

MÜHATEM Yazılım Programlama Kütüphanesi

Çağatay Ündeğer ^(a), Altan Yavaş ^(b),
Mustafa Ocak ^(b), Umut Akın ^(b), Ziya İpekkan ^(b)

^(a) STM A.Ş., Kafkas Sk. No:56, Beştepe, Ankara, cundeger@stm.com.tr

^(b) Genelkurmay BİLKARDEM Bşk.lığı, Bakanlıklar, Ankara

ÖZ

Müşterek Harekat Alanı Simülasyon Sistemi: Temsili Ortam Standart ve Uygulamaları (MÜHATEM) Projesi, ODTÜ TSK Modelleme ve Simülasyon Araştırma ve Uygulama Merkezi (MODSİMMER) Çalışma Planında yer alan hedeflerden biri olan Müşterek Harekat Alanı Simülasyon Sistemi (MÜHASİS)'nin geliştirilmesi kapsamında, bir alt yapı projesi olarak bir yılı aşkın bir ön çalışmanın ardından Kasım 2002 yılında başlatılmıştır. Bu bildiride, MÜHATEM Projesi kapsamında geliştirilen yazılım programlama kütüphanesi özetle tanıtılmaktadır.

Anahtar Kelimeler: Müşterek Harekat Alanı Simülasyon Sistemi, Temsili hareket ortamı, Temsili ortam standartları, Temsili ortam veritabanı, Temsili ortam yazılım kütüphanesi, Temsili ortam modelleme.

MÜHATEM Software Programming Library

ABSTRACT

Following a one-year of preliminary study, the project, *Common Mission Space Simulation System: Synthetic Environment Standards and Applications (MÜHATEM)*, was started in November 2002 as an infrastructure project in the scope of development of *Common Mission Space Simulation System (MÜHASİS)*, which is one of the objectives of METU TUF Modeling and Simulation Research and Development Center (MODSİMMER) Working Plan. In this paper, the application programming library developed in MÜHATEM Project is briefly introduced.

Keywords: Common Mission Space Simulation System, Synthetic theater of war, Synthetic environment standards, Synthetic environment database, Synthetic environment software library, Synthetic environment modeling.

1. GİRİŞ

Bilgisayar benzetimi (simülasyonu), gerçek veya teorik bir fiziksel sistem modelinin bilgisayar ortamında tasarlanması, çalıştırılması ve çalıştırma sonuçlarının analiz edilmesi disiplindir. Benzetim, gerçek hayatta oluşturulması zor veya imkansız durumları, sınırları çizilmiş bir düzeyde doğru ve gerçekçi şekilde bilgisayar ortamında

yaratmayı sağlar ve denemeye dayalı öğrenme prensibini kullanır. Benzetimlerin, sistemin gerçeğinin yerini tutması hiç bir zaman beklenemez çünkü bir şeyin yüzde yüz benzeri ancak o şeyin kendisi olabilir. Askeri alanda, temsili ortam modelleme, benzetimi yapılacak bir hareketin gerçekleştirileceği sentetik ortamın bilgisayarda sanal olarak oluşturulmasıdır.

MÜHATEM Projesi, TSK'nin tüm modelleme ve simülasyon projelerinde ortak bir ihtiyaç olarak öne çıkan temsili ortam modellemesinin, tekrarların önüne geçilerek sağlanması amacıyla, ODTÜ TSK MODSİMMER tarafından hazırlanan Modelleme ve Simülasyon Çalışma Planında yer alan hedeflerden biri olan MÜHASİS'nin geliştirilmesi kapsamında, bir alt yapı projesi olarak Kasım 2002'de başlatılmıştır. MÜHATEM Projesi ile;

1. MÜHASİS'in ihtiyaç duyacağı hareket alanı unsurlarının, benzetimi yapılmış hareket ortamında en gerçekçi şekilde bir araya getirilmesine yönelik alt yapının tesis edilmesi,
2. Temsili ortam veri standartlarının belirlenmesi,
3. Temsili ortam verilerinin toplanması, derlenmesi ve bir veritabanı oluşturularak kullanılabilir hale getirilmesi,
4. Tekrarların önlenmesi ile kaynak tasarrufu sağlanması ve
5. Benzetim sistemlerinin birlikte çalışabilirliğinin kolaylaştırılması hedeflenmiştir.



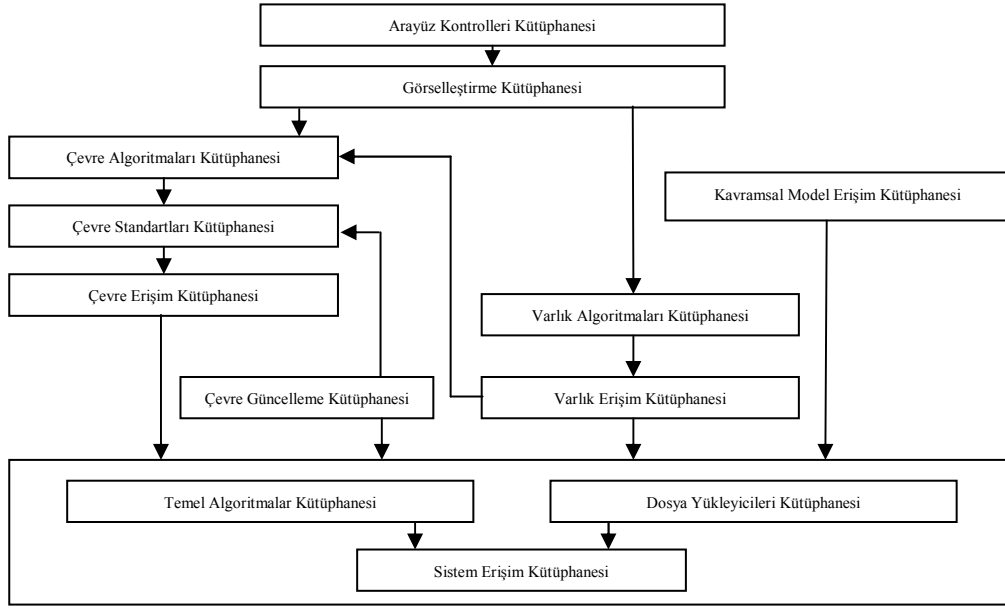
Şekil 1. MÜHATEM Kapsamı

Proje kapsamında (bkz. Şekil 1), tüm hareket ortamının (çevre, varlıklar, insan davranış ve görevleri) temsili için gerekli veriler toplanmakta, derlenmekte ve bir temsili ortam veritabanı hazırlanmaktadır. Ayrıca söz konusu veritabanının benzetim sistemlerince kullanılabilmesi için gerekli yazılım kütüphaneleri (API) ve temsili ortam veritabanının idamesi için gerekli sistem uygulama yazılımları geliştirilmektedir. Bu bildiride, geliştirilen yazılım programlama kütüphaneleri kısaca anlatılmakta olup detaylı bilgi için [1]'ye bakılması önerilir.

MÜHATEM kapsamında geliştirilen kütüphanelerin genel yapısı Bölüm 2'de, alt yapı kütüphaneleri Bölüm 3'te, çevre ilişkili kütüphaneler Bölüm 4'te, hareket alanı varlıklarına ilişkin kütüphaneler Bölüm 5'te ve görselleştirme ile arayüz kontrollerine yönelik kütüphaneler Bölüm 6'da anlatılmaktadır. Bölüm 7'de, sonuç ve ileriye yönelik çalışmalar yer almaktadır.

2. MÜHATEM YAZILIM PROGRAMLAMA KÜTÜPHANESİ

MÜHATEM yazılım programlama kütüphaneleri C++ tabanlı olarak geliştirilmektedir. Şu an için API, Windows işletim sisteminde Borland C++ Builder ile Microsoft Visual C++ ve Linux tabanlı işletim sistemlerinde Borland Kylix'te çalışabilmektedir. Bunların dışında bir JNI ara katmanı yazılarak hem Windows'da hem de Linux'te JAVA tarafından kullanılabilirliği test edilmiştir.



Şekil 2. MÜHATEM Yazılım Kütüphaneleri

Hız ihtiyacı çok yoğun olan benzetim sistemlerinde, C++ kullanımı JAVA ve C# gibi yorumlayıcılara oranla çok daha fazladır. JAVA gibi dillerin birden fazla işletim sistemine kolay taşınabilirliği bu dillerin kullanımını cazip kılabilir ancak bu dillerde hazırlanan benzetim sistemlerinin performansları beklenen düzeyde olmamaktadır. C++ ile hazırlanan yazılımlarının standard C/C++ kütüphaneleri dışındaki komutları yoğun olarak kullanmamaları farklı işletim sistemlerine taşınabilirliklerini kolaylaştırmakta ve genelde sadece kullanıcı arayüzleri açısından geçişler zaman almaktadır. Hatta, platform bağımsız kullanıcı arayüz tasarım kütüphaneleri (TCL/TK [2], vb.) kullanan uygulamalarda bu geçiş de zaman almamaktadır. MÜHATEM'in geliştirilmesinde işletim sistemi, derleyici ortamı ve kullanıcı arayüzlerine özel komutların kullanımından azami düzeyde kaçınılmış, kullanımının gerekli olduğu durumlarda ileride anlatılacak olan *Sistem Erişim Kütüphanesi* üzerinden bir ara katman aracılığıyla erişim sağlanmıştır. MÜHATEM yazılım kütüphaneleri Şekil 2'de gösterilen katmanlı yapıya sahiptir. Bu katmanlar, müteakip bölümlerde anlatılan *alt yapı*, *çevre*, *varlıklar* ve *görselleştirme/arayüz kontrolleri* kütüphaneleri olarak dört temel kategoride ele alınmaktadır.

3. ALT YAPI KÜTÜPHANELERİ

Alt yapı kütüphaneleri, benzetim sistemleri tarafından yoğunlukla ihtiyaç duyulan genel fonksiyon ve algoritmaları içerir. Bu bölümde genelde MÜHATEM yapısına özelleşmiş kısımlar yer almaz. İlerleyen alt bölümlerde alt yapı kütüphaneleri özetlenmiştir.

3.1 Sistem Eriřim Kütüphanesi

Bu kütüphane, işletim sistemi kaynaklarına (metin, tarih, görüntü, ses, disk, hafıza, veritabanı, vb. işlemler) erişimi sağlayan fonksiyon ve nesne sınıflarını içerir ve derleyici/işletim sistemi özelinde kodları barındıran tek kütüphanedir. Diğer yazılım kütüphaneleri bu kütüphane üzerinden sistem kaynaklarına erişir. Böylece kodların derleyici ve platform bağımsız olması sadece bu kütüphanenin güncellenmesi ile sağlanır.

3.2 Temel Algoritmalar Kütüphanesi

Bu kütüphane, genel maksatlı algoritmaları ihtiva eder. Koordinat dönüşümleri (Coğrafi/UTM/kartezyen koordinat sistemi ve WGS-84/ED-50 datum dönüşümleri, vb.), vektör (çevirme, açı hesaplama, çizgi kesişim testleri, matris işlemleri, vb.) ve çokgen işlemleri (delaunay üçgenleştirmesi, çokgen üçgenleştirmesi, kesişim testleri, vb.), yapay sinir ağları, karar ağaçları, kořum zamanı deęişkenleri ve kořum zamanı yorumlayıcıları (run-time interpreter - script programlama) bunlara örnek olarak verilebilir. Ayrıca bu kütüphane içinde bütün üst katmanlardaki nesne sınıflarının kalıtım yolu ile türetildeęi *TObjectBaseClass* adlı temel nesne sınıfı bulunmaktadır. Bu nesne sınıfı, derleme (compile-time) ve kořum zamanı (run-time) deęişkenleri ile metodlarına kullanıcı arayüzleri ve script programlama ile erişimi sağlayan alt yapı metodlarına sahiptir. Bu nesneler üzerinde, C++ benzeri bir dilde script metodlar tanımlamak ve çalıştırmak mümkündür.

3.3 Dosya Yükleyicileri Kütüphanesi

Bu kütüphane, çeşitli formatdaki dosyaların yüklenmesini sağlayan nesne sınıflarını içerir. İmaj yükleme (Bitmap-*.bmp, Truevision TARGA-*.tga, Audio Video Interleave-*.avi, vb.), üç boyutlu model yükleme (3D Studio-*.3ds, Quake 2 Model File-*.md2, vb.), raster coğrafi veri yükleme (Digital Elevation Data-*.dt?, Digital Elevation Model-*.dem, Tagged Image File-*.tif, vb.), vektör coğrafi veri yükleme (ESRI Shape-*.shp, S-57, vb.), sembol yükleme (vektör ve imaj semboller) ve XML dosya yükleme olarak altı ana kategoride geliştirilmiştir. İleride ses dosyalarını yüklemeye ilişkin bir kısmın da eklenmesi planlanmaktadır.

4. ÇEVRE KÜTÜPHANELERİ

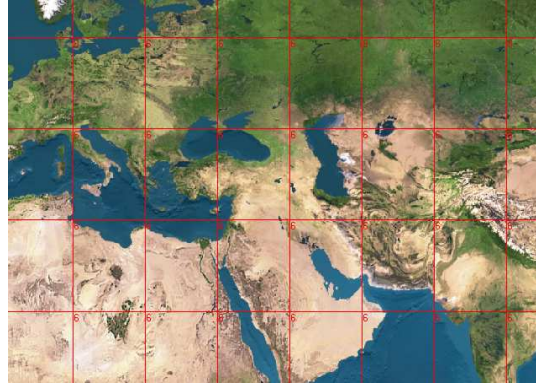
Bu kütüphane, kara, su alanları, atmosfer ve astronomiye ilişkin MÜHATEM veritabanlarının yaratılması ve kullanılmasını sağlayan nesne sınıflarını içerir. Çevre erişim, çevre standartları, çevre algoritmaları ve çevre güncelleme olmak üzere dört alt kütüphaneden oluşur.

4.1 Çevre Eriřim Kütüphanesi

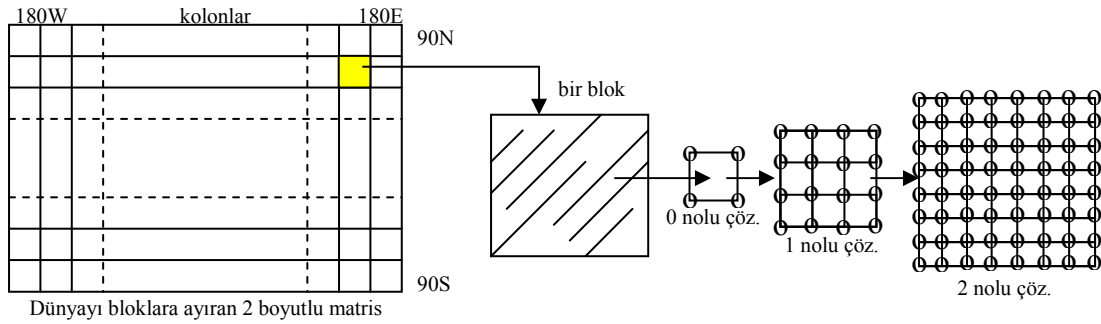
Çevre erişim kütüphanesi, çevre verilerini oluşturmak, bütünleştirmek, saklamak ve geri yükleyerek kullanabilmek amacıyla hazırlanmış jenerik bir yazılım kütüphanesidir. Raster tabanlı harita verileri, vektör tabanlı harita verileri ve nokta tabanlı zaman ilişkili coğrafi veriler olarak üç alt kütüphaneye bölünür.

4.1.1. Raster Tabanlı Harita Verileri

Raster tabanlı harita verileri alt kütüphanesi, tüm dünyayı coğrafi koordinatlar cinsinden eşit boyutlu bloklara bölerek 2 boyutlu bir matrise (bkz. Şekil 2) dönüştürür. Matrisin her elemanı (bloğu) kendi içinde çok çözünürlüklü olarak tutulan 2 boyutlu bir dizi matris daha barındırır. Bir blok 0 ile 9 arasında bir çözünürlükte yaratılabilir, kayıt edilebilir ve geri yüklenebilir. Çözünürlük, bir bloğun verilerini tutan 2 boyut matrisin boyutlarını ifade eder. Çözünürlüğü n olan bir bloğun verilerini tutan matrisinin boyutları $2^{(n+1)} \times 2^{(n+1)}$ dir. Dolayısıyla 9 nolu çözünürlüğe sahip bir matrisin boyutları 1024x1024'dür. Matrisin her elemanı bir coğrafi noktaya ilişkin istenilen tiplerde, bir yada daha fazla alanda veri tutar (yükseklik, derinlik, renk (RGB), vb.). Bir blok matrisinin en kuzey ve doğu kenarlarındaki elemanlar kuzey, doğu ve kuzey-doğudaki komşu blokların elemanlarıyla çakışır. Dolayısıyla n çözünürlüklü bir matris, bir bloğu $(2^{(n+1)}-1) \times (2^{(n+1)}-1)$ parçaya böler (bkz. Şekil 3).



Şekil 2. Dünyanın bloklara ayrılması



Şekil 3. Blokların çözünürlükleri içermesi

Bir bloğun maksimum 10 çözünürlük ile sınırlandırılmasının nedeni, bir blok için gerekli hafıza ihtiyacının çok artmamasının ve verileri yükleme ile işleme performansının çok fazla düşmemesinin garanti edilmesidir. Ancak bu sınırlama ile çok yüksek çözünürlüklü verilerin (örneğin 1 metre) tutulabilmesi için dünyanın çok fazla sayıda bloğa bölünmesi gerekir. Çok fazla blok kullanımı, yüksek çözünürlüklü ama dar alanda çalışan benzetim sistemleri (örneğin kara aracı simülatörleri) için idealdir ancak çok düşük çözünürlüklü ve geniş alanda çalışan bir benzetim sisteminde (örneğin JTLS) küçük küçük bir çok bloğun işlenmesinden dolayı performans çok fazla düşmektedir.

Dolayısıyla MÜHATEM’de dünya *stratejik*, *operatif* ve *taktik* olarak üç farklı detay grubunda bloklamaya tabi tutulmuştur. Yazılım geliştiriciler, ihtiyaçlarına göre bu detay gruplarının bir yada daha fazlasını aynı anda kullanabilmektedir.

Stratejik detay grubu, dünyayı 32x16 adet bloğa böler ve her blok en fazla 39,5 saniye (yaklaşık 1 km) çözünürlüklü veri tutabilir. Operatif detay grubu, dünyayı 2048x1024 adet bloğa böler ve her blok en fazla 0,6 saniye (yaklaşık 17 m) çözünürlüklü veri tutabilir. Son olarak taktik detay grubu, dünyayı 131072x65536 adet bloğa böler ve her blok en fazla 0,009 saniye (yaklaşık 27 cm) çözünürlüklü veri tutabilir. İleride daha yüksek çözünürlüklü verilere ihtiyaç olduğu durumda, mühendislik detay grubu eklenerek 4 mm düzeyine inmek mümkündür.

Anılan bloklar, kullanıcı açısından 2 boyutlu bir matris üzerinden erişiliyor gibi gözükse de aslında hafızada yer kaplamamaları açısından ağaç yapısında tutulmaktadır. Her blok bir birinden bağımsız olarak yaratılabilmekte, veritabanından yüklenebilmekte ve hafızadan silinebilmektedir. Bloklar istenilen çözünürlükte (veri mevcutsa) yüklenebilmektedir. Bir çözünürlük yüklendiğinde o çözünürlükten küçük olan tüm çözünürlüklerde hafızaya yüklenmektedir. Örneğin 3 nolu çözünürlüğün yüklenmesi aynı zamanda 0, 1 ve 2’nin de yüklenmiş olması anlamına gelir. Bu kütüphane kişiselleştirilebilir blok verileri tutmaya yönelik olup, yüklenen verilerin yorumlanmasını “Çevre Standartları Kütüphanesi”ne bırakır.

4.1.2. Vektör Tabanlı Harita Verileri

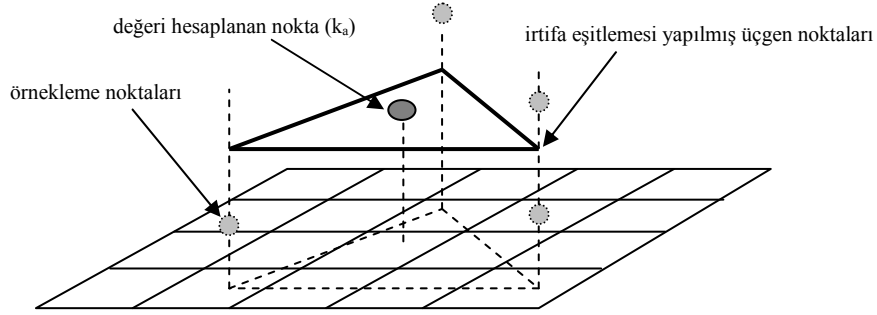
Vektör tabanlı harita verileri alt kütüphanesinin, tüm dünyayı adresleyebilecek şekilde vektör verileri tutabilmesi planlanmaktadır. Ancak bu kütüphanenin tasarımı henüz tamamlanmadığı için bu bildiride anlatılmayacaktır.

4.1.3. Nokta Tabanlı ve Zaman ilişkili Coğrafi Veriler

Nokta tabanlı ve zaman ilişkili coğrafi veriler kütüphanesi, tanımlanan bir dizi parametrenin zamana ve konuma bağlı olarak değişimini interpolasyon teknikleri ile hesaplayabilen kişiselleştirilebilir bir veri yapısı sunar. Tanımlanan parametreler (örneğin nem, sıcaklık, bulut yüksekliği, akıntı yönü vb.), sabit bir değere sahip olabilir, benzetim sistemi zamanına bağlı olarak önceden belirlenmiş bir plan (değişim çizelgesi) dahilinde değişebilir veya zaman ile coğrafyaya bağımlı olarak değerleri hesaplanabilir. Coğrafyaya bağımlı olarak değer tespiti için arazide belirli noktalar ile irtifalarda örneklenmiş ve ölçüm zamanları belirtilmiş parametre değerleri dizisi kullanılır. Bu değerler dizisi sayesinde, ölçümleri olmayan ara noktalar ve zamanların değerleri interpolasyon teknikleri ile hesaplanır.

Ara değerlerin hesaplanması için, öncelikle farklı koordinatlardaki ölçüm noktaları Delaunay üçgenleştirmesi ile bir üçgen ağına dönüştürülür. Ardından değeri hesaplanacak koordinatın (k_a) denk düştüğü üçgen bulunur. Sonra, üçgenin köşelerindeki noktaların, k_a noktası ile aynı irtifadaki değerleri interpolasyon teknikleri ile hesaplanır. İrtifa eşitlemesi yapıldıktan sonra, k_a noktasının değeri, irtifası eşitlenmiş nokta değerleri ile yapılan ikinci bir interpolasyon ile belirlenir (bknz. Şekil 4).

Bu veri kütüphanesi, katmanlardaki parametrelerin neler olacağı ve hangi verileri tutacağı konusunda her hangi bir kısıt getirmedigi için bu parametrelerin belirli bir standarda oturtulması “Çevre Standartları Kütüphanesi” aracılığıyla sağlanır.



Şekil 4. İrtifa eşitlemesi ve üçgen enterpolasyonu ile ara değerlerin hesaplanması

4.2 Çevre Standartları Kütüphanesi

Bu kütüphane, çevre erişim kütüphanesinin sunduğu kişiselleştirilebilir veri yapılarının kullanımını standartlara bağlar (sayısal yükseklik verilerinin “Elevation” adlı bir alanda 4 byte float olarak tutulması, uydu fotoğrafı verilerinin “R”, “G”, “B” adlı alanlarda 1’er byte tam sayı olarak tutulması, nem parametresinin isminin “Nem”, biriminin “%” olması, vb.). Çevre standartları kütüphanesi, yüzey, meteoroloji ve astronomi verileri olarak üç grupta ele alınmaktadır.

4.2.1. Yüzey Verileri

Kara ve su alanlarının yüzey/dip bilgilerine ilişkin raster harita standartları yüzey verileri kategorisinde ele alınır. MÜHATEM’in raster harita standartları; *sayısal yükseklik haritası*, *raster harita görüntüsü* (taranmış kağıt haritalar), *görünür ışıık bandı görüntüsü* (uydu görüntüsü, vb.), *kızılötesi görüntü*, *raster yüzey örtüsü haritası* (bitki örtüsü), *yüzey normalleri haritası* (yüzey bakış yönü) ve *su derinlik haritası* olarak belirlenmiştir. Vektör haritalara ilişkin standartlar ise henüz netleşmemiştir.

4.2.2. Meteoroloji Verileri

MÜHATEM’in meteoroloji verilerine ilişkin standartların belirlenmesinde Devlet Meteoroloji İşleri Müdürlüğü ile koordineli çalışılmıştır. Meteorolojik bilgiler, Zamana dayalı parametre veritabanı kullanılarak saklanmış ve Türkiye’nin 2000-2004 yılları arasındaki ölçümlerine dayalı istatistikler çıkarılmıştır. Bu sayede atmosfer verilerinin uzun dönem ortalama ve standart sapma bilgileri elde edilmiş ve benzetim sistemlerinde kullanılmak üzere, enterpolasyon teknikleri ile farklı zaman ve konumlardaki parametre değerleri gerçekçi bir şekilde tahmin edilebilmiştir.

MÜHATEM’in standart meteoroloji parametre ve birimleri; *sıcaklık* (°C), *mahalli basınç* (Mb), *buhar basıncı* (Mb), *nispi nem* (%), *görüş mesafesi* (m), *rüzgar hızı* (m/saat), *rüzgar yönü* (derece, kuzey 0 ve (+) açı saat yönünde), *bulut yoğunluğu* (kapalılık oranı, 0..1), *bulut yüksekliği* (m, deniz seviyesinden), *yağış türü* (0 = yağmur, 1 = kar, 2 = dolu) ve *yağış yoğunluğu* (mm) olarak belirlenmiştir. Sis yüzdesi, parametre olarak kullanılmamaktadır çünkü sis yatay görüş mesafesi parametresi içine dolaylı olarak ithal edilmiş durumdadır. Yatay görüş mesafesi parametresinin yağışsız bir durumu kapsadığı, yağışın görüşe etkisini içermediği varsayılmaktadır.

4.2.3. *Astronomi Verileri*

Astronomi verileri ay, güneş ve yıldızlarını durumunu içermektedir. MÜHATEM’de, güneş ve ayın konumu, durumu ve gökyüzü aydınlanmasına etkileri, verilen bir konum ve zamana (tarih, saat ve GTM) göre belirlenmektedir. Güneşin ve ayın gökyüzündeki konumu, *pusula açısı* (0 derece kuzey ve (+) açı saat yönünde artar) ve *yükseliş açısı* (0 derece ufuğa, 90 derece gökyüzüne ve -90 derece yer yüzüne doğru), ayın durumu ise *aydınlık yüzey oranı* (0 ile 1 arası) ile ifade edilmektedir. Aydınlatma etkileri ise *genel aydınlanma* (ambient) ve güneş ile ay için *doğrudan aydınlatma* (diffuse) olarak üç değer ile belirlenir. Genel aydınlanma, çevrenin ortalama aydınlanma miktarıdır ve güneş ile ay ışığının doğrudan vurmadığı noktalarda bile yansımalar dolayısıyla en az genel aydınlanma kadar bir ışığın olacağı varsayılır. Doğrudan aydınlatma ise ışık kaynağından gelen ışığın yüzeye dik olarak bir engele raslamadan çarptığı durumda meydana gelen aydınlanma miktarıdır. Toplam aydınlanma değerinin 1 olduğu noktalarda, yüzey rengi en parlak halinde (örneğin uydu görüntüsünün orjinal renkler tonlarında) görülür.

4.3 Çevre Algoritmaları Kütüphanesi

Bu kütüphane, Çevre Standartları Kütüphanesi tarafından sunulan veri yapıları üzerinde çalışan bir dizi algoritma sunar. Algoritmalar, nokta tabanlı, çizgi tabanlı, alan tabanlı ve diğer olmak üzere dört kategoride ele alınır. Nokta tabanlı algoritmalar, belli bir noktadaki yükseklik, derinlik, bitki örtüsü gibi verileri okuyan fonksiyonları, çizgi tabanlı algoritmalar, iki nokta arasındaki görüş hattı testi, profil çıkarma gibi işlemleri yapan fonksiyonları ve alan tabanlı algoritmalar, pencereleme mekanizmaları kullanımı ile hızlı alan yükleme/kaydırma, bentler ile bölgeyi su doldurma, kapsama alanı hesaplama gibi çeşitli işlevleri sunmaktadır.

4.4 Çevre Güncelleme Kütüphanesi

Bu kütüphane, Çevre Standartları Kütüphanesinde tanımlanan veri standartlarına uyumlu çevre verilerinin kaynak veri dosyalarından (DEM, DTED, Geo-Tif, vb.) yüklenerek veritabanının güncellenmesini sağlamaktadır.

5. VARLIK KÜTÜPHANELERİ

Varlık kütüphaneleri, varlıkları tanımlamak ve üç boyutlu modellerini oluşturmak için gerekli her türlü işlevi içermektedir. Varlıklar iki farklı yolla tanımlanabilmektedir. Bunlardan ilki, *harekat alanı kavramsal modeli katmanı* ile tanımlama, ikincisi ise *varlık erişim katmanı* ile tanımlamadır.

Kavramsal model katmanı kullanıldığında, varlıkların öznitelikleri ve diğer varlıklarla olan ilişkileri kavramsal olarak ifade edilir. İki yada daha çok varlık arasında cümleler şeklinde ilişkiler tanımlanması ve bu ilişkilerin detaylı olarak sorgulanabilmesi mümkündür. Bu yetenek fazla kullanılmamaktadır.

Varlık erişim katmanı ise temsili ortamda yer alan varlıkların çalışabilir modellerinin oluşturulabilmesi için geliştirilmiştir. Varlık erişim katmanlarının en önemli özelliği eklenti (plug-in) mekanizması ile sınırsız genişleyebilme kabiliyetidir. Bu kabiliyet sayesinde, hem MÜHATEM geliştirici ekibinin hem de diğer kurum/kuruluş ve firmaların, MÜHATEM’e yeni ve özelleşmiş varlık tipleri ekleyebilmesi veya eskiden

tanımlanmış varlıkları değiştirebilmesi mümkün olmaktadır. Hali hazırda, model, çizim, sembol, resim, ışık, ses, parçacık efektleri gibi özellikleri sunacak eklentilerin geliştirilmesine devam edilmektedir. Varlık erişim kütüphanesi ile eklenti alt yapısı;

- *TMhtmlObjectLayerContainer*,
- *TMhtmlObjectLayer*,
- *TMhtmlObject* ve
- *TMhtmlObjectResource* adında dört temel nesne sınıfından oluşmaktadır.

Söz konusu nesne sınıflarından son ikisi olan *TMhtmlObject* ve *TMhtmlObjectResource*'dan kalıtım yolu ile yeni nesneler türetilmesi ve DLL (Dynamic Link Library)'ler yolu ile varlık erişim katmanına (*TMhtmlObjectLayer*'a) tanıtılması mümkündür.

5.1 TMhtmlObjectLayerContainer

TMhtmlObjectLayerContainer, varlık katmanlarının listesini barındıran ve bu katmanları ekleme, silme, arama, yükleme, kayıt etme gibi işlemler yapabilen nesne sınıfıdır. Görevi sadece bir dizi katmanı bir arada tutabilecek bir depolama ortamı sunmak ve katmanlara biricik (unique) tanımlama numaraları (ID'ler) atamaktır.

5.2 TMhtmlObjectLayer

TMhtmlObjectLayer, bir dizi varlığı (tank, uçak, duvar, bina, ışık, algı cihazı, silah, sembol, yazı, vb.) barındırabilen katmanların yaratıldığı nesne sınıfıdır. Katmanlara varlıklar eklenebilir, varlıkların birbirine kalıtım ve parça-bütün ilişkileriyle bağlanması sağlanabilir ve katmana atanan simülasyon koşumu başlangıç ve anlık tarih/saati ile tanımlanmış varlık davranışlarının zamana dayalı olarak ilerletilmesi sağlanabilir. Katmanlar başka katmanlara bağlantı kurarak bu katmanlardaki varlıklar ile kendi içlerindeki varlıklar arasında kalıtım ilişkileri kurabilir. Örneğin "Classes" adlı bir katman ile benzetim sisteminde kullanılacak varlık tiplerini (sınıflarını) tanımlamak ve "Instances" adlı ikinci bir katman ile "Classes" katmanına bağlanarak buradaki varlık tiplerinin kopyalarını "Instances" katmanına aktarmak, böylece bir senaryo oluşturmak mümkündür. Başka bir varlıktan kopya çıkarılırken, kopyası çıkarılan varlığa ek olarak kopyalanan varlık üzerinde yer alan diğer varlıkların da (örneğin bir uçağın üzerindeki pervaneler) toplu olarak kopyalanması mümkün olmakta, benzer şekilde bir varlık silindiğinde ona bağlı tüm alt varlıkların silinmesi de sağlanabilmektedir.

5.3 TMhtmlObject

TMhtmlObject, varlıkların türetildiği temel eklenti sınıfıdır. Bu sınıf, işlevleri devir alınarak değiştirilebilir bir dizi *virtual* metod sunmakta, böylece farklı işlevlere sahip eklentilerin oluşturulması mümkün olmaktadır. Katmanlara eklenecek varlıklar, *TMhtmlObject* sınıfından türetilmiş eklentiler ile yaratılabilmekte, varlıkların isim, açıklama ve coğrafi konumları tanımlanabilmekte ve varlıklar başka varlıklarla kalıtım ve parça-bütün ilişkileri kurabilmektedir. Varlıkların coğrafi konumları (koordinat ve doğrultuları), nokta, çoklu nokta, çoklu çizgi, kapalı uçlu çoklu çizgi veya çokgen olarak tanımlanmakta ve bir varlık başka bir varlığa bağlanarak onunla birlikte hareket edebilmektedir. Varlıklar, kendilerinin eklenti olmasına ek olarak, varlık özelliklerini

genişletmeyi amaçlayan bir dizi alt eklenti de (kaynak eklenti - resource plug-in) içerebilmektedir. Kaynak eklentiler, müteakip bölümde anlatılmaktadır.

5.4 TMhtmlObjectResource

TMhtmlObjectResource, varlıkların şekil, sembol, ışık, algılama, yorumlama, haberleşme gibi özelliklerini belirleyen eklentilerin (kaynak eklentilerin) türetildiği nesne sınıfıdır. Bu eklentiler genişleyebilirliğin sağlanmasında varlık eklentilerine oranla çok daha fazla öneme sahiptir çünkü varlık çeşitliliği ancak varlık eklentilerinin sayısı kadar olabilir ve bir varlık aynı anda birden fazla eklenti sınıfından türetilemez. Bununla birlikte tek bir sınıftan türetilmiş varlık, farklı kombinasyonlardaki kaynak eklentileri ile çok farklı özellikte varlığa dönüşebilir.

6. GÖRSELLEŞTİRME VE ARAYÜZ KONTROLLERİ KÜTÜPHANESİ

Görselleştirme Kütüphanesi, raster/vektör haritalar, arazi, atmosfer, varlıklar, karar ağaçları gibi çeşitli verilerin görselleştirilmesine yönelik sınıflar sunmaktadır. Arayüz Kontrolleri Kütüphanesi ise görselleştirme kütüphanesini kullanarak pratik 2/3 boyutlu ortam görselleştirmesi yapabilen ve etkileşimli şekilde pencere görünüm bilgilerini (harita konumu ile yakınlığı, koordinat çizgileri vb.) değiştirebilen nesne sınıflarını içermektedir. Bu nesne sınıfları, programlama platformu bağımlı olacak şekilde doğrudan kullanıcı arayüzlerine erişim yapmamakta ancak arayüzlerinden çeşitli metodlar ile bilgiyi alıp kullanıcı arayüzlerine callback fonksiyonları ile geri bildirimler sağlamaktadır.

7. SONUÇ

Bu bildiride, ODTÜ TSK MODSİMMER tarafından 2002 yılından beri geliştirilmesine devam edilen MÜHATEM Projesi kapsamında oluşturulmuş yazılım programlama kütüphanesi kısaca tanıtılmıştır. Bu projenin temel amacı, tüm modelleme ve simülasyon projelerinde ortak ihtiyaç olan temsili ortam modelinin, tek bir alt yapı tarafından sunulmasıdır. Kütüphanenin ara sürümleri, “Küçük Ölçekli Harekatın Modellenmesi ve Simülasyonu (BASKIN)” ile “Sınır Gözetleme ve Kontrol Sistemi C4ISR Faaliyetlerinin Modellenmesi ve Simülasyonu (SGKS C4ISR-MOS)” projelerinde kullanılmış, “Müşterek Görev Kuvvetinin Modellenmesi ve Simülasyonu (MGKMOS)” ile “Satıhta Konuşlu Aktif Hava Savunma Harekatının Modellenmesi (SAMMOS)” projelerinde kullanılmakta ve “Deniz Harp Oyunu Simülasyon Sistemi (DEHOS)” projesinde kullanımı planlanmaktadır.

KAYNAKÇA

- [1] ODTÜ TSK MODSİMMER, (2006), “MÜHATEM Yazılım Geliştirici Kılavuzu”, ODTÜ TSK Modelleme ve Simülasyon Araştırma ve Uygulama Merkezi.
- [2] Brent Welch, Ken Jones, and Jeff Hobbs, (2003), “Practical Programming in Tcl and Tk”, 4th Edition ISBN: 0-13-038560-3.

EK BİLGİ

Dr.Çağatay Ündeğer

Adres : STM A.Ş., Kafkas Sk. No:56, Beştepe, Ankara

e-posta : cundeger@stm.com.tr

Telefon : 0 (312) 402 38 08

Fax : 0 (312) 425 24 65

Bnb.Altan Yavaş

Adres : Genelkurmay BİLKARDEM Bşk.İği, Bakanlıklar, Ankara

e-posta : ayavas@tsk.mil.tr

Telefon : 0 (312) 402 38 07

Fax : 0 (312) 425 24 65

Bnb.Mustafa Ocak

Adres : Genelkurmay BİLKARDEM Bşk.İği, Bakanlıklar, Ankara

e-posta : mocak@tsk.mil.tr

Telefon : 0 (312) 402 38 44

Fax : 0 (312) 425 24 65

Yb.Umut Akın

Adres : Genelkurmay BİLKARDEM Bşk.İği, Bakanlıklar, Ankara

e-posta : uakin@tsk.mil.tr

Telefon : 0 (312) 402 38 20

Fax : 0 (312) 425 24 65

Alb.Ziya İpekkan

Adres : Genelkurmay BİLKARDEM Bşk.İği, Bakanlıklar, Ankara

e-posta : zipekkan@tsk.mil.tr

Telefon : 0 (312) 402 38 05

Fax : 0 (312) 425 24 65