COĞRAFİ BİLGİ SİSTEMLERİNDE GEOMETRİ KÜTÜPHANESİNİN GELİŞTİRİLMESİ

Murat Hacıömeroğlu[1], Hayri Sever[2], Haşmet Gürçay[3]

[1]University of East Anglia, İngiltere, m.haciomeroglu@uea.ac.uk

[2]Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü, sever@hacettepe.edu.tr

[3]Hacettepe Üniversitesi, Matematik Bölümü, gurcay@hacettepe.edu.tr

ÖZET

Coğrafi Bilgi Sistemleri (CBS) dünyada ve ülkemizde sürekli gelişen bir ilgi alanı halindedir. Coğrafi Bilgi Sistemlerinde verilerin yapısı diğer sistemlerden bazı farklılıklar göstermektedir. CBS'ler de veriler konumsaldır (konumsal geometriler) ve geleneksel yöntemlerle işlem yapmak zordur. Birçok araştırmacı geometrik yapılar ve aralarındaki ilişkileri tanımlamak için çalışmalar yapmıştır. Bu çalışmalar nihayet günümüzde ihtiyaçları karşılayabilecek bir standarda kavuşmuştur. Geometrik yapıları ve aralarındaki ilişkileri, geçmiş çalışmalar temelinde standarda kavuşturmak üzere Open Geospatial Consortium (OGC) kurulmuştur. Konumsal verilerin işlenebilmesi için ise bu verileri tanımlayacak ve aralarındaki işlemleri yürütebilecek konumsal bir sınıf kütüphanesinin gerçekleştirilmesi gerektiği açıktır.

Anahtar Kelimeler: Konumsal Veri, Geometrik Sınıf Kütüphanesi, Coğrafi Bilgi Sistemi.

ABSTRACT

DEVELOPMENT OF GEOMETRIC LIBRARY IN GEOGRAPHIC INFORMATION SYSTEMS

Geographical Information Systems (GIS) has been a progressively developing interest area in the world and in our country. In Geographical Information Systems, the structure of the data indicates some differences from other systems. In GIS, data are spatial (spatial geometry) and it is difficult to make operations with traditional methods. Many researchers have made studies to describe the geometric structures and the relationships between them. Today, these studies finally reached the standards that meet the needs. On the basis of previous studies, in order to reach the standards for geometric structures and the relationships between them, Open Geospatial Consortium was founded. It is clear that a spatial class library should be formed in order to describe the data for dealing with this spatial data and to process the operations between them.

Keywords: Spatial Data, Geometric Class Library, Geographical Information Systems

1.GİRİŞ

Çeşitli uygulama alanlarında geometrik ya da konumsal veriyi (, yani uzay ile ilişkili olan veriyi) kullanmak ve işlemek gerekmektedir. Konumsal verileri ifade edebilmek, sorgulayabilmek ve değiştirebilmek amacıyla var olan veri tabanı sistemlerine ek özellikler kazandırılarak konumsal veritabanı sistemleri tasarlanmıştır. Bu tür konumsal veritabanları *Coğrafi Bilgi Sistemleri* (CBS) gibi uygulamaları destekleyen veritabanı teknolojilerine sahiptir. *Nokta, doğru* ve *bölge* gibi Konumsal Veri Türleri (KVT) aralarında çeşitli bağıntılar, özellikler ve bu KVT üzerlerinde tanımlı olan bazı matematiksel işlemler mevcuttur. KVT 'ler geometrik varlıkların yapısını modellemek için kullanılırlar. Bu amaçla KVT 'lerin genel bir tanımlanması (soyutlanması) gereklidir. KVT 'lerin tanımlanması ve elde edilmesi konumsal veritabanı sistemlerinin elde edilmesinde belkide en önemli aşama olmaktadır.

Konumsal veri türü sistemlerini ya da bir başka deyişle konumsal cebirleri sorgulama diline gömen yani sorgulama dilinin matematiksel ifadesi olan ilişkisel cebir ile uyumlu olan bazı tasarımlar geliştirilmiştir. Bu konuda detaylı bilgi için Güting 'in ([11], [14]) yayınlarına bakılabilir. Bu tasarımlardan biri ve en önemlisi olan yine Güting [11] tarafından geliştirilmiş olan **ROSE** cebiridir. ROSE (RObust Spatial Extension) Cebiri bilgisayarlı hesaplamayı kolaylaştırmak için geliştirilmiş **realm** [12] veri yapılarını taban almıştır. realm' ler tüm veri yapılarını bir ızgaraya oturtan ve veri değerlerinin bu ızgara noktası dışında bir nokta olmasını engelleyen koordinat ve veri türleri tanımıdır. Ayrıca ROSE Cebiri ikinci-mertebenden forma **(second-order signature)** [13] gösterim modelini kullanmaktadır. Aslınada Rose Cebiri konumsal bir sorgulama dilinin matematiksel modelidir.

Veri yapılarının aralarındaki ilişkileri tıpkı klasik matematikteki toplama çıkarma işlemleri gibidir. Bu bakımından 4-kesişim modeli [9] geliştirilmiştir. 4-kesişim modeli nokta kümesi topolojisini [9] taban almakta ve geometrik nesnelerin içlerini ve sınırlarını karşılaştırarak sınıflandırma yapmaktadır. 9-kesişim modeli ise geometrik nesnelerin dışlarını da kullanan konumsal veri yapılarının aralarındaki geçerli ilişkileri tanımlayan 4-kesişim modelinin bir genişletilmesidir.

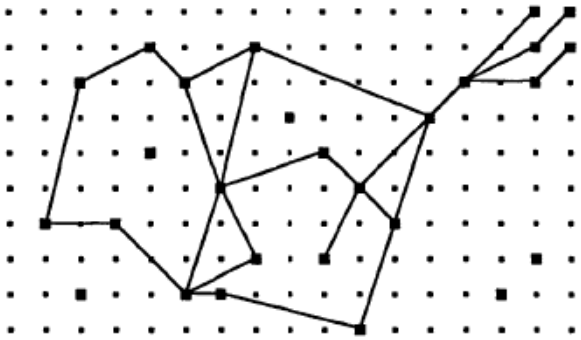
2. KONUMSAL VERİ YAPILARI VE KONUMSAL CEBİR

Konumsal veri yapıları ya da konumsal cebirler, nokta, doğru ve bölgenin ana yapılarını ve aralarındaki ilişkileri yürütebilecek sistemlerdir. Konumsal cebirde en önemli örnek ROSE Cebiridir [11] . Genel olarak bir KVT 'nün aşağıda belirtilen özellikleri sağlaması istenir:

- *Genel olmalıdır:* KVT olarak kullanılan geometrik nesneler mümkün olduğunca genel olmalıdır. Örneğin, bir bölge değeri delik içeren ya da içermeyen ayrık alanların kolleksiyonu şeklinde yazılabilir olmalıdır. Başka bir deyişle, *point, line* ve *region* gibi veri türlerinin tanım kümesi birleşim, kesişim ve fark işlemleri altında kapalı olmalıdır.
- *Özenli tanımlanmalıdır:* KVT 'lerin semantik yapısı kullanıcı ve geliştirici için ne olduğu belirsiz tür ve fonksiyonlardan arındılmış olmalıdır.
- *Sonlu çözünürlük özelliğine sahip olmalıdır:* Biçimsel tanımlamalar bilgisayarda kullanılabilecek şekilde sonlu gösterime sahip olmalıdır.
- *Geometrik tutarlı olmalıdır:* KVT 'ler geometrik olarak tutarlı olmalıdır.
- *Genel nesne model arayüz özelliğini sağlamalıdır:* KVT tanımı VTYS veri modeli soyut arayüzüne dayanmalıdır.

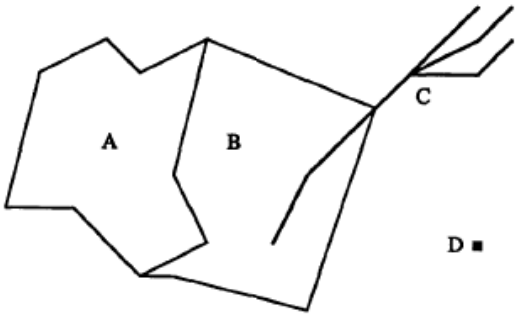
İşte realm' lar yukarıdaki özellikleri sağlayan üzerinde bazı işlemlerle tanımlı olan bir soyut KVT 'dür. Amaç realm kavramına dayanan bir Veri Tabanı Yönetim Sistemi (VTYS) tanıtmaktır. realm genel olarak bir ya da daha fazla veri türleri sistemi için taban olan sonlu, kullanıcı tanımlı bir yapıdır. realm 'lar programlama dillerindeki numaralandırılmış türlere (enum değerler) benzerler. ROSE Cebiri realm uzay yapısını temel alan bir KVT ve onların üzerinde tanımlı olduğu işlemler sistemidir. realm iki boyutlu uzayda özel bir uygulama alanının içinde yatan geometriyi tanımlayan bir kavramdır. Daha genel olarak realm Şekil 2.1 de görüldüğü gibi ayrık bir ızgara üzerinde tanımlı noktalar ve doğru parçalarının sonlu kümesidir. Bir realm 'in özelliği ayrık bir alan içinde noktalar kümesi ve kesişmeyen doğru parçacıklarından oluşan Şekil 2.1'deki gibi bir ızgaradır [12] . realm için bazı kurallar bulunmaktadır:

1. Her nokta ya da her doğru parçasının bitim noktaları bir ızgara noktasıdır.
2. Her realm doğru parçasının bitim noktası da yine realm noktadır.
3. Herhangi bir realm noktası, bitim noktaları hariç bir realm doğru parçasının üzerinde bulunamaz.
4. Hiçbir realm doğru parçası bitim noktaları dışında kesişmez.



Şekil 2.1 Bir Realm örneği.

Amaç realm nokta ya da realm doğru parçası gibi realm ilkeller ve bunların bileşkeleri kullanarak konumsal nesnelerin geometrisini oluşturmaktır. Şekil 2.1 de, bu yolla tanımlanabilecek bazı konumsal nesneler gösterilmektedir.



Şekil 2.2 Şekil 2.1'deki realm örneğinden tanımlanan nesneler.

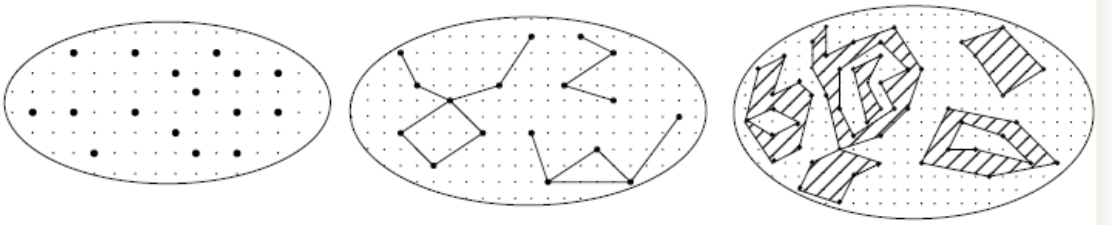
Realm'in altındaki ızgara basit olarak sayıların bilgisayar hafızasında sonlu sayıda temsilleri olduğu gerçeğinden doğmuştur. Pratikte bu temsiller sabit uzunlukta olur ve programlama dillerinde bulunan INTEGER veya REAL veri türlerine (ya da daha özel olarak, sayı sistemlerine) karşılık gelmektedir.

KVT 'lerin temeli olarak realm kavramı aşağıdaki belirtilen özelliklere sahiptir:

- realm'ler ilişkili olduğu konumsal nesne ile geometrik olarak tutarlıdır. Örneğin, A ve B gibi iki ülkenin sınırlarının ortak kısımları hem konumsal nesne için ve hem de realm için aynıdır.
- realm'ler üzerindeki işlemlerle kapalılık özelliği sağlanır. Örneğin, B bölgesinin C doğrusu ile kesişimi (burada C doğrusu B ülkesinin içinde olan bir nehir parçasını gösterebilir) yine realm tabanlı bir doğrudur.
- realm'lar bizi sorgulama sürecindeki geometrik hesaplamalar esnasında ortaya çıkabilecek sayısal doğruluk ve algoritmaların sağlamlığının bozulması gibi problemlerden korur. Örneğin, doğru parçalarının kesişim noktalarının hesabı normalde ızgara üzerinde bulunmayabilir. Ama realm 'lara dayanan KVT 'leri ile sorgulama yapıldığında henüz hesaplanmamış yeni bir kesişim noktası ortaya çıkma ihtimali yoktur. Bunun yerine, realm düzeyinin altında sayısal problemler yerine getirilir.

2.1. ROSE Cebiri

ROSE cebiri değerleri realm-tabanlı olan ve adına *points, lines* ve *regions* denilen üç adet veri türünü kullanıma sunar.



Noktalar **Doğru Parçaları** **Bölgeler**

Şekil 2.3 : Konumsal veri türü örnekleri

Konumsal nesnelerin yapısı bir *R-point,* bir *R-block* ve bir *R-face* notasyonları türünden tanımlanır. Verilen bir realm için, bir *R-point* R noktasıdır. Bir *R-block* R yi oluşturan doğru parçalarının bağlantılı kümesidir. Bir *R-face* realm doğru parçalarıyla oluşturulan delikler içerebilen bir poligondur. Bu halde *point* türünden değer *R-point* 'ların bir kümesi, *line* türünden değer ayrık *R-block* 'ların bir kümesi ve *region* türünden bir değer kenar-ayrık (yani iki yüzün ortak bir köşeye sahip olabildiği durum) *R-face* 'lerin bir kümesidir.

Konumsal işlemlerin dört ayrı sınıfı vardır:

1. Topolojik ilişkiyi gösteren konumsal yüklem (ör. ***inside, intersect, meets*** gibi)
2. Atomik konumsal veri türü döndüren işlemler (ör. ***intersection, plus, minus, contour*** gibi)
3. Sayısal değer döndüren konumsal işlemler (ör. ***dist, perimeter, area*** gibi)
4. Veritabanı nesnelerinden oluşan kümeler üzerinde tanımlı konumsal işlemler (ör. ***sum, closest, fusion*** gibi).

ROSE cebiri dilin değişkenleri ve ifadeleri nasıl sınıflandıracağını tanımlayan ikinci-mertebenden forma [11] tür sistemini kullanmaktadır. İkinci-mertebeden forma sistemi cebir ile birlikte tür sistemi üzerinde yeni bir tür sistemi

tanımlamamıza izin verir [13] . Özellikle bu *kinds* (bu bir şekilde tür sistemidir) 'lar üzerinde polimorfik işlemleri tanımlamamıza izin verir. Basit bir örnek aşağıda verilmektedir. İki küme *EXT* = {*lines, regions*} ve *GEO* = {*points, lines, regions*} olsunlar. İşlemlerin dört farklı sınıflandırılması olabilir:

1) Topolojik ilişkiyi gösteren konumsal yüklem:
   $\forall geo$ in GEO.$\forall$ *ext*1, *ext*2 in EXT.$\forall$ *area* in $regions^{area\text{-}disjoint}$.
   geo × *regions* → *bool* **inside**
   *ext*1 × *ext*2 → *bool* **intersects**, **meets**
   area × *area* → *bool* **adjacent**, **encloses**

   Burada tür değişkeni *geo* çeşit GEO kümesi içinde yer alan üç tür türünden biri olabilir. Bu sebeple ***inside*** işlemi *points, lines* ya da *regions* değerlerini bir *region* değeri ile karşılaştırabilir**.** ***intersect*** işlemi çeşit EXT kümesi içinde yer alan farklı türlerden herhangi ikisine uygulanmaktadır. $region^{area\text{-}disjoint}$ notasyonu tür sisteminde bölüntülerin yapısını yakalamak içindir. Bu tüm bölüntüler için bir çeşit tanımlar, herbir bölüntü bu türden bir çeşit içinde değerleri bölüntü içinde kalan bölgeler olan bir türdür. Böylece *area* tür değişkeni tek bir bölüntüyü alabilir ve ***adjacent*** işlemi bölüntünün herhangi iki bölgesine uygulanabilir.

2) Atomik KVT değerlerini döndüren işlemler:
   $\forall$ geo in GEO.
   lines× lines → points **intersection**
   regions × *regions*→ *regions* **intersection**
   geo × *geo* → *geo* **plus**, **minus**
   regions → *lines* **contour**

   Burada ***plus, minus*** aynı türden iki değer için birleşim ve fark işlemlerini göstermektedir.
3) Sayı döndüren konumsal işlemler:
   $\forall$ geo1 × geo2 in GEO.
   geol × *geo2* → *real* **dist**
   regions → *real* **perimeter**, **area**
4) Nesne kümeleri üzerinde tanımlı konumsal işlemler:
   $\forall$ *obj* in OBJ. $\forall$ *geo*, *geo*1, *geo*2 in GEO.
   set(obj) × (*obj* → *geo*) → *geo* **sum**
   set(obj) × (*obj* → *geo*1) × *geo*2 → *set(obj)* **closest**

   Burada ***sum*** işlemi "*konumsal toplam*" işlemini göstermektedir. Bu *geo* türünden nesnelerin bir konumsal yüklemiyle birlikte nesnelerin bir kümesini alır ve geriye tüm niteliksel değerlerin geometrik birleşimini döndürür. Örneğin, bu bir ülkenin alanını hesaplamak amacıyla o ülkede bulunan tüm iller kümesinin birleşimi olabilir. ***closest*** işlemiyse bir nesneler kümesi içinde olan ve konumsal nitelik değeri diğer geometrik nesnelerden en az uzaklıkta bulunan nesneyi döndürür.

3. KONUMSAL İLİŞKİLER ve İLİŞKİSEL OPERASYONLAR

Konumsal cebir'in sağladığı işlemler arasında en önemlilerinden biri konumsal ilişkilerdir. Örneğin, bir sorgulama nesnesi aracılığıyla verilen bir ilişkinin içerdiği tüm nesneler istenebilir (fare ile çizilen bir pencerenin içindeki tüm nesneler gibi). Bazı araştırmacılar bu ilişkileri sınıflara ayırmışlardır (Egenhofer, 1989 [7] ; Worboys, 1992 [17]):

- ***Topolojik ilişkiler****:* öteleme, derecelendirme ve döndürme gibi topolojik dönüşümler altında değişmez kalan *komşuluk, içerme* ve *ayrık olma* bağıntıları birer topolojik ilişkidir.
- ***Yön ilişkileri****: yukarı, aşağı, kuzey-doğu, güney-batı* gibi ilişkiler birer yön bağıntısıdır.
- ***Metrik ilişkiler****: uzaklık> 100 km* gibi bir ilişki sınıfına örnek olabilir.

Geometrik nesneler arasındaki ilişkileri tanımlamak aralarındaki geçerli olan işlemleri tanımlamak anlamına gelmektedir. İlişkiler ve işlemler iki geometri arasında belirtilen topolojik konumsal ilişkinin varlığını test etmek için kullanılır. İki geometrik nesne arasındaki topolojik konumsall ilişki literatürde geniş kapsamlı olarak çalışılmıştır ([8] , [9] , [1] , [2], [3]) . İki geometrinin karşılaştırılmasını içeren bu temel yaklaşım, iki geometrinin *İçleri, Sınırları* ve

Dışları arasındaki kesişmelerin ikili testlerini gerçekleştirmek ve sonuçta elde edilen "kesişim" matrisindeki girdilere dayalı iki geometri arasındaki ilişkiyi sınıflandırmaktır.

İki geometrinin aralarındaki ilişkileri geometrinin iç ve sınır değerlerinin etkileşimi olarak değerlendirip aralarındaki ilişkileri sınıflandırmak için 4-Kesişim Modeli tanımlanmıştır [9] . Bu model sonradan, geometrinin dışınıda değerlendirecek şekilde genişletilmiş, bu da bir 9-Kesişim Modeli ile sonuçlanmıştır [8] Bu model, delikli alanlar ve çoklu bileşenli çizgiler ve alanlar dahil olmak üzere, noktalar, çizgiler ve alanlar arasındaki konumsal ilişkilerin ifade edilmesine imkân tanır.

3.1. 4-Kesişim Modeli

4-Kesişim Modeli Egenhofer [9] tarafından iki geometrik nesne arasındaki kesişimleri değerlendirip ilişkileri sınıflandırmak için geliştirilmiştir. İki nesnenin iç ve sınır değerleri bu modelin temel aldığı nesne özellikleridir. Bir A geometrik nesnesinin içini A^o ve sınırını ∂A göstermek üzere bu iki nesne için aralarında dört adet ilişki kümesi bulunmaktadır bunlar: $\partial A_1 \cap \partial A_2$, $\partial A_1 \cap A_2^o$, $A_1^o \cap \partial A_2$ ve $A_1^o \cap A_2^o$ ilişkileridir. Bu durumda karşımıza
Tablo **33**' deki gibi 2^4=16 farklı olasılık ortaya çıkmaktadır.

$\partial A_1 \cap \partial A_2$	$\partial A_1 \cap A_2^o$	$A_1^o \cap \partial A_2$	$A_1^o \cap A_2^o$	İlişki
Ø	Ø	Ø	Ø	A_1 ve A_2 ayrık
Ø	Ø	Ø	≠Ø	
Ø	Ø	≠Ø	Ø	
Ø	Ø	≠Ø	≠Ø	A_2 , A_1 'in içinde
Ø	≠Ø	Ø	Ø	
Ø	≠Ø	Ø	≠Ø	A_1 , A_2 'in içinde
Ø	≠Ø	≠Ø	Ø	
Ø	≠Ø	≠Ø	≠Ø	
≠Ø	Ø	Ø	Ø	A_1 , A_2 'ye değer
≠Ø	Ø	Ø	≠Ø	A_1 ve A_2 eşittir
≠Ø	Ø	≠Ø	Ø	
≠Ø	Ø	≠Ø	≠Ø	A_1, A_2 'yi kapsar
≠Ø	≠Ø	Ø	Ø	
≠Ø	≠Ø	Ø	≠Ø	A_2, A_1 'yi kapsar
≠Ø	≠Ø	≠Ø	Ø	
≠Ø	≠Ø	≠Ø	≠Ø	A_1 , A_2 'yi örter

Tablo 33 4-Kesişim Modelindeki 16 değişik olasılık

Bu olasılıkların 8 tanesi geçersiz, iki tanesi simetriktir. Geriye altı ayrı ilişki kalmaktadır bunlar da *ayrık*, *içinde*, *değen*, *eşit*, *kapsar* ve *örtüşür* olarak adlandırılmaktadır.

3.2. 9-Kesişim Modeli

Max J. Egenhofer [8] tarafından geliştirilmiş bu modelin amacı iki boyutlu cisimlerin etkileşimlerini formalize etmektir. Bu model temel olarak 4-Kesişim Modeline, geometrik nesnelerin dışının ilave edilmesidir. A ve B gibi iki nesne arasındaki bir ikili topolojik ilişki R olmak üzere ve A ile B geometrik nesnelerinin içi, dışı ve sınırı sırasıyla A^o, ∂A, A^- , B^o, ∂B ve B^- olmak üzere 9-Kesişim Matrisi aşağıdaki 3x3 lük matris ile verilir:

$$R(A,B) = \begin{pmatrix} A^o \cap B^o & A^o \cap \partial B & A^o \cap B^- \\ \partial A \cap B^o & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^o & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

9-Kesişim matrisinin her bir sonucu değişik bir topolojik ilişkiyi ifade eder. Matristeki her bir kesişimin sonucu ya boş (Ø) ya da boş değil (¬∅) değerlerinden oluşur. Örneğin, A bölgesi B bölgesi kapsıyorsa bunun dokuz kesişim matrisindeki karşılığı

$$R(A,B)=\begin{pmatrix} A^{\circ}\cap B^{\circ}=\neg\varnothing & A^{\circ}\cap\partial B=\varnothing & A^{\circ}\cap B^{-}=\varnothing \\ \partial A\cap B^{\circ}=\neg\varnothing & \partial A\cap\partial B=\neg\varnothing & \partial A\cap B^{-}=\varnothing \\ A^{-}\cap B^{\circ}=\neg\varnothing & A^{-}\cap\partial B=\neg\varnothing & A^{-}\cap B^{-}=\neg\varnothing \end{pmatrix}$$

ya da kısaca

$$R(A,B)=\begin{pmatrix} \neg\varnothing & \varnothing & \varnothing \\ \neg\varnothing & \neg\varnothing & \varnothing \\ \neg\varnothing & \neg\varnothing & \neg\varnothing \end{pmatrix}$$

matrisi olmaktadır.

3.2.1. 9-Kesişim Modelinde Topolojik Özellikler:

İki nesne arasındaki topolojik özellikler 9-kesişim modeli ile ifade edilebilir. Değerinin ne olduğu önemli olmayan kesişimler (-) sembolü ile ifade edilir.

a_i ve b_i , A ve B nesnelerinden rasgele seçilmiş parçalar olsun.

- Eğer a_i parçası b_i parçasından ayrık ise, iki parçanın kesişimleri boş olmalıdır. Bu halde diğer kalan 8 kesişim önemsiz değerler olabilir. Örneğin, A nın sınırları B nin içinden ayrık ise, A ile B arasındaki 9 kesişim şu kalıpta olmalıdır:

$$R\{\varnothing,\neg\varnothing\}(A,B)=\begin{pmatrix} - & - & - \\ - & - & \varnothing \\ - & - & - \end{pmatrix}$$

- Eğer a_i parçası b_i parçasını kesiyorsa, bu iki parçanın kesişimleri boş olmamalıdır. Örneğin, A nın içi B nin sınırlarını kesiyorsa A ile B arasındaki 9 kesişim şu kalıpta olmalıdır:

$$R\{\varnothing,\neg\varnothing\}(A,B)=\begin{pmatrix} - & \neg\varnothing & - \\ - & - & - \\ - & - & - \end{pmatrix}$$

- Eğer a_i parçası b_i parçanın alt kümesi ise, bu iki parçanın arasındaki kesişim boş olmamalı, bununla birlikte a_i ve diğer iki parçanın (b_k ve b_l) kesişimleri boş olmalıdır. Örneğin, A nın sınırları B nin içinin alt kümesi ise, A ile B arasındaki 9 kesişim şu kalıpta olmalıdır:

$$R\{\varnothing,\neg\varnothing\}(A,B)=\begin{pmatrix} - & - & - \\ \neg\varnothing & \varnothing & \varnothing \\ - & - & - \end{pmatrix}$$

- Eğer a_i be b_j parçaları için $a_i \not\subset b_J$ ve $a_i \not\subset b_k$,(j ≠ k) olduğunda a_i parçası hem b_J parçasının ve hem de b_k parçasının alt kümesi ise, b_k , a_i ve B'nin üçüncü bir parçası arasındaki kesişim boş iken, bu iki parça arasındaki kesişim boş olmamalıdır. Örneğin, $\partial A \not\subset \partial B$ ve $\partial A \not\subset B^{o}$ olacak şekilde eğer $\partial A \subseteq (\partial B \cup B^{o})$ ise, Şekil 3.1 'deki A ve B 'nin arasındaki 9 kesişim şu kalıpta olmalıdır.

Şekil 3.1 A ve B alanlarının kesişimi

$$R\{\varnothing,\neg\varnothing\}(A,B)=\begin{pmatrix} - & - & - \\ \neg\varnothing & \neg\varnothing & \varnothing \\ - & - & - \end{pmatrix}$$

- Eğer a_i ve b_i nesne parçaları çakışıyorlarsa aralarındaki kesişim boş olmamalıdır, ama bu arada diğer dört kesişim boş olmalıdır. Örneğin, eğer A ve B'nin sınırları çakışıyorsa, A ve B 'nin arasındaki 9 kesişim şu kalıpta olmalıdır.

$$R\{\varnothing, \neg\varnothing\}(A,B) = \begin{pmatrix} - & \varnothing & - \\ \varnothing & \neg\varnothing & \varnothing \\ - & \varnothing & - \end{pmatrix}$$

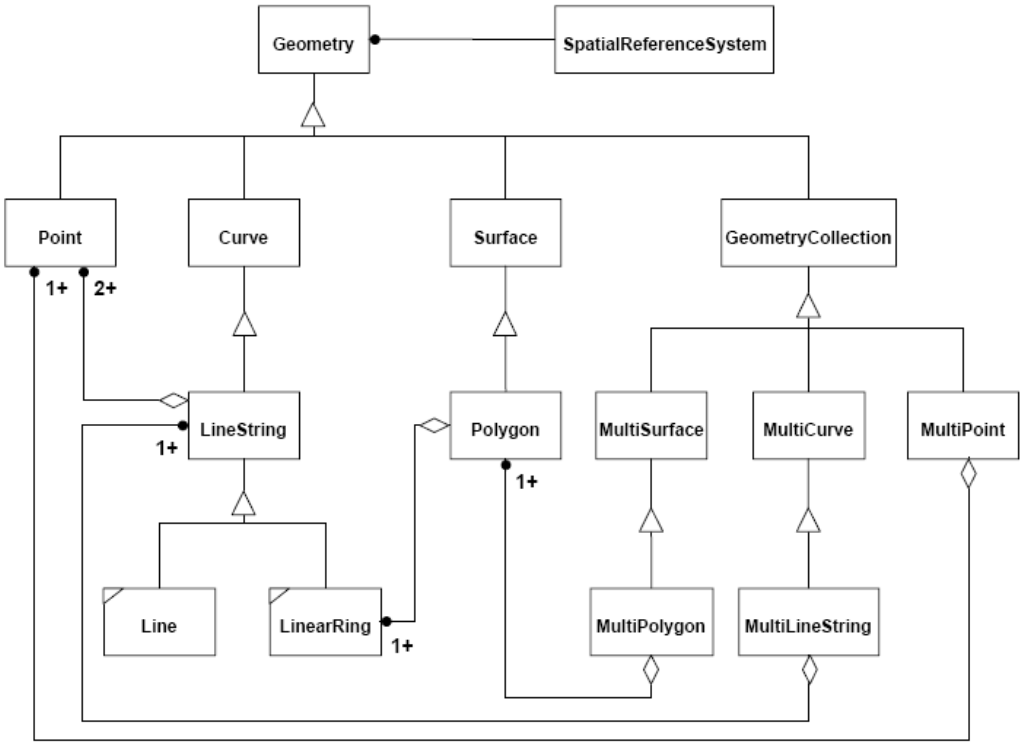
4. GEOMETRİK SINIF KÜTÜPHANESİ

Konumsal verileri işleyebilme yeteneği geleneksel Veri Tabanı Yönetim Sistemlerinde (VTYS' de) bulunmamaktadır. Bir coğrafi bilgi sistemi geliştirmek istendiğinde firmaların bir kısmı kendi veritabanı yönetim sistemlerini geliştirmek yolunu seçmektedirler. Bu zahmetli ve çoğunlukla da geleneksel VTYS'ler kadar kararlı ve güvenilir olmamaktadır. Bunun nedenleri açıktır; geleneksel veritabanı yönetim sistemleri (Oracle, MS SQL) bilgi teknolojileri ile birlikte paralel gelişmiş ve yılardan beri belirli bir kararlılığa ulaşmış sistemlerdir. Gerek güvenilirlik, gerek hız ve gerekse kararlılık yönlerinden geleneksel sistemlere yaklaşabilecek veri yönetim sistemleri geliştirmek zordur. Ayrıca esas amaçlanan veritabanı yönetim sistemi yapmak değil de coğrafi bilgi sistemi yapmak olunca, vakit ve zaman kaybı ciddi düzeylere ulaşmaktadır.

Bu sebeplerden dolayı var olan geleneksel bir veritabanı yönetim sisteminin üzerinde çalışacak bir ara katman geliştirilme yolu birçok CBS için daha uygun bulunmaktadır. Bunu başarmanın ilk şartı ise konumsal sorgulama yapabilecek bir SQL dilinin geliştirilmesidir.

Konumsal SQL dilinin sahip olması gereken özellikleri OGC (Open Geospatial Consortium) tarafından *Open GIS Consortium, Inc. OpenGIS Simple Features Specification For SQL Revision 1.1* [16] adlı yayımlanan belgede açıkça belirtilmektedir. Buna göre konumsal bir sorgu dilini işlevsel hale getirebilmek için iki temel aşamadan geçilmelidir. Birincisi sorguyu ayıklayıp, konumsal ve konumsal olmayan sorguları birbirinden ayırıp sonuçları ilişkisel cebir ile birleştirmek, ikincisi ise ayıklanan konumsal sorguları işlemek üzere bir sınıf kütüphanesi oluşturmaktır. Bu sınıf kütüphanesinin temel yapısı yine Open GIS Consortium tarafından aynı adlı belgede [16] verilmiştir.

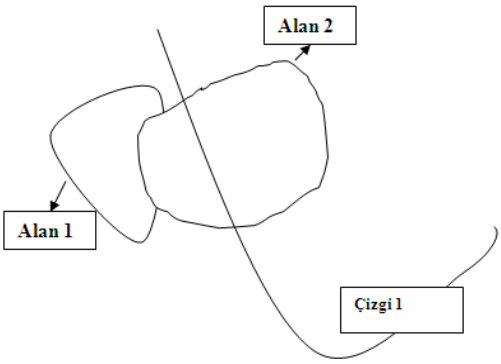
Bu Geometrik Sınıf Kütüphanesi tek başına bir şey ifade etmeyebilir. Dış dünyadan gelen konumsal sorguları, konumsal ve konumsal olmayan sorgular olarak birbirlerinden ayrılmalı daha sonra da bu kütüphane aracılığıyla konumsal sorgular işlenmeli ve son olarak da sonuç kümeleri birleştirilmelidir.



Şekil 4.1 ***Open GIS Consortium, Geometrik Sınıf Kütüphanesi***

OGC yine konumsal sorgu standartlarının ortaya konulduğu gibi nesne modeli de açıkça belirtimiştir [15] (Şekil). OGC bu kütüphaneyi tamamen nesneye dayalı tasarlamıştır. Bunun nedeni tüm geometrik veri yapılarının aslında bir *geometry* olmasını yani birbirlerine dönüşebilir olabilmelerini sağlamaktır. Gerçekten de bu gerek diğer kaynaklardan gelen verilerin türünün önceden bilinememesi, gerekse veri tabanındaki verilerin tür dönüşümleri açısından bu çok faydalı olabilmektedir.

OGC konumsal veri yapısını açıkça göstermiştir. Bu sınıf yapısında her nesne kendi türünün özeliklerini tutmakla sorumlu olduğu gibi, kendi türünden ya da farklı türlerden nesneler ile etkileşimlerinden de sorumludur. Bu etkileşimlere örnek vermek gerekirse:



Şekil 4.2 Değişik Geometrik Şekiller ve Etkileşimleri

Şekil 4.2 ‘de üç adet geometrik şekil görünmektedir. Bu geometrik şekillerden *Çizgi 1* nesnesi *Alan 1* ve *Alan 2* ile etkileşmektedir. Bu etkileşim tipinin ve koordinatlarının tespit edilmesi gerektiğinde Geometrik Sınıf Kütüphanesine başvurulacaktır.

Geometrik Sınıf Kütüphanesi OGC tarafından standartlaştırılmadan önce birçok araştırmacı tarafından çeşitli araştırmalar yapılmıştır ve bu çalışmaların sonucunda bu sınıf kütüphanesi çatısı ortaya çıkmıştır.

Bu sınıf kütüphanesinin amacı geometrik verileri sorgulamak ve işlemektir. Konumsal verilerin sorgulanabilmesi için öncelikle geleneksel verilerin sorgulama dili olan SQL taban alınmış ve SQL’ e çeşitli eklentiler tanımlanmıştır [5] . SQL dilinin matematiksel ifadesi ilişkisel cebir olduğundan konumsal sorgulama dilinin de bir konumsal cebire ihtiyacı vardır. İşte bu noktada ROSE cebirinin gerekliliği ortaya çıkmaktadır.

Veri yapılarının aralarındaki ilişkilerinin tıpkı matematikteki alışılmış toplama çıkarma işlemleri gibi formalize edilmesi gerekliliği gözden kaçırılmamalıdır. Öncelikle Dört Kesişim Modeli [9] geliştirilmiştir. Daha sonra nesnelerin tamlayanlarının (iç ve sınırının dışında) da bu sınıflandırmada gerektiği ortaya çıkmış ve Egenhofer Dokuz Kesişim Modelini geliştirmiştir [6] . Bilindiği gibi dokuz kesişim modeli konumsal veri yapılarının aralarındaki geçerli ilişkileri tanımlamaktadır

Bu noktadan sonra karşımıza konumsal verilerin geleneksel veritabanlarında nasıl tutulacağı sorunu çıkmaktadır. Bir geometrik nesne için ne kadar veri tutulacağı önceden kesin olarak bilinememektedir. Bu sebeple Şekil 4.3 deki gibi klasik tablo oluşturup verileri onun içine atma yaklaşımı konumsal veriler için uygun değildir.

Orders : Table

OrderID	total	dateordered
2	$3.00	1/1/2002
3	$4.00	1/10/2002
4	$5.00	2/3/2002
5	$6.00	2/6/2002
6	$7.00	2/13/2002
7	$9.00	3/2/2002
8	$3.00	4/18/2002

Record: 1 of 23

Şekil 4.3 Geleneksel bir veri tabanı tablosu örneği

Bu soruna yine OGC bir çok CBS geliştiricisi ile birlikte geometrik verileri tek bir tablo alanında toplama çözümünü getirmiştir. Konumsal veri türleri örneklerinin sayısı önceden bilinmeyen bir sistemde bu verilerin tek bir tablo alanında tutulmasını sağlayan standardın adı ise [15] Well-Known Text'dir (WKT). WKT geometrik verilerin Şekil 4.4 'deki gibi (örneğin, Ankara İl Sınırı, Tuz Gölü'nün Topolojik Şekli, binaların koordinatları, haritada bir nokta, vb. gibi) tek bir yalın katar şeklinde tutulmasını sağlamaktadır.

GID	Xmin	Ymin	Xmax	Ymax	WKT_Geometry
1	5	5	3	6	LINESTRING(0 0, 10 10, 20 25, 50 60)
1	-	-	-	-	POINT(15 20)

Şekil 4.4 WKT kullanan konumsal bir veri tabanı örneği

Yukarıdaki nedenlerden dolayı WKT' lerden geometrik nesneler oluşturalacak bir yordamlar kümesinin Geometrik Sınıf Kütüphanesi ile uyumlu bir sürümünün yazılması gerekliliği ortaya çıkmıştır.

Ayrıca geometrik sınıf kütüphanesinin ihtiyaç duyduğu birçok geometrik algoritmanın gerçekleştirilmesi de gerekmektedir (örneğin; dışbükey kabuk bulma algoritmaları [10], nokta poligon içindemi? [4] vb. gibi).

5. SONUÇ

Tüm bu çalışmaların sonucunda geometrik sınıf kütüphanesi standartları ortaya çıkmıştır. Bu standartları ortaya çıkaran ve değinilen çalışmaların anlaşılması ve özümsenmesi, sınıf kütüphanesinin geliştiricisi için vazgeçilmez bir ön koşuldur. İleride Geometrik Sınıf Kütüphanesi Standartlarının değişmeyeceği garanti edilemez, araştırmacılar daha iyi modeller ortaya atabilirler, ama elbette ki bu modeller eski modelleri taban alacak modeller ve teoremler olacaktır. Bu durumda hangi modeller, veri yapıları ve nesne uzayları üzerine geliştirilme yapıldığının iyi anlaşılması sadece geometrik sınıf kütüphanesi geliştiricileri için değil, tüm CBS geliştiricileri için bir şarttır.

KAYNAKLAR

[1] **CLEMENTINI E.; DI FELICE P**., (1994) A Comparison of Methods for Representing Topological Relationships, *Information Sciences* 80, (1-34)

[2] **CLEMENTINI, E.; DI FELICE, P.,** (1996) A Model for Representing Topological Relationships Between Complex Geometric Features in Spatial Databases, *Information Sciences* 90 (121-136)

[3] **CLEMENTINI, E.; DI FELICE, P.; VAN OOSTEROM**, **E**, (1993) A small set of formal topological relationships suitable for end-user interaction. *Proceedings of the Third International Symposium on Large Spatial Databases, Singapore.*

[4] **CROMLEY, R. G**., (1992), Digital Cartography, *Prentice Hall, New Jersey, 316p.*

[5] **EGENHOFER, M.,** (1994) Spatial SQL: A query and presentation language. *IEEE Transactions on Knowledge and Data Engineering,* 6 (86-95)

[6] **EGENHOFER, M.**,(1989) A formal definition of binary topological relationships. *Proceedings of the Third International Conference on the Foundations of Data Organization and Algorithms, Paris.*

[7] **EGENHOFER, M. J.; FRANK, A.; JACKSON, J.P**., (1989) A topological data model for spatial databases. *Proceedings of the First International Symposium on Large Spatial Databases, Santa Barbara.*

[8] **EGENHOFER, M. J.; HERRING, J. R.** (1991) Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases. *Technical report, Department of Surveying Engineering, University of Maine, Orono, ME..*

[9] **EGENHOFER, M. J.; FRANZOSA, R. D.**, (1991) `Point-Set Topological spatial relations', *International Journal for Geographical Information Systems* 5(2),(161—174)

[10] **GRAHAM R. L.,** (1972) An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. *Information Processing Letters,* 1 (132-133*)*

[11] **GÜTING R. H.; M. SCHNEIDER.,** (1995) Realm-Based Spatial Data Types: The ROSE Algebra. *VLDB Journal,* 4(2), (243-286)

[12] **GÜTING, R. H.; SCHNEIDER, M.**,(1993a) Realms: A foundation for spatial data types in database systems. *Proceedings of the Third International Symposium on Large Spatial Databases, Singapore.*

[13] **GÜTING, R. H.,** (1993) Second-order signature: A tool for specifying data models, query processing, and optimization. *Proceedings of the ACM SIGMOD Conference, Washington, DC.*

[14] **GÜTING, R. H**., (1994) An Introduction to Spatial Database Systems. *VLDB Journal 3, 4 (Special Issue on Spatial Database Systems)*, (357-399)

[15] **OGC REFERENCE MODELL**, (2005) Open Geospatial Consortium Inc.; *Date: 2003-09-16; Reference number: OGC 03-040 ; Version: 0.1.3,* http://www.opengis.org (14.01.2005).

[16] **OPEN GIS CONSORTİUM, INC.** (1999) *OpenGIS Simple Features Specification For SQL Revision 1.1,* http://www.opengis.org (05.02.2005).

[17] **WORBOYS, M. E.,** (1992) A generic model for planar geographical objects. *International Journal of Geographical Information Systems*, 6, (353-372)