

Dağıtık Simülasyon Sistemlerinde Performans Analiz Altyapısı

Cemil Akdemir¹ Cüneyt Ay² Hüsnü Karaküçük³ Yasemin Timar⁴

Öz

Dağıtık simülasyon sistemlerinin önemli özelliklerinden biri tekrarlanması zor veya imkânsız olan durumların simülasyonunu gerçekleştirmeleridir. Simülasyon gerçekleşirken bu durumların kaydedilmesi, bu durumların simülasyon sonrası pek çok kez tekrarlanabilmesini, incelenebilmesini ve sistemin performansının analiz edilebilmesini sağlamaktadır. Simülasyon sonrası oluşturulan analiz verileri karar destek sistemleri için önemli girdiler oluşturmaktadır. Bu bildiri kapsamında HLA1516 tabanlı dağıtık simülasyon sistemlerinde kullanılabilen senaryo koşumu boyunca simüle edilen platform bilgilerinin ve gerçekleşen önemli olayların, senaryo koşumu sonrasında analiz edilmesi için geliştirdiğimiz performans analiz altyapısı anlatılmaktadır. Performans analiz yazılım altyapısının öne çıkan özelliklerinden biri senaryo koşumu esnasında gerçek zamanlı olarak iletilen mesajları simülasyon ortamının performansını olumsuz yönde etkilemeyecek, sistem yükünü arttırmayacak şekilde kayıt işlemini gerçekleştirmesidir. Altyapının önemli bir modülü olan kayıt veritabanı ise, ham hâlleri ile sayısallaştırma, grafik çizimi ve karşılaştırma işlemlerine uygun olmayan kayıt verilerini, analize uygun biçimde yapılandırmak için kullanılan veriyapılarını ve fonksiyonları içermektedir. Yazılım altyapısının grafik modülünde ise, yapılandırılan verilerin 2 boyutlu grafiklerinin çizilmesi ve karşılaştırılması sağlanmaktadır. Geliştirdiğimiz performans analiz altyapısı tanımlanan özelliklerin tümünü birleştirmesi ile benzer eylem sonrası gözden geçirme (After Action Review, AAR) sistemlerinden büyük farklılık göstermekle birlikte dağıtık tabanlı simülasyon sistemlerine uyarlanabilir olmasıyla geniş bir kullanım alanına sahiptir.

Anahtar Kelimeler: Eğitim Simülasyonu, HLA1516, Kayıt ve Oynatma, Performans Analizi, Simülasyon Veritabanı

Performance Analysis Framework for Distributed Simulation Systems

Abstract

One of the most important properties of the distributed simulation systems is to provide simulating systems and situations which hardly reoccur. Recording of the simulation data during an exercise makes it possible to inspect and analyze the system performance repeatedly after the simulation. Analysis of the collected simulation data produces important inputs for After Action Review and Decision Support Systems. Within the scope of this paper, the after action review software framework, which we have developed for the purpose of recording platform information and

¹ Yazışma adresi: TÜBİTAK BİLGEM Bilişim Teknolojileri Enstitüsü, Gebze, Kocaeli, cemil.akdemir@bte.tubitak.gov.tr

² TÜBİTAK BİLGEM Bilişim Teknolojileri Enstitüsü, Gebze, Kocaeli.

³ TÜBİTAK BİLGEM Bilişim Teknolojileri Enstitüsü, Gebze, Kocaeli.

⁴ TÜBİTAK BİLGEM Bilişim Teknolojileri Enstitüsü, Gebze, Kocaeli.

important events occurring within a scenario-run in an HLA1516-based simulation system, is explained in detail. One important feature of our framework is collecting simulation data in real-time during an exercise without causing any performance penalty or affecting the overall system. The essential data subsystem has a relational database system which contains the data structures and functions suitable for analytical comparison and graphical statements, since with their initial format, the recorded data is not ready for use analytical drawing and comparison operations. Within the graphics subsystem user interfaces for drawing 2D graphics of the processed data for analyzing operations are developed in the framework. Our performance analysis framework is distinguished from the similar works in AAR by combining all of those features and it has a wide range of application area as being customizable to be used in distributed simulation systems.

Keywords: After Action Review (AAR), HLA1516, Performance Analysis, Simulation Database, Training Simulation.

Giriş

Özellikle eğitim amaçlı kullanılan benzetim uygulamalarının önemli bir özelliği, eğitim seansının ardından önemli bir değerlendirme safhası içermesidir. Değerlendirme safhası eğitim boyunca gerçekleşen durumların, olayların, tepkilerin ve etkileşimlerin gözden geçirilmesini, bunların analiz edilmesini ve sayısal olarak eğitimin değerlendirilmesini içerir. Dağıtık simülasyon sistemlerinde yaygın olarak kullanılan HLA/RTI altyapılarında değerlendirme süreci üç ana adımda yapılmaktadır; (1) Eğitim boyunca kayıt yapılması, (2) Kayıt verilerinin veri tabanına aktarılması, (3) Veri tabanından kayıtların analiz ve tekrar oynatma amacıyla okunması. Var olan eğitim sonrası değerlendirme sistemlerine örnek olarak; General Dynamics tarafından geliştirilmiş olan S2Focus “Analyzer” (<http://www.gdc4s.com>, 2011), MAK tarafından geliştirilmiş olan MAK Data Logger (<http://www.mak.com/products/datalogger.php>, 2011) ve Pitch firması tarafından geliştirilmiş olan Pitch Recorder (<http://www.pitch.se/products/recorder>, 2011) sayılabilir. Bu araçların tümü eylem sonrası gözden geçirme (After Action Review, AAR) amacıyla geliştirilmiştir.

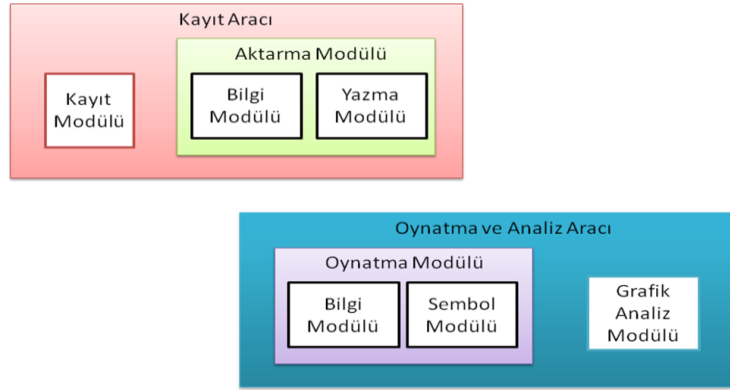
Her birinde kayıt edilen simülasyonun yeniden oynatılması sağlanabilir. Ancak bu yazılımlar, sayısal verilerin analitik değerlendirilmesini sağlayacak 2 boyutlu grafikler içermemektedir. Veritabanı kullanım imkânları kısıtlıdır. Performans analizinde temel kriterlere göre sayısal verilerin incelenmesi, karşılaştırılması ve analitik çıkarımların yapılması esastır. Veritabanı kullanımı S2Focus'ta mümkün değilken MAK Data Logger uygulaması ise sadece verileri veritabanına aktarma yeteneğine sahiptir. Yapılan çalışmada diğer AAR yazılımlarından farklı olarak tüm kayıt, oynatma ve analiz imkânlarını beraber içeren bir yazılım altyapısı geliştirilmiştir.

Bu bildiriye tanıtılan eğitim sonrası değerlendirme altyapısının ayırt edici teknik özellikleri; (1) HLA/RTI altyapısında kullanılan Federate Object Module (FOM) da tanımlı veri yapıları ile otomatik veri tabanı şeması oluşturulması, (2) veri tabanından analiz işlemlerinde kullanılacak C++ sınıflarının otomatik oluşturulması, ve (3) analiz işlemlerinde faydalı olan grafiksel gösterim olanaklarının sunulması, olarak sıralanabilir.



Şekil 1. Kayıt ve Analiz İşlemleri

Performans analiz altyapısında kayıt, veri aktarma, oynatma ve analiz işlemlerini gerçekleştiren 2 ana yazılım geliştirilmiştir; (1) Kayıt Aracı ve (2) Oynatma ve Analiz Aracı. Bu yazılımların alt modülleri Şekil 2’de görülmektedir.



Şekil 2. Performans Analiz Araçları

Veri Yapıları

Dağıtık simülasyon altyapılarında kullanılan HLA, 90’lı yılların ortasında geliştirildi ve 1998’de HLA 1.3 (DMSO, 1998) olarak standard hâline getirildi. Bu standart, 2 önemli revizyon geçirdi: IEEE 1516-2000 (IEEE, 2000) ve IEEE1516-2010 (IEEE, 2010). HLA standartlarında, birlikte çalışacak simülasyon sistemlerinin veri alışverişi için kullanacakları veri yapıları ve modellerin standartları Nesne Modeli Şablonu (Object Model Template-OMT) tanımlamaları ile belirlenmiştir (IEEE Std. 1516.2, 2000). HLA içinde Federasyon Nesne Modeli (Federate Object Model -

FOM) olarak adlandırılan bu model, HLA federasyonuna dahil olan federeler tarafından ortaklaşa kullanılır. Federasyonun alt yapısı için yapısal bir obje modelinin bulunması ve bu modelin FOM standardına uygun olarak tanımlanması:

- Federasyona dâhil federelerin birbirleri ile iletişimi ve veri alışverişinde ortak ve anlaşılır bir yapı içerdiği,
- Federelerin yayımladıkları ve aldıkları bilgiler nazarında yeteneklerinin tanımlanmasını sağladığı,
- HLA nesne modelleri yaratılması için yazılım araçları geliştirmesine olanak verdiği için vazgeçilmez bir önem taşımaktadır (IEEE Std. 1516.2, 2000).

Federasyon geliştirme sürecinin (FEDEP) kavramsal modeli geliştirme safhasında FOM oluşturulmaya başlanır. Kavramsal modelleme sürecinde kullanılabilen pek çok araç arasında Temel Nesne Modeli (Base Object Model) de sayılabilir (Timar Y., Bıkmaz İ., Taşdelen İ., Akgün S., Dikenelli O., 2007).

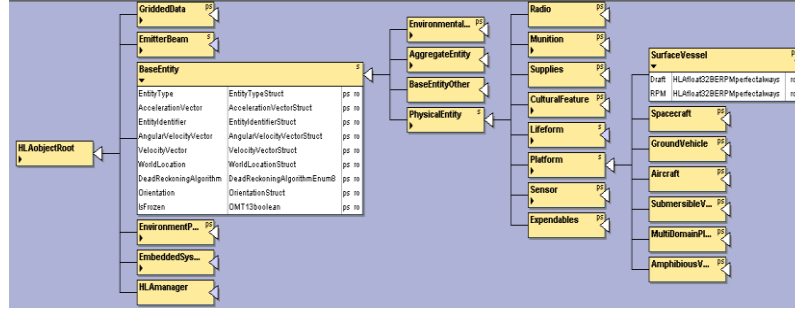
Simülasyon koşumu esnasında HLA federasyonunda kullanılan veri yapıları, koşum sonrası yapılacak değerlendirmeler için esas veri yapılarını oluşturmaktadır. Bu veri yapılarının değerlendirme altyapısına en uygun formatlardan biri olan veri tabanı yapılarına aktarılması analiz olanaklarını büyük ölçüde geliştirmekte ve zenginleştirmektedir.

FOM Nesne ve Mesaj Tanımları

HLA federasyonuna katılan federeler simülasyon koşumu boyunca yayımlayacakları ve abone olacakları (publish/subscribe) nesne ve etkileşim sınıfları ile ilişkilendirilirler. Bir federe için oluşturulan nesne modeline Simülasyon Nesne Modeli (Simulation Object Model) adı verilirken tüm federelerin ilişkilendirildikleri nesne ve etkileşim modelleri FOM'da sınıf tanımlaması olarak bulunmaktadır. HLA1516 FOM tanımlaması nesne tabanlı modellerle uyumlu olan XML formatında yapılmaktadır.

FOM oluşturmada varolan FOM'ların temel olarak kullanılması ve geliştirilmesi yöntemi tercih edilebilir. Bu yöntem federasyonun ve/veya simülatörlerin birlikte çalışabilirlik ve tekrar kullanılabilirlik özelliklerini pekiştirmektedir. Özellikle askeri benzetim projelerinde Realtime Platform Reference FOM (RPR FOM) (SISO, 2003) yaygın olarak kullanılmaktadır. Aşağıdaki şekillerde RPR-FOM'un geliştirilmiş bir durumunu içeren bir FOM'un bir bölümü görülmektedir. Şekil 3'te görülen nesnelerin hiyerarşik

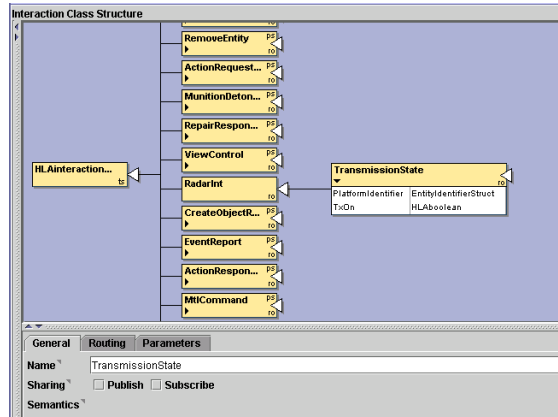
olarak tanımlanması ile alt sınıfların, üst sınıfların tüm özniteliklerine sahip olması sağlanmıştır. SurfaceVessel nesnesi, BaseEntity'de bulunan WorldLocation, VelocityVector, Orientation vb. bilgileri içermektedir.



Şekil 3. FOM Nesne Tanımları

Şekil 4'te görülen etkileşim tanımlamalarında hiyerarşinin yanında nesnelere ilişkili parametrelerin nasıl tanımlandığına örnek verilmektedir. SurfaceVessel nesnesinin radarının transmisyon durumu bu etkileşim ile ifade edilmektedir.

Nesnelerin öznitelikleri ve etkileşimlerin parametrelerinin veri tipleri temel basit veri tipleri (doğal sayı, gerçel sayı, boolean vb.) olabildiği gibi, diziler veya birkaç basit veri tipinin birlikte bulunduğu karmaşık veri tipleri de olabilir. Karmaşık veri tiplerinin kayıt, oynatma ve analiz işlemlerinde işlenmesi basit tiplere indirgenerek yapılmaktadır.

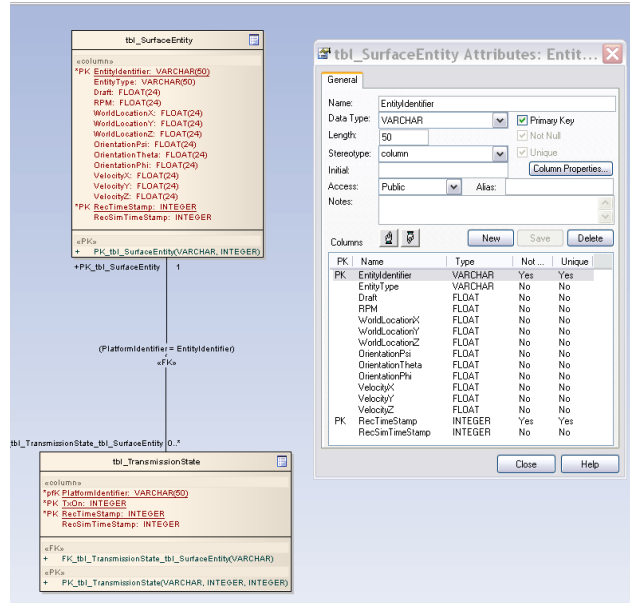


Şekil 4. FOM Etkileşim Tanımları

FOM-Veritabanı Eşleşmesi ve Veritabanı Oluşturulması

Analiz işlemleri için veritabanı kullanmanın çok daha verimli olduğu açıktır. FOM'da tanımlı nesne ve etkileşimlerin veritabanına aktarılması için

ilk adım her nesne ve etkileşim için tablo tanımlamaktır. Bu tablonun sütunları kaydedilecek, oynatılacak ve analiz adilecek öznitelikleri içerecek biçimde tanımlanır. Tüm tablolarda kayıt zamanı ve simülasyon zamanı bilgilerini içerecek sütunların bulunması önemlidir. Bölüm 2.1’de örnek olarak verilen nesne ve etkileşim için örnek veritabanı şablonu Şekil 5’teki gibidir. Bu tanımlamalar Unified Modelling Language (UML) tabanlı yazılımlar kullanılarak yapılmıştır. UML aracının otomatik olarak veritabanı şeması oluşturma özelliği ile güncelleme ve tekrar kullanılabilirlik özellikleri sağlanmıştır. Karmaşık tipler basit tiplere indirgenerek tablolarda ifade edilmektedir. Benzetimi yapılan nesnenin koşum esnasında birden fazla örneği oluşmuş ise her bir varlık için benzer yapıda tablolar otomatik olarak oluşturulabilmektedir. Karmaşık tiplerin yapısına göre her karmaşık tip için ayrı tablo tanımlanarak birincil ve yabancı anahtar (Primary Key-Foreign Key) ilişkisi ile tablolar arası bağlantı sağlanabilmektedir.



Şekil 5. Veritabanı şeması

Veritabanı şemasının belirlenmesinden sonra analiz amaçlı sorgular için gerekli prosedürler (Stored Procedure) veritabanına eklenir. Her koşum için ayrı bir içerik tanımlıdır. İlerleyen bölümlerde anlatılacağı gibi kayıt yükleme işlemleri esnasında tablolar temizlenmekte ve kayıt verileri ile doldurulmaktadır.

Kayıt

Kaydedilecek Verilerin Belirlenmesi

Kayıt uygulaması tarafından performans ve analiz işlemlerinde değerlendirilmek üzere simülasyon ortamından hangi nesne ve etkileşim sınıflarının kayıt edileceği bir konfigürasyon dosyası ile belirtilmektedir. Bu dosyada nesne sınıflarına özellik düzeyinde abone olunurken, etkileşim sınıflarına sınıf düzeyinde abonelik söz konusudur. Özellikle kaydedilen verinin veritabanına aktarımından sonra veriler arasında ilişkisel sorguların yapılabilmesi amacıyla, nesne ve etkileşimler içerisinde yer alan bu tipte özellik ve parametrelerin bu konfigürasyon dosyasında belirtilerek kayıt edilmesi önem taşımaktadır.

Şekil 6, kayıt uygulaması tarafından kayıt edilen bir etkileşim ve nesne sınıfı bilgisini içermektedir. Bu örnekte SampleObject sınıfının primaryKey özneliği ile SampleInteraction etkileşiminin foreignKey parametresi simülasyon ortamında aynı tip veri ile temsil edilip veri, veritabanına aktarıldıktan sonra da bu iki tablo arasındaki sorgular içerisinde ilişki kurulması amacıyla kullanılan alanları temsil etmektedir.

```
<Interaction
FullName="HLAinteractionRoot.SampleInteraction">
  <Parameter ParamName="foreignKey" />
  <Parameter ParamName="parameter2" />
</Interaction>

<Object FullName=" HLAobjectRoot.SampleObject">
  <Attribute AttribName="primaryKey" />
  <Attribute AttribName="attribute2" />
</Object>
```

Şekil 6. Nesne ve Etkileşim Tanımı

Kayıt İşlemi

Kayıt federesinin dahil olduğu federasyon tarafından ihtiyaç duyulan yaşam döngüsü akışını sağlaması önemlidir. Bu akış simülasyonda belirli bir işlemin devam edebilmesi amacıyla federasyona dahil olan tüm federelerden bir yanıt alınmasını bekleyen işlemleri içermektedir. Kayıt federesi, bu tip işlemler için gerekli olan nesne ve etkileşimleri alarak, çevrimin uygun şekilde tamamlanabilmesi amacıyla gerekli etkileşim ve nesnelere simülasyon ortamına gönderebilme yeteneğine sahiptir.

Simülasyon verilerinin analizlerinin düzgün şekilde yapılabilmesi düzgün sırada ve şekilde etiketlenmelerini gerektirmektedir. Simülasyon

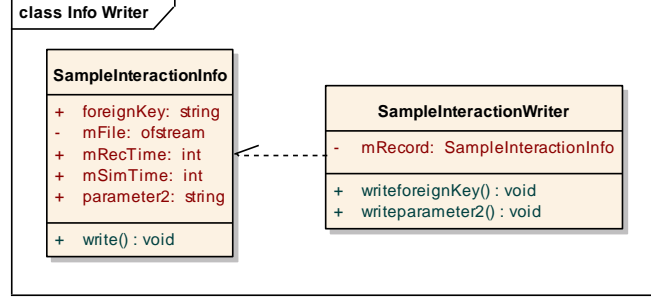
sisteminde zaman işletimi, sistem gereksinimlerine bağlı olarak gerçek zamandan farklı bir şekilde olabilmektedir. Örneğin, koşum esnasında hızlı şekilde gerçekleşmesi istenen akışlarda simülasyon zamanı gerçek zamana göre daha büyük ölçekte işletilebilmektedir. Bazı sistemlerde, simülasyon akışı durdurulup kaldığı yerden tekrar işleme devam etme olanağı sunulduğu gibi daha karmaşık sistemler etiketlenen önemli olay anlarına koşum esnasında ileriki bir zamanda tekrar dönüşü de mümkün kılmaktadır. Simülasyon zamanı olarak bahsedilen bu zaman yukarıda belirtilen olaylar gözönüne alınarak işletilmektedir.

Simülasyon zamanından bağımsız olarak veriler gerçek zaman değerlerine göre analiz edilmek istenebilir. Bu durumda simülasyon saatinin durdurulması ya da geri alınması gibi işlemlerin analiz zaman etiketlerini etkilememesi gerekir. Bu amaçla kayıt federesi simülasyon zamanının yanında ayrı bir zaman işleterek kayıtları hem simülasyon zamanı ile hem de gerçek zamanlı olarak etiketlemektedir. Analiz esnasında ise kayıtlar için hem gerçek zaman etiketi hem de simülasyon zamanı etiketi sergilenmektedir. Örnek olarak ilk 10 dk. boyunca koşturulan sonraki 5dk. boyunca durdurulan bir koşum için, toplam kayıt zamanı 15 dk. olacak ve analiz sırasında ilk 10 dk. boyunca gerçek zaman ve simülasyon zamanı etiketleri aynı seyrederken 10.dk'dan itibaren simülasyon zamanı değeri 00:10:00 değerinde kalırken gerçek zaman etiketi 00:15:00 değerine kadar ilerleyecektir.

Kayıt federesi konfigürasyon dosyasında belirtilen tüm nesne ve etkileşimleri herhangi bir çözümlenme (decode) işlemi yapmadan belirtilen iki zaman etiketi ile birlikte bir dosyaya aktararak kayıt etmekte ve koşumun sonlanması ile birlikte kayıt işlemini de sonlandırmaktadır.

Kayıt Verilerinin Veritabanına Aktarılması

Kaydedilen her nesne ve etkileşim sınıfı için bu sınıfların veritabanındaki ilgili tabloya aktarılmasını sağlayacak C++ sınıfları otomatik kod üretme teknikleri ile üretilmektedir. Her bir nesne ve etkileşim sınıfı tipi için 2 ayrı sınıf üretilmektedir. Bu sınıflardan ilki nesne veya etkileşimin sahip olduğu özniteliklere karşı gelen alanları içeren bilgi sınıfı olup bu sınıf ayrıca kaydın zaman etiketlerini de saklayacak alanları içerir. Bu sınıf sahip olduğu alanları bir dosyaya virgül ayrımlı olarak yazar ve üretilen bu dosya daha sonra veritabanındaki ilgili tabloya veri olarak aktarılır. Üretilen diğer sınıf olan çözücü sınıf ise, kayıt dosyasından okunan her bir çözümlenmemiş simülasyon verisini çözümlüyüp her bir parametrenin bilgi sınıfı içindeki karşı gelen alana atanmasını sağlar.



Şekil 7. Bilgi ve Çözücü Sınıfları

Çözömlenen ve dosyalara aktarılan veriler bu hâlleri ile veritabanında karşı gelen tablolara aktarılmaya hazır hâle gelmiş olurlar.

Çözömlenen ve dosyalara aktarılan verilerin analiz amaçlı veritabanına aktarılmasından önce veritabanında bu verileri tutacak olan tablolar ve bu tablolar üzerinde çalıştırılacak olan sorgular hazır hâle getirilmelidir. UML tasarım aracı kullanılarak eşleştirilmesi yapılan nesne ve etkileşimlerin aktarılacağı tablo scriptleri bu araç kullanılarak oluşturulur. İhtiyaca uygun olarak yazılmış olan saklı yordamlar da ayrı bir script dosyasında saklanmaktadır. Veri çözümleme işlemini bitiren kayıt uygulaması daha sonra bu scriptler vasıtasıyla analiz veritabanının sunucuda yapılandırılmasını, tabloların ve prosedürlerin oluşturulması işlemlerini gerçekleştirir. Kayıt uygulaması son olarak hazır hâle getirdiği veritabanı içerisine kayıt dosyasından oluşturduğu verileri içeren virgül ayrımlı dosyaları aktarır. Böylece kaydedilen verilerin analiz amacıyla veritabanına aktarımı tamamlanmış olur.

Kaydedilen ve veritabanına aktarılmış olan veriler üzerinde performans ve analiz işlemlerine başlanmadan önce bazı işlemlerin yapılması gerekebilir. Bu işlemler, koşum esnasında bazı zamanlarda geçersiz değer alabilen kayıtların temizlenmesi, sorguların hızlandırılması amacıyla belirli tablolarda yer alan belirli alan değerlerinin başka tablolarda yer alan değerler ile değiştirilmesi ya da geçiş amacıyla yeni tabloların/görünümlerin üretilmesi gibi işlemleri içermektedir. Bahsedilen tüm bu işlemler, ihtiyaçlar doğrultusunda yazılmış bir script dosyasının veri tabanı aktarma işleminden hemen sonra uygulama tarafından çalıştırılması ile sağlanmaktadır. Bu işlem sırasında güncellenen ya da silinen tüm veriler kullanıcıya işlem sonrasında raporlanarak kayıt edilen veri üzerindeki değişikliklerin takip edilebilmesi sağlanmıştır. Bu işlemin uygulama tarafından yapılması kullanıcı seçimine bağlı olup, kayıt dosyası üzerinde

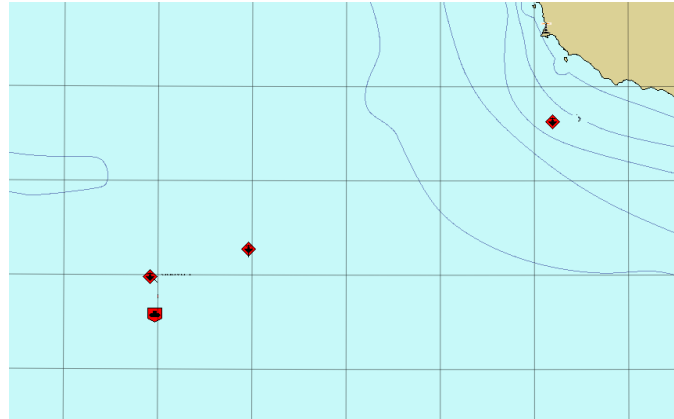
değil de aktarılmış veri üzerinde işlem yapıldığı için istenen herhangi bir zamanda orjinal kayıt tekrar kullanıma alınabilmektedir.

Tekrar Oynatma

Çalışma kapsamında kaydedilen ve veritabanına aktarılan verilerin koşum sonrasında tekrar oynatılması ve analiz edilebilmesi amacıyla Tekrar oynatma ve analiz uygulaması da geliştirilmiştir. Bu uygulama temel olarak hareketli platformların harita alanı üzerinde sergilenmesi ve belirlenen olay ve bilgilerin yine zaman ekseninde 2 boyutlu grafikler olarak analiz edilmesi yeteneklerini içermektedir. Platform hareketlerinin harita üzerinde görsellenmesi amacıyla S57 harita formatını destekleyen bir harita kütüphanesi geliştirilmiş ve platformlar bu kütüphane kullanılarak oluşturulan taktik saha üzerinde görsellenmiştir.

S57 harita kütüphanesi, S-57 veri formatındaki dosyaların katmanlarını okumak, coğrafi koordinat piksel, piksel coğrafi koordinat dönüşümlerini yapmak ve harita verilerini iki boyutlu grafiksel olarak gösterime hazır hâle getirmek amacıyla hazırlanmış bir yazılım kütüphanesidir. S57 kütüphanesi birden fazla S-57 dosyası okuma imkanı sağlar. Ayrıca kuzey batı ve güney doğu köşe coğrafi koordinatları verilerek S-57 harita verilerini alt parçalara bölebilir. Bundan başka haritayı istenilen çözünürlükte ve koordinat aralığında resim olarak da çıktı verebilme yeteneklerini içermektedir.

Veritabanında bulunan her bir platform haritada tipine göre değişik semboller ile ifade edilir. Ayrıca kayıt oynatma arayüzü kaydı ileri, geri sarma ve kayıta istenilen zamana gitme olanağı verir.



Şekil 8. Harita Alanı

Sergilenen her platform tipi için gereken C++ sınıfları, otomatik kod üreten bir şablon ile yaratılır. Şablon konfigürasyon dosyası varlıkların haritada gösterilecek sembol resmi, veritabanı ilişkilendirilmesi (sorgu, tablo ismi vb.) gibi parametrelerini içerir. Otomatik kod üretme sayesinde kayıttaki yeni bir platformun verilerinin arayüzde gösterimi oldukça basit hâle gelir. Üretilen bu platform sembol sınıfları veri tabanından çekilen kayıt verisini tutmak için ikili buffer mekanizması kullanılır. Sembol sınıfları içerisinde tutulan bu bilgi sınıfları kayıt sırasında üretilen bilgi sınıfları ile benzerlik gösterir ve yine otomatik kod üretme teknikleri kullanılarak üretilirler.

Kayıt oynatımı esnasında veritabanından sürekli olarak veri çekilmesi süreksizliğe ve performans problemlerine neden olabilir. Bu yüzden veritabanından parametrik olarak belirlenen sayı kadar platform verisi çekilmekte ve bu veriler platform sembol sınıflarındaki ikili buffer mekanizmasına aktarılmaktadır. Böylelikle bir yandan belirli bir ana ait kayıtlar harita alanında sergilenirken diğer buffer kullanılarak ileriki zamana ait veriler veritabanından çekilebilmektedir. Görsellenen platform sayısı fazla olabileceği için sorgulama işlemleri bir thread havuzu tarafından yapılır. Bu havuzdaki yeralan thread sınıfları platform nesnelere gelen sorgu ve zaman bilgisini kullanarak platform sembollerini içerisinde kullanımında olmayan buffer'ın güncellenmesini sağlarlar.

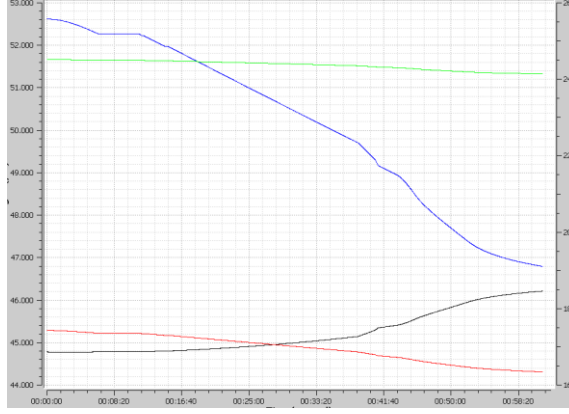
Kayıt oynatma arayüzünde bulunan harita alanı üzerine çeşitli şekilleri çizilerek saha alanları belirlenebilir. Oluşturulan bu şekiller ayrı bir katman olarak kaydedilebilir ve tekrar harita alanı üzerine yüklenebilir özelliklerine sahiptirler.

Ayrıca analiz işlemlerini kolaylaştırmak amacıyla harita alanı içerisinde bulunan bir platformun başka bir platforma veya sabit bir noktaya göre mesafe ve kerteriz ölçümü yapılabilir. Kerteriz ve mesafe bilgisi kayıt süresince anlık olarak güncellenerek kullanıcıya sergilenmektedir.

Grafiksel Analiz

Kayıt edilen verilerin harita alanında tekrar oynatma amacıyla görsellenmesinden başka bu verilerin zaman içerisindeki değişimlerini 2 boyutlu grafiklerle de sergilenmesi mümkündür. Verilerin grafiksel olarak görsellenmesi için kullanımı kolay bir parametre seçme arayüzü oluşturulmuştur. Arayüz ile görsellenmek istenilen platformlar ve bu platformlara ait görsellenebilir bilgiler seçilerek grafiksel analiz yapılabilir. Analiz edilecek verinin zaman aralığını belirlemek için zaman çizelgesi

kullanılır. Uygulama seçilen zaman aralığı için seçilen platformların seçili özelliklerini kullanıcıya sergiler.



Şekil 9. Mesafe – Kerteriz Grafiği

Grafiksel Analiz ekranlarında, platformların birbirlerine göre durumları değerlendirilebilir. Parametre seçim arayüzü kullanılarak seçilen iki platformun birbirlerine göre mesafe ve kerteriz bilgisinin görsellenmesine olanak sağlanır. Benzer olarak analiz amaçlı simülasyonlarda gerçek verinin sistem tarafından üretilen bir çözüm ile kıyaslanması gibi işlemler amacıyla da grafiksel analiz pencereleri kullanılabilir. Grafiksel Analiz ekranları, yaklaşma, gezinme gibi analiz işlemlerini kolaylaştıracak işlevselliğe sahiptir.

Grafik Analiz kapsamında ayrıca S57 kütüphanesi kullanılarak verilerin belirlenen zaman aralıklarında belirlenen askeri formlara uygun olarak istenen ölçekte çıktılarının alınması ya da farklı resim formatlarında kayıt edilmesi imkânı da kullanıcıya sunulmaktadır.

Sonuç

Eğitim amacıyla kullanılan dağıtık simülasyon uygulamalarının da dahil olduğu pek çok sistemde simülasyon sonrası değerlendirme ve analiz önemli bir yere sahiptir. Mevcut eylem sonrası değerlendirme sistemleri simülasyonun tekrarlanmasını sağlarken gerekli olan analiz işlemlerini tam anlamıyla içermemektedirler. Bu eksikliği kapatmak amacıyla geliştirdiğimiz performans analiz aracı kayıt, veri tabanına aktarma ve oynatma ve grafiksel işlemlerinde yüksek bir performans göstermektedir.

Simülasyon koşumu esnasında gerçekleşen kayıt işleminin varolan iletişim trafiğine minimum yük getirmesi sadece gerekli bilgilerin kayıt

edilmesiyle önemli ölçüde sağlanmıştır. Saatlerce süren bir senaryonun kayıt verilerin veritabanına aktarılması sadece dakikalar sürmektedir. Kayıtların oynatılması, ileri-geri alınabilmesi ve istenilen zamana atlaması işlemlerinin verimli biçimde gerçekleştirilmesi kullanılan veritabanı ve sorguların yüksek performanslı çalıştığını göstermektedir. Veritabanı kullanımı ile analiz imkânları genişletilmiş ve grafiksel analiz imkânı istenen öznelikler bazında olduğu gibi hesaplanabilir büyüklükler (mesafe, kerteriz) bazında da yapılabilir duruma getirilmiştir.

Geliştirilen performans analiz altyapısı, diğer AAR yazılımlarından farklı olarak tüm kayıt, oynatma ve analiz imkânlarını beraber içeren yüksek performanslı bir analiz yazılımı alt yapısı sunar.

Kaynakça

- High Level Architecture Version 1.3, DMSO (1998).
- IEEE 1516-2000, High Level Architecture (HLA) (2000).
- IEEE Std. 1516.2-2000. "IEEE Standard for Modelling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specification".
- IEEE 1516-2010, High Level Architecture (HLA) (2010).
- MAK Data Logger, MAK Technologies (2011). 10 Şubat 2011'de <http://www.mak.com/products/datalogger.php> adresinden alınmıştır.
- PITCH Recorder, Pitch Technologies (Pitch) (2011). 10 Şubat 2011'de <http://www.pitch.se/products/recorder> adresinden alınmıştır.
- S2Focus "Analyzer", General Dynamics C4 Systems (2011). 10 Şubat 2011'de <http://www.gdc4s.com/> adresinden alınmıştır.
- Simulation Interoperability Standards Organization (SISO) (2003). Guidance, Rationale, and Interoperability Manual for the Real-timePlatform reference Federation Object Model (RPR FOM).
- Timar Y., Bıkmaz İ., Taşdelen İ., Akgün S. ve Dikenelli O. (2007). Dağıtık Simülasyon Projelerinin kavramsal Modelini Geliştirme çalışmalarında BOM Kullanımı, USMOS 2007 Bildiriler Kitabı.