

GELENEKSEL KONUMSAL VERİ DEĞİŞİMİNİN SORUNLARI VE FME YAZILIMININ SUNDUĞU YENİ OLANAKLAR

Çetin CÖMERT
Halil AKINCI

ÖZET

Konumsal Veri Yönetiminde (KVY) hızlı, sağlıklı ve ekonomik çözümler üretilebilmesi için konumsal verinin “etkin” bir biçimde paylaşımı, bugün her zamankinden daha çok kaçınılmaz olmuştur. Bunun nedeni, günümüz uygulamalarının çok çeşitli türde veri gerektirmesi, gerekli verinin ilk elden toplanmasının çok pahalı, üstelik çoğu durumda, sayısal formda dış kaynaklardan sağlanabilir olmasıdır. Ancak KVY de “veri paylaşımının” geleneksel uygulanış şekli olan Konumsal Veri Değişimi (KVD), bugün hala istenen düzeyde gerçekleştirilememektedir. Bunun nedeni, kullanılan iki yöntemin de sorunlu olmasıdır; İki format arasında “doğrudan” değişim yöntemi, eksiksiz bir değişim sağlamakla birlikte, kullanımdaki formatlara yapılacak olası değişikliklerden doğrudan etkilenmesi nedeniyle pratik değildir. İki format arasında bir “ara” format üzerinden veri değişimi yöntemi ise, bu ara formatın çok çeşitli formatlardaki veri temsillerini kapsayacak kadar “genel” fakat aynı zamanda “özel” olmasını gerektirmesi nedeniyle sorunludur. Özetle, bugün pratikte uygulanan şekliyle KVD basit, hızlı ve eksiksiz transferlere olanak tanımamaktadır. Bu çalışmada, pratikte uygulanan şekli ile KVD nin sorunları incelenmiş, 1998 yılı sonlarında piyasaya sürülen FME (Feature Manipulation Engine) yazılımının bu sorunlara ne derece çözüm getirdiği irdelenmiştir. Değerlendirme, belirlenen bir dizi ölçüt ve farklı Coğrafi Bilgi Sistemi (CBS) formatlarındaki test verileri kullanılarak yapılmıştır. Sonuçta FME yazılımının geleneksel KVD de yaşanan sorunlar için oldukça geçerli çözümler sunduğu, ancak “ideal” bir KVD ortamına kıyasla bir takım eksiklikleri olduğu belirlenmiştir.

ABSTRACT

It has long been realized that timely and economical solutions are only possible through an “effective” sharing of data. This is much strongly felt today than ever before. This is because of a number of reasons; First, Today’s challenging applications require data from many different sources. Second, collecting spatial data first hand is a costly and time consuming operation. And third, in most of the cases, digital data required for an application could be gathered from external sources. Nevertheless, Spatial Data Interchange (SDI), the traditional way of spatial data sharing, has always been problematic. This is due to the problems of the methods used; The “direct” method, where the transformation is performed directly between the two formats of interchange, is not practical since it would directly be affected from highly likely modifications of the formats in use. And the “indirect” method, where the transformation is carried out over an “intermediary” format, has long suffered the difficulty of comprimising many different formats while at the same time enabling dedicated translations. In brief, SDI, as it is performed today, can not be effected at the desired level. In this work, issues of traditional SDI have been examined first. For the solutions to these issues, FME (Feature Manipulation Engine), a spatial data manipulator and translator software, first released at the end of 1998, has been evaluated. Evaluation has been carried out according to an evaluation criteria, using some test data of several GIS formats. Consequently, it has been found that FME offers valuable solutions to many problems of traditional SDI. However, it still lacks some of the features of an “ideal” SDI environment.

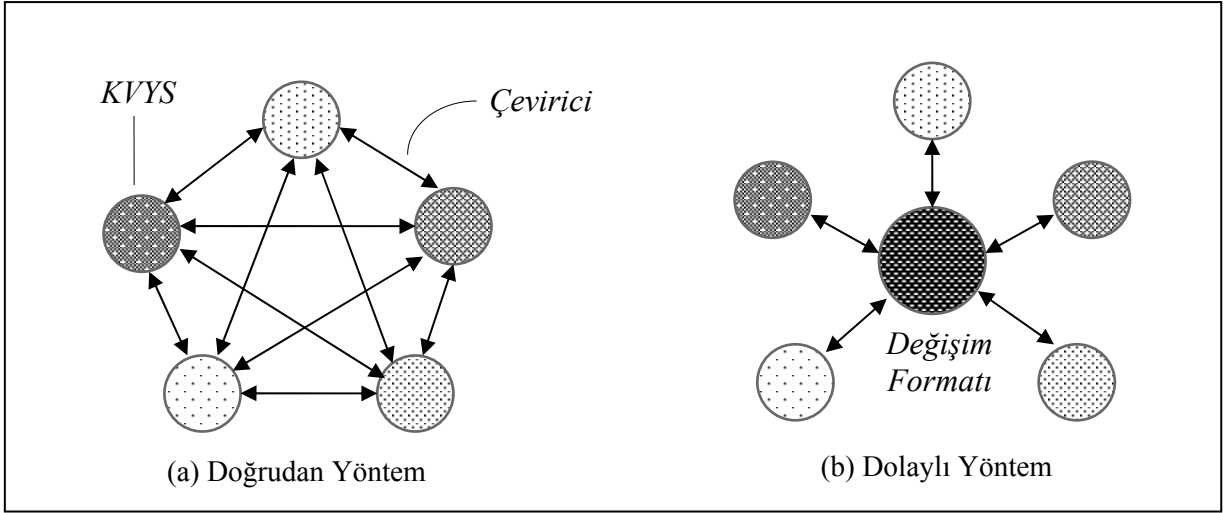
1. GİRİŞ

Konumsal Veri Yönetiminde (KVY) hızlı, sağlıklı ve ekonomik çözümler üretilebilmesi için konumsal verinin “etkin” bir biçimde paylaşımı, bugün her zamankinden daha çok kaçınılmaz olmuştur. Bunun nedeni, günümüz uygulamalarının çok çeşitli türde veri gerektirmesi ve konumsal verinin ilk elden toplanmasının, kullanılan yöntem ne olursa olsun, oldukça pahalı ve zaman alıcı olmasıdır. Aslında ilk elden veri toplama, çoğu durumda uygulama yürütücüsünün kendi olanakları ile veri toplama kapasitesini aşması nedeniyle de kabul edilemez bir yoldur.

KVY de veri paylaşımının alışlagelmiş yolu, Konumsal Veri Değişimi (KVD), yani gerekli verinin, sayısal formda bir dış kaynaktan transfer edilmesidir. Bir KVD de iki taraf vardır; Bunlardan biri veriyi sağlayan, “sunucu”, diğeri ise veri ihtiyacında olan “istemci” dir. İstemci ve sunucu iki farklı Konumsal Veri Yönetim Sistemi (KVYS) olabileceği gibi, sunucu bir veri sağlayıcı kurum da olabilir. Burada “KVYS”, bir KVD nin tarafları olabilecek Coğrafi Bilgi Sistemi (CBS), Bilgisayar Destekli Haritacılık (BDH) ve Sayısal Görüntü İşleme Sistemlerini (SGİ) kapsamak üzere kullanılmıştır.

KVD aynı “coğrafi gerçeğin” iki farklı formattaki temsilleri arasında bir “çeviri” (translation) gerektirir. Temsildeki farklılık ile, aynı coğrafi gerçeğin “tanımında”, bu tanım için “kullanılan dilde”, tanımın “gerçekleştirmede” ve nihayet tanımın “sunumunda” olabilecek farklılıklar kastedilmektedir. Coğrafi gerçeğin tanım farklılıklarından biri, coğrafi gerçeğin tanımlandığı bilgi kapsamıdır. Bir “parsel” tanımında yer alan parselin “toprak cinsi” bilgisinin, diğeri bir tanımda ise bulunmaması, buna örnektir. Bu sınıflamaya göre, coğrafi gerçeğin vektör ya da raster formunda, yalnızca metrik ya da metrik ve topolojik olarak tanımlanması tanım farklılıkları olarak görülmektedir. Dolayısıyla, raster ve vektör tanımlamaların kendi içindeki farklılıkları da bu gruba girmektedir. “Tanımlamada kullanılan dil” ile kastedilen veri modelleridir. Konumsal veri için Hiyerarşik, Ağ, İlişkisel gibi geleneksel ve “Nesne Tabanlı” veri modelleri kullanılabilir /1/. Bu modellerin her biri farklı kavram ve kurallar içerirler. Gerçekleştirim farklılıkları ile, veri yapıları farklılıkları, birim farklılıkları (örn. metre-inch), veri tipi farklılıkları (örn. parsel numarası için bir sitemde “tamsayı” diğeri “karakter dizisi” kullanılması) vs. kastedilir. Nihayet sunum farklılıkları ile coğrafi gerçeğin farklı kartoğrafik sunumları yani farklı semboller, projeksiyonlar, renkler, fontlar vs. kastedilmektedir.

Genel olarak, KVD iki şekilde gerçekleştirilebilir. Bunlar “Doğrudan” ve “Dolaylı” yöntemlerdir. Dolaylı yöntemde iki format arasındaki çeviri bir “ara format” üzerinden gerçekleştirilir (Şekil-1b). Sunucu verisi önce ara formata dönüştürülür, alıcı ara formatta veriyi transfer eder ve kendi formatına dönüştürür. Buradaki ara format çoğunlukla “Değişim Formatı” (DF) olarak anılır /2/. Doğrudan yöntemde ise, iki format arasında doğrudan bir çeviri uygulanır (Şekil-1a). Doğrudan yöntemde çeviri yalnızca iki formata yönelik olduğundan, “çeviri kalitesi” yüksektir. Dolaylı yöntemdeki gibi olası bir “bilgi kaybı” riski yoktur. Dolaylı yöntemde DF nin yeterince “genel” olmaması bilgi kaybına yol açacaktır. Buna çok basit bir örnek olarak, 3 boyutlu koordinatlar kullanan tarafların yalnızca 2 boyut içeren bir DF üzerinden KVD gerçekleştirilmesi verilebilir.



Şekil-1: Konumsal veri değişim yöntemleri

Dolaylı yöntemde sorun, DF nin “genel”, fakat aynı zamanda “esnek” olması gereğidir. Formatın çok sayıda format arasında “eksiksiz” çevirilere olanak tanıyan kapsamlı bir DF, özellikli transferler için “hantal” kalacaktır. Örneğin yalnızca vektör veri içeren bir değişim için, hem raster hem vektör veri için tasarlanmış bir DF nin kullanılması verimsizliğe yol açacak, DF nin farklı kullanıcı kesimlerinden kabul görmesini engelleyecektir. O nedenle formatın özellikli değişimler için uyarlanabilir, yani esnek olması gerekir. Örneğin değişik çözünürlüklerde koordinat gösterimi mümkün olmalı, veya belirli bir değişim için yalnızca 2 boyutlu koordinatların geçerli olduğu belirtilebilmelidir.

Doğrudan yöntem ise, çok sayıda sistemin veri değişiminde bulunabilmesi için gerekli çevirici sayısı ve çeviricilerin yenilenmesi yada yeni çeviricilerin eklenmesi bakımından sorunludur; Şekil-1 de mevcut sistemlere, yeni bir sistem eklenmesi durumunda yeniden geliştirilmesi gerekli çevirici sayısı dolaylı yöntemde yalnızca bir iken, doğrudan yöntemde mevcut sistem sayısı kadardır. Çevirici geliştirme, her iki formatın da çok iyi bilinmesini gerektirdiğinden, oldukça zaman alıcı bir işlemdir. O nedenle, doğrudan çevirici geliştirme çoğu zaman eldeki proje süresi açısından geçerli bir çözüm olmayacaktır. Dolayısıyla, KVD de geleneksel olarak daha yaygın kullanılan yöntem, dolaylı yöntem olmuştur. Ancak, konumsal veri kullanıcılarının çeşitliliği, bütün kullanıcı gruplarının gereksinimlerini karşılayacak bir DF tasarımı çok güç, belki de olanaksız yapmıştır. Dahası, tasarımcılarına göre “çok iyi” tasarlanmış olması, DF nin standartlaşmasını garanti etmemiştir. Bunun en çarpıcı örneği A.B.D. de dokuz yıl gibi bir sürede geliştirilen ve bütün yasal yaptırımlara rağmen bir türlü yaygın kabul görmeyen SDTS (Spatial Data Transfer Standard) formatıdır. Dolayısıyla, bugüne kadar pek çok DF geliştirilmiş olmasına rağmen bugün KVD için tek bir standart mevcut değildir. Bir anlamda herkes “kendi standardını” kullanmaktadır. Örneğin pek çok bakımdan yetersiz olmakla birlikte DXF (Drawing Exchange Format), bugün KVD için en yaygın kullanılan formatlardan biridir. Sonuç olarak, geleneksel KVD de kullanılan DF lerin yukarıda anılan farklılıklar içerecek kadar genel ve aynı zamanda özel olamaması nedeniyle çeşitli sorunlar yaşanmaktadır. İzleyen kısımda bu sorunlar ele alınacaktır.

2. GELENEKSEL KONUMSAL VERİ DEĞİŞİMİNİN SORUNLARI

Bu kısımda geleneksel konumsal veri deęişiminin sorunları, belirli ölçütlere göre incelenmiştir. Bu ölçütler aynı zamanda FME nin deęerlendirilmesinde kullanılan ölçütler oldukları için, bunlara göre ideal bir konumsal veri dönüştürücünün sahip olması gereken özellikler de bu kısımda belirtilmiştir. Bu ölçütler /1/ den yararlanılarak belirlenmiştir. /1/ Bir DF nin deęerlendirilmesine yönelik bir dizi ölçüt belirlemiştir. FME bir KVÇ olduğundan, /1/ deki ölçütlerin, bir KVÇ deęerlendirmesinde kullanılabilir olanları seçilmiştir. Bu ölçütler aşağıda kısaca tanımlanmıştır. Daha ayrıntılı açıklamalar için /1/'e başvurulabilir.

a. Veri Modeli

Konumsal veri modelleri “Nesne-Tabanlı (NT)” ve “Kayıt-tabanlı (KT)” olarak ikiye ayrılabilir /1/. Geleneksel model, genellikle İlişkisel Veri Modeline dayanan KT modellerdir. KT modellerde coęrafi detaylar, temel modelleme elemanları “temel geometrik elemanlar” nokta, çizgi ve poligonlardır. Coęrafi detaylar ve aralarındaki ilişkiler, bu elemanlar üzerinden temsil edilir. İşlemler, temel geometrik elemanlar bazında gerçekleştirilir. NT modeller ise Nesne Yönelimli (NY) teknolojinin özellikle 1980 lerin 2. yarısından sonraki hızlı yükseliş ile gündeme gelmiştir. NT modellerde temel modelleme elemanı “nesne” yani coęrafi detaylardır. Coęrafi detaylar geometri ve öznitelik bileşenleri ile tanımlıdır. Kullanıcı coęrafi detaylar bazında uygulamalarını modeller. Bu açıdan NT modeller, KT modellerden daha yüksek düzeylidir. Bugün piyasada KT ve NT CBS ler mevcuttur. Ayrıca geleneksel KT CBS lerin NY arabirimleri geliştirilmiştir. Dolayısıyla KVÇ nin KT ve NT coęrafi veri temsilleri arasında dönüşüme olanak tanınması gerekir. Geleneksel konumsal veri deęişiminde, transferde nesne tanımlama olanağı olmadığından, NT sistemlere doğrudan dönüşüm yapılamaz.

b. Öznitelik Veri Transferi

İdeal bir konumsal veri dönüştürücü, coęrafi varlıklara ait konumsal ve konumsal olmayan verinin transferine olanak tanımalıdır. Geleneksel olarak, konumsal veri deęişimi konumsal veri transferi üzerinde yoğunlaşmış, öznitelik veri transferi adeta göz ardı edilmiş, ayrıca yazılmış batch programlar yardımıyla transferi yoluna gidilmiştir. Oysa, bir konumsal veri deęişiminde, sayısal formatta öznitelik verisinin transfer edilememesi çok büyük bir eksikliklerdir. Çünkü bu durumda bu verilerin elle tek tek girilmesi ya da batch programlarla bu sorunun çözülmesi gerekir. Her iki çözüm şekli de, özellikle çok miktarda veri içeren uygulamalar açısından (örn. çok sayıda parsel verisi içeren kadastral bir veri tabanı), oldukça bıkırtıcı, zaman istemci ve hata-eęilimlidir. Bu nedenle konumsal veri deęişiminde, öznitelik verisinin doğrudan transfer edilebilmesi gerekir.

c. Kartografik Gösterim

Coęrafi veri kartografik olarak çok çeşitli şekillerde gösterilebilir. Örneğin akarsu isimleri bir sistemde akarsu üzerinde tek bir açı altında gösterilebilirken, dięer bir sistemde daha estetik olarak akarsuyun eğriliğini izleyen bir biçimde gösterilebilir. KVÇ nin bu farklı gösterimleri tanınması gerekir. Benzer şekilde, coęrafi detaylar çok çeşitli şekil ve renklerde sembollendirilebilir. Örneğin kentler nüfuslarına göre deęişik yarıçaplı dairelerle, otoyollar kırmızı çizgilerle, göller mavi taralı poligonlarla gösterilebilir. Gösterim (Annotation) ve sembollendirme, özellikle çok sayıda sembol içeren haritalar için oldukça zaman alıcı bir

işlemdir. Dolayısıyla, geleneksel yaklaşımda olduğu gibi, sembollendirmenin transferden sonra istemcide yeniden yapılması, özellikle seri harita üretimi yapan bir sistem açısından hiç te pratik olmayacaktır. Bu bakımdan KVC nin sunucu ve istemci sembolleri arasında bir eşleştirme yapmaya olanak tanınması gerekir. Kartoğrafik projeksiyonlar açısından, KVC de çok sayıda projeksiyon sisteminin tanımlı olmasından daha önemlisi, transfer veri grubunda istenen bir projeksiyona ait parametrelerin belirtilmesine ve transferde herhangi iki projeksiyon arasında dönüşüme izin vermesidir. Geleneksel uygulamada konumsal veri değişim formatları yalnızca popüler sistemleri (örn. Mercator, UTM) tanımakta, yeni bir sistem tanımlamaya ve transferde koordinat dönüşümüne olanak tanımamaktadır.

ç. Koordinatlar

Her koordinat sisteminin bir referans sistemi vardır. Farklı referans sistemlerindeki koordinatların birbirlerine dönüştürülebilmesi için referans sistemi parametreleri bilinmelidir. Değişik koordinat sistemlerinde koordinatların belirtilebilmesi ve bu koordinat sistemleri arasında koordinat dönüşümlerinin gerçekleştirilebilmesi için dönüştürücünün öncelikle yaygın kullanılan koordinat, projeksiyon ve referans sistemlerini tanıması ve daha önemlisi, gerektiğinde yeni bir sisteme ait parametrelerin belirtilmesine izin vermesi gerekir. Ayrıca koordinatların transferde en azından sunucudaki duyarlılıkla belirtilebilmesi gerekir. Aksi takdirde, istemcide yanlış sonuçlar (örn. sunucuda bir noktada kesişen iki çizgi parçasının bu durumunun istemcide değişmesi gibi) ortaya çıkabilir. Gerek koordinat sistemleri ve gerekse koordinat duyarlılığı açısından geleneksel DF ler sınırlı olanaklar sunmaktadır.

d. Topoloji

Topolojik veri transferinin gerekliliği tartışmalı bir konudur. Topoloji transferi iki bakımdan gereksiz görülmektedir. Birincisi, topoloji istemcide yeniden oluşturulabilir. İkincisi, dönüştürücünün topolojik veri yapısının sunucu ve istemcininkinden oldukça farklı olması durumunda, topolojinin istemcide yeniden oluşturulması çok daha pratik olabilir. Buna karşılık topoloji transferini destekleyen etmenler vardır. Öncelikle, topoloji oluşturma zaman alıcı bir işlemdir. İkinci olarak, istemcinin topoloji oluşturma yeteneği bulunmayabilir. Sonuç olarak, KVC nin topoloji desteğinin bulunması arzu edilen bir özelliktir. Bununla birlikte, istemci topolojik veriye gereksinim duymayabilir ya da yalnızca metrik veriyi transfer etmek isteyebilir. Bu örneğin, istemcinin transfer ettiği veriyi yalnızca harita üretiminde kullanması ya da topolojiyi kendi içerisinde ayrıca oluşturmak istemesi durumunda söz konusu olabilir. Bu nedenle konumsal veri dönüştürücünün geometrik veri transferi yanında, yalnızca metrik yada topolojik veriyi transfer edebilme seçeneğini de sunması gerekir. Geleneksel uygulamada yalnızca metrik bilgiler transfer edilir. Gerekirse topoloji istemcide yeniden kurulur.

e. Nesne Bazında Veri Değişimi

Geleneksel olarak, konumsal veri değişimi bütün bir dosyanın transferi şeklinde gerçekleştirilmektedir. Oysa istemcinin gereksinim duymadığı veriyi transfer etmesi insan ve bilgisayar kaynaklarının israfına yol açacaktır. Çünkü bu durumda hem transfer zamanı artacak, hem de gerekli seçimin istemcide yapılması gerekecektir. İdeal olarak kullanıcı, bütün bir dosyayı transfer edebileceği, yalnızca istediği detayları, örneğin bir karayolunu yada

bir kent sınırını transfer edebilmelidir. Bu nedenle ideal bir KVC nin nesne bazında veri değişimini desteklemesi gerekmektedir.

f. Karmaşık Nesnelere

Coğrafi nesnelere basit ya da karmaşık olabilir. Karmaşık nesnelere daha basit nesnelere içerirler. Örneğin bir “karayolu” nesnesi basit, birden çok karayolundan oluşan bir “karayolu_ağı” ise karmaşık bir nesnedir. Bir KVC nin karmaşık nesne desteğinin olması, basit ve karmaşık nesnelere arasında geçişe olanak tanıması anlamına gelir. Diğer bir ifadeyle, “birleştirme” ve “ayrıştırma” gibi nesnelere basit ya da karmaşık nesne oluşturma mekanizmalarını içermesi gerekir. Çünkü bugün CBS piyasasında katman-tabanlı popüler yazılımların büyük çoğunluğunun, nesne-tabanlı yapıya geçme eğiliminde oldukları, yeni geliştirilenlerin de çoğunlukla nesne-tabanlı oldukları gözlenmektedir. Bu nedenle bir KVC nin karmaşık nesne desteğinin bulunması çok büyük bir önem taşımaktadır. Bu destek KVC nin veri modeli düzeyinde ya da, dönüşümde birleştirme/ayrıştırma mekanizmalarının kullanılması şeklinde olabilir. Geleneksel konumsal veri değişiminde karmaşık nesne oluşturma mekanizmaları yoktur.

g. Nesne Paylaşımı

Nesne paylaşımı, birden fazla nesnenin, veri tekrarına yol açmaksızın aynı bir nesneyi paylaşmalarıdır. KT sistemlerde iki poligonun ortak sınır çizgisini paylaşması, NT sistemlerde aynı nesnenin iki ya da daha çok nesneye ait olabilmesi gibi. Nesne paylaşımı günümüzde çok popüler hale gelen, internet üzerinden konumsal veri dağıtımını açısından önemlidir. Çünkü bu açıdan nesne paylaşımı, daha küçük boyutlu bir transfer veri seti ve dolayısıyla daha hızlı bir transfer anlamına gelmektedir. Bu nedenle, ideal bir KVC nin nesne paylaşımını desteklemesi gerekir. Nesne paylaşımı zaten NY veri modeli ile gündeme gelen kavramlardan biri olduğu için, geleneksel yaklaşımda nesne paylaşımı uygulanmaz. Çünkü geleneksel konumsal veri değişim formatları bu amaca yönelik olarak geliştirilmemişlerdir.

ğ. Raster Veri

Konumsal veri, vektör ya da raster formda temsil edilebilir. Vektör formda coğrafi gerçek (veri), temel geometrik elemanlar nokta, çizgi ve poligonlarla temsil edilir. Raster temsilde ise, coğrafi gerçek, üzerine bir anlamda izdüşürüldüğü, bir grid sistemine izdüşen karakteristikleri yardımıyla temsil edilir. Raster veri bugün, özellikle çeşitli veri sağlayıcılardan alınan verinin (örn. uydu görüntüleri) büyük oranda raster veri modunda olması nedeniyle çok popülerdir. Konumsal veri bugün, vektör temsilde olduğu gibi, raster temsilde de çok çeşitli formatlarda gelebilmektedir. TIFF, JPEG, RLE, Landsat TM, SPOT, GIF, BMP, PICT ve TARGA en yaygın raster formatlara örnektir. Geleneksel olarak konumsal veri değişim formatları çoğunlukla vektör veri için geliştirilmiştir. Raster verinin yaygınlık kazanması ile, yakın zamanda geliştirilen formatlara (SDTS, SAIF) yaygın kullanılan raster formatlara yönelik bir destek eklenmiştir. Aslında yukarıda anılan popüler raster format bugün kullanımda olan pek çok CBS ve Sayısal Görüntü İşleme yazılımı tarafından tanınmaktadır. Ancak ideal bir konumsal veri çeviricinin, bu yeteneği olmayan sistemler açısından yaygın kullanılan bu raster formatları için gerekli desteği sunması ve daha önemlisi, gerektiğinde yeni bir format tanıtmaya da olanak tanıması gerekir.

h. Değişim Ortamı

Konumsal veri deęişiminin bugün pratikte uygulanışına bakıldığında, veri gruplarını görme, sorgulama ve ardından istenen alt grupları ya da nesnelere transfer etmeye olanak tanıyan Grafik Kullanıcı Arabirimlerinin (GKA) eksikliği göze batmaktadır. Oysa ideal bir veri deęişim ortamında istemci, sunucu verisini bir GKA vasıtasıyla görüp, sorgulayarak yalnızca gereksinim duyduğu veriyi transfer edebilmelidir. Diğer bir ifadeyle “seçmeli transfer” ya da “nesne bazında transfer” olanaklı olmalıdır. Seçmeli transfer, transfer edilecek detayların tek tek (örn. “1212” nolu parsel) ya da grup olarak (“örn. Hazineye ait bütün parseller”) belirtilmesine olanak tanımalıdır. Seçim, özniteliklere ya da konumsal kapsama göre veya grafik görüntü üzerinden yapılabilir. İdeal bir KVC tüm bu seçenekleri sunmalıdır. GKA ya da seçmeli transfer araçları gibi deęişim ortamı araçlarının eksikliği, geleneksel Konumsal veri deęişiminin en önemli sorunlarından biridir. Kullanıcının, genellikle “batch” modunda çalışan dönüştürücülere müdahalesi ya hiç yoktur ya da çok sınırlıdır.

3. FME YAZILIMI

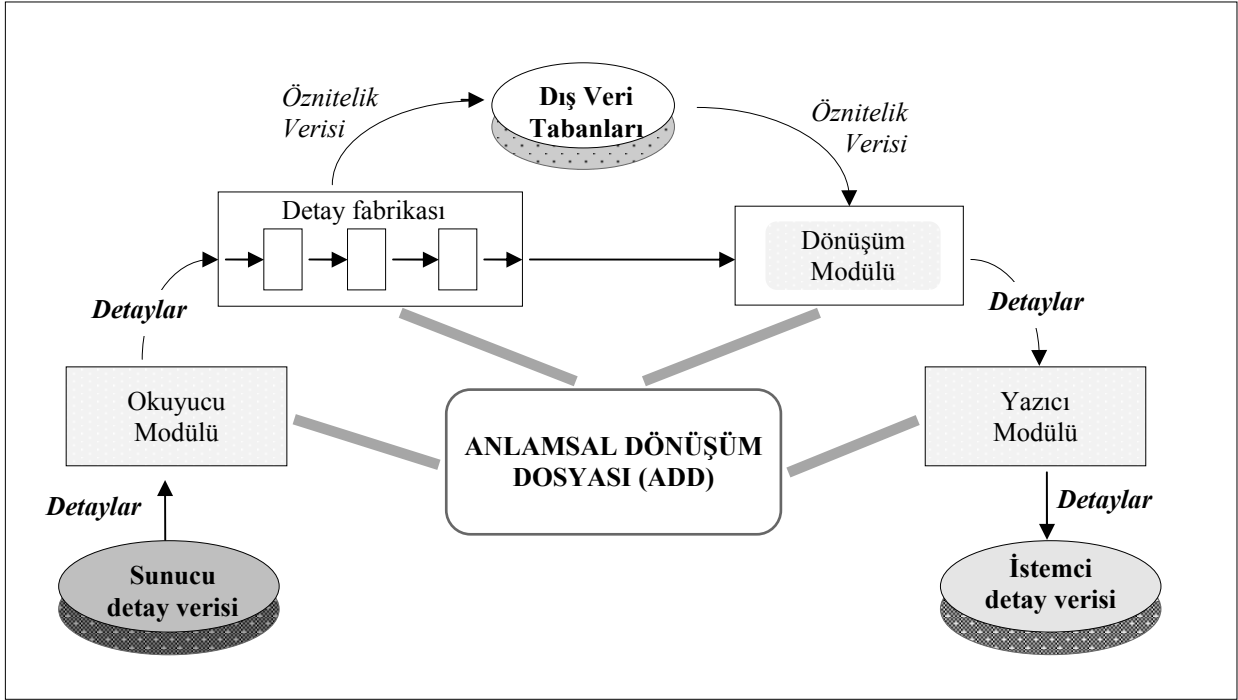
FME (Feature Manipulation Engine), Kanada'nın “Safe Software” yazılım şirketi tarafından geliştirilen bir Konumsal Veri Çevirici (KVC) dır. Ancak FME, verinin yalnızca bir formattan diğerine dönüştürülmesine izin veren geleneksel KVC lerden farklıdır. FME, sunucu formatındaki verinin bir takım işlemlerden geçirilmesine de olanak tanımaktadır. Bu ayırımı vurgulamak üzere FME geliştiricileri, “KVC” yerine tam karşılığı “Detay İşleme Motoru” olan “FME” ismini tercih etmişlerdir. FME ile veriye topolojik işlemler, geometri işlemleri, öznitelikler üzerinde işlemler ve koordinat dönüşümü işlemleri uygulanabilir /3/.

FME, yaygın kullanılan 35 CBS, BDH ve İlişkisel Veri Tabanı Yönetim Sistemi (İVTYS) veri formatı arasında dönüşüm yapabilmektedir /3/. Bunlardan bazıları, AutoCAD Formatları (DWG, DXF), Arc/Info Formatları (Export (E00), Generate, SDE 2.x ve 3.x), ArcView Shape File, Intergraph Formatları (IGDS, MGE, FRAMME), MapInfo Formatları (MIF, MID, TAB), Smallworld, PHOCUS PHODAT, İVTYS Formatları (DBF, CAT, CSV, ASCII), Oracle Spatial, SAIF (Spatial Archive and Interchange Format), Vector Product Format (VPF) dır. Java veya C++ programlama dilleri kullanılarak sisteme yeni bir format desteęi de eklenebilmektedir. Ayrıca kullanıcılar kendi koordinat ve projeksiyon sistemlerini tanımlayabilirler.

FME'nin, FME Desktop ve FME Professional ana modülleri mevcuttur. Sıradan kullanıcılar için FME Desktop yeterlidir. Bu modül ile kullanıcılar iki format arasında otomatik dönüşüm seçeneğini kullanabilecekleri gibi, kendi amaçlarına uygun özelleştirmeler de yapabilirler. FME Professional ise örneğin yeni bir format için destek eklenmesi gibi işler için daha üst düzey kullanıcılara yönelik olan modüldür. Ayrıca pek çok kullanıcının aynı anda FME yi kullanmalarına olanak tanıyan FME Server modülü vardır. FME Desktop ve Professional için Windows NT/95, FME Server için popüler UNIX sistemleri ve Windows NT sürümleri mevcuttur. FME'nin, en son sürümü, Haziran'99 da çıkan 2.3 sürümüdür. FME hakkında daha detaylı bilgi için Safe Software internet sitesine (www.safe.com) başvurulabilir.

a. FME Mimarisi

FME verileri işlemek için çeşitli modüllere sahiptir (Şekil -2). Okuyucu (Reader) modülü, sunucudan gelen verileri okur. Fabrika (Factory) modülü, değişik yollarla detayları birleştirir ya da daha basit bileşenlere ayırır. Dönüşüm modülü, veriyi bir formattan başka bir formata çevirir. Yazıcı modülü (Writer) ise FME tarafından desteklenen bir formatta verileri yazar. Tüm bu işlem adımları FME' nin dönüşümü yönettiği **Anlamsal Dönüşüm Dosyasında (ADD)** tanımlanır. Tüm modüllerdeki istatistik bilgileri, işlem bilgileri, uyarılar ve hatalar dönüşüm kütüğünde (log file) kullanıcıya sunulur.



Şekil -2: FME Mimarisi /4/

b. FME Fonksiyonları

FME fonksiyonları, dönüşüm sırasında detaylara özel işlemler uygulanmasına olanak tanır. FME fonksiyonları bir "@" ile başlar, parametreler kapalı parantez içerisinde belirtilir. Parametreler sabit, transfer değişkeni (veri setinden okunan değerlerin yüklendiği değişkenler), öznelik değerleri, yine bir fonksiyon, ya da bu dördünün karması olabilir. FME iki tür dönüşüm fonksiyonuna sahiptir. Bunlar öznelik değerleri için "öznelik fonksiyonları" ve detayları işleyen "detay fonksiyonları" dır. Bir fonksiyonun genel yapısı aşağıdaki gibidir:

@ Fonksiyon([< parametre > [, < parametre >]])

FME iki çeşit dönüşüm fonksiyonuna sahiptir. Bunlar öznelik değerlerini hesaplayan "öznelik fonksiyonları" ve detayları şekillendiren, işleyen "detay fonksiyonları" dır. Tablo-1, dönüşüm fonksiyonları arasındaki farkları göstermektedir.

Tablo-1: Dönüşüm fonksiyonları arasındaki farklar

Öznitelik fonksiyonları	Detay fonksiyonları
Bir değer hesaplar ve döndürür	Değer döndürmez
Detayı doğrudan değiştiremez	Detayın hem geometrisini hem de özneliğini değiştirebilir
Detaya yan etkileri yoktur	Kalıcı yan etkiler bırakabilir
Detayı, ortamı ve dış veri kaynaklarını etkilemez	Harici veri kaynaklarını okuyabilir ya da bu kaynaklara veri yazabilir
Dönüşüm özellik satırlarında, öznitelik isminin sağına yazılırlar	Dönüşüm özellik satırlarında bütün öznitelik değerlerinden sonra yazılırlar

FME fonksiyonları sunucu detaylarına uygulanarak yeni değerler hesaplanabilir ya da detaylar üzerinde işlemler yapılabilir. Fonksiyonlarla hesaplanan değerler birer öznitelik değeri olarak istemci detaylarına eklenebilir ya da bunlar üzerinden transfer kontrol edilebilir. Örneğin “Area” ve “Length” fonksiyonları ile detayların alan ve uzunluk değerleri hesaplanabilir. Bu değerlere göre yalnızca istenen detaylar transfer edilebilir. Buna bir örnek aşağıdaki (3.3) Test fabrikası örneğinde fabrikaya giren detaylara “Area” (alan) fonksiyonunun uygulanarak alanların hesaplanması ve yalnızca 2000 m² den büyük parsellerin transfer edilmemesi verilebilir. FME de fonksiyonlarla detaylar üzerinde çeşitli işlemler yapılabilmektedir. Bunlara örnek olarak ta, “Generalize” fonksiyonu verilebilir. Bu fonksiyonla, çizgi detaylar inceltme suretiyle genelleştirilebilir. Bu fonksiyon örneğin eşyükseklik eğrilerinin yumuşatılması amacıyla kullanılabilir.

FME’de iki format arasındaki dönüşüm için aynı ADD kullanılabilir. Yani aynı ADD de sunucu ve istemci formatlarının isimleri evirilebilir. Bu durumda fonksiyonlar, tanımlı ise ters işlemi yerine getirir. Fonksiyonun ters işlemi tanımlı değilse, hiç bir işlem yapılmaz.

c. Detay Fabrikaları

Fabrikalar, fonksiyonlardan sonra FME' nin ikinci ana işlem birimleridir. Detay fabrikaları öznetelik ve detay fonksiyonları ile birleştirilebilir çeşitli dönüşümler gerçekleştirilebilir /4/. Genel olarak bir fabrika 3 birimden oluşur. Giriş birimi, fabrikaya giren detayları, çağırılan öznetelik ve detay fonksiyonlarını kontrol eder. "işlem birimi", fabrikaya giren detayların işlendiği birimdir. İşlem birimi, detayları birbirleriyle birleştirir, parçalara ayırır, siler, fabrikaya giren detaylardan yeni detaylar üretir yada görevi dahilinde ise detayların geometrilerini veya özneteliklerini değiştirir. Çıkış birimi ise, işlenen detayların fabrikadan çıkışını kontrol eder. Çıkış birimi, detay tiplerinin ve öznetelik değerlerinin, istenildiğinde silinerek, detayların yeni değerlerle fabrikayı terk etmelerine olanak tanır. Şekil 5.de fabrikaların genel yapısı görülmektedir.



Şekil -3: Detay fabrikalarının yapısı

ADD'de, fabrikalar hangi sıraya göre tanımlanmış ise, detaylar o sıraya göre bir fabrikadan diğerine geçerler. Bir fabrikadan çıkan detay, hemen başka bir fabrikaya gönderilebilir.

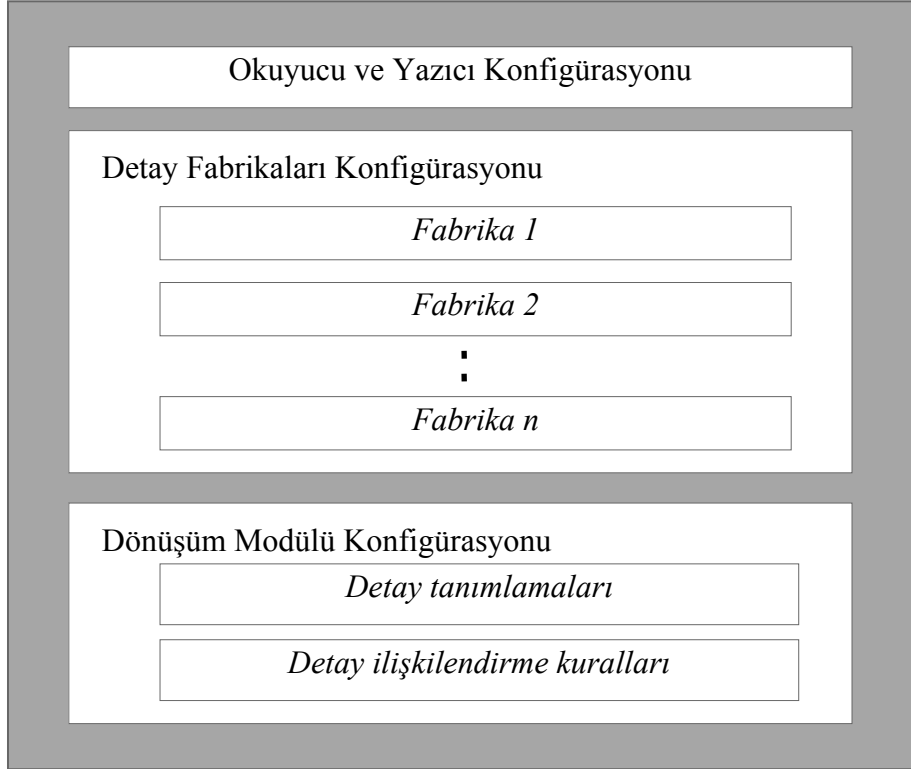
FME de çeşitli detay fabrikaları tanımlanmıştır /4/. En yaygın kullanılan fabrikalardan biri, "TestFactory" fabrikasıdır. Bu fabrikanın kullanımına bir örnek aşağıdadır:

```
FACTORY_DEF E00 TestFactory \
    INPUT FEATURE_TYPE * fme_geometry fme_polygon \
    TEST @Area() > 2000 \
    OUTPUT PASSED FEATURE_TYPE *
```

Burada, fabrikaya giren poligon detaylardan (örn. parseller) yalnızca alanı 2000 m² den büyük olanlar fabrikadan çıkabilir. Bu fabrika tanımını ADD ye ekleyerek kullanıcı, yalnızca alanı 2000 m² den büyük parselleri transfer etmek istediğini belirtir.

d. FME ile Çeviri İşlemi

FME veri dönüşümü bir "Anlamsal Dönüşüm Dosyası (ADD)" (Semantic Mapping File) kontrolünde yapılır. Bu dosya sunucu ve istemci format isimlerini, sunucu ve istemci detay tipi tanımlarını, bu tipler arasındaki dönüşüm tanımlarını ve fabrika tanımlarını içerir (Şekil - 4). Dönüşüm tanımlamaları sunucu ve istemci detayları arasındaki dönüşümü tanımlar. Bir dönüşüm tanımlaması, alt alta sunucu ve istemci formatlarını satırlardan oluşur /4/. Bu satırlardan sunucu formatı için olanlar "sunucu satırı", istemci formatı için olanlar da "istemci satırı" olarak anılacaktır.



Şekil - 4: FME Anlamsal Dönüşüm Dosyası yapısı /4/

FME sunucu veri setini tarararken sunucu satırındaki tanıma uyan detaya rastlandığında, detayı istemci satırındaki tanıma göre istemci formatına yazar. Her bir detay tek bir tanımlama satırı ile eşleşmelidir. Bir detayın birden çok satırdaki tanıma uyması durumunda, ADD deki “ilk” tanım kullanılır. Dönüşüm tanımlamasının genel formatı aşağıda verilmiştir. Bu tanımlamada, “sunucu detay tipi” sunucu satırındaki tanıma uyan detayın tipini, “istemci detay tipi” ise sunucu detayının dönüştürüleceği istemci detayı tipini belirtmektedir. “Öznitelik ismi” ve “öznitelik değeri”, “detay tipi” tipindeki detayların, sırasıyla öznitelikleri ve değerlerini belirtir. “Detay fonksiyonu”, varsa dönüşümde kullanılmak üzere FME fonksiyonunu belirtir.

```
<Sunucu formatı ismi> < Sunucu Detay Tip > \
[<öznitelik ismi> <öznitelik değeri>]* \
[<detay fonksiyonu>]*
< İstemci formatı ismi > < İstemci Detay Tipi> \
[<öznitelik ismi> <öznitelik değeri>]* \
[<detay fonksiyonu>]*
```

Aşağıdaki ADD parçası, IGDS ve SDE arasındaki dönüşümü tanımlamaktadır. FME de dönüşüm tanımlamaları çift yönlüdür. Dolayısıyla FME, her iki yöndeki (IGDS ten SDE ye ve tersi yönde) dönüşüm için de aşağıdaki tanımlamayı kullanılır. Aşağıdaki tanım, 12 nolu katmanda yer alan renk değeri 10, tipi 3 olan IGDS çizgi detayının, şerit sayısı 2, asfalt kaplamasız ve ROAD (YOL) katmanında yer alacak bir SDE detayına dönüştürüleceğini belirtmektedir.

```
# -----
IGDS 12 igds_type igds_line igds_color 10 igds_style 3
SDE ROAD NUMLANES 2 PAVED false
```

FME, ADD nin pratik bir biçimde oluşturulmasına olanak tanımaktadır. Bunun için, FME arabiriminden sunucu ve istemci format isimleri belirtilerek, “otomatik dönüşüm” seçeneği uygulanır. Belirtilen sunucu ve istemci formatları FME nin tanıdığı formatlar ise, ADD otomatik olarak üretilir. Bundan sonra kullanıcı kendi transfer verisine özel tanımlamaları varsa, gerekli değişiklikleri bu ADD üzerinde yapabilir. Ardından yine FME arabiriminden, sunucu ve istemci dosyalarının belirterek, FME yi tekrar koşturur. FME sunucu verisini tarayarak, ADD de tanımlı eşleştirme ve dönüşümleri yapar ve veriyi istemci formatında, belirtilen dosyaya yazar.

4. FME YAZILIMININ DEĞERLENDİRİLMESİ

Bu kısımda FME'nin ideal bir KVC nin sunması gereken özellikler bakımından bir değerlendirmesi yapılmıştır. Bunun için, belirlenen ölçütlere göre, çeşitli formatlardaki veri setleri kullanılarak, FME'nin değerlendirmesi yapılmıştır. Değerlendirme, KTÜ Jeodezi ve Fotogrametri Mühendisliği Bölümü, CBS Laboratuvarı'nda yapılmış, ARC/INFO .E00, ArcView Shape, AutoCAD DXF ve DWG formatlı veri setleri kullanılmıştır.

a. Veri Modeli

FME nin, geleneksel veri değişim formatlarındaki gibi etraflıca tanımlanmış bir veri modeli yoktur. Referans kitapçığında FME'de tanımlı geometri tipleri “fme_point”, “fme_line”, “fme_polygon”, “fme_donut” ve “fme_aggregate” olarak listelenmiştir. Bunlar sırasıyla nokta, çizgi, poligon, ada(lar) içeren poligon ve karmaşık detayları temsil etmek içindir. FME transferde bu geometrilerle temsil edilen herhangi bir detaya istenen sayıda öznitelik ekleyebilme olanağı sunmaktadır. Geleneksel veri değişim formatlarının tanımlamalarında olduğu gibi, her bir detay tipi (Karayolu, Akarsu, Göl, Parsel vs.) için, detayın sahip olabileceği özniteliklerin önceden, veri modelinde tanımlanmış olması böyle bir esnekliğe izin vermeyeceğinden FME tasarımında böyle bir yola gidildiği anlaşılmaktadır. Bu tasarım felsefesi aynı zamanda, geleneksel veri değişim formatı tasarımındaki en önemli problemlerden biri olan, “genel” fakat aynı zamanda “esnek” bir format tanımlamanın güçlüğü ortadan kaldırması açısından da çok anlamlıdır. Herhangi tipte bir detaya istenen bir özneliğin transfer anında atanabilmesi ile, detaylar bu öznitelikler bazında sorgulanabilmekte, transfer edilmesi gerekenler seçilerek, istenirse belirli işlemlerden geçirilerek istenen formata dönüştürülebilmektedir.

b. Öznitelik Veri Transferi

FME konumsal ve öznitelik veri transferine olanak tanımaktadır. Ancak, özellikle öznitelik verisi için özel veri yapılarının yer aldığı transferlerde, ilişkisel tabloların transferden önce sunucu sistemde birleştirilmesi gerekebilir. Örneğin, FME bir Arc/Info dosyasının transferinde sadece PAT (poligon öznitelik tablosu) ve AAT'yi (çizgi parçaları öznitelik tablosu) doğrudan dönüştürür. Veri tabanında bulunan uzantısı örneğin .TAB olan dosyalar ise ancak, dönüşümden önce PAT veya AAT ile birleştirilerek dönüştürülebilir. Bu tablolar dönüşümde FME'nin “Relate” fonksiyonunu tersi yönde kullanarak yine ayrı ayrı tablolar şeklinde istemciye gönderilebilir. Ayrıca, Relate fonksiyonu ile harici veri kaynaklarından okunan ilişkisel tablolardaki öznitelik verileri FME detaylarına eklenebilir.

FME'nin öznitelikler konusunda oldukça esnek bir yapısı vardır. Öznitelik fonksiyonlarının hesapladıkları değerler birer öznitelik değeri gibi detaylara eklenebilir (örn. Area (alan) ve Length (uzunluk) fonksiyonlarının hesapladıkları değerler) ya da bu değerler TEST fabrikasında seçmeli transfer için kullanılabilir.

c. Kartoğrafik Gösterim

FME, detayların kartoğrafik özniteliklerini, detay öznitelikleri gibi ele alır. Kartoğrafik öznitelikler, gösterim ve sembollendirme karakterini belirleyen boyut, renk, genişlik, yönelme, metinler için font vs. gibi özelliklerdir ve bazen görüntü karakteristikleri ya da grafik öznitelikler olarak anılır. FME'de kartoğrafik öznitelikler aynen detay öznitelikleri gibi sunucu ve istemci arasındaki karşılıkları ADD de belirtilerek dönüştürülebilir. Bunun için kullanıcının her iki sistemdeki grafik öznitelik tanımlamalarını bilmesi gerekir. Çünkü FME kartoğrafik öznitelikleri otomatik olarak aynen koruma özelliğine sahip değildir. Örneğin, bir ArcView Shape dosyasının AutoCAD DWG'ye dönüşümü yapılırken renk bilgisi otomatik olarak DWG dosyasına taşınmaz. FME'de AutoCAD için varsayılan renk "kırmızı" dır. Bu nedenle de, ArcView'de mavi renkte gösterilen detaylar dönüşümden sonra AutoCAD'de kırmızı renkte görünür. Ancak FME renk bilgisini de bir öznitelik olarak tanıdığı için, kullanıcı ADD'de "color" özniteliğinin değerini "mavi" olarak değiştirerek, ilgili detayların AutoCAD'te de mavi renkte gösterilmesini sağlayabilir. Aynı yaklaşım çizgi tipi, çizgi kalınlığı, font ve diğer kartoğrafik öznitelikler için uygulanabilir. Benzer şekilde sunucu ve istemci sembol eşlemeleri de ADD de belirtilir. İstemci detaylarına sembol eklemek veya sunucu veri setindeki belirli bir sembole, istemci veri setinde karşılık gelecek olan sembolü belirtmede, FME'nin @Lookup fonksiyonu kullanılabilir. Böylece sembol eşleştirmeleri bir defa tanımlanarak sunucu veri setinin tamamı için uygulanır. FME'nin projeksiyon sistemi desteği oldukça güçlüdür. Tanımlı 800 koordinat sistemi, 40 tan fazla projeksiyon, 130 dan fazla datum ve 29 ölçü birimi tanımını içermektedir. FME sunucu veri setinden koordinat sistemi parametrelerini otomatik olarak okumaktadır. Daha önemlisi FME, kullanıcıların kendi koordinat sistemlerini tanımlamalarına izin vermektedir.

ç. Koordinatlar

FME'de koordinatlar iki ve üç boyutlu belirtilebilir. "Function builder" iletişim kutusunda yer alan "scale" fonksiyonu ile istenen koordinat bileşeni belirtilen ölçek faktörüne göre değiştirilebilir. Aynı kutuda yer alan "offset" fonksiyonu ile koordinat bileşenleri istenen miktarda ötelenir. Affin veya Helmert dönüşümleri yapılabilir. Koordinat temsili için ayrılan standart alan virgülden sonra 15 karakter olmak üzere toplam 31 karakterdir. Koordinat temsili için ayrılan alan, ADD'de değiştirilerek istenen duyarlılık sağlanabilir.

d. Topoloji

FME, topoloji-geometri ayırımını desteklemektedir. Topolojik veri gerektirmeyen durumlarda, topolojik veriler göz ardı edilir. Bununla birlikte, "TopologyFactory" fabrikası ile topolojik veri yapısını desteklemeyen sunucu sistemlerden alınan çizgi parçası ve poligon detaylar için, topoloji oluşturulabilir. FME, sahip olduğu formatlardan hangisinin topolojik veri yapısını kullandığını hangisinin kullanmadığını bildiği için topoloji geometri ayırımını otomatik olarak yapmaktadır. Örneğin AutoCAD DWG/DXF formatından ArcInfo Export formatına veri transferi yapıldığında, E00 formatındaki verinin topolojisi otomatik olarak

kurulur. Tersine durumda da AutoCAD topolojik veri yapısını kullanmadığı için topoloji bilgilerinin dönüşümü yapılmaz.

e. Nesne Bazında Veri Değişimi

FME nesne bazında veri değişimini desteklemektedir. ADD’de yapılacak düzenlemelerle istenen detayların (örn. sadece nokta detaylar) transferi sağlanabilir. Ayrıca, TestFactory fabrikası kullanılarak detaylar öznitelik değerlerini göre teste tabi tutulur ve sadece testi geçen ya da testi geçemeyen detayların transferi yapılabilir. FME’ nin bu fabrikası özellikle nesne bazında veri değişimi konusunda çok büyük kolaylıklar sağlamaktadır.

f. Karmaşık Nesnelere

FME’de tanımlı konumsal nesne tiplerinden biri de karmaşık nesne tanımlamada kullanılan “Aggregate” dir. FME’nin tanıdığı formatlardan sadece SAIF ve SHAPE’in karmaşık nesne desteği vardır. SAIF’te karmaşık nesne farklı detay tipleri içerebilir. Yani nokta, çizgi parçası ve poligonlar birleştirilerek karmaşık nesne oluşturulabilir. SHAPE’te ise bir karmaşık nesne aynı tip detaylardan, yani yalnızca nokta, çizgi parçası ya da poligonlardan oluşabilir.

FME, “AggregateFactory” fabrikası ile öznitelik bazında gruplama yoluyla detayları birleştirerek karmaşık nesne oluşturmaya olanak tanır. Eğer istemci karmaşık nesne yapısını tanımlamıyor ise “DeaggregateFactory” fabrikası ile karmaşık nesnelere ayrıştırılır. Ayrıca tek tek nesne bazında karmaşık nesne tanımlamak için önce “TestFactory” sonra da “AggregateFactory” kullanılabilir. Örneğin, kadastro parsellerini içeren bir veri setinde “1250”, “1268” ve “1300” nolu parseller önce bir Test fabrikasından geçirilip hepsine “passed_a_test” özniteliği eklenebilir. Ardından, tüm parseller bir AggregateFactory’ ye gönderilerek burada “GROUP_BY passed_a_test” tanımlaması ile bu parsellerden oluşan bir karmaşık nesne tanımlanabilir.

g. Nesne Paylaşımı

FME nesne paylaşımını desteklemektedir. Bunun için detay fabrikaları kullanılır. Örneğin, “TopologyFactory” fabrikasında yer alan “REMOVE_DUPLICATE_ARCS” rutini topoloji kurulmadan önce farklı nesnelere tarafından kullanılan ortak çizgi parçalarının yalnızca bir kez kodlanmasını sağlar.

ğ. Raster Veri

FME, popüler raster formatlarındaki veriyi okuma yeteneğine sahip değildir. Yalnızca GIF formatında çıktı verebilmektedir.

h. Değişim Ortamı

FME, seçmeli transfer için gerekli araçlar sunmaktadır. Kullanıcı, çeşitli FME fonksiyonları ve fabrikalarını kullanarak, hangi detayları hangi öznitelikleri ile transfer etmek istediğini ADD de belirtebilmektedir. Örneğin, sadece nokta detayları transfer etmek için, ADD de diğer detaylara ait olan tanımlamaları silmek yeterlidir. Detayların bu şekilde bir

grup halinde seçime tabi tutulması yanında, “TestFactory” fabrikası kullanarak detaylar öznitelikleri bazında tek tek seçilebilirler. Bununla birlikte, FME’nin detayları görüntüleyip doğrudan seçerek dönüştürmeye olanak tanıyan bir Grafik Kullanıcı Arabirimi (GKA) yoktur. İşlemler detay tipleri ve detay öznitelikleri bazında belirtilmektedir. Oysa bir GKA, başta seçmeli transfer olmak üzere, karmaşık nesne oluşturmada da büyük kolaylık sağlayacaktı.

1. Dökümantasyon

FME’nin dökümantasyon desteği çok iyi değildir. FME Dokümanları (Reference Manual, User Manual ve Tutorial) kolay anlaşılır değildir. Bazı önemli ayrıntılar yeterince vurgulanmamıştır. Verilen örnekler kullanıcıların karşılaştığı sorunları çözecek yeterlikte değildir. Örneklerde çoğunlukla aynı formatlar kullanılmıştır. Bu bakımdan yazılımın öğrenilmesi, özellikle başlangıç aşamasında oldukça zordur.

5. İRDELEME

Bu bölümde, 4. bölümde elde edilen bulgular ışığında FME’nin KVD de geleneksel yöntemle üstünlükleri, ideal bir KVÇ ye göre eksiklikleri irdelenmektedir.

a. FME’nin Geleneksel Yöntemle Veri Değişimine Üstünlükleri

- Geleneksel yöntemde, bir dosyanın tamamı transfer edilir. Kullanıcılar dönüşüme müdahale edemediklerinden, seçmeli transfer gerçekleştirilemez. FME, nesne bazında veri değişimini desteklemektedir. Detaylar tek tek ya da grup olarak transfer edilebilir.

- Geleneksel yöntemde konumsal veri ve öznitelik verisi ayrı ayrı programlarla transfer edilebilir. FME ile her iki tür veri de bir arada transfer edilebilmektedir. Harici dosyalardan DBF, CAT, CSV, ASCII formatındaki öznitelik verisi okunabilir ve FME detaylarına eklenebilir. Ters yönde, detay öznitelikleri bu formatlardaki harici dosyalara yazdırılabilir ya da istemci formatı NT ise ilgili detaylara atanabilir.

- Geleneksel yöntemde kartoğrafik öznitelikler renk, çizgi tipi, çizgi kalınlığı ve semboller transferden sonra çoğunlukla değişir. Bunların istemcide yeniden düzenlenmesi, özellikle yoğun kullanıldıkları dosyalar için istemciye ağır bir yük getirecektir. FME transferde kartoğrafik özniteliklerin kontrolüne olanak tanımaktadır; Kullanıcı ADD’de sunucu ve istemci için kartoğrafik öznitelik eşleştirmelerini belirtebilmektedir.

- Geleneksel uygulamada, transfer aşamasında projeksiyon ve koordinat sistemi dönüşümleri yapılamaz. Ayrıca kullanıcı transferde kendine özel bir sistem tanımlayamaz. FME’de transfer verisine projeksiyon ve koordinat sistemi dönüşümleri uygulanabilir. Ayrıca, kullanıcılar kendi koordinat sistemlerini tanımlayabilir.

- FME, topoloji-geometri ayırımını desteklemektedir. Topolojik veri kullanmayan bir sisteme dönüşüm yapılırken (örn. AutoCAD), topolojik veri göz ardı edilir. Bununla birlikte, topoloji fabrikası (TopologyFactory) ile topolojik veri yapısını desteklemeyen sistemlerden alınan çizgi ve poligon detaylar için topoloji oluşturulabilir. Geleneksel yöntemde topoloji-geometri ayırımı yapılamaz. Ayrıca topoloji transfer edilemez.

- FME’de basit detaylar birleştirilerek karmaşık nesne oluşturulabilir. Bu özellik, karmaşık nesne yapısını destekleyen formatların (SAIF, ArcView Shape) yer aldığı veri değişimleri açısından önemlidir. Geleneksel yöntemde karmaşık nesne tanımlanamaz.

- Geleneksel yöntemde nesne paylaşımı söz konusu değildir. FME, nesne paylaşımını desteklemektedir.

- FME fonksiyonları (Area, Length, vd.) transfer verisine uygulanabilir, hesaplanan değerler birer öznitelik değeri olarak transfer veri setine eklenebilir ya da bunlar üzerinden transfer kontrol edilebilir. Geleneksel yöntemde bu tür olanaklar yoktur, transfer verisi herhangi bir işleme tabi tutulmadan olduğu gibi transfer edilir.

- FME dönüşüm kütüğünde (log file), transferde ne kadar detayın işlendiği, hangi detayların dönüştürüldüğü, hangilerinin silindiği gibi bilgiler, uyarılar ve transfer zamanı görülebilir. Böylece yapılan transfer hakkında, transfer aşamasında bilgi edinilebilir. Geleneksel yöntemde, transfer edilen dosya istemcide açılmadan transferle ilgili bir bilgi edinilemez.

b. FME' nin İdeal bir KVCÇ ye Göre Eksiklikleri

- İdeal bir KVCÇ, etkileşimli veri transferine olanak tanınmalıdır. Kullanıcılar, bir GKA aracılığı ile detayları görme ve sorgulama ve yalnızca istedikleri detayları transfer edebilmelidir. FME'nin böyle bir GKA desteği yoktur.

- İdeal bir KVCÇ, kartoğrafik öznitelikleri otomatik olarak koruyabilmelidir. FME bu özelliğe sahip değildir. Bu sorun FME'nin tanıdığı sistemlerin renk, sembol ve font tanımlamalarının programa tanıtılmasıyla giderilebilir. Ancak, özellikle semboller açısından standartlaştırmanın çok zor, belki de imkansız olması nedeniyle otomatik sembol dönüşümü çok iddialı bir hedef ve tek başına FME'nin problemi değildir.

- İdeal bir KVCÇ, öznitelik veri transferini doğrudan yapabilmelidir. FME ile, öznitelik verisini kendi özel veri yapısında tutan sistemlerin yer aldığı transferlerde bu yapılamamaktadır. Örneğin, sunucunun ARC/INFO sistemi olması durumunda, ilişkisel tablolardaki (örn. TAB uzantılı tablolar) öznitelik verisinin ARC/INFO sisteminde, Poligon Öznitelik Tablosu (PAT) veya Çizgi Parçaları Öznitelik tablosu (AAT) ile birleştirilerek transfere sokulması gerekir.

- FME vektör formatındaki verinin dönüşümünü yapmakta, raster formatındaki veriyi okuyamamaktadır. Yalnızca, vektör veriyi GIF formatında yazabilmektedir.

- FME'nin dokümantasyon desteği çok iyi değildir. Sunduğu dokümanlar (Reference Manual, User Manual ve Tutorial) kullanıcıların karşılaştığı sorunları giderecek ayrıntıda değildir.

6. SONUÇ

Bu çalışmada, belirlenen bir dizi ölçüte göre geleneksel KVD nin sorunları ve FME nin bu sorunlar için sunduğu çözümler incelenmiştir. Buna göre, FME nin KVD için sunduğu yeni olanaklar bir kaç grupta toplanabilir: Birincisi, iki format arasındaki dönüşüm tanımlamaları pratik bir yolla oluşturulabilir, gerekirse kullanıcı tarafından değiştirilebilir. Bu yolla örneğin sunucu tarafında kartoğrafik özniteliklerin nasıl olması gerektiği belirtilebilir, kullanıcı kendi projeksiyon ve koordinat sistemi tanımlamalarını belirtebilir. İkinci olarak, sunucu verisi fonksiyonlar ve detay fabrikaları kullanılarak çeşitli işlemlere tabi tutulabilir. Bu işlemler sonucunda yeni değerler hesaplanabilir, bu değerler istemci detaylarına öznitelik değeri olarak eklenebilir, ya da bu değerler üzerinden transfer kontrol edilebilir. Örneğin hesaplanan değerlere göre seçmeli transfer yapılabilir. Ayrıca bu işlemler kullanılarak detaylar üzerinde değişiklikler yapılabilir. Üçüncüsü, geleneksel KVD de önemli sorunlardan

biri olan tüm dosyanın transferi yerine, FME de seçmeli transfer yapılabilir. Son olarak, konumsal ve öznelik bilgileri bir arada transfer edilebilir. Gerçi bu açıdan kendi özel veri yapılarını kullanan sistemlerin yer aldığı transferlerde tabloların sunucuda birleştirilmesi gibi bir zorluk vardır ancak, bu yine de geleneksel yöntemle göre daha avantajlıdır. Sonuç olarak, ideal bir KVC ye göre bir takım eksikleri olmasına rağmen, geleneksel yöntemle sağladığı üstünlükler bakımından KVD için, FME nin kullanımındaki yöntemlere tercih edilmesi çok yararlı olacaktır.

KAYNAKLAR

- /1/ **Cömert, Ç.** : Ulusal Konumsal Veri Altyapısı İçin Veri Değişim Standardının Belirlenmesi, Doktora Tezi, KTÜ, FBE, Trabzon, 1996.
- /2/ **Lee, Y.C.** : A Framework for Evaluating Interchange Standards, CISM Journal, **Coleman, D.J.** Cilt: 44, Sayı: 4, Sayfa: 391-402, 1990.
- /3/ **Safe Software** : FME User Manual, Safe Software Inc., www.safe.com, 1999.
- /4/ **Safe Software** : FME Reference Manual, Safe Software Inc., www.safe.com, 1999.