ANALYSIS PATTERN OF ŞANLIURFA HARRAN PLAIN IN UML AND ITS IMPLEMENTATION IN GEODATABASE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ULAŞ ÇUBUK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF GEODETIC, GEOGRAPHICAL INFORMATION TECHNOLOGIES

APRIL 2004

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Assoc. Prof. Dr. Oğuz Işık
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Assist. Prof. Dr. Zuhal Akyürek
Co-Supervisor

Assoc. Prof. Dr. Nurünnisa Usul
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Sibel Kalaycıoğlu

Assoc. Prof. Dr. Nurünnisa Usul

Assist. Prof. Dr. Zuhal Akyürek

Assoc. Prof. Dr. Nazife Baykal

Assoc. Prof. Dr. Ali Doğru

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

ANALYSIS PATTERN OF ŞANLIURFA HARRAN PLAIN IN UML AND ITS IMPLEMENTATION IN GEODATABASE

Çubuk, Ulaş

MS, Department of Geodetic, Geographical Information Technologies

Supervisor: Assoc. Prof. Dr. Nurünnisa Usul
Co-Supervisor: Assist. Prof. Dr. Zuhal Akyürek

April 2004, 121 pages

An emerging trend in GIS is the adoption of object oriented concepts for both logical and physical design phases. Extensive research has been conducted on logical design of GIS and several conceptual models have been proposed. Classical data models like the relational data model have proven to be insufficient for the conceptual modeling of spatial data. Therefore among other object oriented modeling tools, a new modeling language, Unified Modeling Language (UML) has also become a popular modeling tool in the GIS domain due to its wide acceptance in industry. In this thesis ArcInfo UML Model, which is an extension of UML and proposed by ESRI is utilized to create analysis pattern of Şanlıurfa Harran Plain within the scope of GAP (In Turkish, Güneydoğu Anadolu Projesi). The proposed analysis pattern mainly characterize the irrigation drainage system, social environment and irrigation management activities of the Şanlıurfa Harran Plain and incorporate the interactions between, and among, features of the system. At the implementation phase ESRI's geodatabase model is used. This is an object relational spatial database utilizing the full functionalities of the underlying relational DBMS. This hybrid approach claims to be the most promising approach to deal with the complex data types found in GIS applications and allows the storage of spatial and non-spatial data together. Besides many advantages of geodatabase, ability to implement it in a multi-user environment is the most remarkable one. In this thesis both the personal and multi-user approaches, in which the underlying DBMSs are MS access and Oracle, are tested. The multi-user geodatabase is built in three-tier architecture with ESRI's ArcSDE (Spatial Database Engine) as the middleware.

Keywords: GIS, Object Oriented GIS , ArcInfo UML Model, Geodatabase, ArcSDE

ÖZ

ŞANLIURFA HARRAN OVASI'NIN UML'DE ANALİZ ŞABLONU VE GEODATABASE'DE UYGULAMASI

Çubuk, Ulaş

Yüksek Lisans, Jeodezi, Coğrafi Bilgi Teknolojileri Bölümü

Tez Yöneticisi: Doç. Dr. Nurünnisa Usul
Ortak Tez Yöneticisi: Dr. Zuhal Akyürek

Nisan 2004, 121 sayfa

Mantıksal ve fiziksel tasarım evrelerinde nesneye yönelik kavramların benimsenmesi Coğrafi Bilgi Sistemlerinde (CBS) yükselen bir eğilimdir. CBS'in mantıksal tasarımı üzerine kapsamlı bir araştırma yürütülmüş ve pekçok kavramsal model önerilmiştir. İlişkisel veri modeli gibi klasik veri modelleri, mekansal verinin kavramsal modellemesinde yetersizliğini kanıtlamıştır. Bu nedenle yani bir modelleme araçı olan Bütünleşik Modelleme Dili (UML) endüstrideki geniş kabulünden dolayı. CBS alanında da populer bir modelleme aracı haline gelmiştir. Bu tezde GAP kapsamındaki Şanlıurfa Harran Ovası'nın analiz şablonunu oluşturmak için ESRI tarafından önerilen ve UML'in bir uzantısı olan "ArcInfo UML Model" kullanılmıştır. Önerilen analiz şablonu temel olarak Şanlıurfa Harran Ovası'nın sulama drenaj sistemi, sosyal çevresini, ve sulama yönetimi faaliyetlerini karakterize eder ve sistemin objeleri arasındaki etkileşimleri içerir. Uygulama evresinde ESRI'nin "geodatabase" modeli kullanıldı. Bu tabanındaki ilişkisel veri tabanı yönetim sisteminin (VTYS) işlevselliğini kullanan bir nesne ilişkisel mekansal veritabanıdır. Bu hibrit yaklaşım, CBS uygulamalarında bulunan kompleks veri türlerinin üstesinden gelmede en umut verici yaklaşım olduğu iddasıyla ortaya çıkar ve mekansal ve mekansal olmayan verinin bir arada tutulmasına izin verir. "Geodatabase" modelini çok kullanıcılı bir ortamda uygulayabilme özelliği, onun pekçok avantajı yanında en çarpıcısıdır. Bu tezde tabanındaki VTYS sırasıyla MS Access ve Oracle olan hem tek kullanıcılı hem de çok kullanıcılı yaklaşımlar test edilmiştir. Çok kullanıcılı "geodatabase" üç katmanlı mimaride ESRI'nin ArcSDE ürünü "middleware" olarak kullanılmıştır.

Anahtar Kelimeler: CBS, Nesneye Yönelik CBS , ArcInfo UML Model, Geodatabase, ArcSDE

In the memory of my bright days that slipped through my fingers with the sudden disappearance of my grandmother from the world in my eyes…
Her bitter end dragged me into a lasting strange nightmare…
Her devotion to life until her last breath gave me the inspiration to move on…

ACKNOWLEDGEMENTS

This thesis cannot be credited solely to the author. Through the stages of research and writing of this dissertation, I have become indebted to numerous individuals. First, I would like to express my largest gratitude to my supervisor, Assoc. Prof. Dr. Nurünnisa Usul for his continuous encouragement and guidance throughout this investigation. I cannot fully express appreciation to her for her kind help both during the corrections of this study and for her initial guidance for my steps towards academic life. I am grateful for many valuable comments and simulating discussions to Assist. Prof. Dr. Zuhal Akyürek. I express my sincere thanks to Assoc. Prof. Dr. Sibel Kalaycıoğlu for her open door, her ideas, her constructive criticism, and many discussions about GAP and Şanlıurfa Harran Plain. I also would like to thank Assoc. Prof. Dr. Ali Doğru for his expertise that shed valuable insight into this study. My knowledge about object orientation and UML consolidated with his courses at METU Computer Engineering Department. My parents have been a constant source of support – emotional, moral and of course financial – during my postgraduate years, and this thesis would certainly not have existed without them. Finally my thanks go to Can Gökçay for his support and understanding at the mostly miserable moments during these months. He has been invaluable throughout the long haul of this thesis; whenever I lapsed into depression

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Water resources of the Earth are distributed in time and space and are not infinite. Their availability may vary greatly from time to time and place to place. This variability causes problems such as drought or flood, and therefore water resources are at the heart of sustainable development in many regions of the world (Seckler et al., 1998). A growing population leads to increasing demand for water. Water of sufficient quantity and quality is an essential resource for agriculture, industry, and tourism, but also for everyday life in cities and villages (Brookfield and Byron, 1993).

Water Resources Management, which is an integrating concept for a number of water sub-sectors such as irrigation and drainage, ensures that social, economic, environmental and technical dimensions are taken into account. Mechanisms to improve water use efficiency gain more importance as the economic pressures on water resources increase. This is especially true for irrigated agriculture, a major consumer of water. In fact, agriculture is the largest water user, accounting for 70% of world water withdrawals (FAO Aquastat, 2000).

Irrigation, an input to agriculture, provides supplementary water where rainfall is not sufficient for the need of crops. Adequate and reliable water resources are a pre-requisite for efficient and effective irrigation and at the heart of sustainable agricultural development. Good irrigation depends on managers being able to apply the correct amounts of water accurately and efficiently. Because of the climatic

conditions; rainfed agriculture is very limited in Turkey. Hence, irrigation plays an important role in the agricultural sector. The largest ongoing irrigation project in Turkey is the Southeastern Anatolia Project (GAP), which is also one of the biggest of its kind in the World (Ünver and Gupta, 2002).
GAP is the most multifaceted development project in the context of sustainable development in Turkey. Its water resources program envisages the construction of 22 dams and 19 power plants and irrigation schemes on an area extending over 1.7 million hectares. The basic objectives of the project can be summarized as the improvement of living standards, poverty reduction and broad-based economic development. The Şanlıurfa Harran Plains Irrigation Project can serve as an adequate illustration of the GAP because of being the oldest component of GAP (Southeastern Anatolia Project Regional Development Administration, 2003).

The Şanlıurfa Harran Plains Irrigation Project is the first and the most important component of GAP for enabling irrigation of 152,353 hectares (GAP RDA, 1998). It is a comprehensive project that covers not only irrigation and agriculture, but also transportation, settlements, infrastructure, health and education. Within the scope of Şanlıurfa Harran Plains On-Farm and Village Development Project (GAP RDA, 1998), whose main purpose is to assist in on-farm and village development, some data have been gathered. Nevertheless there are many key development agencies that gather and consume data about the region. These agencies are; GAP RDA (GAP İdaresi), General Directorate of Rural Services (KHGM), State Hydraulic Works (DSİ), General Directorate of Agrarian Reform (Tarım Reformu), Ministry of Agriculture and Rural Affairs. Interested parties should not be limited with those, universities that conduct research projects and private companies may also concern with the data about the region. New technologies need to be used for the storage of existing data in both graphical and tabular format and conveyance of reliable data to policy makers. In this respect Geographical Information Systems (GIS) is quite powerful and effective tool. “In the strictest sense, a GIS is a computer system referenced information, i.e. data identified according to their locations." (USGS, 2003)

Up to now many projects considering various aspects have been started by GAP RDA to serve the sustainability of GAP. From the Internet address of GAP RDA (www.gap.gov.tr) finished and on-going projects can be viewed. Among these projects only some of them aim to utilize GIS. A project whose purpose was to develop an Information System of a pilot Irrigation Association (Tahılalan), could not progress because of technical problems.

As stated before multidimensionality is the most remarkable characteristic of GAP. Irrigation management is one of these dimensions which DSİ and GAP RDA pay much attention. In a project conducted by GAP RDA together with Ministry of Agriculture and Rural Affairs, cropping pattern is tried to be determined with the aid of satellite images. In a study performed by DSİ, irrigated areas are determined using Remote Sensing (RS) techniques and by this way irrigation performance can be estimated. Again in a project conducted by a private company, HATGIS, farm boundaries are determined using RS. The project and its consequences have also found a place in media (www.hatgis.com).

However, none of these studies are integrated by a comprehensive uniform GIS. The difficulties encountered through data collection phase and deficiencies in the GIS technology and design methodologies are the main causes of unsuccessful attempts. But using RS technology, it is supposed to overcome or reduce the problems faced with during data collection phase (Parcel boundaries, cropping pattern). Emergence of Object Relational Databases bring the possibility to move GIS into IS mainstream. By this way GIS data can be stored in a RDBMS, which provides a more reliable environment. Besides through the utilization of Object Oriented and Component Based approaches, development of GIS applications is getting easier.

Şanlıurfa Harran Plain that is the oldest component of GAP is selected as study area in this research. GIS for the region that is intended to be designed in this thesis, is supposed to be a precedent to other components of GAP. The problems, which may possibly arise through putting such a massive development project into practice, are first observed in the region. Many things have been written on the subject and as stated before many projects have been proposed. So the importance of accessing up-to-date information could not be underestimated. And in this thesis it is aimed to design the GIS for the region using a sound design methodology.

While designing GIS for the region mainly three aspects are taken into consideration; Irrigation and Drainage System, Irrigation Management, and Sociological matters. Irrigation drainage network being the initiator of winds of change is considered as the principal component of the system. It has become a necessity to start irrigation management activities with the construction and operation of irrigation and drainage network. Collection of irrigation water charges, which is the main concern in the GIS to be designed, is one of these activities and is the responsibility of irrigation associations. And of course observation of some sociological changes is unavoidable with the availability of water. Certainly it is not proper to restrict the database extent with only these three aspects due to the multidimensionality of the project. So it is critical to choose a GIS that is extendable.

Because of the complexity of the system the analysis phase has gone through a hard and long period of time. Many interviews are performed in DSİ and GAP RDA. Map sheets related to irrigation and drainage canals are obtained from DSİ and are used to identify the attributes of canals and hydraulic structures. Information about irrigation management is obtained through interviews and also through a research about irrigation water prices, which was conducted in METU Civil Engineering department by Böke (Böke, 1997). To form a sociological perspective about the region assistance from Assoc. Prof. Dr. Sibel Kalaycıoğlu of METU Sociology Department, who made many visits to the region, is taken. Furthermore, the web sites of governmental organizations like Ministry of Health and Ministry of National Education give an idea about the attributes of the data that need to be stored.

In this thesis ArcInfo 8.x, which is a product of ESRI, is chosen as a GIS tool. ESRI has unveiled many new features to ArcInfo in this version. Before all else designers may benefit from object oriented analysis and design methodologies. The conceptual model of the GIS to be developed can be created using Unified Modeling Language (UML) which is the most popular object oriented modeling tool in today's Information Technologies (IT) world. This is possible through utility named ArcInfo UML Model, which is installed with ArcInfo. It is somewhat an enhanced UML with added constructs for spatial features through stereotypes and tagged values.

At the end of the analysis phase a conceptual model is created using ArcInfo UML Model. Although it is critical for the success of the Information Systems (IS) being developed, in GIS community it is not very common to form a conceptual model. This is mainly because that the GIS users are from disciplines other than IT and general purpose modeling tools are not well suited for geographical applications (Filho and Iochpe, 1999). Besides the time it takes to build a conceptual model for a new application can be very long. Analysis patterns can be a solution to that problem. Analysis pattern is simply a conceptual model of a part of application, which distills the knowledge, and experience of other designers (Fernandez, 1998). It is preferable to use models or part of models that have been made for similar applications instead of starting from scratch. For example ESRI proposes many domain specific data models to provide ready to use frameworks for various industries and applications such as Hydro Data Model for hydrological applications (www.esri.com/software/arcgisdatamodels/). Following these concepts, in this thesis, Şanlıurfa Harran Plains data model is supposed to be an analysis pattern for the interested parties.

At the implementation phase ArcInfo's new geographical database model named geodatabase, which is based on object, relational technology is used. It is a revolutionary step of ArcInfo to migrate from previous version's georelational architecture in which spatial and non-spatial data are stored in separate places, to object relational architecture, which offers uniform storage of spatial and non-spatial data. By this way it could be possible to design a multi-user GIS. Geodatabase grant two options for the design; personal and multi-user. In this thesis, both of them are tried. For personal geodatabase the underlying RDBMS is Microsoft Access and for multi-user it is Oracle. ArcSDE, which is a middleware that provides the communication between ArcInfo and Oracle, is used for the multi-user geodatabase. So it would be possible to share data among many users. Also through the utilization of subtypes and domains, which are special features of geodatabase that provide easy editing and integrity constraints, it is aimed to increase the quality of the GIS.

In summary main objectives of this research is:

- To develop an analysis pattern for Şanlıurfa Harran Plains using ArcInfo UML Model as a modeling tool

- To design the geodatabase of the Şanlıurfa Harran Plains utilizing a sound design methodology, OO analysis and design methods

- To achieve an extendable and reusable data model

- To support multi-user database capabilities through implementing geodatabase in Oracle with ArcSDE as an application server

- To increase the quality of the Şanlıurfa Harran Plains geodatabase by ensuring integrity constraints

- To research object-oriented concepts and UML class diagrams
- To explore conceptual models proposed for GIS applications

1.1 Study Area

The Southeastern Anatolia Project (GAP) is Turkey's largest and most multifaceted development project in the context of sustainable development, and also, one of the largest development projects in the World. Its basic objectives include the improvement of living standards, poverty reduction and broad-based economic development. Consequently this will eliminate regional imbalances and encourage rural migrants to return home by enhancing employment opportunities in the rural sector.

The project area covers 9 administrative provinces (Adıyaman, Batman, Diyarbakır, Gaziantep, Kilis, Mardin, Siirt, Şanlıurfa and Şırnak) in the basins of the Euphrates and Tigris and in Upper Mesopotamia (Figure 1.1). Compared to the other regions of Turkey, the GAP region is disadvantaged in terms of socioeconomic development indicators. This region surrounded by Syria to the south and Iraq to the southeast has a surface area of 75,358 km^2, which corresponds to 9.7 % of the total surface area of the country. 20 % of a total of 8.5 million hectares of irrigable land in Turkey is in the GAP region made up of wide plains in the basins of Lower Euphrates (Fırat) and the Tigris (Dicle) Rivers. Initiated in 1976, the GAP is a combination of 13 projects primarily developed for hydroelectric power generation and irrigation (Table 1.1).

The province of Şanlıurfa possesses the 36 % of cultivated land in the GAP region and the most fertile agricultural land. The Şanlıurfa-Harran Plains Irrigation Project is the first and the most important components of GAP for enabling irrigation of 152,353 hectares. It is surrounded by Şanlıurfa and Harran main irrigation canals and Syrian border (Figure 1.1). The area falls within the scope of the Lower Euphrates

Project and composed of two subprojects. Irrigation water comes from the Atatürk Dam Reservoir and is delivered through the Şanlıurfa tunnels.

GAP development in the Şanlıurfa Harran Plains can serve as an adequate illustration of the potential of the project, as well as a snapshot of the potential problems that arise from such a massive and complex development.

Table 1.1: Projects and Units of GAP (Source DSİ)

	EUPHRATES BASIN		
		7	GAZİANTEP PROJECT
1	KARAKAYA PROJECT		Hancağız Dam and Irrigation
	Karakaya Dam and HEPP		Kayacık Dam and Irrigation
2	LOWER FIRAT PROJECT		Kemlin Dam and Irrigation
	Atatürk Dam and HEPP		Belkıs Nizip Pump. Irrigation
	Şanlıurfa HEPP		Pumping from Birecik Dam Res.
	Şanlıurfa Tunnel and Irrigation		**TIGRIS BASIN**
	Şanlıurfa-Harran Plains Irrigation	8	KRALKIZI-DİCLE PROJECT
	Mardin-Ceylanpınar Gravity Irrigation		Kralkızı Dam and HEPP
	Mardin-Ceylanpınar Pump.Irrigation		Dicle Dam and HEPP
	Groundwater Irrigation		Dicle Right Bank Gravity Irrigation
	Siverek-Hilvan Pump. Irrigation		Dicle Right Bank Gravity Irrigation (P2-P5)
	Bozova Pump. Irrigation		Dicle Right Bank Gravity Irrigation (P6)
3	SINIR FIRAT PROJECT		Dicle Right Bank Gravity Irrigation (P3-P4)
	Birecik Dam and HEPP	9	BATMAN PROJECT
	Karamış Dam and HEPP		Batman Dam and HEPP
4	SURUÇ-YAYLAK PROJECT		Batman Left Bank Irrigation
	Yaylak Plain Irrigation		Batman Left Bank Gravity Irrigation
	Suruç Plain Irrigation	10	BATMAN-SİLVAN PROJECT
5	ADIYAMAN KAHTA PROJECT		Silvan Dam and HEPP
	Çamgazi Dam and Irrigation		Kayser Dam and HEPP
	Gökmikan Dam and Irrigation		Dicle Left Bank Gravity Irrigation
	Koçali Dam and HEPP and Irrigation		Dicle Left Bank Pump. Irrigation
	Sırımtaş Dam and HEPP	11	GARZAN PROJECT
	Fatopaşa HEPP		Garzan Dam and HEPP
	Büyükçay Dam and HEPP and Irrigation		Garzan Irrigation
	Kahta Dam and HEPP	12	ILISU PROJECT
	Pumping from Atatürk Dam Res.		Ilısu Dam and HEPP
6	ADIYAMAN-GÖKSU-ARABAN	13	CİZRE PROJECT
	Çataltepe Dam		Cizre Dam and HEPP
	Gölbaşı, Abbasiye, Besni-Keysun, Araban, Kızılin, Yavuzeli, İncesu, Pazarcık Irrigation		Nusaybin-Cizre-İdil Irrigation
	Erkenek HEPP		Silopi Plain Irrigation

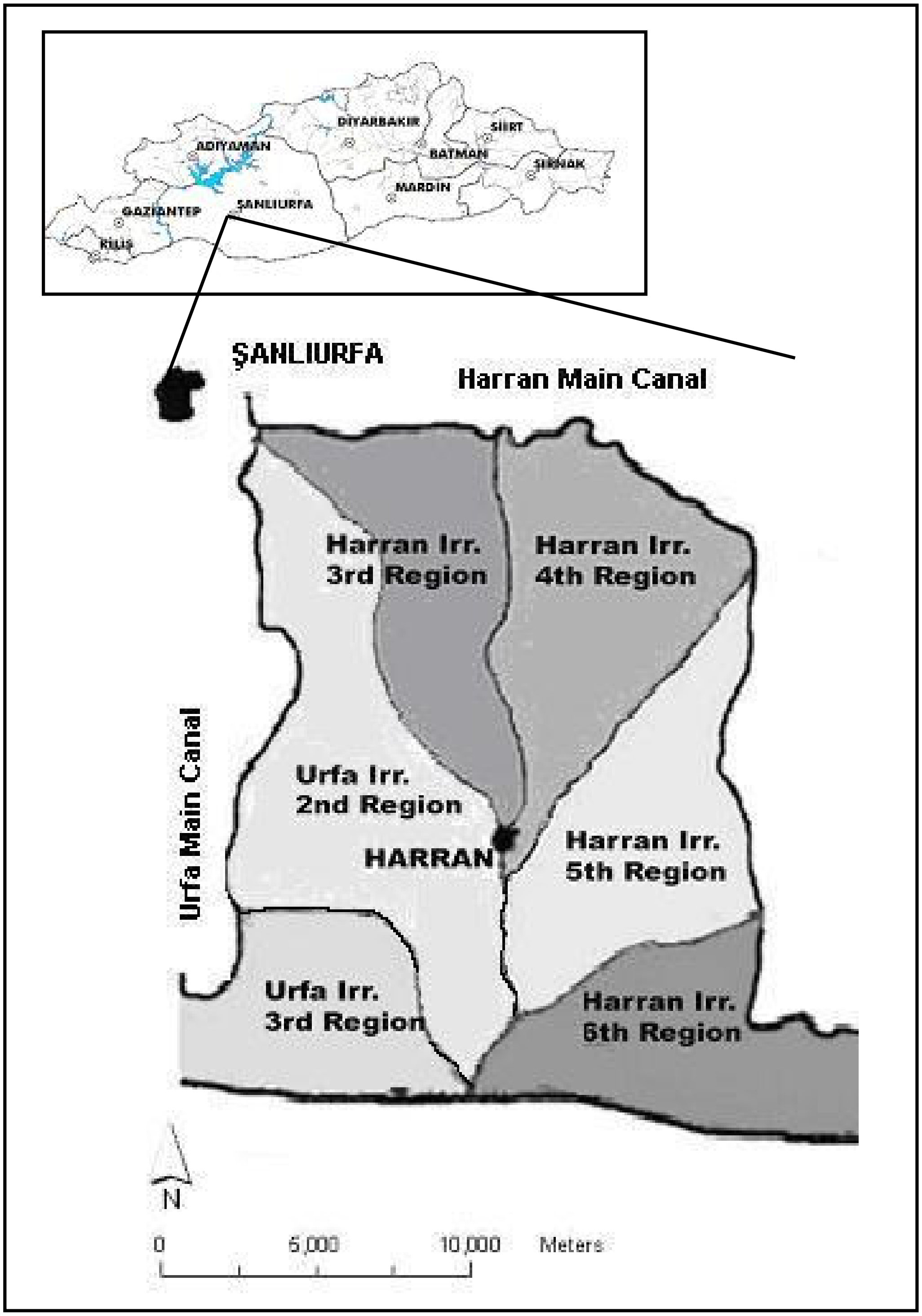


Figure 1.1: Şanlıurfa Harran Plains Irrigation

CHAPTER 2

GIS DEVELOPMENT

In its widest sense, an information system is any system that manages information. When an information system is developed, some formalised information-system development methodology is often used. Such a methodology is a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management and training for developers of information systems. The state-of-the-art of information systems development methodologies is characterised by several hundred, more or less similar, methodologies (Avison and Fitzgerald, 1995).

Techniques and tools are features in each methodology. Each technique may involve using one or more tools. Tools in this connection mean normally computer tools, usually some software used to help the systems development process, i.e. CASE tools (Computer Aided Systems Engineering Tools). Examples of current tools of information systems development are fourth generation programming languages, data dictionary systems, project management facilities and visual modeling tools.

A basic element of development methodologies for information systems is their use of data/information/object models. The model is the basis of the methodologies' view of reality (Avison and Fitzgerald, 1995). The data model describes structural and behavioural properties of the system. The model is an abstraction and a representation important elements needed to communicate between different partners involved in the system development process.

The role of models within the system-development process is similar to the role of drawings within the construction industry or manufacturing industries. The models are the "drawings" of the Information Technologies (IT) industry. Both model and drawing represent a basis and help in constructing the buildings or the IT-applications. To fulfil this communicative function the dominant types of models are iconic, pictorial or schematic.

GIS design encompasses the entire process of information system development life cycle. Geographical database design is usually divided into three major activities: Conceptual data modeling, logical database design, and physical design. Specifically, external modeling provides the scientific basis for studying and understanding the real world. At the conceptual level, organizing principles are established that transform the external data model into functional descriptions and representations of real world entities as well as define the relationships between and among them. Conceptual modeling involves the design of a schema that is an abstraction of the real world situation under consideration. A logical model is a computer model that constitutes a set of mathematical concepts used to represent an explicit form of the conceptual model. Therefore logical database design is the translation of the conceptual database design into the data model of a specific software system. And finally physical design is the representation of the data model in the schema of the software (Elmasri and Navathe, 2000).

In database design, a data model is a set of guidelines for the representation of the logical organization of the data in a database. The purpose of a data model is to find a simple and formal means of representing information and a formal means of manipulating such a representation. A DBMS is a collection of programs that enables users to create and maintain a database. It is possible to model using an object-oriented methodology and implement in a Relational DBMS (RDBMS). Of course, it is always desirable to use a kind of DBMS that can naturally implement all the constructs of the data model.

2.1 Conceptual Modeling for GIS Applications

A conceptual data model is a type of data abstraction that hides the details of data storage. The abstraction of concepts and entities that can be found in the real world is an important part of the creation of information systems. Moreover, the success of a computer implementation of an information system depends on how well the real world entities and their interactions are transposed to a computer database. Abstraction functions as a tool to help us comprehend the system, by dividing it in separate components, which are viewed in different levels of complexity and detail, according to the need to understand and represent the various real world entities and their interactions. The need for better abstraction tools in database management systems (DBMS) is one of the determinant factors of the development of object-oriented technology.

Database models are called conceptual when they depict the client's database purpose and physical when they describe its software implementation. Underlying strategies used in these multi-level ER (Entity Relationship) and OO (Object Oriented) database modeling processes are described by Rumbaugh (1996), and Bedard (1999b). The result of conceptual data modeling is a semantic (or conceptual) schema, i.e., a diagram in the notation of the chosen conceptual data model that captures the desired aspects of the modeled reality at a high level of abstraction.

To achieve a good GIS design, preparation of a conceptual schema that properly represents the entire application domain is the main step. Conceptual data models allow the representation of application domain in an abstract, formal and unambiguous way. They support the communication between the users and designers about miniworld being modeled during the requirements analysis and conceptual design phase (Iochpe and Filho, 1998). The data necessary to meet the systems objectives must be identified. The description of entities (objects), the relationships between the objects and the attributes of objects (or relationships) should be given in complete and unambiguous manner. The data description activity also includes assembly of information about the data objects, i.e., metadata (definition, data type, valid values, data quality, etc.).

Specific requirements posed on spatial database modeling impede the application of some general-purpose conceptual models as successfully as applied to conventional information systems. The main difference between geographic and conventional entities is the position in space. Establishment of a common spatial data model is a necessity for the communication among heterogeneous spatial systems. Special needs peculiar to spatial data calls for additional fundamental semantic constructs for closer representation of the world. However a model with many of these constructs besides its expressive power will increase the complexity so a balancing design decision should be chosen (Hadzilacos and Tryfona, 1997). Filho and Iochpe (1999) identify spatial aspects of data that need to be considered in modeling process: ability to differ geographic object and conventional object, possibility to represent dichotomy between field view and object view, representation of the spatial aspects of the geographic object, representation of spatial relationships between the objects, presentation of theme aspects, ability of multiple spatial representations, inclusion of temporal aspects and multiple data presentations (Davis et al., 1999).

The inadequacy of well-known conceptual data models (e.g. ER, OMT) in modeling of geographical applications stems from the fact that they do not include geographic primitives that would allow for a satisfactory representation of spatial data. In Borges et al. (2001) the difficulties in using these models are mentioned. Inadequacy of general-purpose conceptual models for geographical applications can be coped with enhancing existing models by additional constructs rather than substituting them with new models.

So far numerous conceptual data models for spatial databases have been proposed. Worboys (1995) states that two general approaches to geographic data modeling have been predominated: the Entity Relationship approach and the object-oriented approach. Clear and easy specification of spatial aspects could be possible by using these methods some of which are supported by CASE Tools (Friis-Christensen et al., 2001). Friis-Christensen et al. (2001) note that MODUL-R (Bedard et al., 1996) is the first method that introduces visual representation. Afterwards this type of visual representation has been reused in the object-oriented methods GeoFrame (Filho and Iochpe, 1999), GeoOOA (Kosters et al., 1997), GeoUML (Brodeur et al., 2000), OMEGA (Lbath, 1997; Lbath and Pinet, 2000; Pinet and Lbath, 2000), OMT-G (Borges et al., 1999; Borges et al., 2001) and MADS (Parent et al., 1999).

2.1.1 Review of some of the Proposals for Geographic Conceptual Data Modeling

GeoIFO model (Tryfona and Hadzilacos, 1995) supports the field and objects views, and defines two new concepts: "object position" and "Space-varying attribute".

An extension of ER model **MODUL-R** is proposed by Bedard et al. (1996), which supports multiple representations and temporal aspects. It uses pictograms to represent a rich set of spatial types (point, line, area, complex, alternative and multiple types) and temporality.

CONGOO is an object-oriented formalism for GIS applications, which uses semantic pictograms based on Coad/Yourdon’s (Coad and Yourdon, 1991a) specifications. In this formalism the basic components used to represent data of the real word are layers, classes, objects, relationships between different data, and integrity constraints. CONGOO distinguishes two types of topological relationships between all spatial objects, superposition and neighborhood; these facilitate processing imposed by spatial researches not defined a priori (Pantazis and Donnay, 1996).

Geo-ER (Hadzilacos and Tryfona, 1997) is an extended ER, which models the concepts of position and space depending attributes of geographic objects and includes special entity sets and relationships to express spatial peculiarities. Also "spatial grouping" and "spatial aggregation" constructs are added to model the complex geographic objects.

Geo-OM (Tryfona et al., 1997) is a prototypical object oriented model for the conceptual design of spatial databases. Special object classes and associations are added to OMT model to capture the spatial needs. "Spatial grouping" and "spatial aggregation" constructs are also valid in Geo-OM. Static and dynamic properties of spatial information could be expressed benefiting from the object oriented methodology. In (Hadzilacos and Tryfona, 1996) the comparison of Geo-IFO, Geo-ER and Geo-OM models are made according to the expressiveness, complexity, friendliness and extendibility criteria.

GMOD (Oliveria et al., 1997) data model mainly supports the field and object views. The **GISER** (Shekar et al., 1997) model considers storing metadata about spatial data quality. GISER extends the Enhanced Entity Relationship (EER) model so as continuous fields are included to unify the field and object based approaches. As an add-on to EER diagram notation dashed lines are used for continuous field and relationships.

GeoOOA (Kosters et al., 1997) data model supports topological relationships, fixed set of geometric types and temporality, and allows the representation of topological constraints. It is based on Coad/Yourdon's OOA (Coad and Yourdon, 1991a) with additional GIS-specific primitives.

MADS model, which was introduced by Parent et al. (1998), proposes a conceptual formalism for spatiotemporal data. It offers an object based model enriched by graphical notations to represent spatial types and temporal specifications.

GeoFrame is a conceptual framework that uses UML class diagram notation and provides a solution to be reused among geographic applications (Filho and Iochpe, 1999). The model does not cover semantics for temporal aspects and spatiotemporal associations but the use of analysis patterns makes it an interesting work (Friis-Christensen et al., 2001). Even if the data sharing is not possible for a particular problem, the reuse of existent modeling solution could be appropriate rather than developing from scratch. There is a high tendency to repeat the modeled entities and relationships among them in similar applications (Filho and Iochpe, 1999).

OMT-G (Object Modeling Technique for Geographic Applications), is an extended version of the OMT model for geographical applications that has been adopted by several institutions in Brazil. Although the model initially extends the classic OMT class diagram notation, later some adaptations are made based on UML. OMT-G proposes the use of three different diagrams in the process of designing a geographic application; Class diagram, transformation diagram and presentation diagram.

STER is an icon-based extension to ER model with additional constructs for spatiotemporal entities, attributes and relationships (Tryfona and Jensen, 1999).

Ext.UML (Price et al., 2000) is an extension to UML which covers the semantics of spatiotemporal data by adding a set of constructs for classes, attributes, and associations (Friis-Christensen et al., 2001).

2.2 Object Oriented Development Methodologies

Today, most conceptual modelling for development of information systems is done using the object-oriented paradigm. Object-oriented information systems development methodologies represent a synthesis between conceptual modeling and object-oriented programming. The methodologies are based on the concepts of objects and classes and at least the characteristic features of abstraction, encapsulation and inheritance (Yourdon, 1994). There is no single uniform development methodology of this type.

Currently there are several methodologies available for object-oriented analysis and design. Why is an OO methodology so important? It's a systematic way to do analysis and design. It enables planning and repeatable development. It provides a basis for developer experience. Also it avoids misunderstanding and avoid different notations for the same thing because everyone is speaking the same language.

The following is a list of OO methods from the literature:

- Berard's object-oriented design (Berard, 1991)
- Booch's object-oriented design (Booch, 1994)
- Coad and Yourdon's object-oriented analysis (Coad and Yourdon, 1991a)
- Coad and Yourdon's object-oriented design (Coad and Yourdon, 1991b)
- Jacobson's Objectory (Jacobson, 1992)
- Rumbaugh's Object modelling technique (OMT) (Rumbaugh et al.,1991)
- Object-oriented system analysis (OOA) (Shlaer and Mellor, 1988)
- Object-oriented system development (Henderson-Sellers, 1992)
- Principles of O-O Analysis and Design (Martin and Odell, 1992)
- OO Development: The Fusion method (Coleman et al., 1993)
- UML (Booch et al., 1999)

2.2.1 Fundamental Concepts of Object-Orientation

The starting point for an object-oriented analysis is to observe the external world, looking for objects and events that are relevant for the system to be developed. In principle, any kind of thing occurring in the external world is a plausible candidate to become an object. Each object is unique in the sense that it can be distinguished from all other objects. Choosing objects as the basic structuring elements for a system immediately explains one of the major benefits to be expected from the object-oriented paradigm.

Whenever a new object is created, that object will be different from any already existing object in the universe of discourse. Each object is therefore said to be *unique*: each object will have its own and unique identity, such that it can be differentiated from all other objects. As an example, each person is unique and can therefore be distinguished from any other person, even if the latter person has the same characteristics as the former person, and therefore resembles the former person as if they were twins. Each object is identified using a unique value known as an **object identifier (OID).** The unique identifier enables an object to distinguish itself from other objects when such behavior is specified (Booch, 1994).

Instead of developing descriptions of individual objects, similar objects are grouped into classes. A **class** is a pattern, template, or blueprint for a category of structurally identical items. A class can also be described as the implementation of an abstract data type (ADT). It defines **attributes** and **methods** which implement the data structure and operations of the ADT, respectively. Instances of classes are called objects. Consequently, classes define properties and behaviour of sets of objects (Lee and Tepfenhart, 2002).

The object state is represented by the values of the object's attributes. The goal in O-O analysis is to get a set of attributes such that:

- It is complete. It includes all the information pertaining an object,
- It is factorized. Each attribute captures one separated aspect of the objects' abstraction,
- It is mutually independent. The attributes take on their values independently of one another.

Behavior of objects is realized via **operations**. An operation is a function or transformation that may be applied to or by objects in a class. The execution of a method may change the state of an object.

According to Wirfs-Brock et al. (1990), the concept of **encapsulation** as used in the object-oriented context is not essentially different from its dictionary definition. Encapsulation is a mechanism for improving the flexibility and comprehensibility of a system while allowing the shortening of its development time. The data of the object are hidden from the rest of the system, and made available through the services of the class. This ensures that the integrity of the data can be assured, providing the services are written correctly. For abstraction to work, encapsulation must be present

Polymorphism is another powerful feature of object orientation. Object behavior in response to the same action or message may take many forms, depending on the class of the object. Polymorphism allows a message to be sent to different objects such that each object can respond in a way appropriate to the kind of object it is. As an example, if a message is invoked to draw the object's itself, different procedures will be followed depending on the class of object. Polymorphism reduces the effort to extend the system (Bennet et al., 1999).

Inheritance is a mechanism by which a class (subclass) refines the behavior and properties of some other class (superclass). The subclass is a superclass plus something else. Inheritance allows one class to inherit the functionality of another without having to rewrite the code. In genetics, a child has some of the traits of a parent, while still functioning and appearing quite unique. Inheritance as a programming concept works the same way. An inherited class should contain all of the implementation of its parent class (superclass, base class, etc.) through its inherited interface.

2.3 Unified Modelling Language (UML)

UML (Unified Modelling Language) is a modelling language using text and graphical notation for documenting specification, analysis, design and implementation of object-oriented system development process. UML can be used with all processes throughout the development life cycle and across different implementation technologies. It is an amalgam of three popular modeling techniques; Booch, OMT (Rumbaugh et. al., 1991) and OOSE (Jacobson, 1992) and makes use of the object-oriented paradigm. UML is backed by 12 of the industry's leading software producers some of which are IBM, Microsoft, Oracle, Digital, HP, and Unisys.

2.3.1 UML Class Diagrams

A system's structural model consists of a set of class diagrams. Class diagrams are the backbone of almost every object-oriented method including UML. They identify the class structure of a system, including the properties and methods of each class. Also depicted are the various relationships that can exist between classes. The Class diagram is one of the most widely used diagrams from the UML specification (Booch et al., 1999).

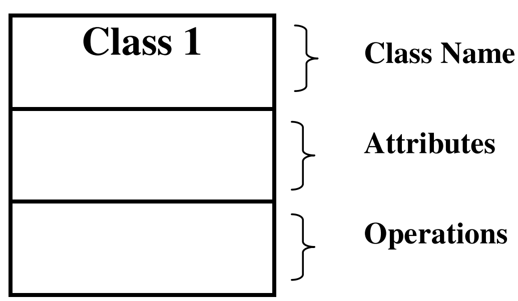


Figure 2.1: UML Class Icon

2.3.1.1 Classes

A **class** is illustrated with simply a rectangle divided into three compartments (Figure 2.1). The topmost compartment contains the name of the class. The middle compartment contains a list of attributes, and the bottom compartment contains a list of operations. Depending on the level of detail desired, it is possible to omit the properties and show only the class name and its methods, or to omit both the properties and methods and show only the class name (Roff, 2003). This approach is common when the overall conceptual relationship is being illustrated.

2.3.1.2 Abstract Classes

A class can also be abstract; which means that no instances can be created of this class. Abstract classes often represent a general term, a generic term for a set of concrete terms. Abstract class is used when the root class is meant to serve as a template for creating various subclasses. An abstract class is represented in the same way as a normal class, but in addition, the tagged value abstract is written below the class name, or the class name is set in italics (Booch et al., 1999).

2.3.1.3 Attributes

The middle section of the class box contains the attributes. An attribute represents some property of the class. The attribute name consists of text and is a noun or noun phrase. Typically, the first letter of each word in the phrase is capitalized, except the first. You can, optionally, specify the type of the attribute and possibly a default value. By specifying the type and default values, you are increasing the information content of the diagram at the possible expense of cluttering the diagram. The complete syntax for attribute is:

[visibility] name [multiplicity] [: type] [= initial-value] [{property}]

The **visibility** of an attribute or operation (Table 2.1) specifies whether it can be used by other classes; the default visibility is public.

Table 2.1: Visibility notations and their descriptions

Visibility	Notation	Description
Public	+	Any outside class can access the feature
Protected	#	Any descendant of the class can use the feature
Private	-	Only the host class can access the feature

Multiplicity allows you to specify collections. The multiplicity of an attribute establishes an upper limit for the number of values of the decorating domain that can be associated with a single object of the decorated class (Roff, 2003). Type can be a primitive type (such as integer) or a Class. Initial-value can be any suitable types. Property can be either changeable, addOnly or frozen (Table 2.2).

Table 2.2: Attribute property values

Property Value	Description
Changeable	default, freely modifiable
AddOnly	refers to multiplicities, saying that you can only add things, not delete or modify them
Frozen	refers to write-once or constant attributes. The value may not change after the object is initialized.

2.3.1.4 Methods

The bottom section of the class box contains the methods. A method is an abstraction of something that an object can do, or have done to it (Quatrani, 2003). The method name consists of text and is a verb or verb phrase, followed by left and right parenthesis. Typically, you capitalize the first letter of each word in the name, except the first. You can, optionally, specify the names, types and default of all method arguments as well as the return type of functions. The complete syntax for operations and parameters (Figure 2.3) are:

[visibility] name [(parameter-list)] [: return-type] [{property}]

[direction] name: type [= default-value]

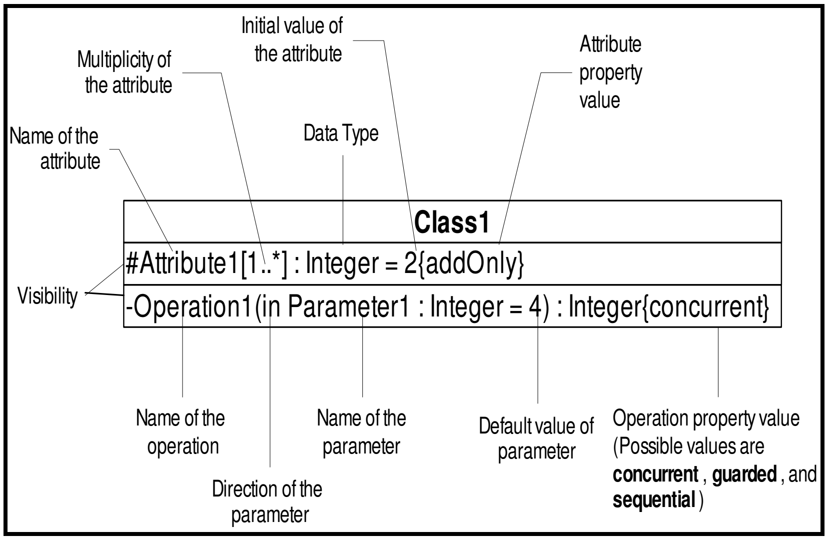


Figure 2.3: Complete syntax for attributes and operations

2.3.1.5 Associations

Associations represent static relationships between classes. Relationships provide a pathway for communication between objects (Muller, 1999). There is an association between two classes if an instance of one class must know about the other in order to perform its work. In set theory terms, each class corresponds to a set of things, and an association corresponds to a mapping between members of these sets. In a diagram, an association is a link connecting two classes.

An association has two ends. Eventually a name can be assigned to the visible ends of the relation. Such a name is called a **role** because it expresses what role the accessible class plays within the context of the other class (Figure 2.4). In other words an Association Role indicates the purpose that the class plays in the association. Roles can be explicitly named, but they don't have to be.

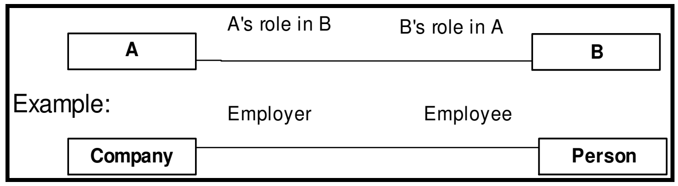


Figure 2.4: Association Roles

The **multiplicity** of an association end is the number of possible instances of the class associated with a single instance of the other end (Table 2.3).

Table 2.3: Multiplicities of an association

Multiplicities	Description
0..1	zero or one instance.
0..* *or* *	no limit on the number of instances (including none).
1	exactly one instance
1..*	at least one instance

2.3.1.6 Aggregation and Composition

Aggregation is an association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole. Hollow diamond is used to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the "part" class, but the two classes are not dependent on each other. Composition is a special type of aggregation that has strong ownership and simultaneity of lifetimes of the classes. That is, if the part class belongs to just one whole class and cannot exist outside the whole object, the relationship is composition. The diamond end in both a composition and aggregation relationship points toward the "whole" class or the aggregate (Figure 2.6).

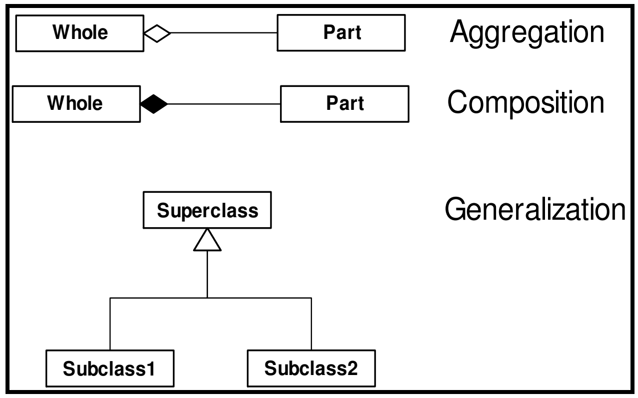


Figure 2.6: Notation for aggregation, composition and generalization

2.3.1.7 Generalization

Generalization is an inheritance link indicating one class is a superclass of the other. A generalization is represented by a line ending on a triangle on the parent class (Figure 6). Subclasses must obey all semantic restrictions of their superclass(es). The attributes, operations, relationships and constraints defined in the superclasses are fully inherited in the subclasses.

2.3.2 UML Packages

Package is a general purpose mechanism for organizing elements into groups. Packages are collections of model elements of arbitrary types which are used to structure the entire model into smaller clearly visible units. A package defines a namespace, that is, the names of the elements within a package must be unique. Each model element can be referenced in other packages, but it belongs to exactly a single (home) package. Packages may be nested within other packages. A system may be thought of as a single high-level package, with everything else in the system contained in it. A model element can be contained in several packages, but each element has its home package. In all other packages, it can only be quoted in the form: PackageName::ClassName. This creates dependencies between the packages: one package uses classes of another package (Booch et al., 1999).

2.3.3 UML Extension Mechanisms

The UML provides a rich set of modeling concepts and notations that have been carefully designed to meet the needs of typical software modeling projects. However, users may sometimes require additional features and/or notations beyond those defined in the UML standard. These needs are met in UML by three built-in extension mechanisms that enable new kinds of modeling elements to be added to the modeler's repertoire as well as to attach free-form information to modeling elements. Stereotypes and Tagged Values are extension mechanisms which can be used separately or together to define new modeling elements that can have distinct semantics, characteristics and notation relative to the built in UML modeling elements.

A stereotype is UML's way of attaching extra classifications to model items; it is one of the ways that UML is made extensible. The stereotype describes a model element, and is placed close to the affected element on a diagram, giving extra information about that element. Some stereotypes are predefined in UML; they are automatically available and cannot be redefined. Double angle brackets (<<...>>), "guillemots" are used to identify stereotypes (Booch et al., 1999).

Tagged values are user-defined, language and tool specific keyword/value pairs which extend the semantics of individual model elements with specific characteristic properties. Tagged values add specific additional properties to existing model elements. They detail the semantics of a model element and can influence code generation. Tagged values consist of a keyword and a value, and are enclosed in braces (Booch et al., 1999).

CHAPTER 3

ŞANLIURFA HARRAN PLAINDATA MODEL AND GEODATABASE CREATION USING CASE TOOLS

3.1 CASE Tools

A computer-based product aimed at supporting one or more activities within any aspect of the software development process is called CASE (Computer Aided Software Engineering) Tool. Usually, CASE tools are also defined as browsers and editors for models in graphical and textual form. Drawing graphical models on paper is time consuming and error prone, particularly as they need to be frequently revised. Thus dedicated drawing tools can support a similar level of support to software engineers that CAD tools provide in other disciplines.

CASE tools provide support for modeling notations and methodologies. Many of these modeling CASE tools also support relational databases by performing logical and in some cases physical database modeling and design, including schema generation and reverse engineering of RDBMS tables and other elements. The Unified Modeling Language (UML) is an industry standard to support this process.

CASE tools offer many benefits for developers building large-scale systems. Continuous changes to user requirements drive system complexity to new levels. CASE tools enable us to abstract away from the entanglement of source code, to a level where architecture and design become more apparent and easier to understand and modify. The larger a project, the more important it is to use CASE technology. It enhances reuse of models or models' components leading to time and effort reduction. Moreover CASE tools help the standardization of diagrams and communication between developers and clients. (Pressman, 2001)

3.2 CASE Tools and GIS

The need for automatic production of complex and time consuming GIS applications force designers to utilize CASE tools (Lbath and Pinet, 2000). Implementation of formal methods with CASE tools yields faster and better results. CASE tools reduce the time required for drawing and editing process of conceptual schema so that more time could be spent on analysis phase (Bedard, 1999a). The review of CASE tools proposed for GIS applications is given below.

Orion (Pageau and Bedard, 1992), which supports the Modul-R formalism, is the fist CASE tool for spatial applications.

COBALT (Parent, 2000) is a CASE tool for spatiotemporal applications which provides a visual schema editor to draw conceptual schemas and also a schema checker module to check the convenience with integrity rules.

AIGLE is a CASE tool, which has been marketed since 1998 and supports OMEGA method; an UML based method for the design of GIS. Generation of prototypes on marketed GIS and Oracle is possible by AIGLE. The main advantages of it are listed in Lbath and Pinet (2000) as

- The independence of target GIS
- Automatic code generation
- An incremental system design
- Integration in a CASE tool

Perceptory is a CASE tool based on user concepts of human "perception" and geospatial "repository" which is a collection of metadata structured to provide the semantics and structure of objects stored in database. Perceptory is expressed with the UML class diagram extended with PVL (Plug-in for Visual Languages) which is a graphical notation depicting geometric, temporal and visual properties of objects and attributes (Bédard, 1999b). It is designed as Visio template.

Some of the other modeling methods supported by CASE tools are GeoOOA (Kosters, 1997) and MADS (Parent et al., 1998).

4.3 CASE Tools Subsystem of ArcInfo

The CASE tools subsystem of ArcInfo 8 has two parts: the Code Generation Wizard and the Schema Wizard. Blueprints of the geodatabase structure can be created graphically with the CASE tools subsystem. Object data model is designed by using UML class diagrams and exported into the Microsoft Repository and into ArcCatalog to create a geodatabase schema.

Microsoft Repository is a software tool through which object-oriented software models can be stored, retrieved, and exchanged. UML models stored in Microsoft Repository retain all the objects, interfaces, data types, properties, and relationships expressed in object-oriented programming languages. Once a model is stored in the Repository, engineers can access it with CASE tools, such as Microsoft Visio, to rapidly generate framework code directly from model components. This enables teams using various modeling or development tools to freely exchange and build on UML components and class diagrams, and thereby expedite development.

Object data model represents the geodatabase elements such as feature classes, subtypes, domains, geometric networks and relationships. Optionally some additional behavior can be modeled by creation of custom features, interfaces and class extensions.

Visio is a drawing and diagramming program with a user-friendly interface that offers a set of tools you can use to graphically organize and display information. With Visio, you can visualize and communicate your information in a concise diagram.

Template is a Microsoft Visio file with .vst extension that presents one or more stencils. The stencil organizes a selection of shapes typically used in the type of diagram you selected. Shapes provide the foundation for your drawing. Visio supplies solution templates for a variety of categories (Figure 3.1). You select an appropriate template for what you want to do and to create your drawing; you simply drag and drop the shape you want from the stencil to the drawing board. You can connect the shapes and add text to enhance the clarity of your drawing. You can also tailor your diagram to add visual impact by changing color, fill patterns, line properties and other shape features.

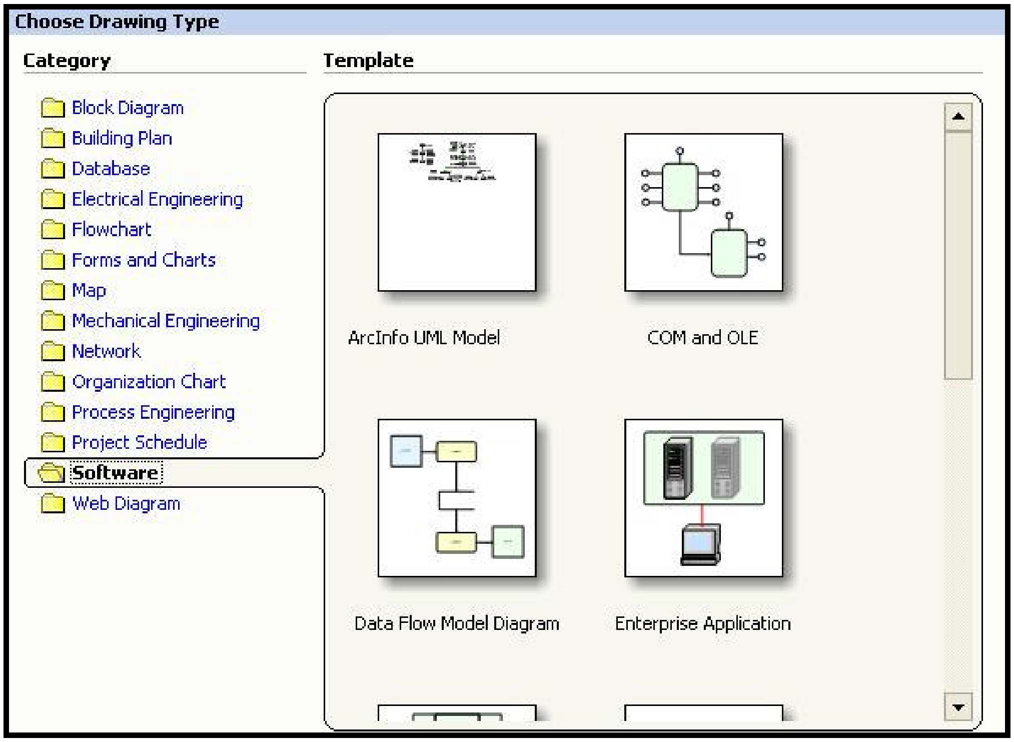


Figure 3.1: Visio Templates

3.4 ArcInfo UML Model

ArcInfo UML Model templates installed with ArcGIS are located in casetools/uml directory. The ArcInfo UML Model contains the relevant parts of COM classes that belong to the geodata access components of ArcInfo. The object model required to design geodatabase has five packages: Logical View, ESRI Classes, ESRI Network, ESRI Interfaces, and Workspace (Figure 3.2).

UML Packages are a grouping of objects into sets of objects that provide related services. Packages act as directory structures to organize the elements of a system into related groups and contain any number of UML elements, such as other packages, classes, interfaces, and diagrams.

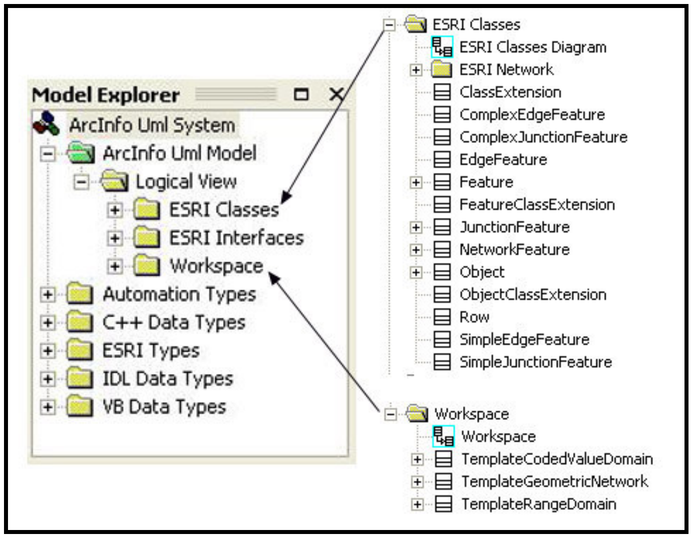


Figure 3.2: ArcInfo UML Model

ArcInfo UML Model has four diagrams: ESRI Classes, ESRI Generic Junction, ESRI Interfaces and Workspace Diagrams.

Logical View is the root level package and contains ESRI Classes, ESRI Interfaces, and Workspace. **Workspace package** represents the geodatabase and it is critical that all diagrams you create are located in this folder otherwise you will not be export them to ArcCatalog. **ESRI Classes package** contains the ESRI Network package, ESRI classes and ESRI Classes Diagram

ESRI Interfaces package contains the ESRI interfaces. An **Interface** is a collection of operations that are used to specify a service provided by a class or component. It represents a contract with the user. In UML an Interface is represented either by a "lollipop" or by a class shape with the stereotype <<interface>>. Interfaces attached to the classes that implement them through UML refinement relationships.

3.4.1 Feature datasets

Feature datasets are represented as stereotyped packages in UML (Figure 3.3). Since geodatabase model do not allow feature datasets to be nested, a feature dataset package cannot be created under another feature dataset package. But other packages can be created under a feature dataset package to enhance the organizational structure of the model. Spatial reference for a feature dataset is not modeled in UML; instead it is set when generating the schema in ArcCatalog.

Figure 3.3: Feature Dataset

Feature classes and tables are modeled as UML classes inherited from the Feature and Object in the ESRI Classes package (Figure 3.4). Note that Feature is inherited from Object and the generalization relationship is automatically inserted when it is added to the diagram. Properties of the object as well as the properties of the base classes are mapped to fields of the table or feature class in the geodatabase.

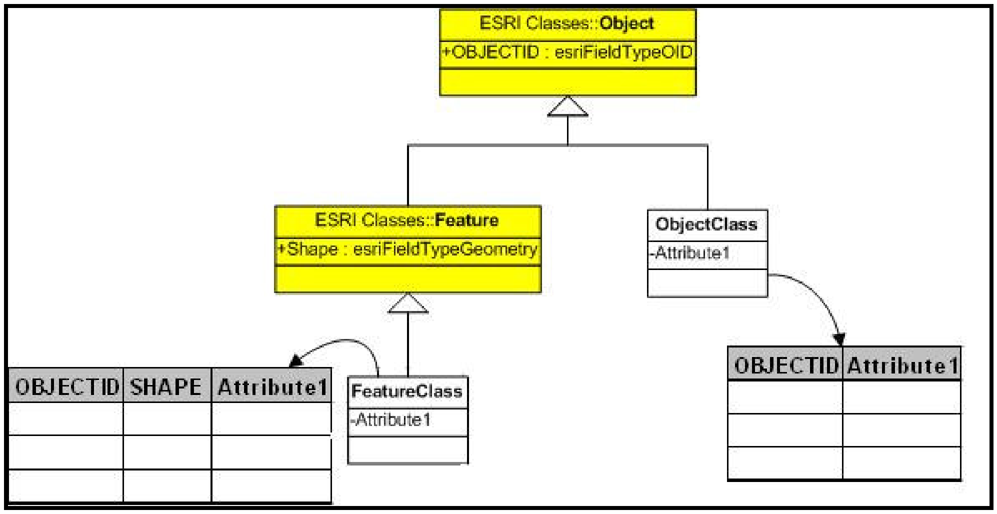


Figure 3.4: Feature and Object Classes

3.4.2 Subtypes and Domains

Coded value and range domains are stereotyped classes. Their valid values, split policies, and merge policies are also specified. In ArcInfo UML Model there are template classes for attribute domains (Figure 3.5). Domains are applied to a particular field by specifying the name of the domain as the field type. Subtypes are represented as classes connected to the parent class through an association stereotyped as "Subtype". Also the subtype field in the parent class is stereotyped as "SubtypeField".

3.4.3 Relationships

Relationship classes are modeled in UML as associations between objects. Tagged values of the UML association are used to specify the primary and foreign keys. Attributed relationships are modeled as classes stereotyped as a relationship class.

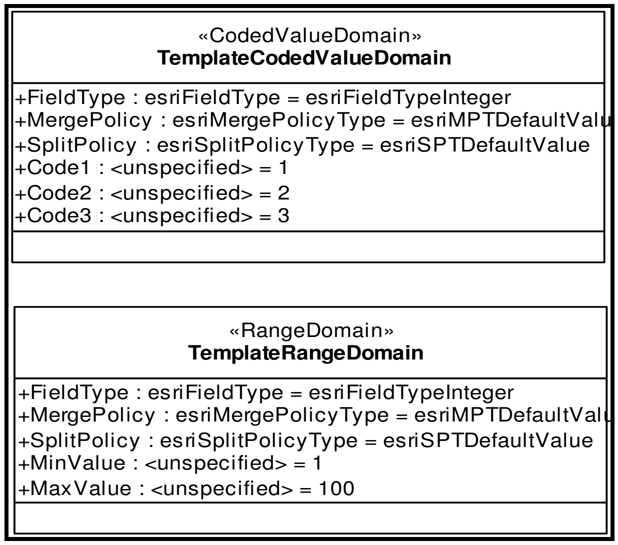


Figure 3.5: Coded value and Range Domains

3.4.4 Tagged values

Tagged values are used to enhance the UML elements. Tagged values are recognized on several other UML elements: class, attribute, associations, and so on. Tables 3.1, 3.2, 3.3, and 3.4 summarize the tagged values used for geodatabase schema elements.

Table 3.1: Tagged Values used for Fields

Tagged value name	Values	Remarks
Precision	Integer value	Integer Fields: Number of the digits Double Fields: Total number of digits
Scale	Integer value	Number of decimal places in single and double fields
Length	Integer value	Width of character fields
AllowNulls	True/False	

Table 3.2: Tagged Values used for Relationship Classes

Tagged value name	Values	Remarks
Notification	esriRelNotificationBackward esriRelNotificationBoth esriRelNotificationForward esriRelNotificationNone	
IsAttributed	True/False	
OriginPrimaryKey	Name of the primary key field of the origin class	
OriginForeignKey	Name of the foreign key field of the origin class	For 1-1 and 1-M relationship classes, this field lives in the destination class. For 1-M/attributed relationship classes, this field lives in the auxiliary <<RelationshipClass>> class.
DestinationPrimaryKey	Name of the primary key field of the destination class	Valid for M-M/attributed relationship classes only.
DestinationForeignKey	Name of the foreign key field of the destination class	Valid for M-M/attributed relationship classes only. This field lives in the auxiliary <<RelationshipClass>> class.

Table 3.3: Tagged Values used for Domains

Tagged value name	Values
Description	String value

Table 3.4: Tagged Values used for Feature and Object Classes

Tagged value name	Values	Remarks
Geometry type	esriGeometryPoint esriGeometryPolygon esriGeometryPolyline esriGeometryMultipoint	Valid only for feature classes
Ancillary role	esriNCARNone esriNCARSourceSink	Valid only for junction feature classes
ConfigKeyword	String value	
HasM	True/False	Valid only for feature classes
HasZ	True/False	Valid only for feature classes
CLSID	GUID in registry format	

3.5 Şanlıurfa Harran PlainData Model Creation using UML

In this study Şanlıurfa Harran PlainGeodatabase is modeled in UML. The ArcInfo UML Model is used as a template. The tree view of the Şanlıurfa Harran PlainData Model in Model Explorer window is shown in the Figure 3.7. There are five packages under the Workspace package, which corresponds to the geodatabase: Domains, Irrigation Drainage System, Irrigation Management, Objects, and Sociological. Irrigation Drainage System, Irrigation Management, and Sociological packages are stereotyped as feature dataset.

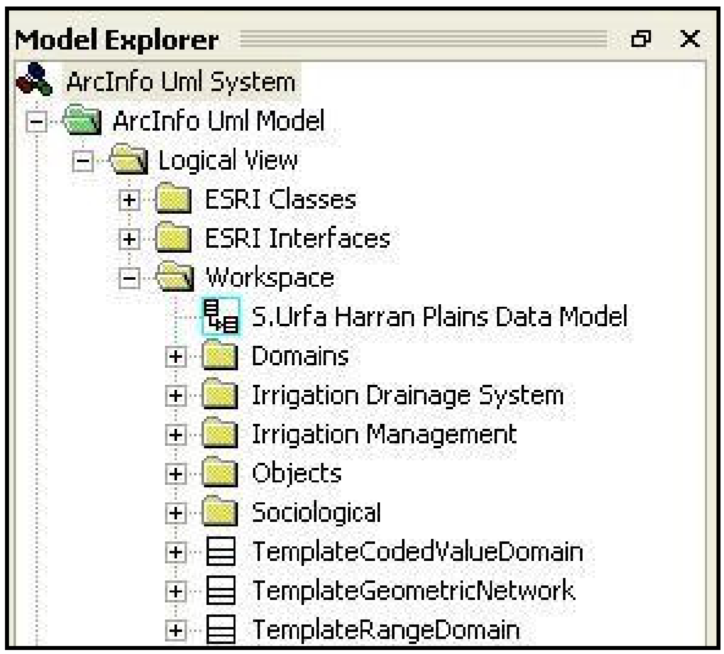


Figure 3.7: Tree View of Şanlıurfa Harran Plain Data Model

Since the object classes are implemented outside the feature datasets, they are organized into a different package called Objects (Figure 3.8). Domains package collects the special classes which will be mapped to geodatabase domains (Figure 3.9).

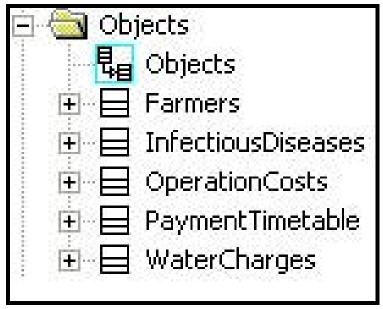


Figure 3.8: Objects Package

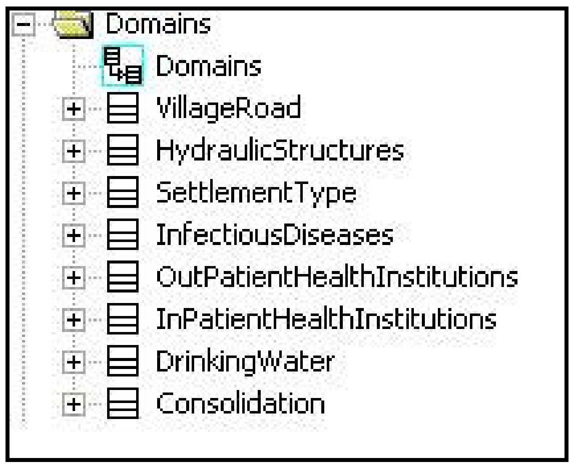


Figure 3.9: Domains Package

The classes created under Irrigation Drainage System package are shown in Figure 3.10. In Figures 3.11 and 3.12, the classes that belong to Irrigation Management and Sociology packages can be seen respectively. Notice that the attributes of the classes can also be viewed (Figure 3.11). The blue-framed icons correspond to the model diagrams, which display the relationships between the classes of the Şanlıurfa Harran Plain Data Model. These diagrams are depicted in Figures 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.19 and 3.20. In Appendix 1 a poster, which integrates all these diagrams into one final analysis diagram, is provided.

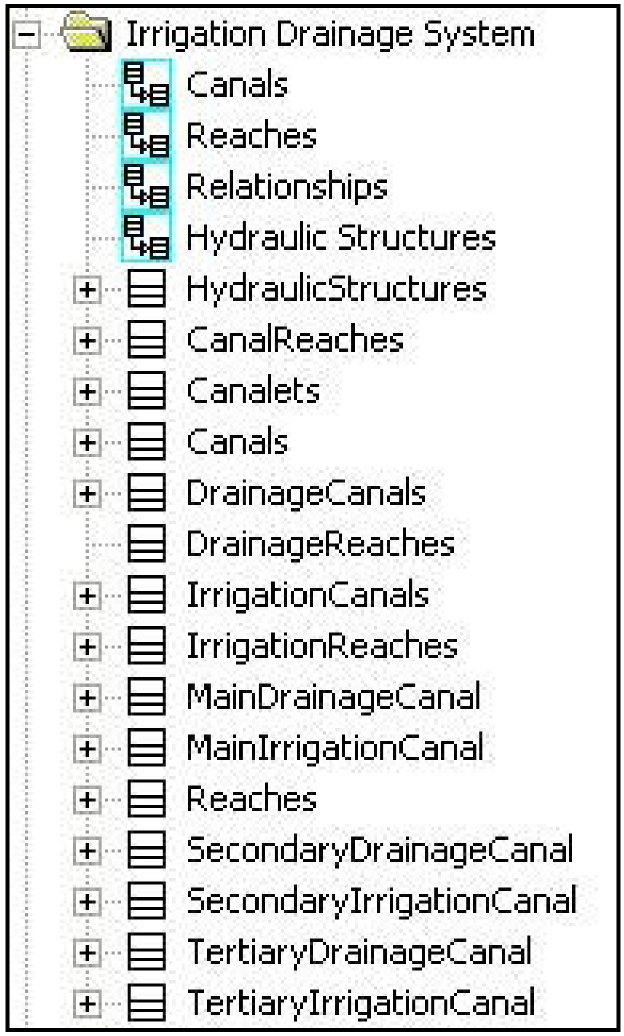


Figure 3.10: Irrigation Drainage System Package

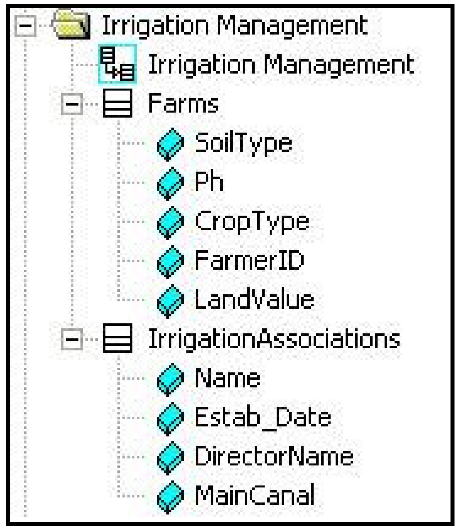


Figure 3.11: Irrigation Management Package

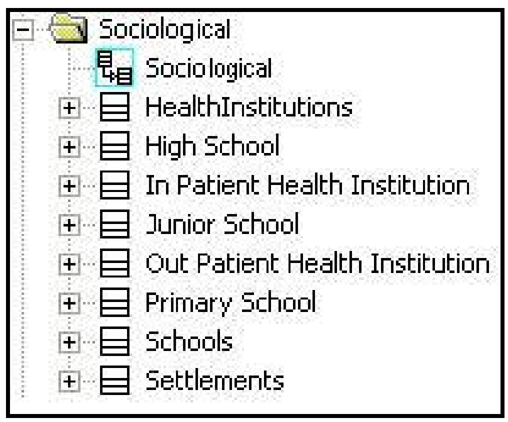


Figure 3.12: Sociological Package

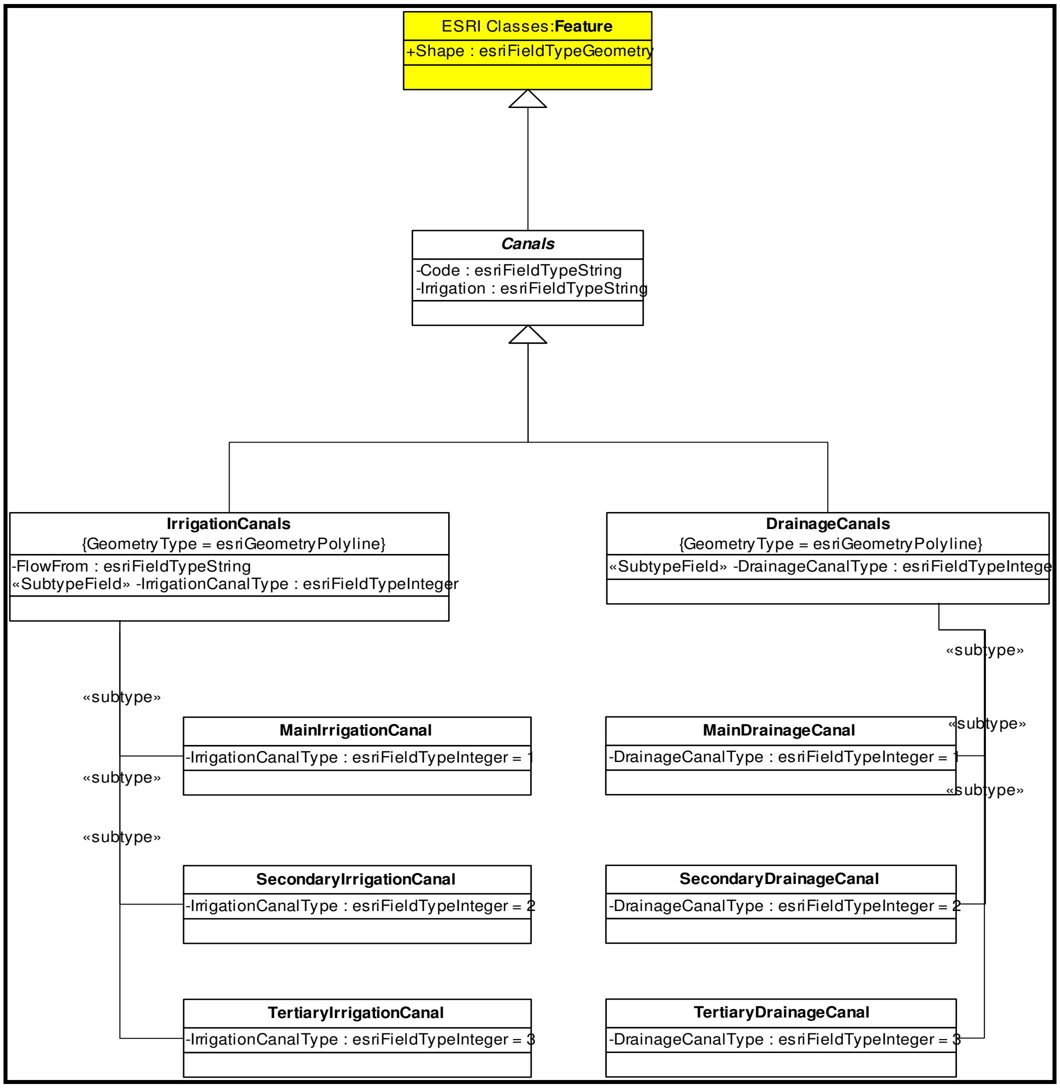


Figure 3.13: Canals Diagram

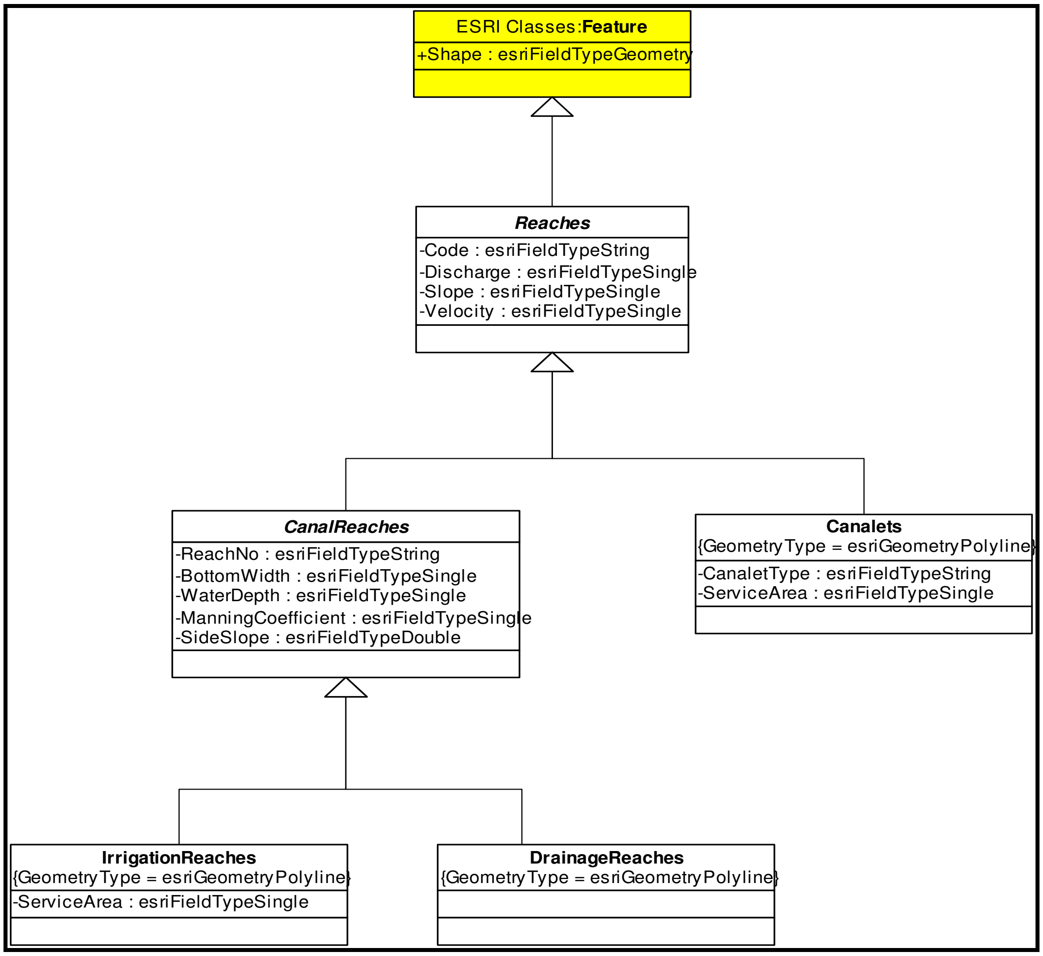


Figure 3.14: Reaches Diagram

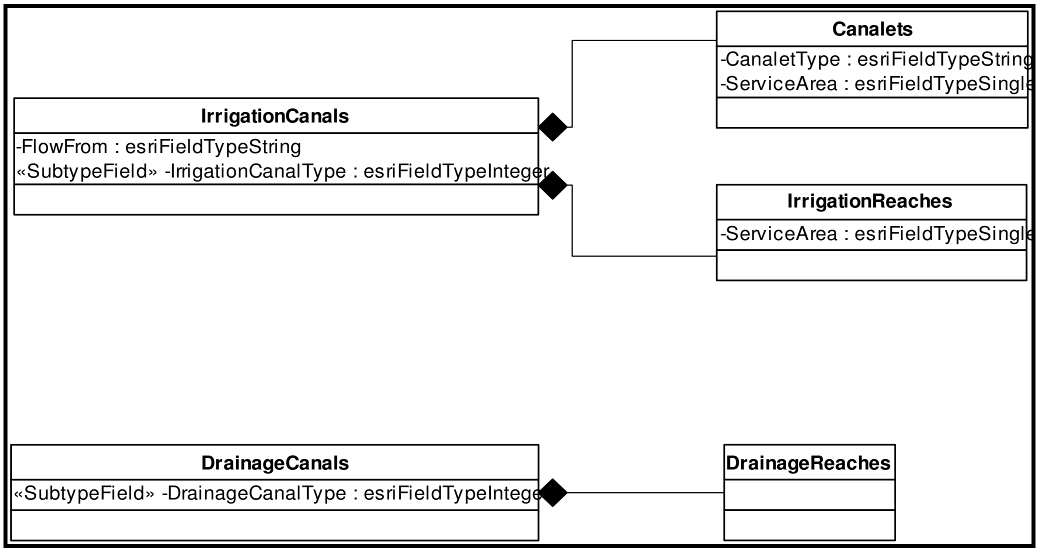


Figure 3.15: Relationships Diagram

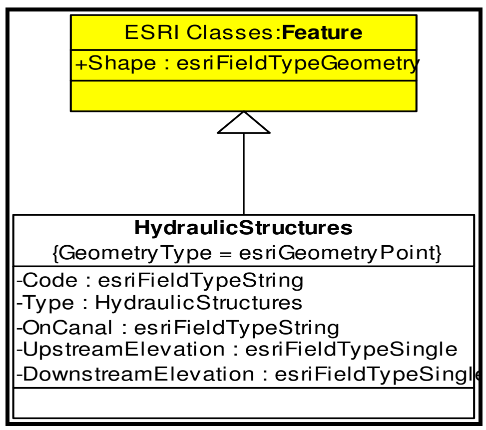


Figure 3.16: Hydraulic Structures Diagram

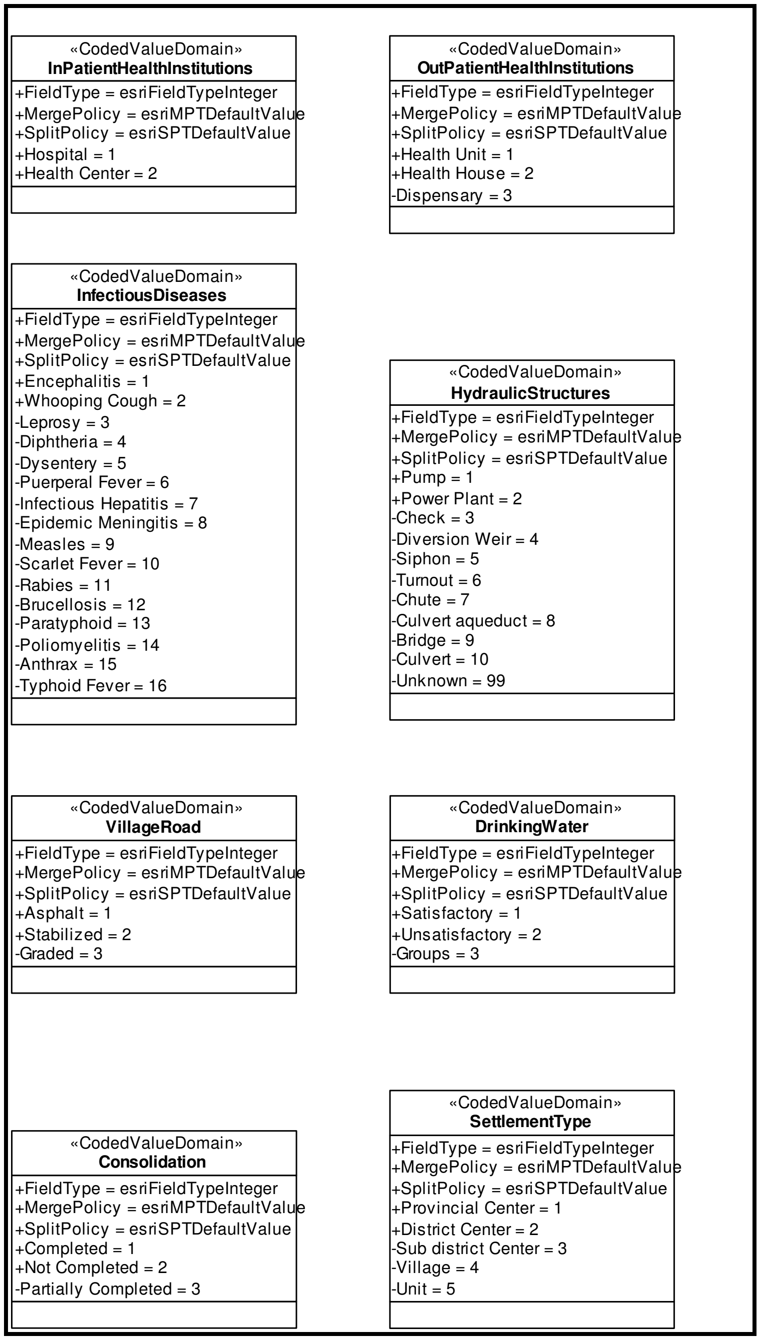


Figure 3.17: Domains Diagram

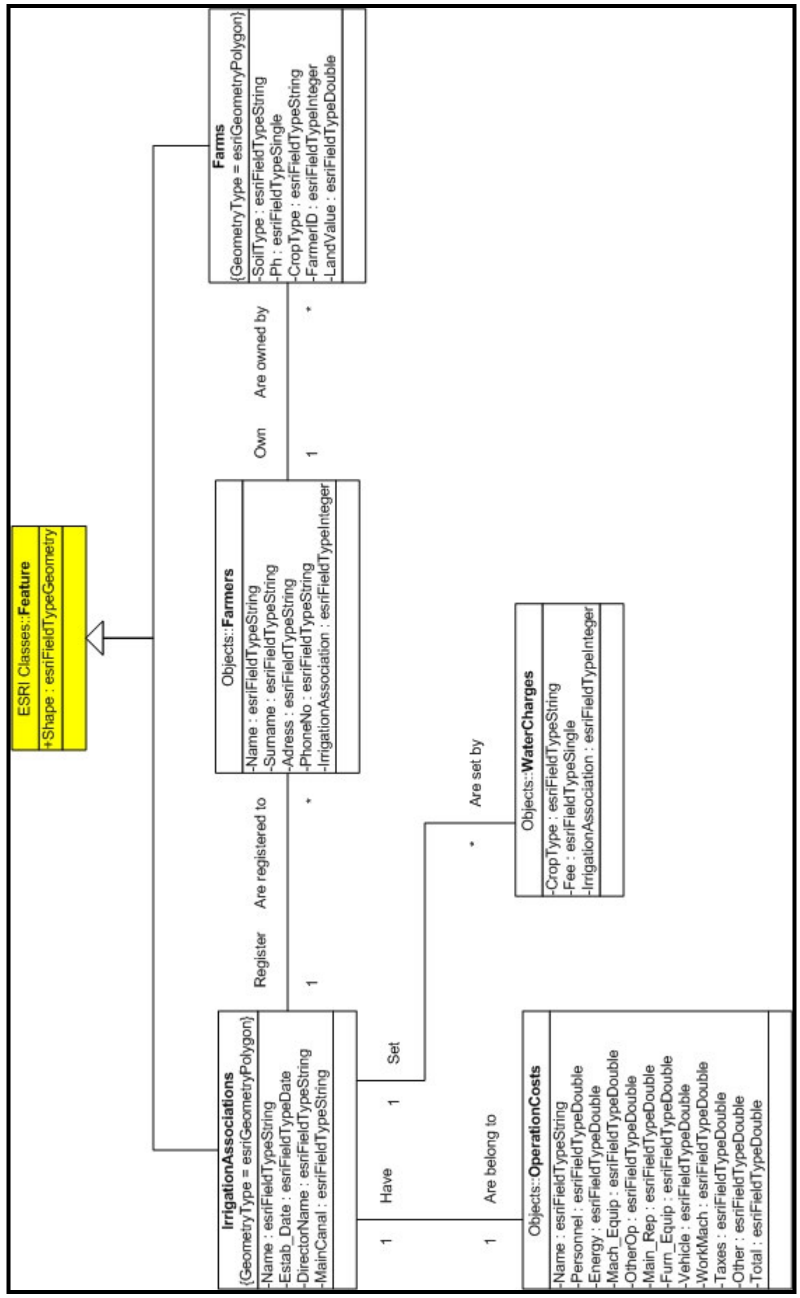


Figure 3.18: Irrigation Management Diagram

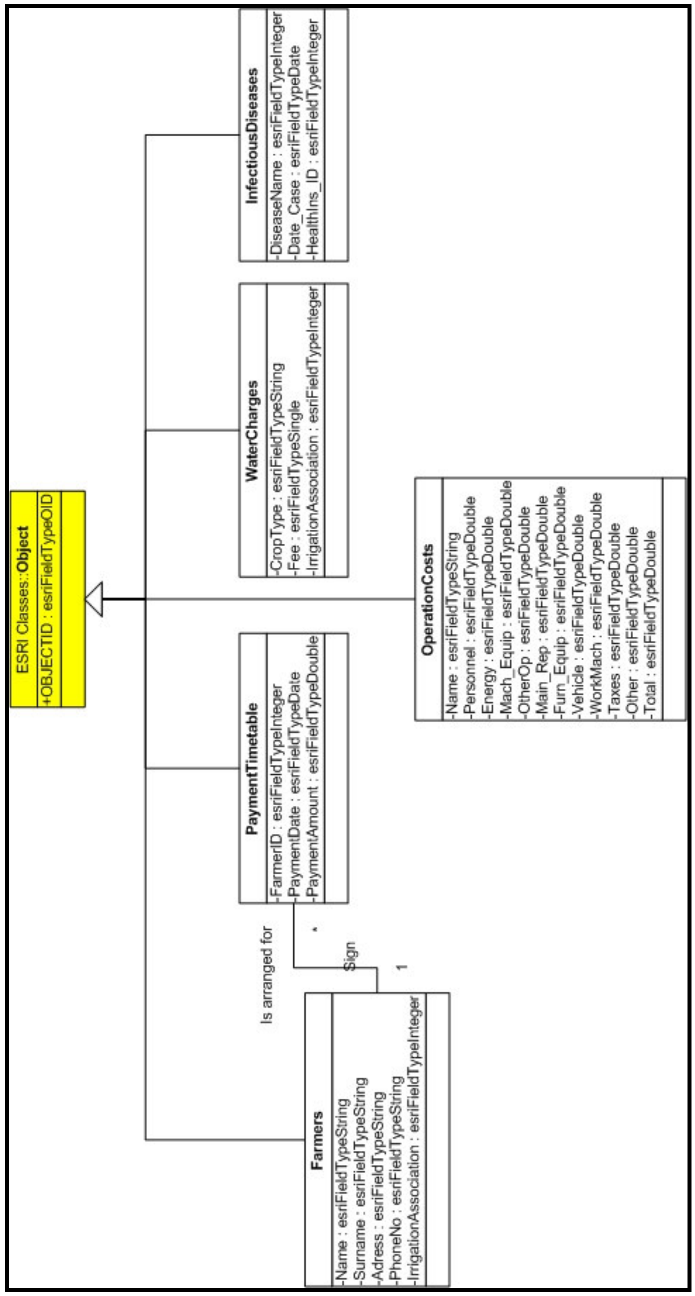


Figure 3.19: Objects Diagram

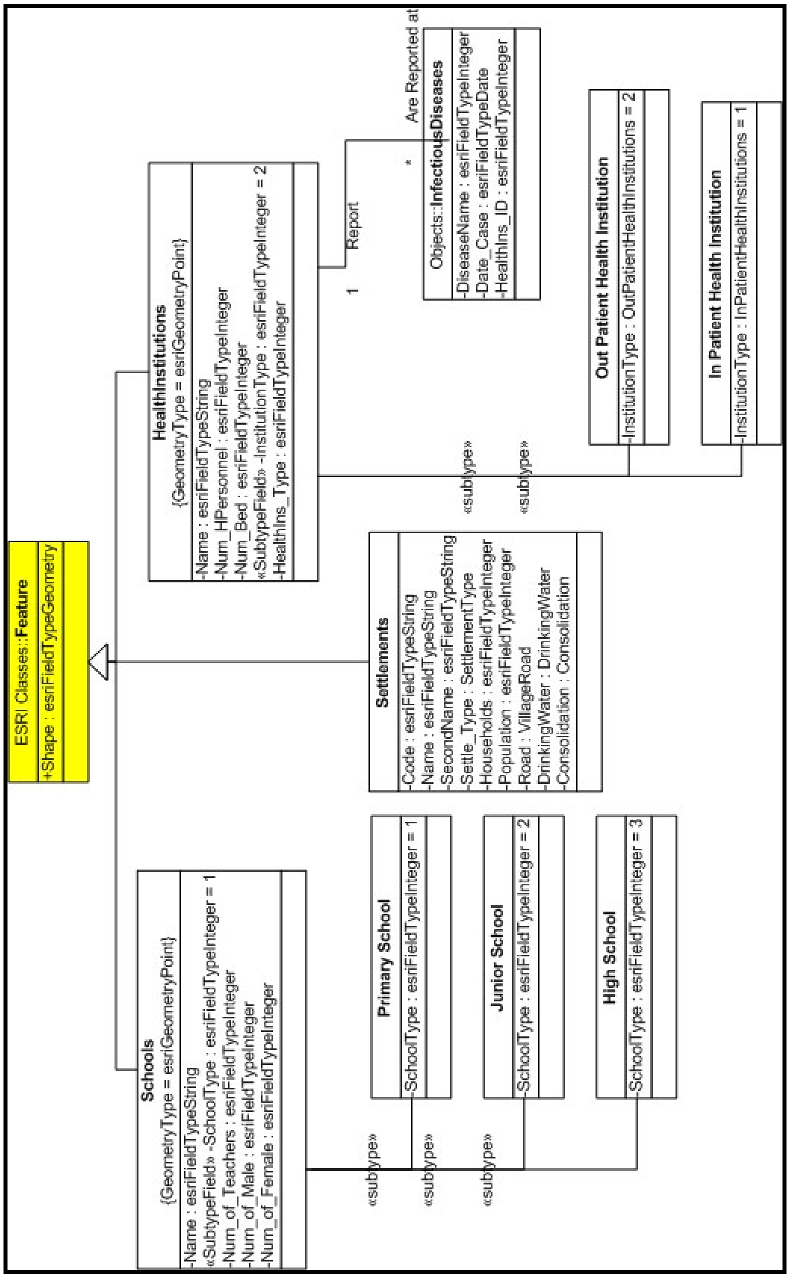


Figure 3.20: Sociological Diagram

3.6 Şanlıurfa Harran PlainGeodatabase Schema

Information system plays vital role in planning and development of rural areas. The invention of high-tech in the field of telecommunication, remote sensing and computers would lend a valuable support to spatial planning process. The most advanced computer based information technology tool for spatial planning is the Geographic Information System, which would become indispensable in planning and management of database. GIS can be used as an effective tool for village and irrigation information system, which will help the planners and administrators to identify the problems associated with rural activities, location and provision of appropriate facilities, monitoring and maintenance management of the assets created in rural areas. As a GIS tool ESRI's software ArcInfo 8.x which offers object oriented analysis and design capabilities and an object-relational data model named as geodatabase, are tried to meet the needs of the region.

The Şanlıurfa Harran PlainData Model created in ArcInfo UML Model is implemented as geodatabase. Geodatabase intuitively represents this complexity as it supports a high level of data representation. This section explains the geodatabase schema and its elements (Figure 3.21). Logically related spatial features are organized into feature datasets; Irrigation Management collects the spatial features about irrigation management activities, Sociological collects features representing sociological structures and Irrigation Drainage System organizes the features participating in the irrigation and drainage system. Domains, which help to ensure database integrity constraints, are also created as property of the geodatabase.

The Şanlıurfa Harran PlainGeodatabase is one to one mapping of the data model created in UML. Packages are implemented as feature datasets. All the classes inherited from Feature class, which is a registered class of ArcInfo UML Model located in ESRI Classes package, are implemented as Feature Class with a proper geometry that is fixed by a tagged value (GeometryType) in UML. All the classes inherited from Object class (again a class of ESRI Classes package) are implemented as Object Classes (non-spatial classes). The associations between classes correspond to relationship classes. The classes linked to feature classes via subtype relationships are not implemented as separate classes; they become a property of the related feature classes. Also the domain classes are not created as separate classes; they become a property of the geodatabase.

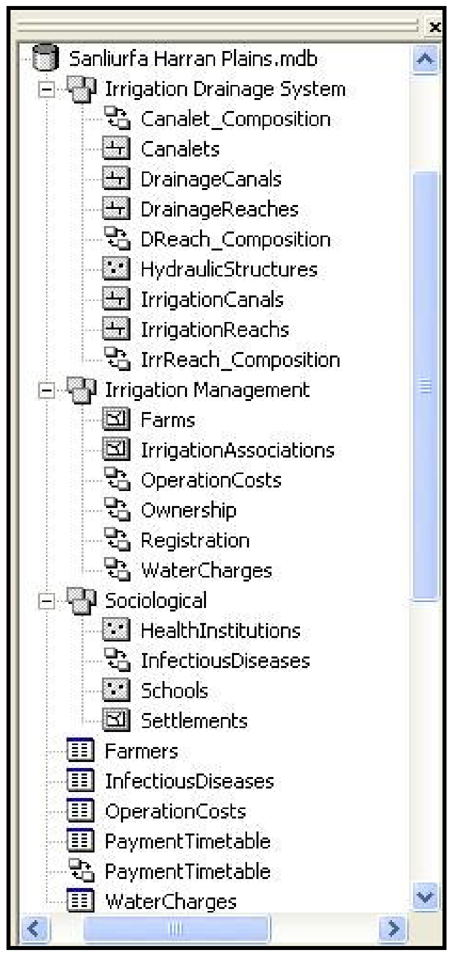


Figure 3.21: Şanlıurfa Harran Plain Geodatabase

3.6.1 Irrigation Management

The importance of relevant and opportune information in decision making for the management of irrigation systems cannot be overemphasized. Water management consists of information about determining when to irrigate, and the operation and maintenance of the irrigation system.

Within the geodatabase spatial entities and relationships related to irrigation management activities are organized into Irrigation Management Feature Dataset and naturally object classes are implemented outside the dataset. All of the geodatabase elements created for this purpose are listed in Table 3.5.

Table 3.5: Geodatabase Elements related to Irrigation Management Activities

Name	Type
Irrigation Management Feature Dataset	
Farms	**Feature Class**
IrrigationAssociations	**Feature Class**
Ownership	**Relationship Class**
Registration	**Relationship Class**
WaterCharges	**Relationship Class**
Farmer	**Object Class**
PaymentTimetable	**Object Class**
OperationCosts	**Object Class**
WaterCharges	**Object Class**
PaymentTimetable	**Relationship Class**

The attributes of the objects in Irrigation Management feature dataset are generated according to the interviews in DSİ and GAP RDA, the thesis work conducted at METU Civil Engineering department by Böke (Böke, 1997), and the Yerel-NET Portal (www.yerelnet.org.tr) which is executed by TODAIE-YYAEM (Public Administration Institution for Turkey and Middle East - Local Managements Research and Education Center) with the resources provided by SPO (State Planning Organization).

3.6.1.1 Farms

Farm is a land devoted to agricultural production and represents the crucial element of it. Within the geodatabase farms are represented as polygon feature class (Table 3.6). The information about farmers is kept within an object class structure (Table 3.7). (Note: The field named as IrrigationAssociation designates the OBJECTID of the Irrigation Association to which the farmer is registered.) And the ownership relationship which exits between farmers and farms is maintained through a relationship class (Table 3.8).

Table 3.6: Attributes of Farms Feature Class

Field Name	Data Type	Instance *
OBJECTID	Object ID	*1*
SHAPE	Geometry	*Polygon*
SHAPE_Length	Double	*0*
SHAPE_Area	Double	*0*
SoilType	Text	*Red Yellow Latasol*
Ph	Float	*8.2*
CropType	Text	*Cotton*
FarmerID	Long Integer	*1*
LandValue	Double	*50000*

Table 3.7: Attributes of Farmers Feature Class

Field Name	Data Type	Instance
OBJECTID	Object ID	*1*
Name	Text	*ERHAN*
Surname	Text	*ULA*
Adress	Text	*KAP*
PhoneNo	Text	*04144411023*
IrrigationAssociation	Long Integer	*2*

* One instance of a class is shown to give an example. In all tables, the attributes which are written in italic are loaded by fictitious data. Note that when an instance of a feature class is created without its geometry, its shape properties are assigned to zero automatically.

Table 3.8: Properties of Ownership Relationship Class

Type	Simple	
Cardinality	1 - M	
Notification	None	
Origin Table	**Name**	Farmer
	Primary Key	OBJECTID
	Foreign Key	FarmerID
Destination Feature Class	**Name**	Farms
Labels	**Forward**	Owns
	Backward	Is owned by

3.6.1.2 Irrigation Associations

The international literature on irrigation management has long used the terms Water Users' Association and Irrigators' Group in discussing irrigation management institutions. The term Water Users' Association (WUA) usually refers to local-level institutions that control the allocation, delivery and management of irrigation water resources. The key concept to these organizations is the active involvement of water users who come together for the purpose of organizing and practicing irrigation system operation and maintenance (Abu-zeid, 2001).

Since WUAs could serve as a collection organization and encourage farmers to participate in on-farm water management, a truly participatory irrigation management can be achieved. However, some problems have been realized for the development of WUAs such as lack of financial support, absence of legal framework and training requirements (Ünal, 2001).

In Turkey the term Irrigation Association (IA) is used to refer to the organizations, which have been formed for the purposes of managing irrigation units covering more than one village or municipality. The council of the IA is made up of village administrators (muhtars) and mayors of participating villages and municipalities plus additional members. DSI, the observer member of the IA identifies boundaries (Ünver and Gupta, 2002).

Irrigation Management Transfer is defined as the turning over of authority and responsibility to manage irrigation systems from government agencies to WUAs and has been performed in many countries (Svendsen and Nott, 1998). Throughout the transfer program, which has been started by DSI in 1950's and accelerated in 1993's, the operation and management of irrigation systems has been transferred to local government units or to Irrigation Associations (IAs). The World Bank played an important catalytic role in this acceleration ("Sulama Birlikleri", 2003). The transfer of an irrigation scheme to an Irrigation Association is appropriate for large areas where there are more than one local administrative units. The approval of Council Of Ministers is a legal requirement for establishment of Irrigation Association (Yazar, 2002). There are 18 IAs located in the Şanlıurfa-Harran Plains which were established as a result of efforts by DSI's 15th Regional Directorate covering Şanlıurfa province. These are Topçu, Yalınlı, GAP, Reha, Tahılalan, Sevimli, Haktanır, Koruklu, Merkez, 13. Yedek, İmambakır, Bereket, Kurtuluş, Fırat, Şuayb, Tektek, Cabirensar, and Kısas (Figure 3.22).

IAs are quasi-public organizations which get approval from local authorities for their budgets and maintain records that are subject to government inspections ("Şanlıurfa & Harran Plains On-Farm and Village Development Project: Inception Report", 1998).

Irrigation Associations are modeled in the geodatabase as a polygon feature class within Agriculture Feature Dataset. Capital and operation costs of IAs are stored as an object class outside the dataset. The data and their description related to these classes are shown in Table 3.9 and Table 3.10.

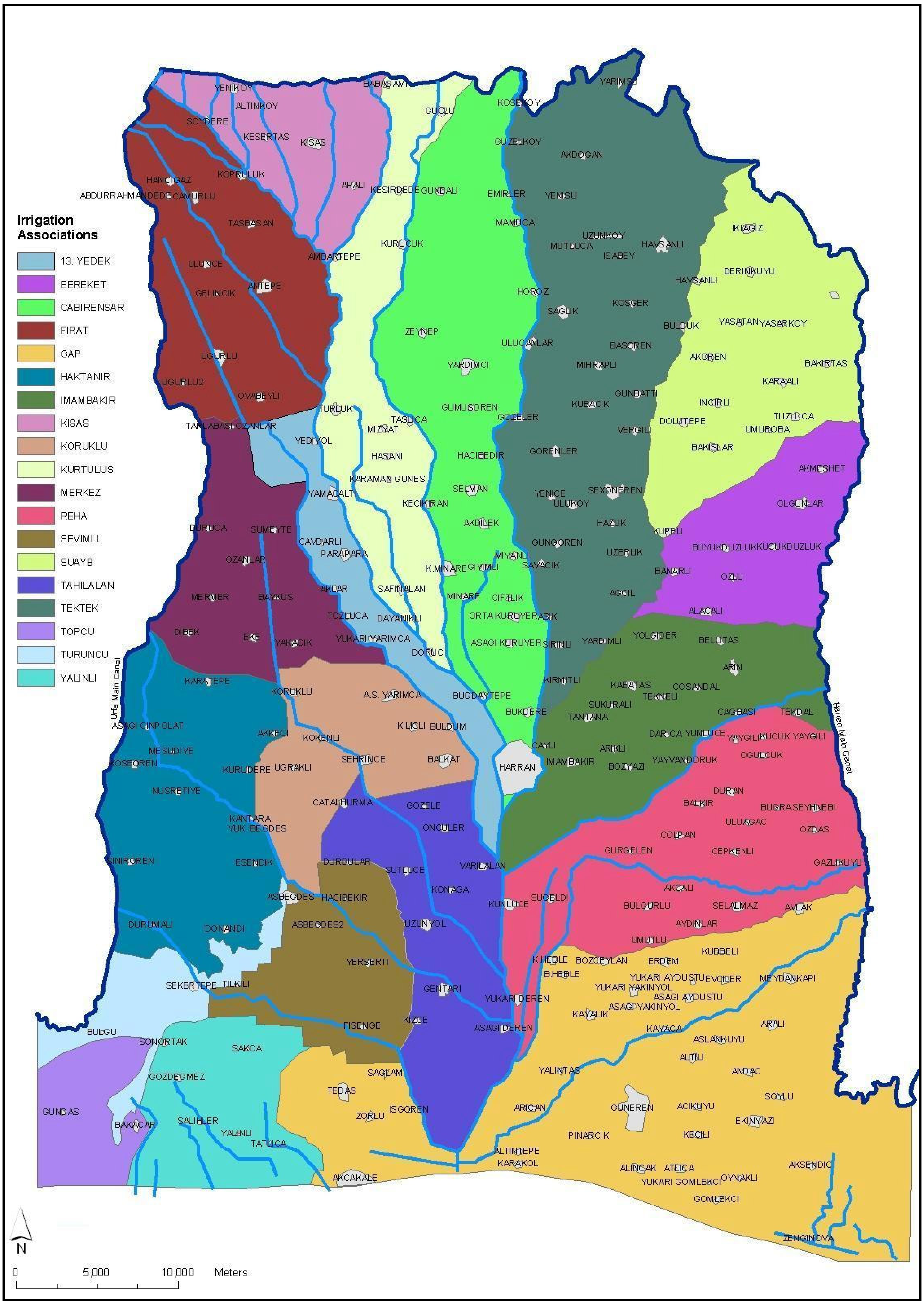


Figure 3.22: Irrigation Associations in Şanlıurfa Harran Plains

Table 3.9: Attributes of IrrigationAssociations Feature Class

Field Name	Description	Data Type	Instance
OBJECTID	Primary Key	Object ID	1
SHAPE	GeometryType	Geometry	Polygon
Name	Name of the Irrigation Association	Text	YALINLI
Estab_Date	Establishment Year of IA	Date	5/25/1995
DirectorName	Name of the IA's Director	Text	MUSTAFA TUR
SHAPE_Length	Length of the IA's Boundary	Double	30722.577
SHAPE_Area	Area of IA	Double	45681332.15
MainCanal	Name of the Main Canal from which irrigation water is supplied	Text	Urfa Main Canal

Table 3.10: Attributes of OperationCosts Object Class

Field Name	Description	Data Type	Instance
OBJECTID	Primary Key	Object ID	*1*
Name	Name of the Irrigation Association	Text	*YALINLI*
Personnel	Personnel Costs	Double	*1000*
Energy	Energy Costs	Double	*1200*
Mach_Equip	Machine and Equipment Costs	Double	*1400*
OtherOp	Other Operation Costs	Double	*1000*
Main_Rep	Maintenance and Repair Costs	Double	*1300*
Furn_Equip	Furniture and Equipment Costs	Double	*1500*
Vehicle	Vehicle Costs	Double	*600*
WorkMach	Work machine Costs	Double	*500*
Taxes	Taxes	Double	*600*
Other	Other Costs	Double	*150*
Total	Total Cost	Double	*8350*

A farmer is registered to the IA within which his farm is. This linkage is stored as a relationship class named as Registration. Properties of this relationship class are shown in Table 3.11.

Table 3.11: Properties of Registration Relationship Class

Type	Simple	
Cardinality	1 - M	
Notification	None	
Origin Feature Class	**Name**	IrrigationAssociations
	Primary Key	OBJECTID
	Foreign Key	IrrigationAssociation
Destination Table	**Name**	Farmer
Labels	**Forward**	Register
	Backward	Are registered to

OperationCosts object class is linked to IrrigationAssociations feature class via OperationCosts relationship class (Table 3.12).

Table 3.12: Properties of OperationCosts Relationship Class

Type	Simple	
Cardinality	1 - 1	
Notification	None	
Origin Feature Class	**Name**	IrrigationAssociations
	Primary Key	OBJECTID
	Foreign Key	OBJECTID
Destination Table	**Name**	Operation Costs
Labels	**Forward**	Have
	Backward	Are belong to

3.6.1.3 Water Charges

Increasing economic pressures on water resources, especially on irrigated agriculture, force countries to improve water pricing policies for the allocation of irrigation water (Johansson et al., 2002). Since large capital investments in infrastructure development are required for irrigation water, governments are often required to allocate water resources. Various mechanisms are used for this purpose by policymakers, some more efficient and some easier to implement than others (Tsur and Dinar, 1997; Dinar, 1998).

Getting prices right and allocating water efficiently will become increasingly important as demand for food and water increases and as water scarcity becomes more of a problem. There are a number of alternative charging mechanisms available in structuring operation and maintenance fees. As a principle, the fee structure should be equitable, administratively simple, and easily understood by both users of a particular service and the staff that will administer and collect the fees. Water pricing mechanisms are mainly categorized as volumetric (based on the quantity of water used) and flat rate (based on area irrigated or households benefited).

In volumetric pricing water is considered as a good delivered to the customer and water charge payments are linked to a measure of irrigation actually, rather than to an intended delivery schedule. While it is the best efficient way for controlling demand, the need for measurement at each delivery point makes the implementation of it most complicated and expensive (Böke, 1997). Volumetric pricing is conducive to creating incentives for efficient allocation and use, but the cost of establishing volumetric water delivery structures is often prohibitive, especially in large and spatially spread surface irrigation systems serving many smallholders. As a result, area based pricing is dominant in most irrigation systems (Johansson, 2000).

Area-based pricing of water is the most common and perhaps the simpler to operate. However, it does not reflect the quantity of water used nor does it provide an incentive to conserve water. Water in this approach is priced per hectare or other unit of area irrigated with minimal or no control at all on the amount of water supplied. Major advantages of this approach are its minimal requirements for monitoring and control of water supplies and its reliance on information that is easily accessible and verifiable (hectares of land cultivated and irrigated). Hence, in terms of cost effectiveness this approach has quite a number of advantages compared with other more sophisticated methods of price assessment.

Crop based water pricing charges water for each crop taking into consideration the crop water requirements. However this mechanism possess some of effects of volumetric pricing in terms of efficiency, it is more difficult to administer than the area based pricing (Abu-zeid, 2001).

In Turkey Irrigation Associations charge water based on area and crop based tariff method in gravity irrigation schemes, while studies for the transition to volumetric pricing have been carried out (Ünver and Gupta, 2002). Charges differ according to the type of crop grown and are fixed seasonally at each Irrigation Association independently. Crop irrigation requirement is considered for the calculation of the water charge, with higher charges for those crops which require higher applications.

Prior to the beginning of irrigation season, farmers fill out a demand form in which the information about the crop to be grown and area to be irrigated is given. The determination of water fees is made using the maps of irrigated farms in the scheme which displays the irrigator name and the crop for each farm (Yazar, 2002). For the collection of water fees, payment timetables are set by Irrigation Associations.

Within the geodatabase payment timetables and water charges are stored as object classes (Table 3.13; Table 3.14).

Table 3.13: Attributes of WaterCharges Object Class

Field Name	**Description**	**Data Type**	**Instance**
OBJECTID	Primary Key	Object ID	*1*
IrrigationAssociation	OBJECTID the Irrigation Association	Long Integer	*1*
CropType	Crop Type	Text	*Cotton*
Fee	Water Charge for the crop type in terms of TL /da	Float	*120*

Table 3.14: Attributes of PaymentTimetable Object Class

Field Name	Description	Data Type	Instance
OBJECTID	Primary Key	Object ID	*1*
FarmerID	OBJECTID of Farmer	Long Integer	*1*
PaymentDate	Date of the payment for the farmer	Date	*2/1/2003*
PaymentAmount	Payment amount for the indicated date	Float	*1000*

The records only for farmers who use irrigation water are maintained in the PaymentTimetable object class and the linkage between these classes is ensured by PaymentTimetable Relationship Class (Table 3.15).

Table 3.15: Properties of PaymentTimetable Relationship Class

Type	Simple	
Cardinality	1 - M	
Notification	None	
Origin Table	**Name**	Farmer
	Primary Key	OBJECTID
	Foreign Key	FarmerID
Destination Table	**Name**	PaymentTimetable
Labels	**Forward**	Signs
	Backward	Is arranged for

WaterCharges object class is related to IrrigationAssociations feature class as shown in Table 3.16.

Table 3.16: Properties of WaterCharges Relationship Class

Type	Simple	
Cardinality	1 - M	
Notification	None	
Origin Table/Feature Class	**Name**	IrrigationAssociations
	Primary Key	OBJECTID
	Foreign Key	IrrigationAssociation
Destination Table/Feature Class	**Name**	WaterCharges
Labels	**Forward**	Sets
	Backward	Are set by

3.6.2 Sociological

Irrigation projects do not only affect economic outcomes, but may have wider sociologic effects. Two very visible effects of irrigation projects, particularly large dams and canal systems are the population growth and displacement and negative health effects associated with increases in infectious diseases. Also educational infrastructure as one of the social indicators plays a crucial role in monitoring the impacts of irrigation. Within the geodatabase spatial entities and relationships related to sociological aspects of the irrigation project are organized into Sociological Feature Dataset (Table 3.17).

Table 3.17: Geodatabase Elements related to Sociological Aspects

Name	Type
Sociological Feature Dataset	
Settlements	**Polygon Feature Class**
HealthInstitutions	**Point Feature Class**
Schools	**Point Feature Class**
InfectiousDiseases	**Relationship Class**
InfectiousDiseases	**Object Class**

3.6.2.1 Infectious diseases

The risk of many infectious diseases is influenced by human alteration of local, regional or global ecosystems. Introduction of new irrigation schemes into previously dry areas and consequent agricultural production dramatically changes the ecosystem. Human activities, such as irrigation, extensive agriculture or building settlements, affect the ecological conditions in which disease-causing microbes thrive. Accompanying these environmental changes is the very real threat of infectious diseases.

However, irrigation, particularly involving canals, reservoirs and tanks, has a downside in terms of health as it encourages waterborne diseases due to inadequate drainage and renders the microenvironment hospitable to anopheles mosquitoes and snails that spread malaria and schistosomiasis. Untreated contaminated water is also responsible for causing serious diseases, from diarrhoea (one of the main proximate causes of child mortality) to cholera. It is likely that the poor are more vulnerable to such water-borne diseases: they are likely to be more exposed to sources through their work and in their homes (e.g. living beside rivers and canals, or on rivers), they are less likely to be able to prevent infection by properly sterilizing water and water utensils, plus they are less likely to have access to prompt, appropriate medical treatment when they are infected, because they live in remote areas and/or they cannot afford the medical fees.

The cost of combating diseases can be very large, whereas the costs of prevention are much lower. Large national projects that anticipate economic benefits may sometimes overlook the distant prospects of disease. For GAP to reach its targeted and sustainable economic aims, projects in various other sectors also need to be considered and integrated. In this context, the public health consequences of emerging diseases in this setting must be anticipated so that appropriate health education and disease prevention measures can be implemented. The establishment of good surveillance and recording systems is an important first step. So new ways can be developed to monitor, assess, predict and respond to emerging infectious disease threats.

InfectiousDiseases object class is created for the purpose of recording the cases of infectious diseases observed in the region (Table 3.18). It is related to HealthInstitutions object class as each occurrence is reported at a health institution (Table 3.20). InfectiousDiseases domain (Table 3.19) is applied to the field DiseaseName to facilitate the name entry process. Infectious diseases used in InfectiousDiseases domain are taken from the Statistical Yearbook of Turkey which is published by State Institute of Statistics (DİE).

Table 3.18: Attributes of InfectiousDiseases Object Class

Field Name	**Description**	**Data Type**	**Domain**	**Instance**
OBJECTID	Primary Key	Object ID		*1*
HealthIns_ID	OBJECTID of Health Institution	Long Integer		*1*
DiseaseName	Name of the Infectious Disease	Long Integer	InfectiousDiseases	*Encephalitis*
Date_Case	Date of the Case	Date		*5/15/2003*

Table 3.19: InfectiousDiseases Domain

<table>
<tr><th>Data Type</th><th>Domain Type</th></tr>
<tr><td>Long Integer</td><td>Coded Value Domain
<table>
<tr><td>Code</td><td>Description</td><td>Code</td><td>Description</td></tr>
<tr><td>1</td><td>Encephalitis</td><td>9</td><td>Measles</td></tr>
<tr><td>2</td><td>Whooping Cough</td><td>10</td><td>Scarlet Fever</td></tr>
<tr><td>3</td><td>Leprosy</td><td>11</td><td>Rabies</td></tr>
<tr><td>4</td><td>Diphtheria</td><td>12</td><td>Brucellosis</td></tr>
<tr><td>5</td><td>Dysentery</td><td>13</td><td>Paratyphoid</td></tr>
<tr><td>6</td><td>Puerperal Fever</td><td>14</td><td>Poliomyelitis</td></tr>
<tr><td>7</td><td>Infectious Hepatitis</td><td>15</td><td>Anthrax</td></tr>
<tr><td>8</td><td>Epidemic Meningitis</td><td>16</td><td>Typhoid Fever</td></tr>
</table></td></tr>
</table>

Table 3.20: Properties of InfectiousDiseases Relationship Class

Type	Simple	
Cardinality	1 - M	
Notification	None	
Origin Feature Class	**Name**	HealthInstitutions
	Primary Key	OBJECTID
	Foreign Key	HealthIns_ID
Destination Table	**Name**	InfectiousDiseases
Labels	**Forward**	Report
	Backward	Are reported at

3.6.2.2 Health institutions

Health institutions provide medical care and other health services. Access to health services, especially in rural areas, is accepted as an important social indicator. (Ünalan, 2003) Health institutions can be grouped as in-patient and out-patient. In-patient medical institutions cover hospitals and health centers, while out-patient institutions health units, health houses, and dispensaries. These institutions are modeled within the geodatabase as point feature class named as HealthInstitutions (Table 3.21). The subtypes associated with this class are shown in Table 3.22. The field InstitutionName possess two different domains according to the subtypes, which are illustrated in Table 3.23 and Table 3.24. Attributes of HealthInstitutions feature class are based on the health statistics of DİE

Table 3.21: Attributes of HealthInstitutions Feature Class

Field Name	Description	Data Type	Instance
OBJECTID	Primary Key	Object ID	1
SHAPE	GeometryType	Geometry	Point
Name	Name of the Health Institutiton	Text	Harran
Num_HPersonnel	Number of Health Personnel	Long Integer	25
Num_Bed	Number of Beds	Long Integer	30
InstitutionType	Whether in-patient or out-patient	Long Integer	In Paitent Health Institution
HealthIns_Type	Health Institution Type	Long Integer	Hospital

Table 3.22: Subtyping Structure of HealthInstitutions Feature Class

Subtype Field	InstitutionType
Default Subtype	Out Patient Health Institution

Subtypes

Code	Description
1	In Patient Health Institution
2	Out Patient Health Institution

Default Values and Domains

Subtype Code	Field Name	Domain
1	HealthIns_Type	In Patient
2	HealthIns_Type	Out Patient

Table 3.23: InPatientHealthInstitutions Domain

Data Type	Domain Type	Code	Description
Long Integer	Coded Value Domain	1	Hospital
		2	Health Center

Table 3.24: OutPatientHealthInstitutions Domain

Data Type	Domain Type	Code	Description
Long Integer	Coded Value Domain	1	Health unit
		2	Health house
		3	Dispensary

3.6.2.3 Education

Education is a prerequisite to building a food-secure world, reducing poverty and conserving and enhancing natural resources. Rural schools traditionally have played a central role in their communities. Besides providing for basic education, they often have served as a cultural center in the community. Thus expanding access to education and improving school attendance in rural areas and also recording these activities will improve the quality of educational services.

Schools are stored in the geodatabase as a point feature class whose attributes are displayed in Table 3.25 and subtypes in Table 3.26. While designing the attributes of schools feature class education statistics of DİE and Ministry of National Education are taken into consideration. Realizing the World Development Indicators of WorldBank in which school enrollment rates of female and male students are also included (http://wbln0018.worldbank.org/dg/povertys.nsf/DownloadSID?openform), related attributes are added to the feature class. As Ünalan, (2003) stated this indicator is a measure of equality in opportunity for boys and girls to benefit from educational services.

Table 3.25: Attributes of Schools Feature Class

Field Name	Description	Data Type	Instance
OBJECTID	Primary Key	Object ID	*1*
SHAPE	GeometryType	Geometry	*Point*
Name	Name of the School	Text	*Cumhuriyet*
SchoolType	School Type	Long Integer	*High School*
Num_of_Teachers	Number of teachers	Long Integer	*12*
Num_of_Male	Number of Male Students	Long Integer	*80*
Num_of_Female	Number of Female Students	Long Integer	*40*

Table 3.26: Subtyping Structure of Schools Feature Class

Subtype Field	SchoolType
Default Subtype	Primary School
Subtypes	

Code	Description
1	Primary School
2	Junior School
3	High School

3.6.2.4 Settlements

The settlements located in the Şanlıurfa-Harran Plains are represented as polygon features. The majority of the population in the region is characterized as rural among which the standard of living is far below that of urban areas. Therefore to monitor the rural development, the information about the existing state of the village infrastructure gains more importance. Attributes of settlements are given in Table 3.27. These attributes are taken from Şanlıurfa Harran PlainOn-Farm and Village Development Project.

Table 3.27: Attributes of Settlements Feature Class

Field Name	**Data Type**	**Domain**	**Instance**
OBJECTID	Object ID		1
SHAPE	GeometryType		Polygon
Code	Text		AKCAKALE_MERKEZ_009
Name	Text		ARICAN
SecondName	Text		ZUNEYBIR
Settle_Type	Long Integer	SettlementType	Village
Households	Long Integer		40
Population	Long Integer		285
Road	Long Integer	VillageRoad	Asphalt
DrinkingWater	Long Integer	DrinkingWater	*Satisfactory*
Consolidation	Long Integer	Consolidation	*Completed*

The domains that the fields Settle_Type, Road, DrinkingWater, and Consolidation possess are described in Tables 3.28, Table 3.29, Table 3.30, and Table 3.31.

Table 3.28: SettlementType Domain

Data Type	**Domain Type**		
Long Integer	Coded Value Domain	Code	Description
		1	Provincial Center
		2	District Center
		3	Sub district Center
		4	Village
		5	Unit

Table 3.29: Road Domain

Data Type	**Domain Type**		
Long Integer	Coded Value Domain	Code	Description
		1	Asphalt
		2	Stabilized
		3	Graded

People living in rural areas in developing countries, sometimes lack access to safe drinking water. Accessible domestic water supplies, even if shared by a large number of households, can make more and better-quality water available for family needs, reduce female drudgery and reduce the incidence of debilitating water-borne diseases. In fact, studies have found that water supplies are often at the top of the list of what the rural poor ask for themselves.

Table 3.30: DrinkingWater Domain

Data Type	**Domain Type**		
Long Integer	Coded Value Domain	Code	Description
		1	Satisfactory
		2	Unsatisfactory
		3	Groups

The impact of land fragmentation on economic growth and social stability in rural areas should not be underestimated. Land consolidation is the main measure, applied so far in Şanlıurfa, in an effort to eliminate land fragmentation because of heritage, sales or irrigation canals. In 1986, Şanlıurfa was declared as the "Reform Region". Land consolidation improves the prevailing defective land tenure structure which is primarily characterized by: a small holding size, intense land fragmentation, mixed land tenures (i.e. land held in undivided form and dual or multiple ownerships), lack of farm roads and irregularly-shaped plots. All these features constitute a major structural and technical obstacle to the rational development of agriculture. Land consolidation measures include also the construction of a proper farm road network for every scheme, which provides access to all consolidated plots.

Table 3.31: Consolidation Domain

Data Type	**Domain Type**		
Long Integer	Coded Value Domain	Code	Description
		1	Completed
		2	Not Completed
		3	Partially Completed

3.6.3 Irrigation and Drainage System

An irrigation distribution system is a network of branching conduits, which may be either of free flow or pressurized type. Several arrangements of irrigation network have been implemented in the world. In one of these, which have received the most widespread application, the network consists of three classes of irrigation canals.

- Main Canal receives water from transmission conduit and feed secondary canals. Normally, no direct irrigation is permitted from the main canal.
- Secondary Canals take off water from main canals and supply water to tertiary canals. They generally cross the irrigation area.
- Tertiary Canals take off water from secondary canals and supply water to outlets and turnouts. These are the canals from which farmers receive their water.

A drainage system is necessary to remove excess water from the irrigated land. This excess water may be e.g. waste water from irrigation or surface runoff from rainfall. It may also include leakage or seepage water from the distribution system. Drainage serves to control the salinity of the soil and prevents the water table to unacceptable levels.

The establishment of the irrigation and drainage network and mapping it into GIS environment are important steps towards transforming high-risk agriculture into highly productive, diversified and sustainable irrigated agriculture.

Irrigation Drainage System Feature Dataset is created for the purpose of organizing topologically related irrigation and drainage network features. Table 3.32 displays the contents of the feature dataset.

To identify the attributes of the objects inside the Irrigation Drainage System feature dataset, the map sheets related to irrigation and drainage canals are obtained from DSİ.

Table 3.32: Contents of the Irrigation Drainage System Feature Dataset

Name	Type
IrrigationCanals	**Line Feature Class**
DrainageCanals	**Line Feature Class**
Canalet	**Line Feature Class**
HydraulicStructures	**Point Feature Class**
DrainageReaches	**Line Feature Class**
IrrigationReaches	**Line Feature Class**
DReaches_Composition	**Relationship Class**
IrrReaches_Composition	**Relationship Class**
Canalet_Composition	**Relationship Class**

Irrigation Canals are stored as line feature class within the geodatabase. Properties of the feature class are given in Table 3.33 and the inherent subtyping structure is given in Table 3.34. DrainageCanals feature class reveals similarities with IrrigationCanals feature class (Table 3.35 and Table 3.36). The field irrigation gives information about whether the canal is part of the Harran Irrigation or Urfa Irrigation.

Table 3.33: Attributes of IrrigationCanals Feature Class

Field Name	**Data Type**	**Instance**
OBJECTID	Object ID	1
SHAPE	Geometry	Polyline
Code	Text	UY-5-1
CanalType	Long Integer	Tertiary Irrigation Canal
Irrigation	Text	Urfa
SHAPE_Length	Double	2179.60 m

Table 3.34: Subtyping Structure of IrrigationCanals Feature Class

Subtype Field	CanalType
Default Subtype	
Subtypes	

Code	Description
1	Main Irrigation Canal
2	Secondary Irrigation Canal
3	Tertiary Irrigation Canal

Table 3.35: Attributes of DrainageCanals Feature Class

Field Name	Data Type	Instance
OBJECTID	Object ID	1
SHAPE	Geometry	Polyline
Code	Text	HT-1
CanalType	Long Integer	Main Drainage Canal
Irrigation	Text	Harran
SHAPE_Length	Double	1448.16 m

Table 3.36: Subtyping Structure of DrainageCanals Feature Class

Subtype Field	CanalType
Default Subtype	Tertiary Drainage Canal
Subtypes	

Code	Description
1	Main Drainage Canal
2	Secondary Drainage Canal
3	Tertiary Drainage Canal

Canals are composed of reaches with different hydraulic properties. Irrigation canals stored in the geodatabase are either canalets or earthen canals, which are implemented as Canalet and IrrigationReaches. Attributes of drainage and irrigation reaches and Canalet are shown in Tables 3.37, 3.38 and 3.39. The composition relationships between the canals and the reaches are implemented as a composite relationship within the geodatabase (Table 3.40, Table 3.41). The relationship between IrrigationCanal and Canalet is also implemented as composite relationship (Table 3.42).

Table 3.37: Attributes of Canalet Feature Class

Field Name	Data Type	Instance
OBJECTID	Object ID	1
SHAPE	Geometry	Polyline
Code	Text	UY-5-1
CanaletType	Text	*1*
Discharge	Float	3.00 lt/s
Slope	Float	0.0054
Velocity	Float	1.58 m^2/s
ServiceArea	Float	*100*
SHAPE_Length	Double	*0*

Table 3.38: Attributes of DrainageReaches Feature Class

Field Name	Data Type	Instance
OBJECTID	Object ID	1
SHAPE	Geometry	Polyline
Code	Text	UTB
ReachNo	Text	7
BottomWidth	Float	1.20 m
WaterDepth	Float	0.79 m
ManningCoefficient	Float	0.028
Discharge	Float	3.00 lt/s
SideSlope	Double	0.666666667
Slope	Float	0.0054
Velocity	Float	1.58 m^2/s
SHAPE_Length	Double	*0*

Table 3.39: Attributes of IrrigationReaches Feature Class

Field Name	Data Type	Instance
OBJECTID	Object ID	1
SHAPE	Geometry	Polyline
Code	Text	UY-4-42
ReachNo	Text	1
BottomWidth	Float	1.20 m
WaterDepth	Float	0.79 m
ManningCoefficient	Float	0.028
Discharge	Float	3.00 lt/s
SideSlope	Double	0.666666667
Slope	Float	0.0054
Velocity	Float	1.58 m^2/s
ServiceArea	Float	*100*
SHAPE_Length	Double	*0*

Table 3.40: Properties of DReaches_Composition Relationship Class

Type	Complex	
Cardinality	1 - M	
Notification	Origin to Destination	
Origin Feature Class	**Name**	DrainageCanals
	Primary Key	Code
	Foreign Key	Code
Destination Feature Class	**Name**	DrainageReaches
Labels	**Forward**	Are composed of
	Backward	Constitute

Table 3.41: Properties of IrrReaches_Composition Relationship Class

Type	Complex	
Cardinality	1 - M	
Notification	Origin to Destination	
Origin Feature Class	**Name**	IrrigationCanals
	Primary Key	Code
	Foreign Key	Code
Destination Feature Class	**Name**	IrrigationReaches
Labels	**Forward**	Are composed of
	Backward	Constitute

There are hydraulic structures located on Canals for the purpose of regulation, measurement and transportation. These structures are modeled as point feature class (Table 3.43). The field Type possesses a domain named as HydraulicStructures that stores possible values of type of the structure (Table 3.44).

Table 3.42: Properties of Canalet_Composition Relationship Class

<table>
<tr><td>Type</td><td colspan="2">Complex</td></tr>
<tr><td>Cardinality</td><td colspan="2">1 - M</td></tr>
<tr><td>Notification</td><td colspan="2">Origin to Destination</td></tr>
<tr><td rowspan="3">Origin Feature Class</td><td>Name</td><td>IrrigationCanals</td></tr>
<tr><td>Primary Key</td><td>Code</td></tr>
<tr><td>Foreign Key</td><td>Code</td></tr>
<tr><td>Destination Feature Class</td><td>Name</td><td>Canalet</td></tr>
<tr><td rowspan="2">Labels</td><td>Forward</td><td>Are composed of</td></tr>
<tr><td>Backward</td><td>Constitute</td></tr>
</table>

Table 3.43: Attributes of HydraulicStructures Feature Class

Field Name	Data Type	Instance
OBJECTID	Object ID	9
SHAPE	Geometry	Point
Code	Text	SIFON 40+485
Type	Long Integer	Siphon
OnCanal	Text	Harran Main Canal
UpstreamElevation	Float	79.239998 m
DownstreamElevation	Float	80.040001 m

Table 3.44: HydraulicStructures Domain

Data Type	Domain Type
Long Integer	Coded Value Domain

Code	Description	Code	Description
1	Pump	6	Turnout
2	Power Plant	7	Chute
3	Check	8	Culvert aqueduct
4	Diversion Weir	9	Bridge
5	Siphon	10	Culvert
		99	Unknown

Check: A structure built across a canal at suitable points to control water levels and regulate and escape supplies.

Turnout: A structure built at the end of an offtaking branch or distributary canal to divert water from a canal or pipeline.

Diversion Weir: A structure built across a canal to measure the rate of flow of water.

Chute: A structure to convey water rapidly from a high elevation to a lower one. Chutes are similar to drops except they carry water over a greater distance.

Culvert: A transverse drain or waterway under a road, railroad, canal, or other obstruction.

Culvert aqueduct: A structure that carries drainage water under the canal section.

Bridge: A structure crested over a depression or an obstacle.

Pump: A device that moves, compresses, or alters the pressure of a fluid, such as water being conveyed through a natural or artificial canal.

Şanlıurfa Harran irrigation and drainage system is demonstrated in Figure 3.23.

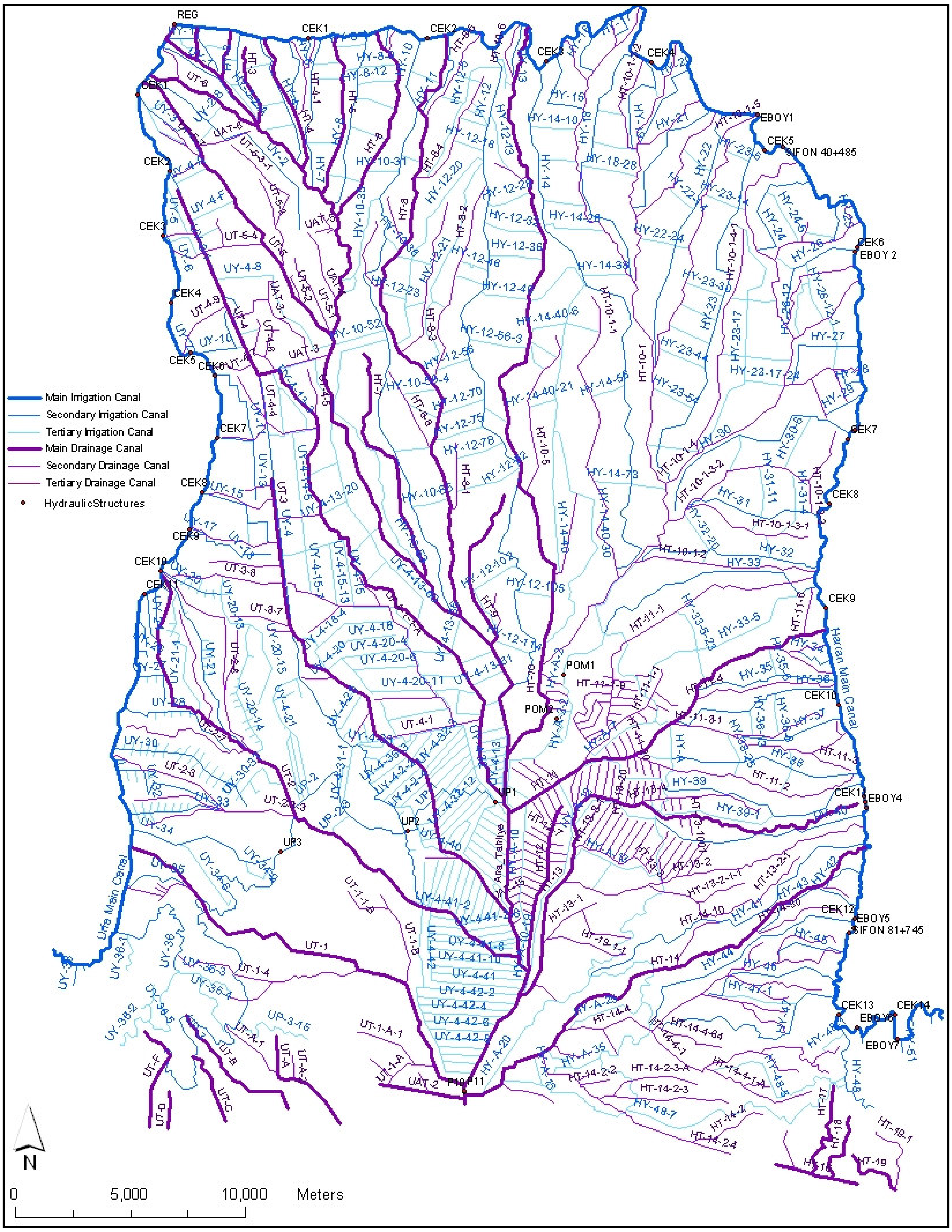


Figure 3.23: Irrigation and Drainage System

CHAPTER 4

ŞANLIURFA HARRAN PLAINS GEODATABASE

4.1 Personal and Multi-User Geodatabases

One task that database administrators must handle is to choose the correct DBMS to use for each new application being developed. There are many aspects to selecting the proper type of DBMS. The supporting architecture for the DBMS environment is critical to the success of the database applications being built. There are mainly two types of DBMSs, personal and multi-user.

A personal database is designed for use by a single person on a single computer. Such a database usually has a rather simple structure and a relatively small size. There is no active database server component that is separate from the application component. Trouble begins to occur when developers begin to use personal databases for multi-user OLTP (Online Transaction Processing) applications. Lotus Approach, Microsoft Access and dBase are examples of personal database software. A multiuser database supports multiple concurrent users.

There are also two types of geodatabases: personal and multi-user. A personal geodatabase has the .mdb file extension and can be read by multiple people at the same time, but edited by only one person at a time. A personal geodatabase has a maximum size of 2 gigabytes (GB) and, currently, can store only vector data.

Multi-user geodatabases are suitable for large workgroups and enterprise GIS implementations and can be read and edited by multiple users simultaneously. They require ArcSDE (Spatial database engine) and a DBMS (such as IBM DB2, Informix, MS SQL Server, or Oracle). ArcSDE provides the interface that allows you to store and manage spatial data in a DBMS. Multi-user geodatabases can store both vector and raster data. A comparison of personal and multi-user geodatabases can be found in Table 4.1.

Table 4.1: Comparison of personal and multi-user geodatabases

	Personal Geodatabase	Multi-user Geodatabase
DBMS	MS Access	IBM DB2, Informix, MS SQL Server, or Oracle
Client/Server	No	Yes
Long Transactions	No	Yes
Editors	Only 1	1 or more
C or Java API	No	Yes
Raster	No	Yes
Size	Up to 2GB	Unlimited

4.1.1 ArcSDE

ArcSDE is a system that stores spatial data in a database management system. ArcSDE manages spatial data using IBM DB2, IBM Informix, Microsoft SQL Server, or Oracle, and allows one to serve spatial data to remote clients. ArcSDE delivers spatial data to ArcGIS Desktop users through a direct, read-only connection from the client's machine to ArcSDE. ArcSDE enhances the RDBMS and SQL by interpreting geometric data, which it stores in an un-formatted, binary column on an RDBMS table. With ArcSDE the GIS software (ArcInfo, ArcView GIS, ArcIMS, and others) can work directly with spatial data managed in Oracle Spatial. ArcSDE provides the gateway between ESRI's GIS software and Oracle Spatial, filling three roles:

- As an integral part of a multi-user ArcGIS system;
- As an application server to deliver spatial data to many users and applications;
- As a developer tool for open access using either ArcObjects or its own Java or C application programming interface (API).

4.1.2 Application Server

ArcSDE is an application server, which is built with client/server technology. Application servers are configured in an n-tier environment, in which they are located in the middle and actually run parts of applications that many users share, and communicate between the desktop and back-end systems. ArcSDE interprets spatial data for RDBMS. It translates client requests for data into SQL statements that the RDBMS can use to read/write geometric data to tables in the database.

In a typical configuration, an ArcSDE application server resides with the relational database on a server platform. Clients communicate with the ArcSDE server over a TCP/IP network. The ArcSDE application server performs spatial searches and sends data that meet the search criteria to the client. For example, a common query handled by the ArcSDE application server is to retrieve all features in a particular map extent to be drawn in the display window. ArcSDE sends data to the client using “data buffering”. Processing and buffering data on the server are much more efficient than sending all of the data across the network and having the client determine which data to send to the end-user application. This becomes critical when applications are simultaneously using thousands of records in the database.

4.1.3 ArcSDE Services

Conveyance of spatial data between ArcGIS and database is accomplished through ArcSDE service. ArcSDE runs as a background service on Windows. Any process that runs in the background on Windows must run as a Windows service. Information about Windows services is stored in the Windows registry, a key component of a Windows operating system. The Windows registry keeps track of everything from computer's hardware to the services that should automatically start on boot.

An ArcSDE instance is a single iomgr (I/O manager) connected to a single DBMS database. It is possible to start multiple iomgrs and connect each of them to a separate database. Each instance exists within its own memory space and executable environment and connects to its own database. ArcSDE instances may share the same executable programs, but require their own home directories.

Starting an ArcSDE instance spawns the giomgr background process which establishes a single connection to the DBMS database. The giomgr process always connects to the DBMS database through the ‘sde’ user. After the giomgr process establishes the connection, it attempts to lock the ‘sde’ user’s VERSION table. If the giomgr process is unable to lock the table, it disconnects and issues an error stating that another giomgr process has already connected to this database. Thus a one-to-one relationship between the ArcSDE instance and the DBMS database must exist. Instances have unique names and home directories. Multiple ArcSDE instances may run on the same node at the same time, but they must always connect to a separate database.
When an ArcSDE instance is started, a giomgr background process establishes a connection to the DBMS database. Once the connection is established the giomgr listens on the port number assigned to the instance for ArcSDE client connection requests. When a connection request is received, the giomgr verifies the user and

password and spawns a gsrvr process and assigns it a new port number to communicate on. Figure 4.1 demonstrates the above process.

The services.sde file contains the unique instance name and the network port number and protocol on which the giomgr accepts connection requests to the instance. By default the services.sde file instance name is esri_sde with a 5151 port number using a TCP/IP protocol.

```
#
# ESRI SDE Remote Protocol
#
esri_sde        5151/tcp
```

The system services file must contain a service name that matches the instance name in the services.sde file in the etc subdirectory of the ArcSDE home directory. When the connection is started, the system services file is searched for a service name that matches the instance name in the services.sde file. If a match is not found, an error is returned.

The system services file contains many services names at least one of which must match your sde instance name. The service name must be entered into the services file on both the ArcSDE server and the client machines.

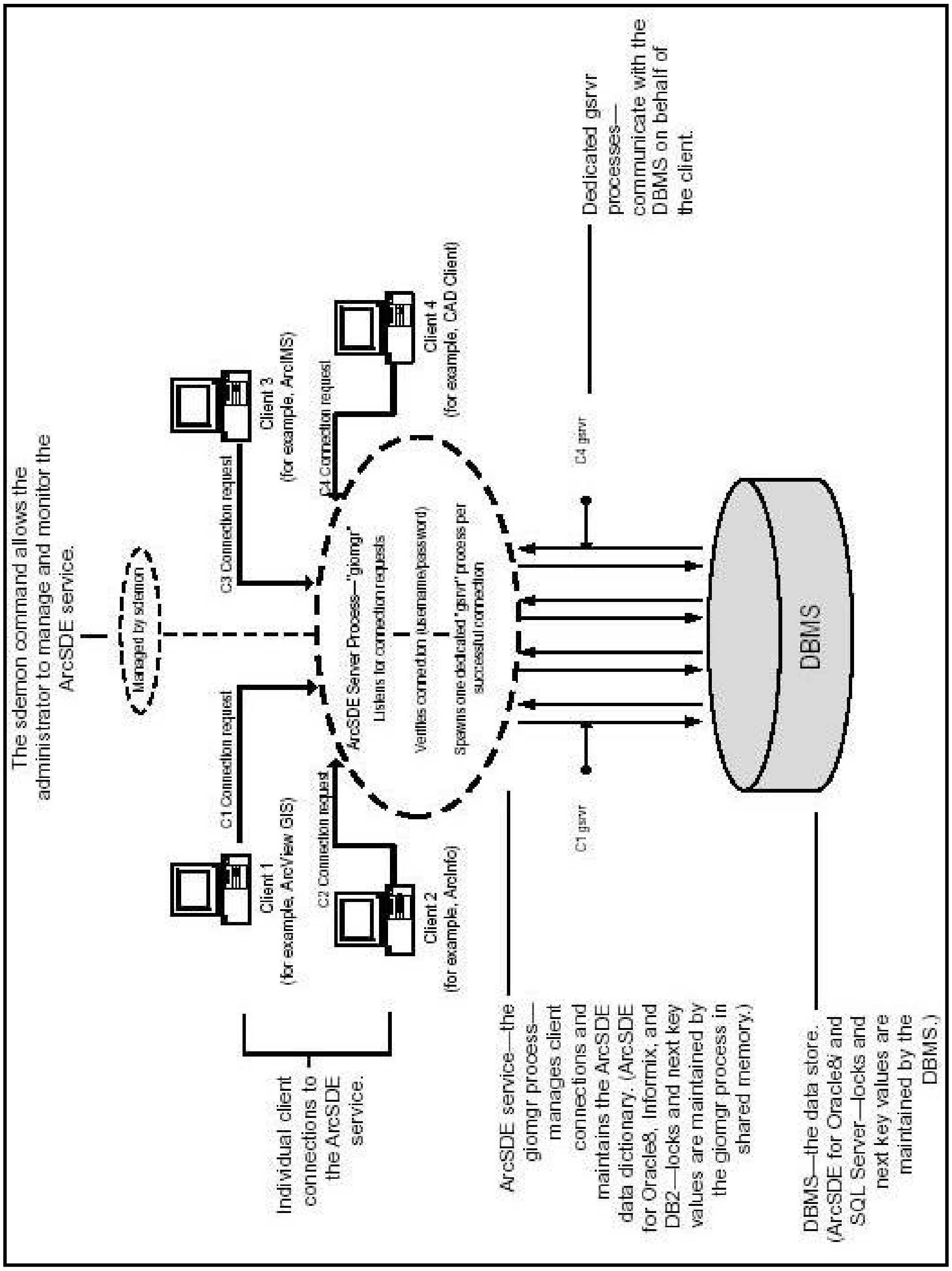


Figure 4.1: ArcSDE Service (ESRI Redlands, 2001)

4.1.4 Connecting to ArcSDE

Connection to ArcSDE is established from ArcCatalog. The SDE Connections dialog box appears after a double-click on the Database Connections followed by a double-click on Add Spatial Database Connection. While completing the form using esri_sde for the service entry is preferred over 5151. The Spatial Database Connection Panel is illustrated in Figure 5.2. The connection is tested using Test Connection. Note that a successful test connection gives no real indication short of a lack of error messages.

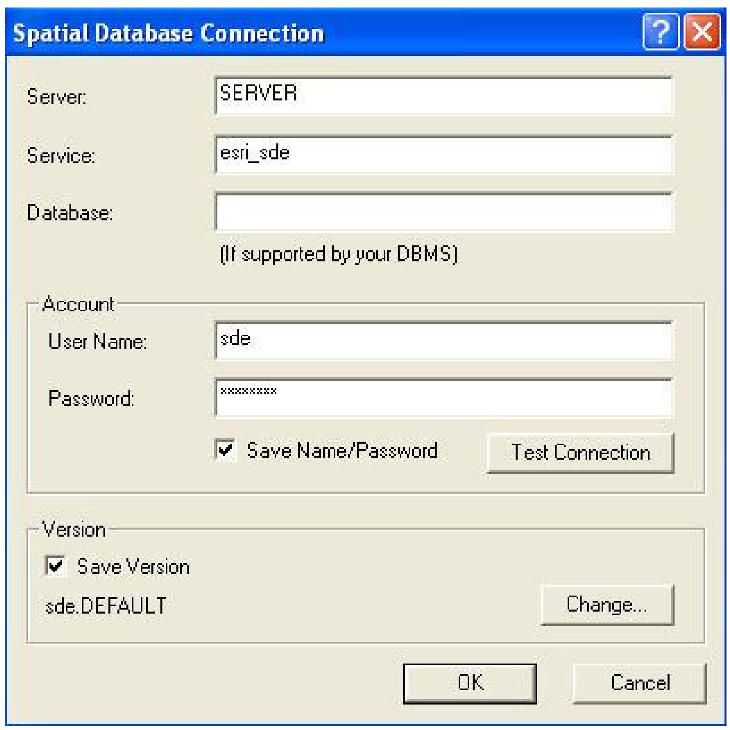


Figure 4.2: Spatial Database Connection Panel

Here the account is the ArcSDE system administrator account, which has a user name of sde. In oracle this user owns a single schema with the same name. In the terminology of Oracle a schema is defined as a logical collection of objects. Tables schema object stores all the geodatabase system tables (Figure 4.3). Some problems were encountered when setting up an SDE connection to the ArcSDE 8.3 server from ArcGIS Desktop on the client machine. The following error messages occurred when the Test Connection option was used.

Error message: Failed to connect to the specified server. Entry for SDE instance not found in the services file.

As indicated earlier the system services file must contain a service name that matches the instance name in the services.sde file. First the services file under C:\WINNT\system32\drivers\etc on the server and then the services file under C:\WINDOWS\system32\drivers\etc is checked. It is observed that the service name on the client machine is missing and is entered manually. So the problem is solved.

Error message: This release of the Geodatabase is either invalid or out of date. Please run the ArcSDE setup utility using the -o upgrade option.

ArcGIS Desktop applications are not forward compatible with major new releases of the ArcSDE server product and the Geodatabase. ArcGIS can only see ArcSDE servers that have the GeoDatabase tables created and that have a compatible version of the GeoDatabase. In the GIS Lab ArcSDE 8.3 is installed on the server, but the client machine has the lower version of ArcGIS. Upgrading the ArcGIS on client machine to 8.3 solves the problem.

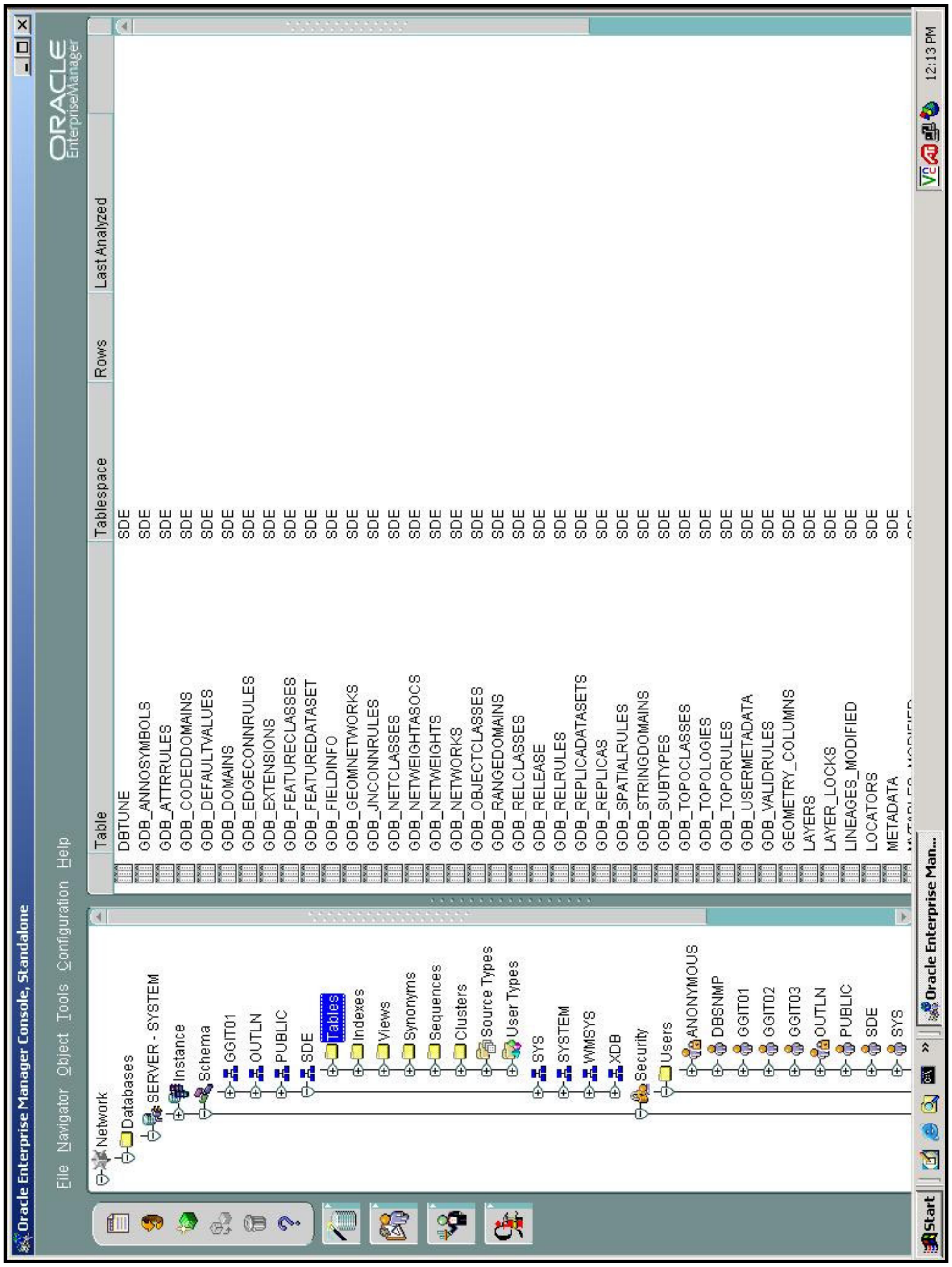


Figure 4.3: ArcSDE Geodatabase System Tables

4.1.5 Converting the Personal Geodatabase to ArcSDE Geodatabase

The Şanlıurfa Harran Plains Data Model created in UML can be exported to Microsoft Repository from which CASE Tools subsystem of ArcInfo converts the model into a personal geodatabase. Altough personal geodatabase is not very meaningful at enterprise wide, for this case because of lack of data the size of the database is not so high and therefore personal geodatabase can be applicable. But it is more convenient to use ArcSDE geodatabase in this study since the data related to Şanlıurfa Harran Plains should be shared and edited by many users and the volume of the data would probably be high.

Since a personal geodatabase in Access is created first, the ways to convert the personal geodatabase to ArcSDE geodatabase in Oracle is investigated. After establishing a connection converting the personal geodatabase to ArcSDE geodatabase is a simple process. Through ArcCatalog the contents of the geodatabase is copied to ArcSDE geodatabase so that an Access database is automatically exported to Oracle database.

4.2 Querying the Data

Naturally, querying the data can be accomplished in both Oracle and ArcMap. In Oracle the queries are created by using the SQL*Plus Worksheet, while in ArcMap the queries are created using the Query Builder from the Selection menu. The main differences are:

- The SQL*Plus Worksheet allows for a whole range of SQL and PL/SQL queries that are not possible to be built in ArcMap, but the results are in text format only.
- The ArcMap Selection menu allows for simplified queries, and the results are visually displayed in the ArcMap panel.

Four simple scenarios are considered in order to query the data.

Scenario 1: Upon a siphon request from farmer, the irrigation canal on which the siphon will be installed has to be determined.

Step 1: Select the farmer who requests a siphon.

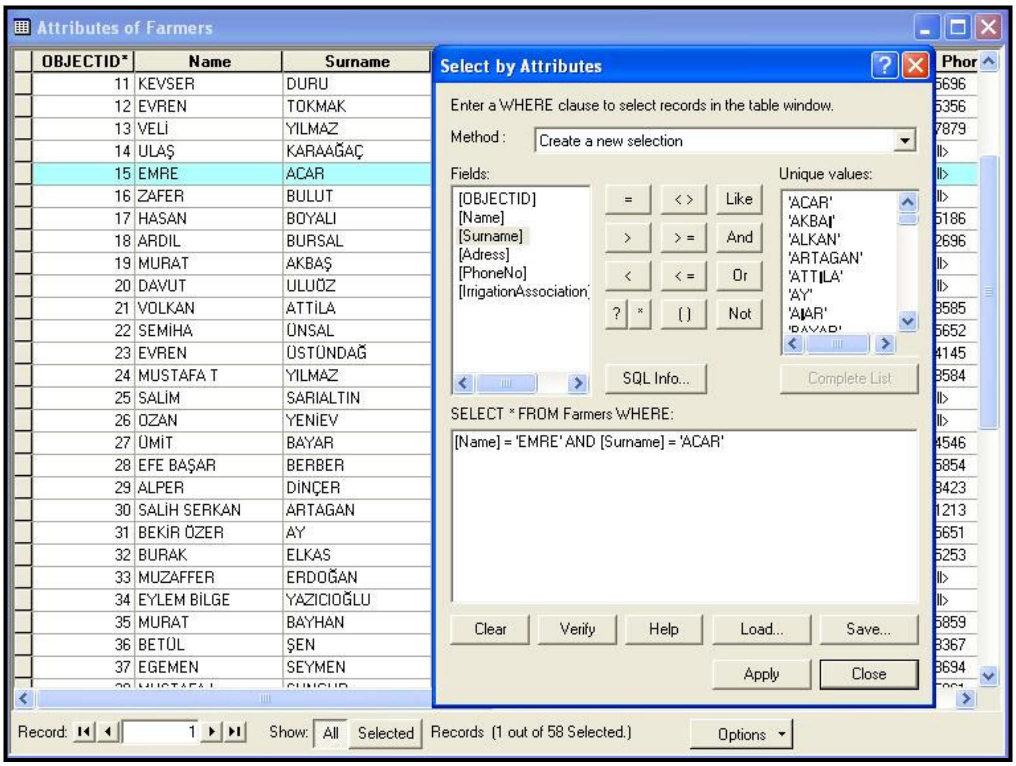


Figure 4.4: Screenshot of Scenario 1 Step1

Step 2: Find the farm which belongs to the farmer. The farm owned by the farmer is automatically selected since there is an Ownership relationship between Farms and Farmers classes (One of the advantageous feature of geodatabase).

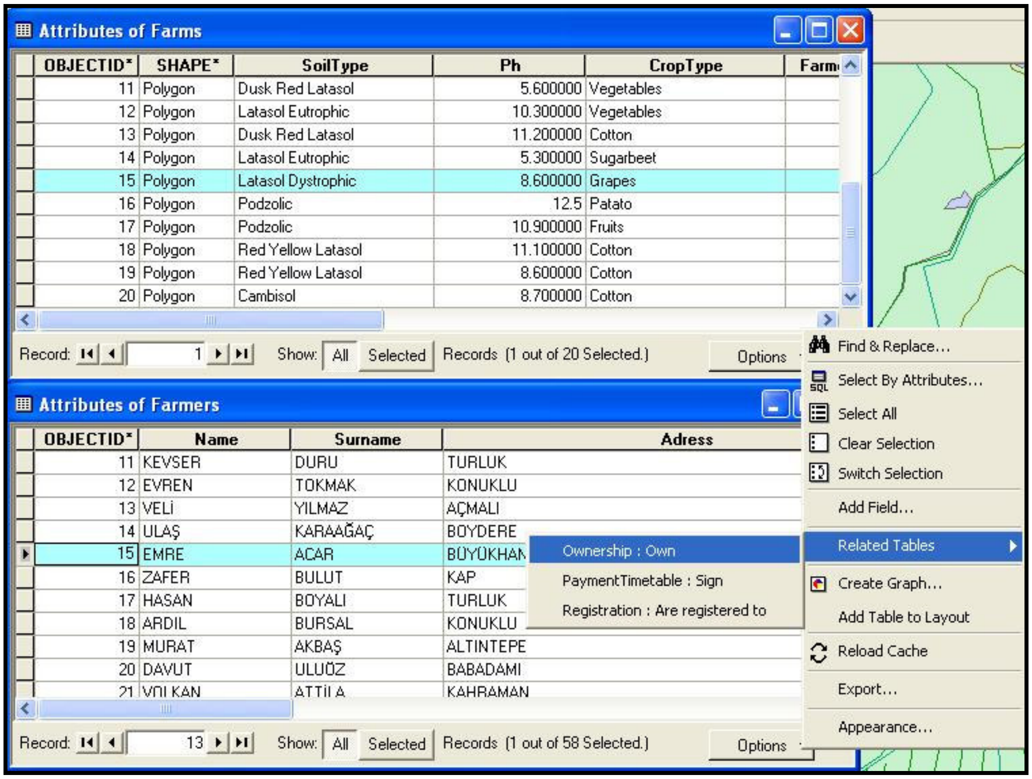


Figure 4.5: Screenshot of Scenario 1 Step 2

Step 3: Select the irrigation canals that are located within the farm. So the irrigation canal on which a siphon may be installed is determined.

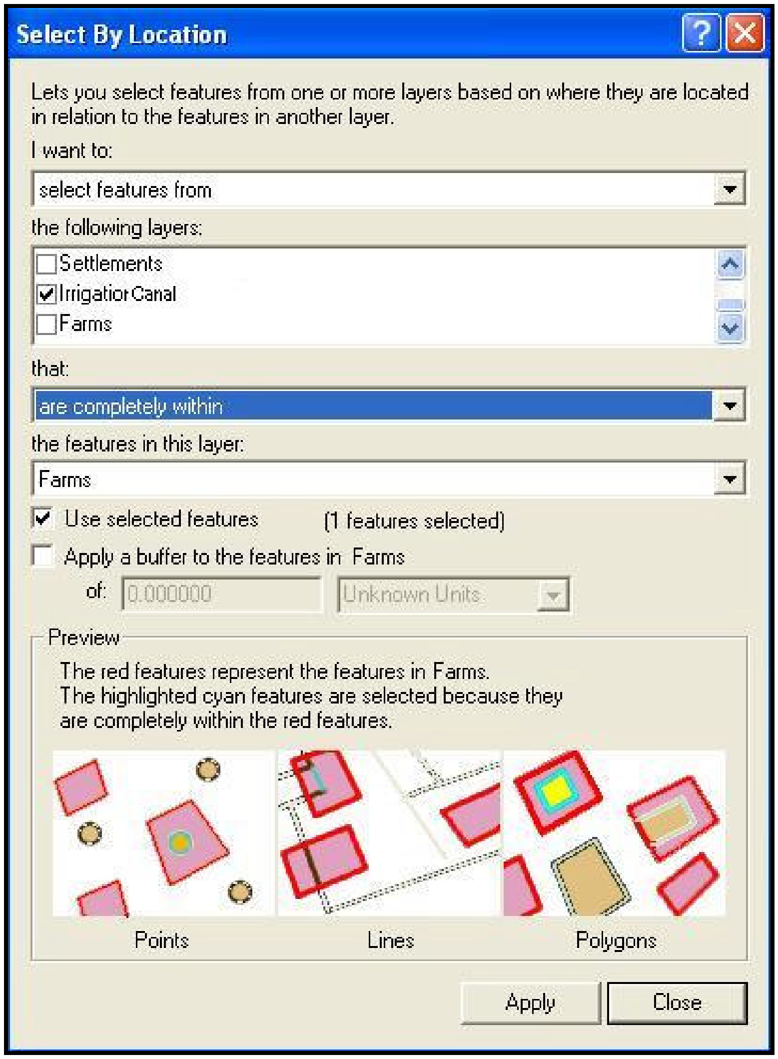


Figure 4.6: Screenshot of Scenario 1 Step 3

Scenario 2: An Irrigation Association wants to learn about the payment amount of a farmer for the water charges.

Step 1: Select the farmer.

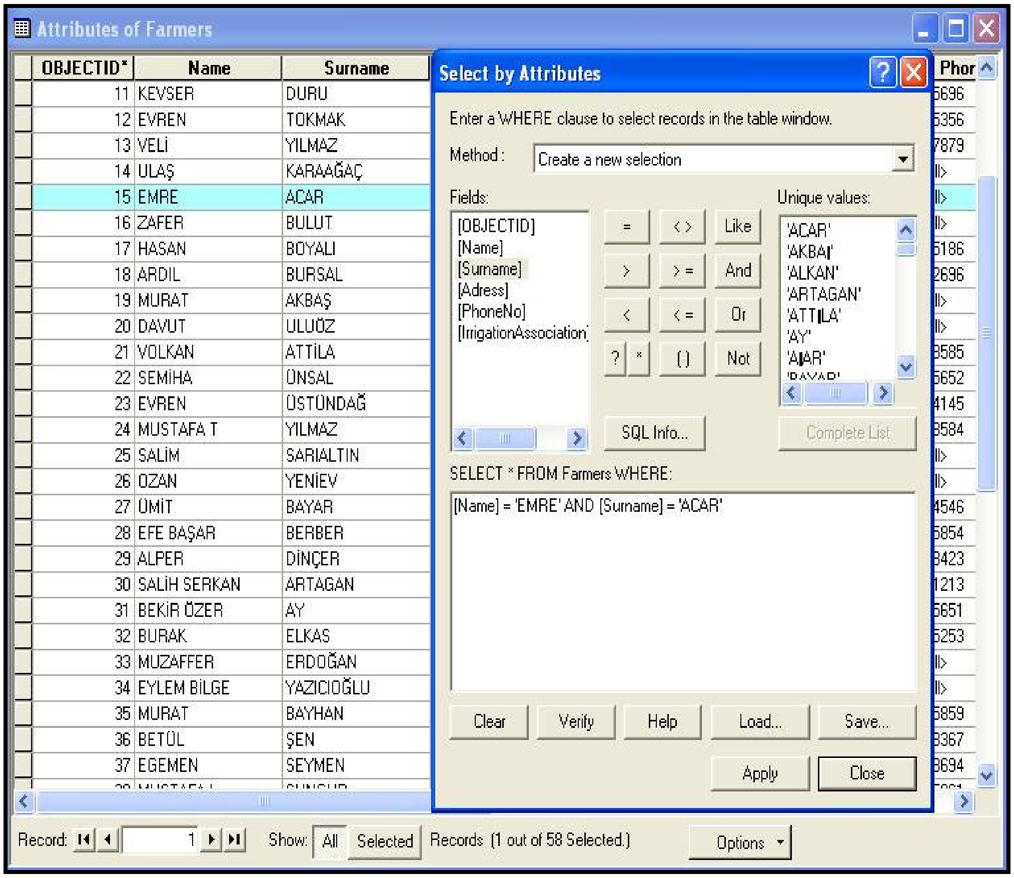


Figure 4.7: Screenshot of Scenario 2 Step 1

Step 2: The record in the Payment Timetable class related to the selected farmer is automatically selected.

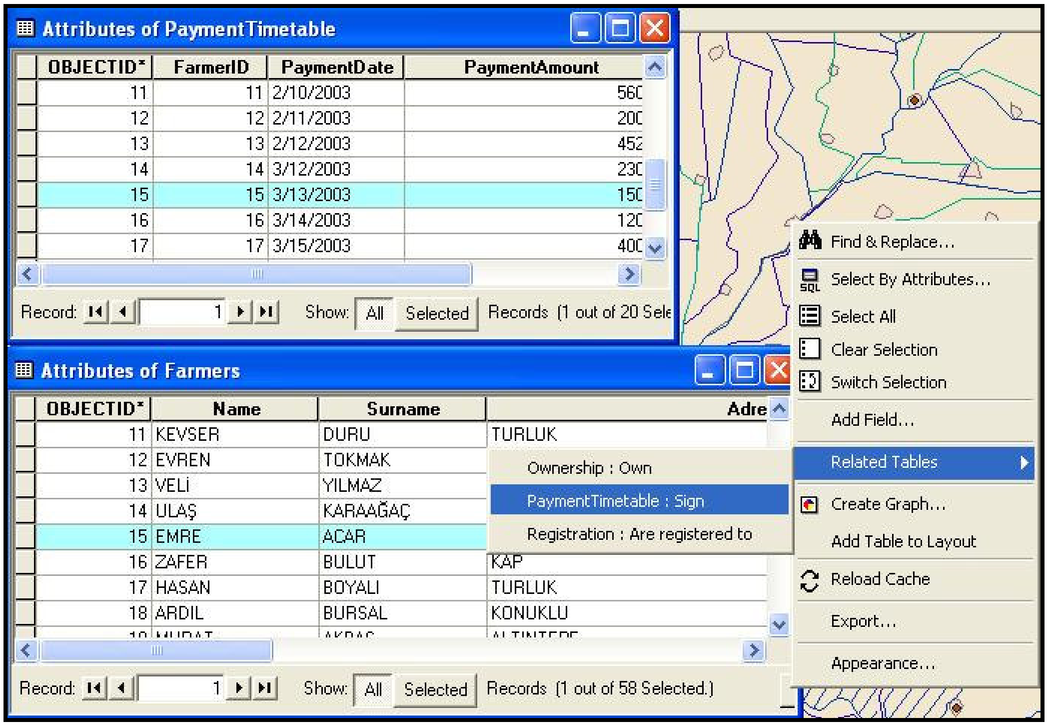


Figure 4.8: Screenshot of Scenario 2 Step 2

Scenario 3: The irrigation and drainage canals, which are located within the boundaries of a selected Irrigation Association, have to be determined for a pilot project that deals with only that region. To gather that information from map sheets is almost impossible.

Step 1: Select the irrigation association.

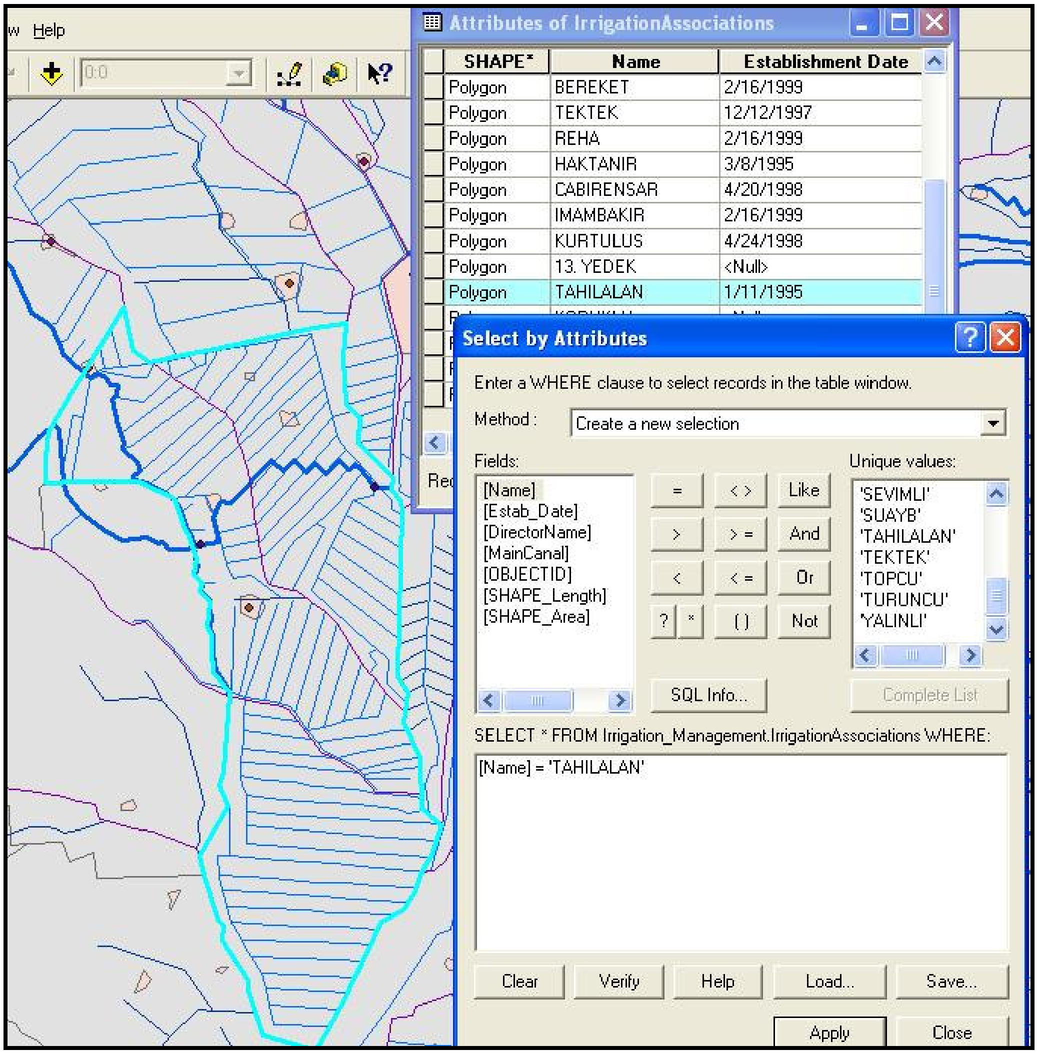


Figure 4.9: Screenshot of Scenario 3 Step 1

Step 2: Select the canals that are located within the selected irrigation association.

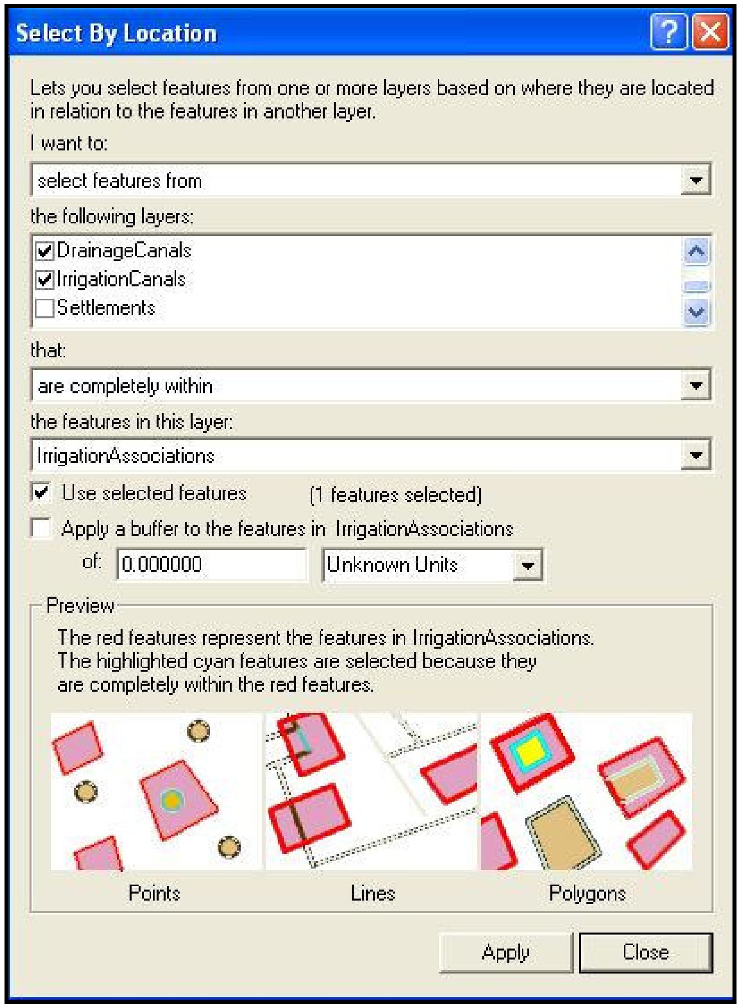


Figure 4.10: Screenshot of Scenario 3 Step 2

Scenario 4: The settlements of an irrigation association, whose drinking water is unsatisfactory, have to be determined by a governmental organization whose responsibility is to provide drinking water to villages.

Step 1: Select the irrigation association. (Same as in Scenario 3)

Step 2: Select the settlements located within the selected irrigation association.

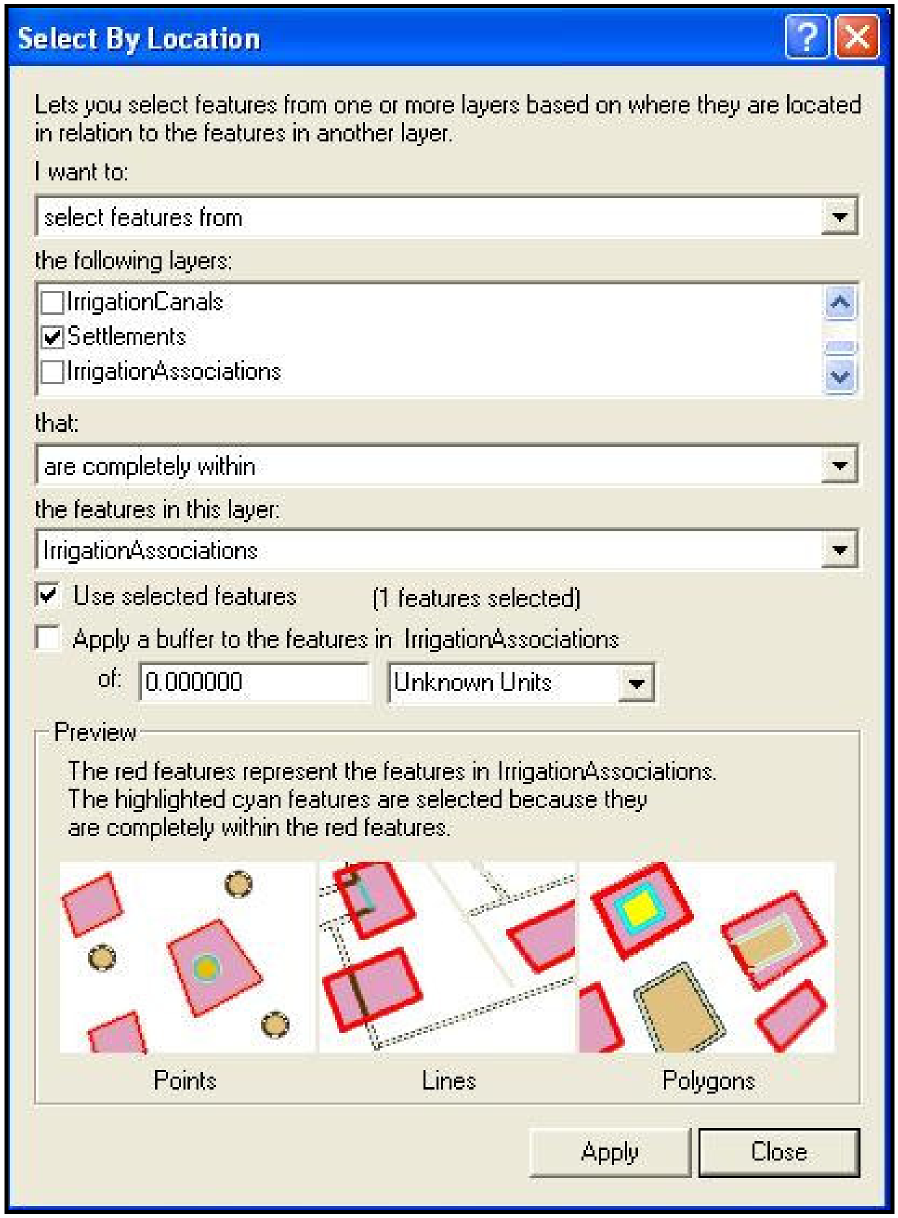


Figure 4.11: Screenshot of Scenario 4 Step 2

Step 3: Select the settlements within that irrigation association whose drinking water is unsatisfactory.

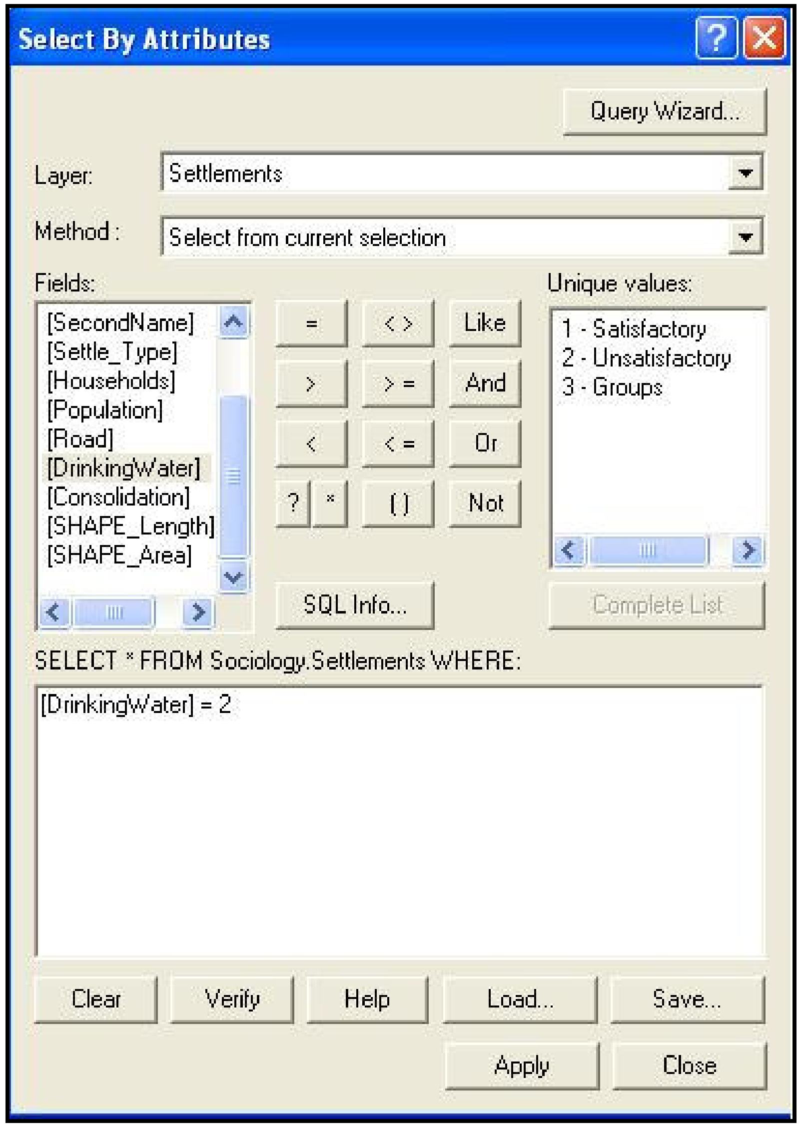


Figure 4.12: Screenshot of Scenario 4 Step 3

Scenario 5: Multi-user editing

Once loaded into a DBMS, ArcSDE feature classes become read-only by default. In order to make these feature classes editable their owner must use ArcCatalog to "Register them as Versioned" (Figure 4.13).

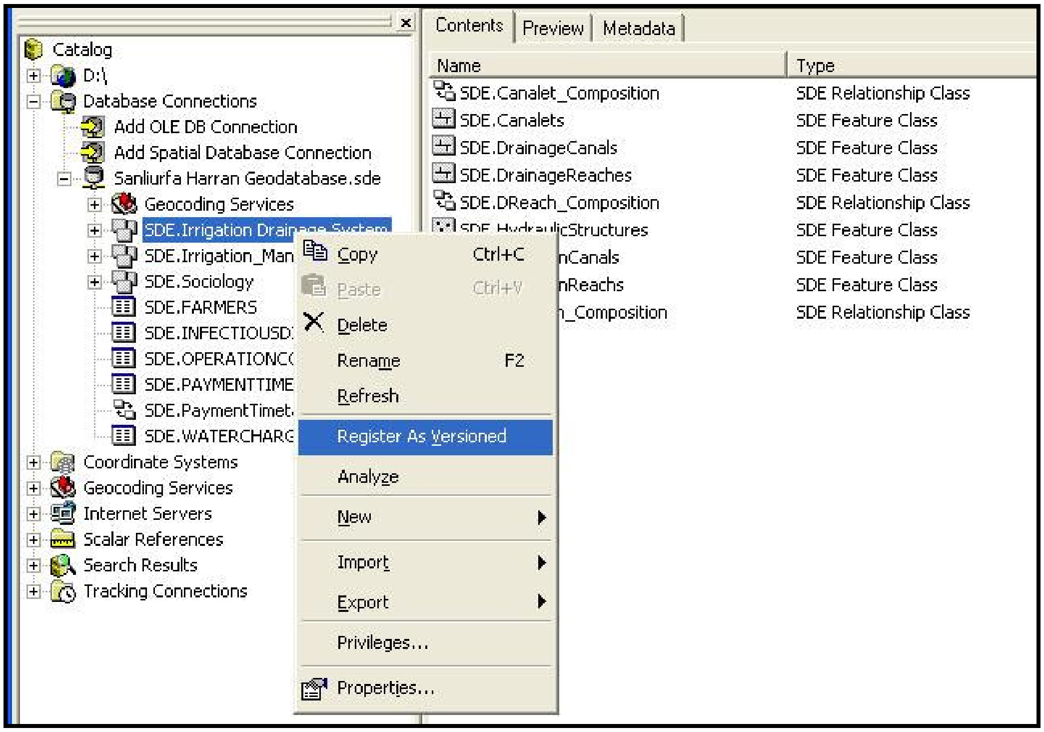


Figure 4.13: Registering ArcSDE tables as versioned

This scenario is based upon a case study of two users simultaneously editing same feature class. Two user accounts (User 1 and User 2) and two instances of ArcMap to simulate a multi-user environment are used (Figure 4.14).

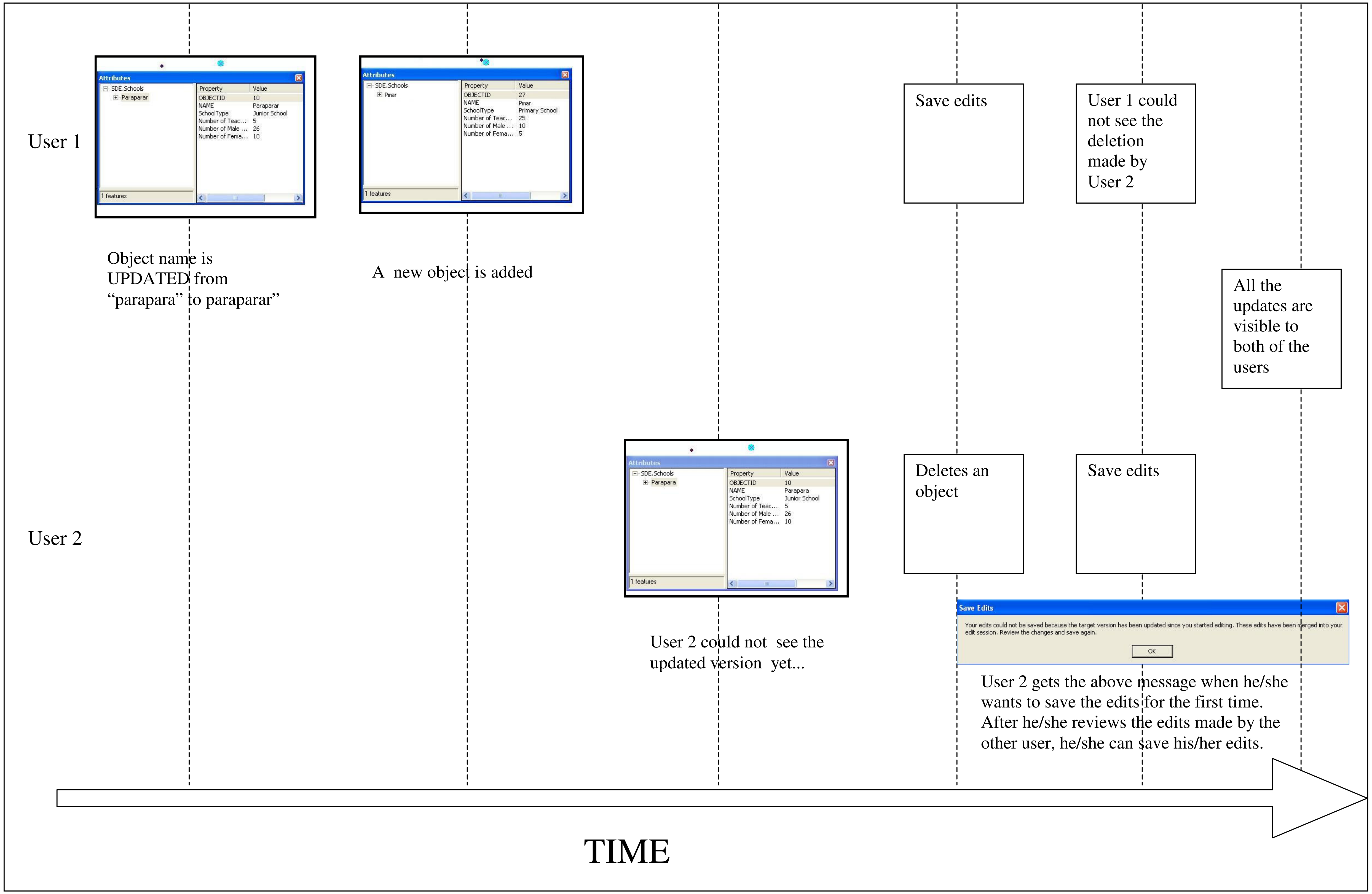


Figure 4.14: A scenario about Multiuser Editing

4.3 Benefits of Geodatabase Model

4.3.1 Schema Management and Database Design

- Centralized Data Management

A key feature of the geodatabase is that all the data are stored in commercial off-the-shelf DBMS. It utilizes the full functionalities of the underlying relational system. Therefore, a geodatabase is composed entirely of tables, but most of these tables are hidden from the user when one works with a geodatabase in ArcCatalog or ArcMap. Because multiple spatial and tabular data formats can be stored in the same geodatabase, managing and accessing GIS data have never been easier. This gives the power to have an integrated data management, which can significantly simplify support and maintenance. For a personel geodatabase, the underlying DBMS is Microsoft Access.

In Figure 4.15 Access reveals all the tables used to maintain the geodatabase. There are only two types of tables that you interact with directly in ArcCatalog or ArcMap: feature class tables and nonspatial attribute tables.

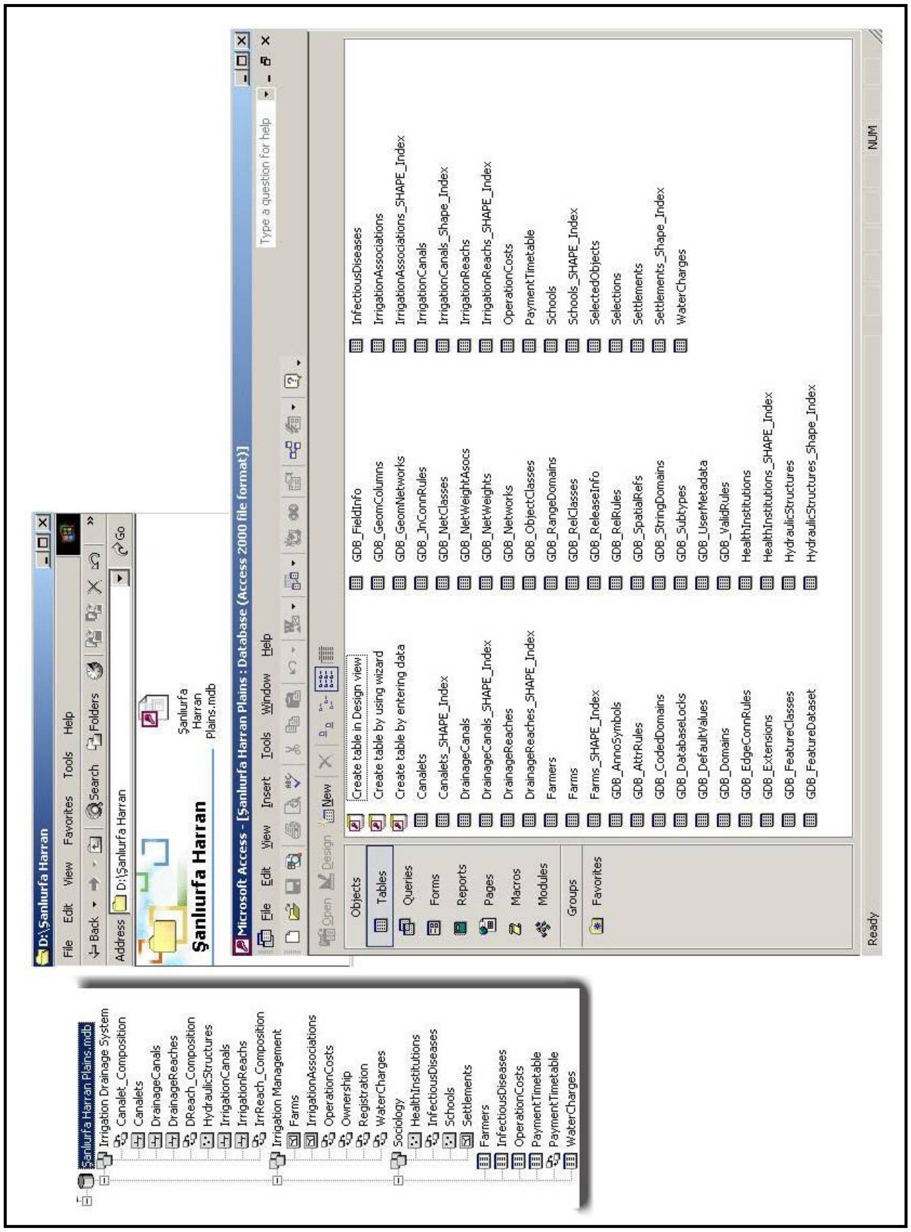


Figure 4.15: Centralized Data Management for Personal Geodatabase

• Centralized Storage of Attribute and Spatial Data

Traditionally spatial and attribute components of geographic data have been stored in separate files linked together by unique identifiers. Fields in a table can only handle standard data types, such as integer and string. However, within the geodatabase spatial data is stored just like an attribute in the database. Object relational technology, which is the basis of geodatabase, allows a table to posses blob fields. Inclusion of a blob field in the feature class table allows the attribute information to be stored alongside its spatial representation. (Figure 4.16)

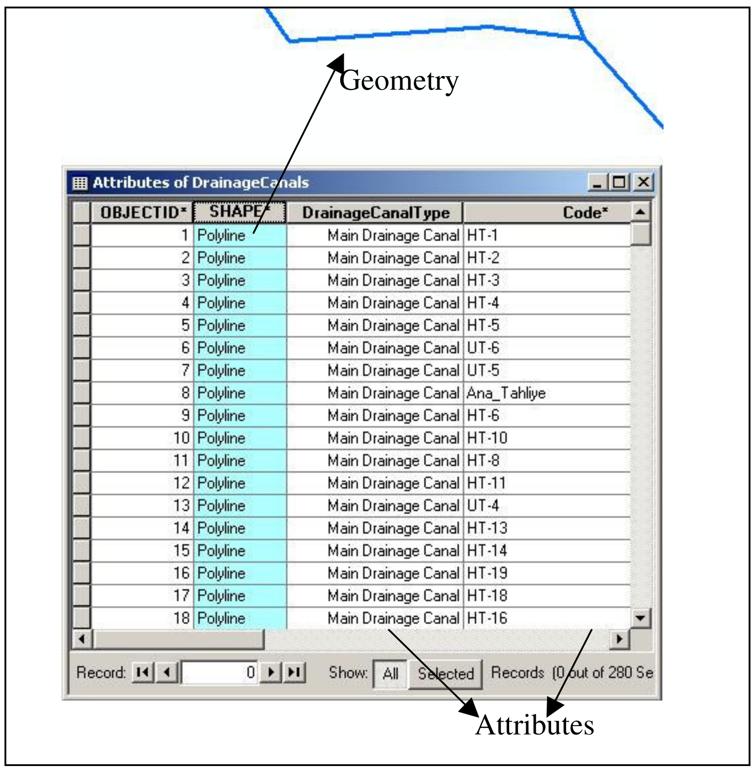


Figure 4.16: Centralized Storage of Attribute and Spatial Data

- Adoption of Object Oriented Concepts

The geodatabase model extends the relational model by integrating it with object oriented concepts. Object oriented design tools can be used to create object models. These tools make use of UML. And with a CASE tool, Visio Enterprise, it is possible to generate geodatabase schema.

Object oriented features such as inheritance lead to smoother and more efficient design reuse. Inheritance is particularly useful for adapting or extending an existing application. Only the differences need to be modified in the schema when new classes are inherited from the existing ones. Object oriented methods provide better a more powerful abstraction of reality and tend to lead better design of geodatabase. They give a far more accurate picture of the users' business operation and its information needs, hence more intuitive data objects can be created with geodatabase. Instead of generic points, lines, and areas the users refer to the objects of the real world such as schools and canals. Features in a goedatabase have their own behavior or methods encapsulated with data. The definition of a class specifies what values and behaviors can be associated with an object.

4.3.2 Multi-User Support

The major contribution of geodatabase model to GIS world is the multi-user support. This is accomplished through the client server architecture that ArcSDE provides. Of course this would be impossible if spatial and non-spatial data are not stored in the same database as the earlier GIS systems in which you need to maintain a link between file-based spatial data and attributes stored in the RDBMS tables. ArcSDE offers the same benefits for your spatial data that an RDBMS does for your non-spatial data. Some of the benefits of ArcSDE are:

- ArcSDE allows remote users to access both raster and vector GIS datasets for use in their projects.
- The use of ArcSDE eliminates the need to download data and store it locally. This saves time and resources as well as simplifies file management for the client.
- An SDE connection keeps an active link to SDE data, regardless of where a map document file (MXD) is located (or moved to) on the client's computer.

4.3.3 Editing

- Subtypes and Domains

Different kinds of features can be classified within a feature class by creating subtypes. Subtype field shows description not code and while editing, you work with the subtype description. Also for each subtype you create, you can specify default attribute values that will be applied automatically when new features are created. Changing the subtype updates the default values. (Figure 4.17) Hence with subtypes, data entry time and the chance of error are greatly reduced, promoting the integrity of your data. Moreover using subtypes instead of creating more feature classes increases performance.

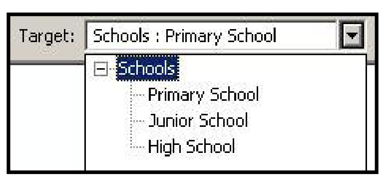


Figure 4.17: Editor Target List

Attribute domains are used to constrain the values allowed in a table field, feature class, or subtype. Coded value domains provide a list of valid attributes values when you are editing inside the attribute editor. (Figure 4.18)

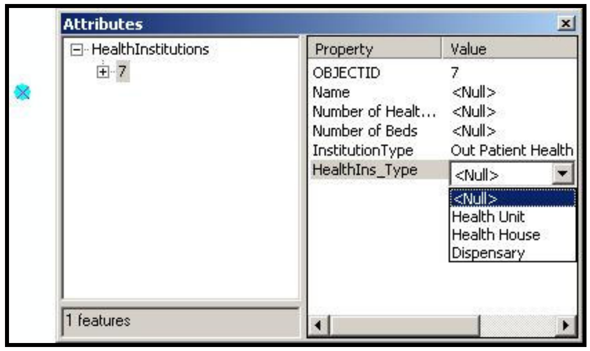


Figure 4.18: Attribute Editor List

- Relationships

Related tables of a class can be directly accessed from the attributes of the class even if the related classes have not been added. (Figure 4.19)

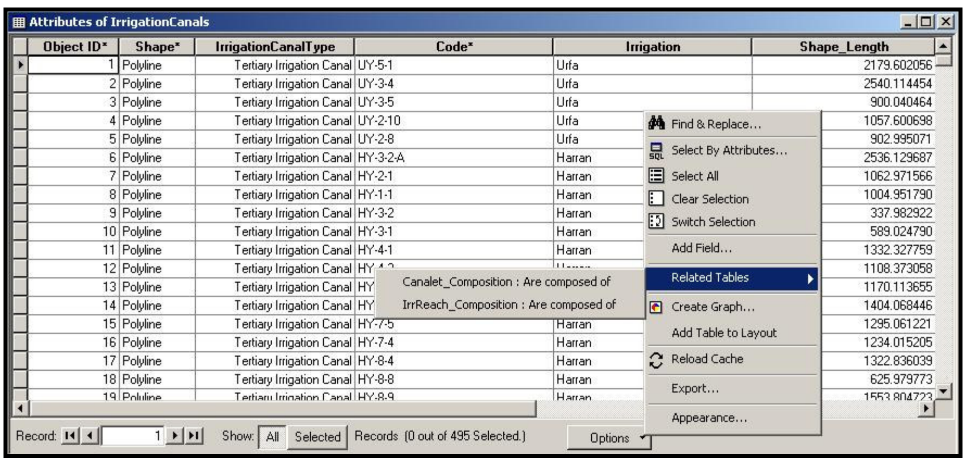


Figure 4.19: Related Tables in geodatabase

Related features of a selected feature can be edited simultaneously through Attribute editor. (Figure 4.20)

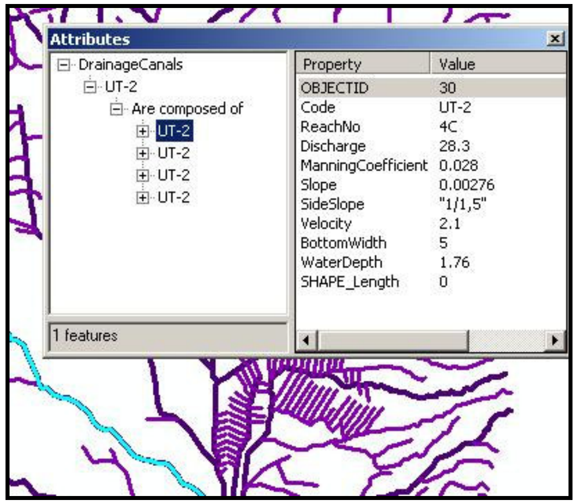


Figure 4.20: Related Data through Attribute Editor

CHAPTER 5

CONCLUSIONS

This research sought to explore object orientation concepts and demonstrate the application of OOGIS in irrigation management and rural information systems. Şanlıurfa Harran Plains within the scope of GAP were selected as the study area. Several research methods were explored. First, the underlying concepts of OO, as well as information systems development methodologies utilizing OO paradigm were addressed. UML, which is the combination of three popular OO methodologies (Booch, OMT (Rumbaugh et al., 1991) and OOSE (Jacobson, 1992)), was introduced and its Class Diagrams are explained thoroughly. Second, after a brief discussion on database design steps, a review of the existing GIS conceptual data modeling approaches was conducted. The evolution of GIS data models is also mentioned.

In this thesis ArcInfo UML Model is used to conceptualize spatial database of Şanlıurfa Harran Plains, and ESRI geodatabase model was used for the implementation. An emerging trend in GIS research includes the possibility of using object oriented GIS to enhance the abstraction and modeling. Several conceptual data models are proposed for geographic applications, which are based on general-purpose conceptual models such as ER and UML. ArcInfo UML Model utilizes UML Class Diagrams that are extended by tagged values and stereotypes to meet the spatial needs.

An attempt has been made to develop a unified spatial data model based on sound mathematics and OO concepts. The OO paradigm has been employed due to the fact that it deals with complexity in a more systematic manner. The OO design provides a cleaner data model, where each component is defined on its own and integrated in a more structured and flexible mode. The mathematical concepts used provide a sound basis for defining the space; this provides an unambiguous definition of the data model for implementation. These concepts are useful, in terms of extensibility and flexibility, for the modeling of space, and attributes. Extensibility means additional classes can be added, e.g., this model can be extended to include branching time or cyclic time. Flexibility means that object attributes and operations can be added or removed at any time. It has been demonstrated that a Şanlıurfa Harran Data Model can be designed and implemented by utilizing these concepts from conceptual level to the physical level.

In this research GIS design follows the IS development life cycle. The conceptual schema of the application domain is built using OOA techniques. Most of the GIS systems exclude the conceptual design phase starting the design with directly data collection phase, which in turn may bring about some maintenance problems and reduce reusability. Especially in Turkey where the general approach in software industry is to pass directly to coding without analysis and design, it is not surprising to observe this approach in GIS community. But the quality of information system solutions will only improve if more effort is devoted to the upstream or early activities in the application development life cycle. That is where conceptual modeling belongs. ArcInfo UML Model, which utilizes UML with proper extensions for spatial applications and supported with a commercial CASE Tool Microsoft Visio, is an excellent tool for conceptual representation. It reveals the modeling strength of UML.

Şanlıurfa Harran Data Model is built upon a set of classes satisfying the needs of the application. It has been demonstrated that the OO concepts underlying ArcInfo UML Model allow the design of these classes by simply inheriting from the predefined classes of ArcInfo UML Model. Additional classes can be added, e.g., this model can be extended to include road feature class. The model also proves to be a flexible model since the classes and object attributes and operations can be added or removed at any time.

Proposed object model of Şanlıurfa Harran will serve as an analysis pattern simplifying the rebuild process of the conceptual model for similar applications. As noted before the analysis phase of the system has been passed through a very long and hard way because of the complexity of the GAP project. Nevertheless the outcome of this process is a good analysis model for a subsystem of a complex system that can be abstracted and can become an analysis pattern for interested parties. It is much easier to use this analysis pattern instead of mapping all the requirements in an ad hoc manner. And extendibility and flexibility of ArcInfo UML Model makes the Şanlıurfa Harran Data Model much more powerful and reusable.

The ArcInfo geodatabase was implemented on Object-Relational databases in MS Access and Oracle using ArcSDE as an application server. The object-relational approach adopted is a hybrid architecture where an OO skin is built on top of an RDBMS. It has two advantages, i.e., it fully utilizes the power and semantics of the OO paradigm and the functionalities of the RDBMS. The advantage of the RDMBS is threefold. First, any commercial RDBMS can be utilized. Second, the power and functionalities (unified data management, data recovery or roll back, data backup, consistency constraints, multi-user access, widespread use, extensive infrastructure of product and standard support etc.) of the RDBMS are advantageous.

Implementing the geodatabase with the ArcSDE application server, which provides a three-tier architecture, allowed providing access to multiple users for both query and update (read/write) of the geographic data. The system espouses a client/server architecture and has several advantages, i.e., data can be located on local or remote machines, many users can access data at one time, and different users can view or update data. This architecture is suitable for designing integrated information systems, because the data related to Şanlıurfa Harran Plains are the fundamental requirement for many parties such as DSİ and GAP RDA. This approach may help to alleviate the accessibility problem of data. It is important to recognize that storing the data in a mature database application such as Oracle have a great impact on data maintenance and data retrieval and editing. The main advantages of using a DBMS as a central repository for spatial data include:

- Easier integration of spatial data with other core organizational data
- The storage of spatial and business data in the same database means reduced complexity, management, and training costs
- Expanded database size limits
- Support for the larger number of users required for enterprise implementations;
- The ability to take advantage of enhanced DBMS features such as administration and maintenance utilities, replication, and faster backup and recovery
- The availability of Java and SQL as open application programming interfaces (APIs);
- The emergence of spatial standards for interoperability
- The ability to publish and distribute spatial data over private Intranets or the public Internet

In the Şanlıurfa Harran geodatabase, through utilizing subtypes and domains the data integrity of the database is ensured. Integrity constraints are a set of data validation rules that you can specify to restrict the data values accepted into a data file. Using integrity constraints preserve the correctness and consistency of stored data meaning that any pieces of data, which violate the integrity, are not passed along to other applications or to end-users. However integrity constraints cannot guarantee data accuracy. There is still plenty of room for human error. Utilizing domains can prevent a data item from straying beyond specified values, but they cannot prevent a person from accidentally entering a wrong value, e.g. entering village for a district for settlements feature class. But it must also be noted that subtypes and domains provide faster and more efficient data entry since the value of a field can be chosen directly from the drop down list instead of typing it.

One of the major deficiencies in the current geodatabase system is lack of data. Mostly fictitious data besides small set of real data were inserted into the database. Obtaining suitable data requires the development of a methodology, which identifies data that can be modeled on a GIS, and/or a methodology to guide new research to collect relevant, spatially referenced data; both tasks are complex and depend upon the availability of significant resources. Most of the time data, especially social data, are not spatially referenced and expensive to produce, and access to such data may also be very difficult. On the other hand, available data may be of poor quality and though they might be spatially referenced they would be of dubious analytical value even when combined with other data sets. In short, data collection and capture are a time consuming and expensive undertaking.

In the geodatabase of Şanlıurfa Harran Plains three aspects are taken into consideration; Irrigation and Drainage System, Irrigation Management, and Sociological matters. Based on these aspects some small-scaled sceniorias are demonstrated using query capabilities of ArcInfo. Querying the data can be accomplished in both Oracle and ArcMap. In Oracle the queries are created by using the SQL*Plus Worksheet which allows for a whole range of SQL and PL/SQL queries. While ArcMap allows for more simplified queries, it is preferrable since the results are visually displayed in the map panel. Also the relationship classes make things easier for implementing a scenerio, since when one object is selected the related object is selected automatically.

The followings are some of the major uses of proposed geodatabase system of Şanlıurfa Harran Plain:

- During the road planning activities of General Directorate of Highways, locations of the roads need to be fixed based on the canals. Traditionally this study is conducted with aid of map sheets of canals requested from DSİ. Within a geodatabase system, on the other hand, a seamless aggregation of many map sheets exits as a feature class.
- Upon a siphon request from farmers, the tertiary canal, which irrigates the field of the farmer, can be determined through the geodatabase system by DSİ.
- Places where the infectious diseases are observed as a large number of cases determined by Ministry of Health. So that appropriate health education and disease prevention measures can be implemented. Maintenance on canals near those places should be increased.
- Data related to water charges will be kept in the geodatabase. This will ease up the process of fixing and collecting water charges which is the responsibility of Irrigation Associations.
- Accurate and up to date maps can be established. For example currently the revisions on canals are shown on the map sheets by a colored pencil in DSİ.
- Storage of data related to Şanlıurfa Harran Plains in a multi-user geodatabase in which the underlying database is Oracle, will speed up exchange of information among many institutions. This way the projects dependent upon the data related to the region will progress rapidly

5.1 Recommendations

The temporal model of Şanlıurfa Harran Plains Geodatabase that enables the storage and analsis of historical data could be developed. The multi-user ArcSDE geodatabase allows multiple versions of the database to coexist. If the concepts developed in this research are combined with the versioning mechanism in ArcSDE geodatabase, a capable temporal GIS may be developed. ArcSDE and ArcInfo support for long transactions through versioning allows users to simultaneously create multiple, persistent representations of the database without data replication.

It is very important to understand that implementing a multi-user GIS system with ArcInfo and ArcSDE requires planning and experimentation to be successful. Each stage of the process, from the data model to database tuning and application design has a significant impact on how well the system performs. In this research default values or the parameters set by a technical person from İşlem (distributor of ESRI's products) are used. Further adjustments may be necessary to improve and test the performance of the system. Also it is recommended that different users (departments in an institution) view or update data, based on rights assigned by a database administrator.

Although basic object behavior was used in this study, OOGIS attains its ultimate utility when the user tailors it to the specific organizational requirements by incorporating advanced object behavior. This may mean using object-oriented languages such as C++ and Visual Basic, additional object behavior for different project needs can be attained and a good graphical user interface can be designed. In the future, additional object behavior may need to be the primary focus of an object-oriented geodatabase if necessary. Objects in the geodatabase can have behaviour associated with them. Integration with object-oriented concepts and COM technology allows great level of customization and reuse of the model to create application-specific models. On a more general note, this research has proven that effective tools can be developed to work in ArcGIS by creating a DLL in the Visual Basic programming environment. Furthermore, the ArcObjects Library is compliant with the OpenGIS simple feature specifications, which may (in the future) provide the framework for interoperability.

It is also recommended to design an Internet/Intranet system that allows different institutions to map, analyze and view various data related to Şanlıurfa Harran. The web-based application of GIS holds great promise for geographic data management whether it is undertaken internally within an organization, externally by a third party, or a combination of the two. Distributing geographic information over the Internet allows for real-time integration of data. Running software programs over the web eliminates the need to install or manage any software, hardware or data locally as all software and data components are requested from a server. ArcIMS can be the solution for distributing mapping and GIS data and services on the Web. Whether you are operating strictly within organization's Intranet or sharing information with hundreds of thousands of people over the Internet, you can use ArcIMS to distribute geographic data to many concurrent users and allow them to do location-based analyses. Also ArcIMS and ArcSDE work together as an integrated back office solution for fast Internet access to data stored in a relational database.

REFERENCES

Abu-Zeid, M. (2001). Water Pricing in Irrigated Agriculture. *International Journal of Water Resources Development*, 17(3), 527-538.

Avison, D. E., and Fitzgerald, G. (1995). *Information Systems Development: Methodologies, Techniques and Tools*. Oxford, UK: Blackwell Scientific Publications.

Bedard, Y., Caron, C., Maamar, Z., Moulin, B., and Valliere, D. (1996). Adapting Data Models for the Design of Spatio-Temporal Databases. *Computer, Environment and Urban Systems Int. Journal, 20* (1), 19-41.

Bedard, Y. (1999a). Principles of Spatial Database Analysis and Design. In P. Longley, M. Goodchild, M. Maguire and D. Rhind (2nd Ed.), *Geographical Information Systems: Principles, Techniques, Management and Applications* (413-424). USA: John Wiley & Sons.

Bedard, Y. (1999b). Visual Modeling of Spatial Databases: Towards Spatial PVL and UML. *Geomatica, 53*(2), 169-186.

Bennett, S., McRobb, S., and Farmer, R. (1999). *Object-oriented systems analysis and design using UML*. London: McGraw-Hill.

Berard, E. V. (1991). *Object-Oriented Design*. Caithersburg: Berard Software Engineering, Inc.

Booch, G. (1994). *Object Oriented Analysis and Design with Applications* (2nd Ed.). Redwood City, California, USA: The Benjamin/Cummings Publishing Company.

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide.* Addison-Wesley.

Borges, K., Laender, A., and Davis, C. (1999). Spatial Data Integrity Constraints in Object Oriented Geographic Data Modeling. *In Proceedings of the seventh ACM* *international symposium on Advances in geographic information systems*, Kansas City, Missouri, United States, 1999. New York, USA: ACM Press. 1-6.

Borges, K., Davis, C., and Laender, A. (2001). OMT-G: An Object-Oriented Data Model for Geographic Applications. *GeoInformatica, 5* (3), 221-260.

Böke, C. M. (1997). Irrigation water pricing in Turkey. *Master's Thesis*, METU, Ankara

Brodeur, J., Bedard, Y., and Proulx, M. (2000). Modelling Geospatial Databases using UML-based Repositories Aligned with International Standards in Geomatics. *In Proceedings of the seventh ACM international symposium on Advances in geographic information systems,* Washington, D.C., United States, 2000. New York, USA: ACM Press. 39-46.

Brookfield, H., and Byron, Y. (1993). *South-East Asia´s Environmental Future: the search for sustainability*. Tokyo, New York, Paris: United Nations University Press.

Coad, P., and Yourdon, E. (1991a). *Object-oriented analysis* (2nd ed.). Englewood Cliffs, N.J.: Yourdon Press.

Coad P., and Yourdon, E. (1991b). *Object-oriented design*. Englewood Cliffs, New Jersey, USA: Prentice-Hall

Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, H., Hayes, F., and Jeremaes, P. (1993) *Object-Oriented Development: The Fusion Method.* Englewood Cliffs: Press/Prentice Hall.

Davis Jr., C. A., and Laender, A. H. F. (1999). Multiple representations in GIS: materialization through map generalization, geometric and spatial analysis operations. *In Proceedings of the seventh ACM international symposium on Advances in geographic information systems*, Kansas City, Missouri, United States, 1999. New York, USA: ACM Press. 60-65.

Dinar, A. (1998). Water policy reforms: Informational needs and implementation obstacles. *Water Policy, 1*(4), 367–382.

Elmasri, R., and Navathe, S.B. (2000). *Fundamentals of Database Systems*. Addison-Wiley.

ESRI, Redlands. (2001). *Managing ArcSDE Services.*

FAO Aquastat, (2000). Retrieved August 24, 2002, From http://www.fao.org/

Fernandez, E. B. (1998). Building Systems Using Analysis Patterns. *In Proceedings of the third international workshop on Software architecture*, Orlando, Florida, 1998. ACM Press. 37-40.

Filho, J. L., and Iochpe, C. (1999). Specifying Analysis Patterns for Geographic Databases on the Basis of a Conceptual Framework. *In Proceedings of the seventh ACM international symposium on Advances in geographic information systems*, Kansas City, Missouri, United States, 1999. New York, USA: ACM Press. 7-13.

Friis-Christensen, A., Tryfona, N., and Jensen, C. S. (2001). Requirements and Research Issues in Geographic Data Modeling. *In Proceedings of the ninth ACM international symposium on Advances in geographic information systems,* Washington, D.C., United States, 2000. New York, USA: ACM Press. 2-8.

GAP RDA. (1998). Şanlıurfa & Harran Plains On-Farm and Village Development Project: Inception Report.

Hadzilacos, T., and Tryfona, N. (1996). Logical Data Modelling for Geographic Applications. *International Journal of Geographical Information Science, 10* (2), 179-203.

Hadzilacos, T., and Tryfona, N. (1997). An Extended Entity-relationship Model for Geographic Applications. [Electronic Version]. *ACM SIGMOD Record, 26*(3), 24-29.

Henderson-Sellers, B. (1992). *A Book of Object-Oriented Knowledge.* New York: Press/Prentice Hall.

Hirschheim, R. A., Klein, H. K., and Lyytinen, K., (1995). *Information systems development and data modeling: conceptual and philosophical foundations*. Cambridge, New York: Cambridge University Press.

Iochpe, C., and Filho, J. L. (1998). BCH/GIS: a basic class hierarchy to support GIS conceptual design. *In Proceedings of International Conference on Modeling Geographic and Environmental Systems with GIS*, Hong Kong, 1998.

Jacobson, I. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.

Johansson, R.C., (2000). Pricing irrigation water: a literature survey. *Policy Research Working Papers, World Bank*

Johansson, R.C., Tsurb, Y., Roec, T. L., Doukkalid, R., and Dinare, A. (2002). Pricing irrigation water: a review of theory and practice. *Water Policy*, 4, 173–199.

Kosters, G., Pagel, B., and Six, H. (1997). GIS-Application Development with GeoOOA. *International Journal of Geographical Information Science, 11* (4), 307-335.

Lbath, A. (1997). AIGLE: a Visual Environment for Design and Automatic Generation of Geographic Applications. Phd Thesis, INSA Lyon, France.

Lbath, A., and Pinet, F., (2000). The Development and Customization of GIS-Based Applications and Web-Based GIS Applications with the CASE Tool AIGLE. *In Proceedings of the eighth ACM international symposium on Advances in geographic information systems,* Washington, D.C., United States, 2000. New York, USA: ACM Press. 194-196.

Lee, R. C., and Tepfenhart, W. M. (2002). *Practical object-oriented development with UML and Java*. Upper Saddle River, NJ: Prentice Hall.

Martin, J., and Odell, J. J. (1992). *Object-Oriented Analysis & Design.* Englewood Cliffs, New Jersey: Press/Prentice Hall.

Muller, R. J. (1999). *Database design for smarties: using UML for data modeling*. San Francisco, California: Morgan Kaufmann Publishers

Oliveira, J. L., Pires, F., and Medeiros, C. B. (1997). An environment for integrated modeling and analysis of geographic information. *GeoInformatica, 1*(1), 29-58.

Pageau, J., and Bedard, Y. (1992). Conception d'un outil CASE pour la modélisation des données à référence spatiale. *Actes de la Conference canadienne sur les SIG,* 381-392

Pantazis, D., and Donnay, J.P., (1996). *Conception de S.I.G.: Méthode et formalisme*. Hermes, Paris: Collection Géomatique.

Parent, C., Spaccapietra, S., Zimanyi, E., Donini, P., Plazanet, C., and Vangenot, C. (1998). Modeling Spatial Data in the MADS Conceptual Model. *In 8th International Symposium on Spatial Data Handling '98 Conference Proceedings*, Vancouver, BC, Canada, July 1998. 138-150.

Parent, C., Spaccapietra, S., and Zimanyia, E. (1999). Spatio-Temporal Conceptual Models: Data Structures + Space + Time. *In Proceedings of the seventh ACM international symposium on Advances in geographic information systems*, Kansas City, Missouri, United States, 1999. New York, USA: ACM Press. 26-33.

Parent, C. (2000). A CASE tool for designing Spatio-Temporal Databases. *International Workshop on Emerging Technologies for Geo-Based Applications,* Ascona, Switzerland, May 22-25. Database Laboratory. 1-19.

Pinet, F., and Lbath, A. (2000). A Visual Modelling Language for Distributed Geographic Information Systems. *In Proceedings of International Symposium on Visual Languages,* Seattle, Washington, USA, September 10 – 13, 2000. USA: IEEE Computer Society. 75-76.

Pressman, R. S. (2001). Software *Engineering: A Practitioner's Approach*. 5th ed. McGraw Hill.

Price, R., Tryfona, N. and Jensen, C. (2000). Extended Spatiotemporal UML: Motivations, Requirements, and Constructs. *Journal of Database Management., 1l* (4), 14-27.

Quatrani, T. (2003). *Visual modeling with Rational Rose 2002 and UML.* Boston: Addison-Wesley.

Roff, J. T. (2003). *UML: a beginner's guide*. Berkeley, California and London: Osborne/McGraw-Hill.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. (1991). *Object Oriented Modelling and Design*. Prentice Hall, Inc.

Rumbaugh, J. (1996). Layered Additive Models: Design as a Process of Recording Decisions. *Journal of Object-Oriented Programming, 9*(1), 2 l-24.

Seckler, D., Amarasinghe, U., Molden, D., de Silva, R., Barker, R. (1998). World water demand and supply, 1990–2025: Scenarios and issues. Research Report #19, IIMI Colombo, Sri Lanka.

Shlaer, S., and Mellor, S. (1988). *Object-Oriented Systems Analysis: Modeling the World in Data.* Engelwood Cliffs, New Jersey: Yourdon Press.

Shekhar, S., Coyle, M., Goyal, B., Liu, D., and Sarkar, S. (1997). Data models in geographic information systems [Electronic version]. *Communications of the ACM, 40*(4), 103-111.
Southeastern Anatolia Project Regional Development Administration, Retrieved Jan 18, 2003, from http://www.gap.gov.tr

Svendsen, M., and Nott, G. (1998). Irrigation Management Transfer in Turkey: Process and Outcomes. *Advanced Short Course on: Capacity Building for Participatory Irrigation Management (PIM). Valenzano, BA (Italy), 7-23 September*.

Sulama Birlikleri. Retrieved Jan 21, 2003, From http://www.yerelnet.org.tr/altyapi/sulamabirlikleri.php

Tsichritzis, D., and Klug, A. (1978). The ANSI/X3/SPARC DBMS Framework: Report of the Study Group on Data Base Management Systems. *Information Systems, 3*(3), 173-191.

Tryfona, N., and Hadzilacos, T. (1995). Geographic Applications Development: Models and Tools at the Conceptual Level. *In proceedings of the 3rd ACM International Workshop on Advances in Geographic Information Systems.* Baltimore, Maryland, 1995.

Tryfona, N., Pfoser, D., and Hadzilacos, T. (1997). Modeling Behavior of Geographic Objects: An Experience with the object modeling technique. *In Proceedings of the 9th International Conference on Advanced Information Systems Engineering, CAiSE'97 (Lecture Notes in Computer Science Vol. 1250)*, Barcelona, Catalonia, Spain, June 16-20, 1997.Springer-Verlag. 333-346.

Tryfona, N., and Jensen, C. (1999). Conceptual Data Modeling for Spatiotemporal Applications. *GeoInformatica, 3*(3), 245-168

Tsur, Y., and Dinar, A. (1997). On the relative efficiency of alternative methods for pricing irrigation water and their implementation. *World Bank Economic Review*, 11, 243–262.

USGS, Retrieved Jan 14, 2003, From http://www.usgs.gov/research/gis/title.html

Ünal, A. (2001). Participatory privatization of irrigation management in Turkey. *International Network on Participatory Irrigation Management (INPIM) Newsletter.*

Ünalan, T. (2003). Population and Development Indicators: National and International Practices, Report Prepared with the support of the UNFPA Sub-Programme on

Population and Development Strategies, Turkish Academy of Sciences Reports No. 1, Tübitak Printing House, Ankara.

Ünver, O., and Gupta, R. K. (2002). Water Pricing: Issues and Options in Turkey. In: *Proceedings of Agadir Conference on Water Management and Pricing in the Mediterranean Region, Morocco, 15-17 June.*

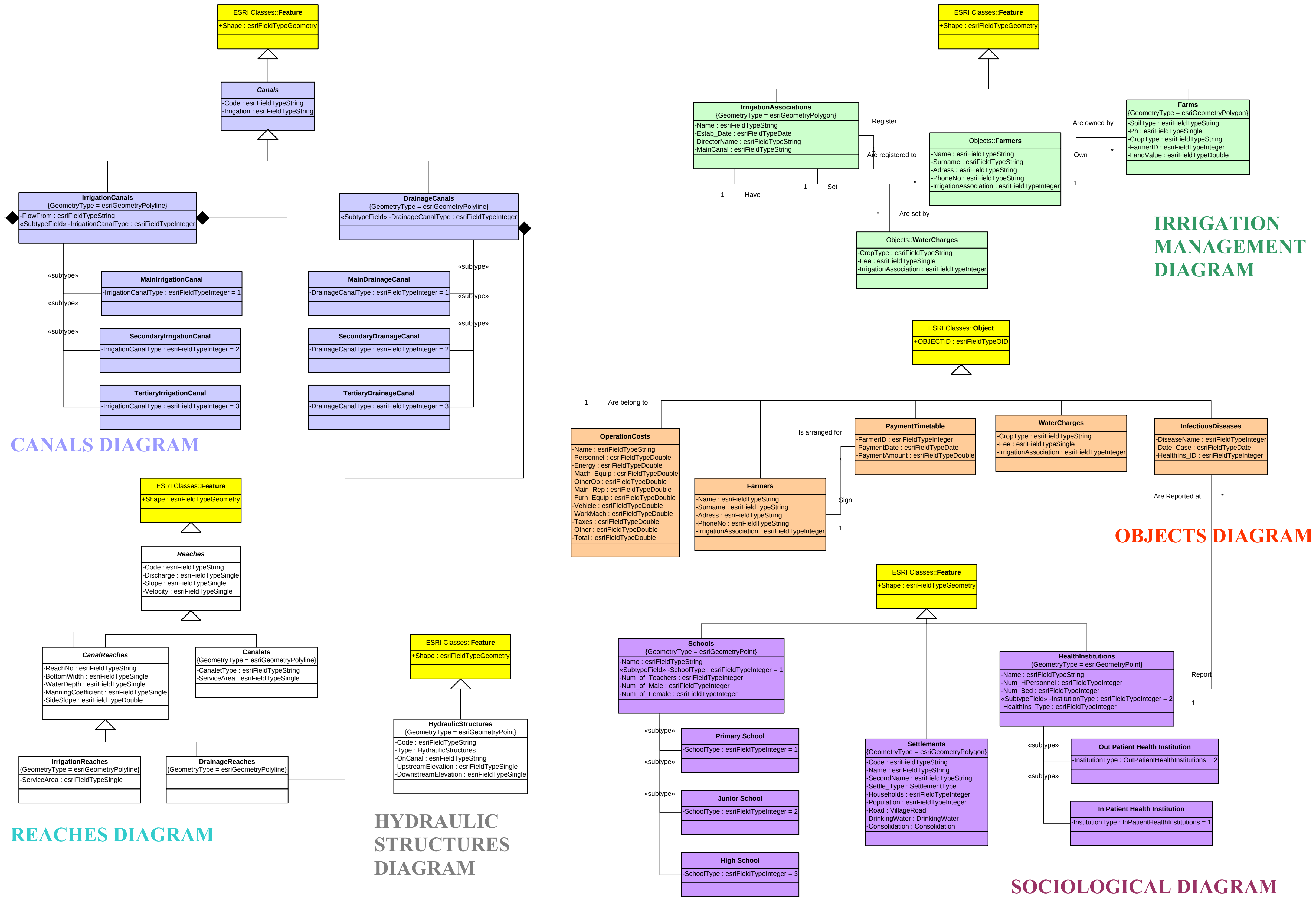
Wirfs-Brock, R., Wilkerson, B., and Wiener, L. (1990). *Designing Object-Oriented Software.* Prentice-hall.

Worboys, M. F. (1995). *GIS: A Computing Perspective*. London, UK: Taylor and Francis Ltd.

Yazar, A. (2002). Participatory Irrigation management (PIM) in Turkey: A Case Study in the Lower Seyhan Irrigation Project. In: *Proceedings of Agadir Conference on Water Management and Pricing in the Mediterranean Region, Morocco, 15-17 June.*

Yourdon, E. (1994). *Object-Oriented Systems Design: An Integrated Approach*. Englewood Cliffs, New Jersey, USA: Prentice-Hall Inc.

APPENDIX



ŞANLIURFA HARRAN PLAIN DATA MODEL