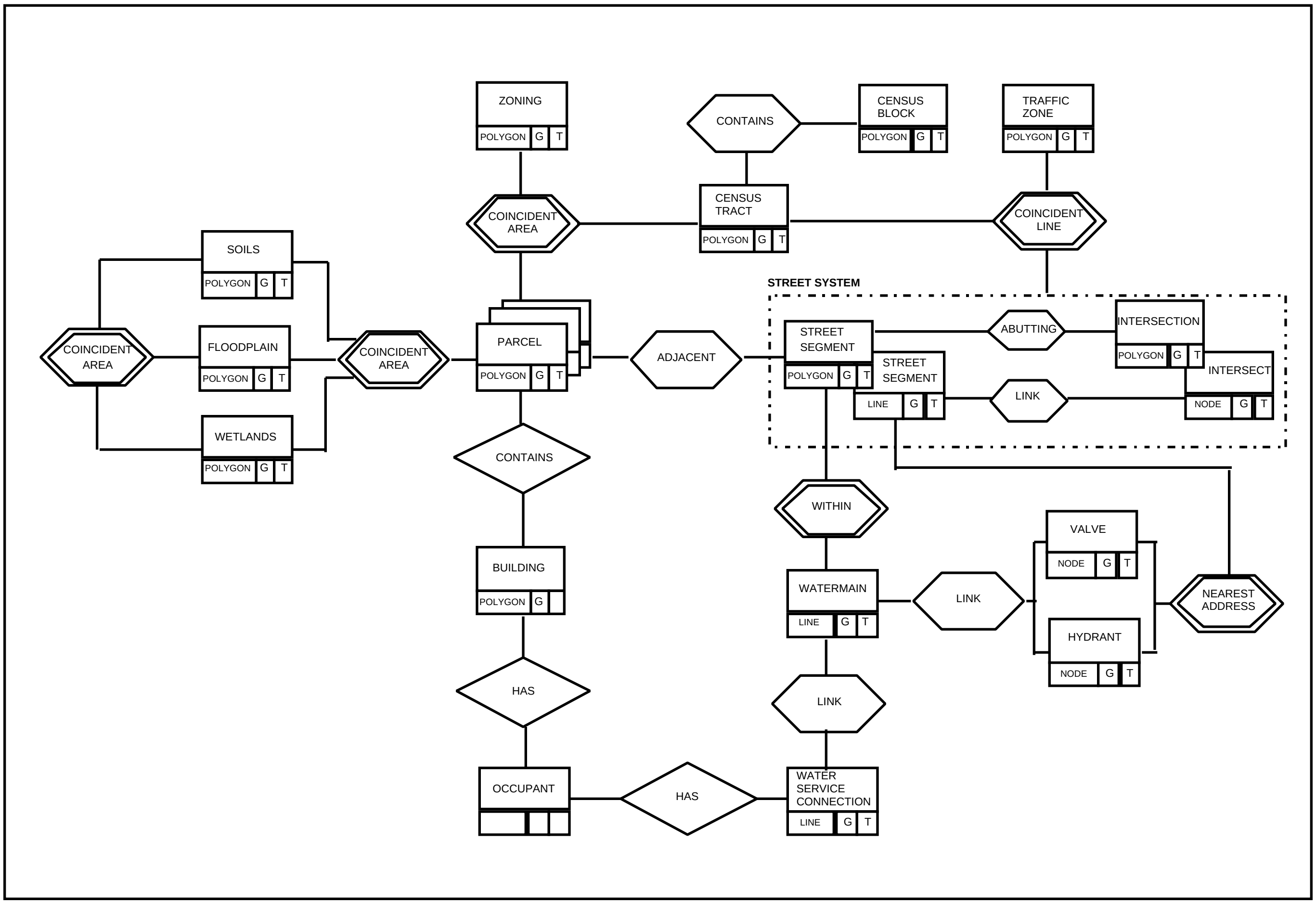


Figure 8 – Example Entity-Relationship Diagram for Local Government

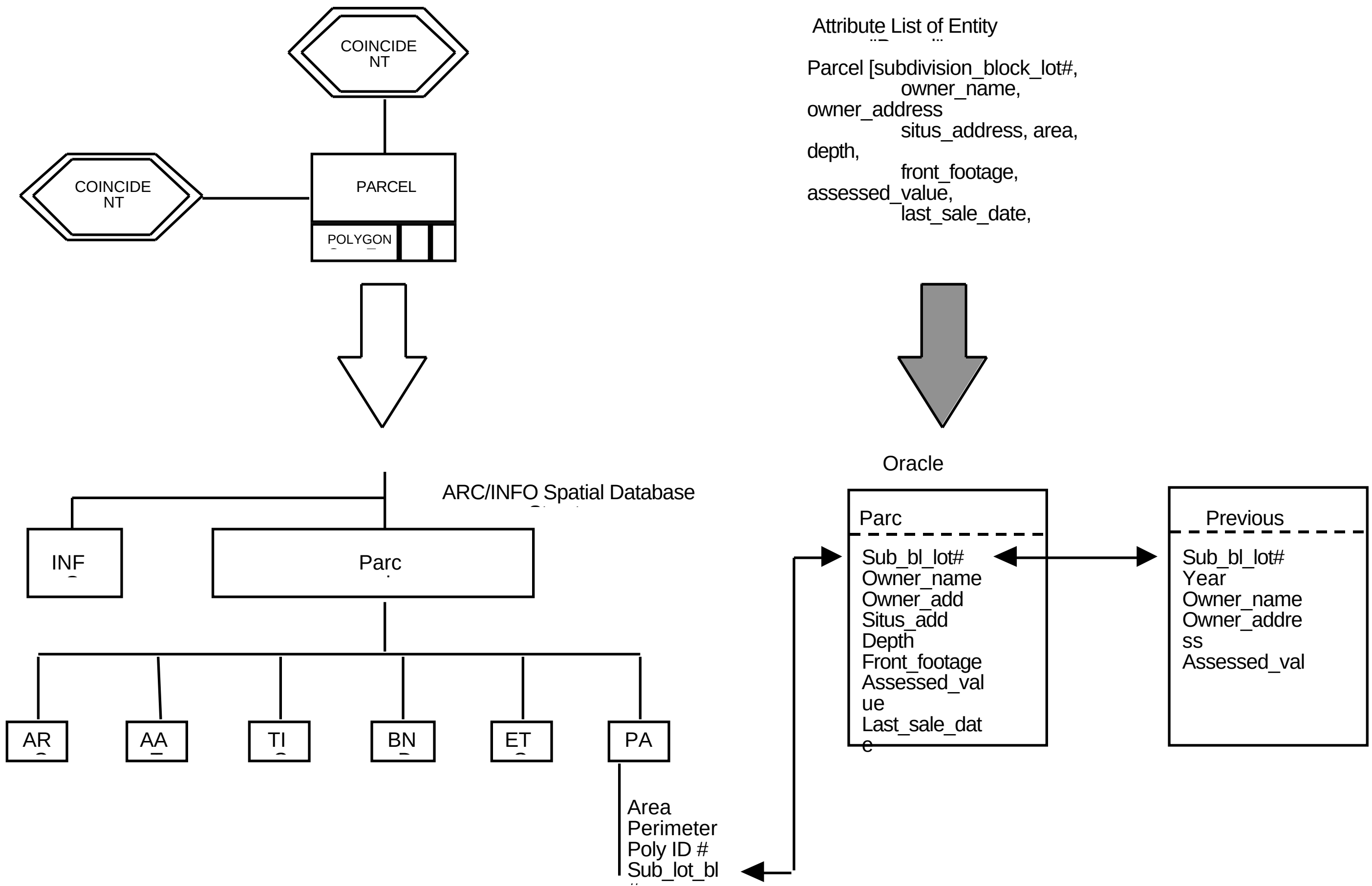


Figure 9 – Example of Mapping of E-R Entity and Attribute List into ARC/INFO & ORACLE Logical Database Design

References

Armstrong, M.P. and P.J. Densham, 1990, Database organization strategies for spatial decision support systems, **International Journal of Geographical Information Systems**, vol. 4, no. 1, 3-20.

Balzer, R.M. and N. Godman, 1979, Principles of good software specification and their implementations for specification languages, **Proceeding**s, Spec. Reliable Software Conference, April 1979, 58-67

Bedard, Y., 1988, Extending entity/relationship formalism for spatial information systems, **ATOCARTO 9**, Baltimore, Maryland, 818-827.

Boehm, B.W., 1981, **Software Engineering Economics** (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.).

Borgida A., 1985, Language features for flexible handling of exceptions in information systems, **ACM Transactions on Database Systems**, vol. 10, No. 4, December 1985, pp. 565-603.

Calkins, H.W. et all, 1972, The economics of geographical data handling, **GEOGRAPHICAL DATA HANDLING**, R. F. Tomlinson, ed., Ottawa: IGU Commission on Geographical Data Sensing and Processing.

Calkins, H.W., 1982, A pragmatic approach to geographic information system design, **Proceedings**, Joint US/Australian Workshop on the Design and Implementation of Geographic Information Systems, International Geographical Union commission on Geographical Data Sensing and Processing.

Chen, P.P., 1976, The entity-relationship model - toward a unified view of data, **ACM Transactions on Database Systems**, vol. 1, no. 1, March 1976, pp. 9-36.

Chen, P.P., 1984, English sentence structure and entity-relationship diagrams, **Information Sciences**, 29, 127-149.

Davis, C., et. al., eds., 1983, **Entity-Relationship Approach to Software Engineering**, Amsterdam, Netherlands: Elsevier Science Publishers B.V.

Elmasri, R. and S.B. Navathe, 1989, **Fundamentals of Database Systems**, Redwood City, California: The Benjamin/Cummings Publishing Company, Inc.

Jajodia, S. and P. Ng, 1983, On representation of relational structures by entity-relationship diagrams, **Entity-Relationship Approach to Software Engineering**, P. Ng and R. Yeh (eds.), Amsterdam, Netherlands: Elsevier Science Publishers B.V., pp. 249-263.

Liskov, B. and S. Zilles, 1977, An introduction to formal specifications of data abstractions, **Current Trends in Programming Methodology - Vol. 1: Software Specification and Design**, R.T. Yeh (ed), Prentice Hall, pp 1-32.

Loucopoulos, P, et. al., 1991, TEMORA – integrating database technology, rule based systems and temporal reasoning for effective information systems, **Journal of Information Systems**, Vol. 1, No. 2.

Loucopoulos, P. and R. Zicari, 1992, **Conceptual Modeling, Databases, and CASE: An integrated view of information systems development**, New York: John Wiley & Sons, Inc.

Mylopoulos, J., 1986, The role of knowledge representation in the development of specifications, **Information Processing 86**, Kugler, H-J (ed), Elsevier B. V., © IFIP.

Teorey, T.J. and J.P. Fry, 1982, **Design of Database Structures**, Englewood Cliffs, NJ: Prentice-Hall, Inc.

Tomlinson, R.F., 1994, **GISPlan**, Redlands: Environmental systems Research Institute.

Ullman, J.D., 1988, **Principles of Database and Knowledge-base Systems**, 2 vols. (Rockville, Maryland: Computer Science Press, Inc.)

Van Griethuysen, J.J., 1982**, Concepts and Terminology for the Conceptual Schema and the Information Base**, Report ISO TC97/SCS/WG3.

Wang, F. and R. Newkirk, 1988, an entity-relationship model for geographical information system development, **ASPRS Annual Convention**, Vol. 5, St. Louis, 162-172.