

Distributed Online Training Simulation for Railway Dispatcher

Nuri Ozalp, Ahmet Basgoze, Ozdemir Kavak, Burcu Kalkan

TUBITAK BILGEM BTE

Informatics and Information Security Research Center

Information Technology Institute

Kocaeli, Turkey 41470

Email: (nuri.ozalp, ahmet.basgoze, ozdemir.kavak, burcu.kalkan)@tubitak.gov.tr

Abstract—Computer Simulations can be considered as a powerful tool for learning such as analysing, designing, and interacting. Especially in the vital criticality level it has become more important tools such as train traffic simulation. The most important purpose of train control system is to prevent train collisions with other trains, keeping them in safe range.

The purpose of this study is to provide train traffic control in a distributed simulation system. The system consists of an instructor, five students and a scenario-editor. The system uses real train route model located in Turkey including three train lines with over 50 stations and interlocking regions. In our simulation environment, dispatchers can control train traffic which have different complexity level. Success of students in educational outcomes can be measured. Instructor console makes decisions about the organization of teaching and learning experiences, classroom management, and responses to individual students. User is able to monitor and track the progress of five targeted students throughout the course of the simulation.

I. INTRODUCTION

Kara ve Demiryolu taşımacılığında önemli bir yeri bulunan tren, bir ya da birkaç lokomotif tarafından çekilen, itilen ve vagonlardan oluşan bir ulaşım aracıdır. Ulaşım ve taşımacılık alanlarında trenlerin çok önemli bir yere sahip olduğu bilinen bir gerçekktir. Artan demiryolu trafiği sonucunda dünya genelinde değişik trafik kontrol sistemleri geliştirilmiştir. Dünyada tren trafiği yönetim metodlarının farklı birçok merkezi tren trafik kontrol sistemleri vardır.

Demiryolu trafiği yönetimi için kullanılan sistemler merkezi kontrol sistemleri olup, kontrolörler (dispeçer) tarafından idare edilmektedir. Dispeçerlerin amaçları trenleri güvenle ve zamanında varması gereken hedeflere ulaştırmaktır.

Tren trafiği kontrol sistemleri çok kritik sistemlerdir. Anlık konum bilgisi konvansiyonel trenlerde bilinmemektedir. Sinyalizasyon sistemleri sayesinde trenlerin yaklaşık konumlari 3km civarında tahmin edilebilmektedir. Bu nedenle tren trafiği kritik bir hal almaktadır. Son yıllarda dispeçer eğitimi, sefer sayılarının ve tren yollarının artması ile daha da önem kazanmıştır. Dispeçerlerin gerçek tren trafiği üzerinde çalışarak tecrübe kazanmaları hem uzaq zaman almakta hem de birtakım riskleri beraberinde getirmektedir. Bu nedenle eğitim simülasyon sistemlerine ihtiyaç duyulmaktadır ve bu konuda birçok eğitim simülasyonu geliştirilmiştir. Özellikle kontrol merkezlerinde birden çok dispeçerin birlikte çalışması gerekliliğinden dolayı dağıtık yapılı ve birden çok kullanıcılı

simülasyonlar önem kazanmıştır. Bu sayede tek bir simülasyonda yoğun trafikli senaryo ile birden çok dispeçerin birlikte tren trafiğini yönetmesi ve birçok probleme başa çıkabilmesi önem arz etmektedir.

Bu çalışmada dispeçerlerin geliştirilen simülasyon sistemi altyapısını kullanılarak eğitilmesi, gerçek tren trafiği yönetimine hazır hale getirilmesi ve kısa sürede alanında uzmanlaşabilmesi amaçlanmıştır. Simülasyon ortamında yoğun trafikli, farklı tip ve hızlardaki trenlerin bulunduğu demiryolu sahası oluşturularak bu saha üzerinde makas, sinyal, ray, hemzemini geçit arızası gibi meydana gelebilecek problemler oluşturulmakta ve yapılan tüm bu işlemler senaryo olarak kaydedilmektedir. Kaydedilen senaryolar eğitmen tarafından bir simülasyonda açılarak aynı anda 5 öğrenciye kadar eğitim verilmesine ve sonrasında öğrencilerin performansları gerçekleştirildikleri işlemler doğrultusunda değerlendirilerek karne verilmesine imkan sağlanmaktadır.

Geliştirilen simülasyon sisteminde Türkiye'deki 3 farklı tipteki gerçek tren hattı gerçeğine bağlı kalarak modellenmiş ve sisteme tanıtılmıştır. Bunlarda 1 tanesi hızlı tren hattı, bir diğerı banliyo hattı ve sonuncusu konvansiyonel tren hattından olmaktadır. 3 farklı tipteki tren hattına ait karakteristik farklılıklar kullanılarak öğrencilerin eğitilmesi halihazırda sistem tarafından gerçekleştirilmektedir.

Geliştirilen simülasyon sistemi 5 modülden oluşmaktadır. Bunlar;

- eğitmen konsolu,
- öğrenci konsolu,
- senaryo editörü,
- performans değerlendirme modülü,
- tren graf modülüdür.

Bunun dışında yardımcı araçlarda geliştirilmiştir. TrackDesigner aracı bunlardan en önemlididir. Bu araç ile modellenecek anlaşılmış sahaları görsel olarak çizilmiş, saha gerçeğine uygun olarak modellenmiştir. TrackDesigner ile bir istasyondaki güzergahlar bulunmuş, sahanın fiziksel özellikleri modellenmiş(sinyal, makas gibi saha ekipmanlarının pozisyonlarının belirlenmesi), anlaşılmış tablolarının üretilmesi gerçekleştirilmiştir. Sistemin ihtiyaç duyduğu verilerin hemen hemen hepsi TrackDesigner kullanılarak üretilmiştir.

Senaryo editörü ile istenilen arazi hazırlanmakta, farklı konumlara istenildiği kadar tren eklenebilmekte ve istenilen bütün

koşullar bu arazide oluşturulmaktadır. Arazide oluşturulabilecek koşullar çarpışma, bölge zaman izni, deray, kırmızı sinyal ihlali, kırmızı geçiş, lokomotif arızası, tren/vagon kaçması, sinyal arızası, makas arızası, sürat tecavüzü, hemzemin geçit arızası gibi durumları içermektedir. Oluşturulan senaryo ile eğitmen ister tek simülasyon isterse farklı 5 simülasyon olarak aynı anda başlatabilmekte ve simülasyonlara müdahale edebilmektedir. İstediği zaman snapshot ile eğitimde bazı önemli gördüğü yerleri tekrar tekrar öğrencilere tecrübe ettirebilmektedir. Öğrenciler yetkileri ölçüsünde sorumlu olduğu alanda tren trafiğini yönetebilmekte ve meydana gelen arızaların çözümü için çalışmalar yapabilmektedir. Simülasyondaki trenlerin anlık hareketleri aynı zamanda trengraf ile takip edebilmektedir. Eğitim sonunda eğitmen öğrencilerin performansını ölçmek amaçlı değerlendirme yapılabilmekte ve yetiştirilen dispeçerin başarısı puanlama sistemi ve başarılı/başarısız olarak ölçülebilmektedir.

2. bölümde related work ler ile ilgili , 3. bölümde sistemin haberleşme alt yapısı hakkında bilgiler verilmiştir. 4. Bölümde sisteme ait 5 modülden bahsedilmiştir. 5. modülde sistem üzerinde yapılan deney ve sonuçları hakkında bilgiler yer almaktadır. 6.bölümde ise sonuç kısmına yer verilmiştir.

II. RELATED WORK

Tren trafiğini yöneten ve birden çok dispeçerin eğitim aldığı dağıtık yapılı simülasyonlar yaygın olarak kullanılmaktadır. Benzer çalışmalara bu bölümde yer verilmiştir.

In Middelkoop and his friend study is a simulator which stands for Flexible Rail Infra Simulation of Operations (FRISO). It includes automatized a simulation model by using a connection to database, editor including generator functions and the possibility to perform single and multiple (stochastic) simulation experiments. FRISO models include following elements track layout, signalling system, route setting, and interlocking. Our system presents multiple screen of regions and to let five students to work at same or different simulation at the same time [1].

In Baohua and his friend's study is about multi-train simulator. It is able to do train performances assessment at the given railway lines, signal layouts optimization, energy-efficient operating strategies , in major terminals exploration of traffic bottlenecks, the evaluation of the reliability of scheduled timetables and train delay propagation. In multi-train simulation does these automatically, in our system dispatcher conduct them [2].

III. SYSTEM DESIGN

A. Simülasyon Yaşam Döngüsü

RAYTES sistemi beş bağımsız simülasyon ortamını eşzamanlı ve birbirinden bağımsız olarak yürütebilmektedir. Aşağıdaki gösterilen Şekil 1' deki yaşam döngüsü bir simülasyon ortamını temsil etmektedir. Sistem ilk çalıştırıldığında IDLE durumda bulunmaktadır. Bir senaryo sisteme yüklenerek simülasyon oturumu oluşturulduğunda LOADED durumuna geçilir. Bu durumda simülasyon başlangıç saatinde beklemekte olup, çalışmaya hazır haldedir. Simülasyon

çalıştırıldığında RUNNING durumuna geçer. RUNNING durumunda simülasyon saatı ilerlemekte, simüle edilen nesneler zamana duyarlı biçimde akmakta ve insan arayüzlerinde sistem ile kullanıcılar arası etkileşim gerçekleşmektedir. Simülasyon herhangi bir zamanında duraksatılabilir, bu durumda simülasyon PAUSE durumuna geçer. PAUSE durumunda iken simülasyon saatı ve simülasyon işlemleri geçici olarak durdurulur. PAUSE durumunda yine simülasyon RUNNING durumuna başlayabilir. Simülasyon sonlandırıldığında TERMINATED durumuna düşer ve simülasyon artık yeniden başlatılamaz.

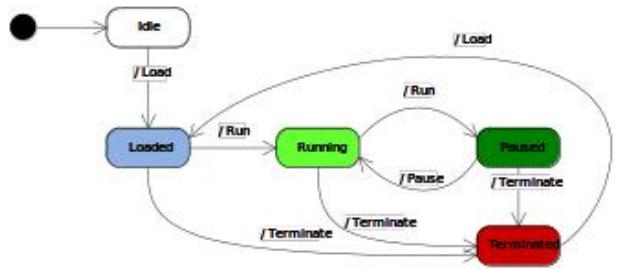


Fig. 1: Representation of Simulation Life cycle

B. Communikation

1) *Simulation Message:* Simulation Message mesaj türü simülasyon sistemin üst seviye kumanda ve kontrol merkezi mesajlaşma türüdür. Bu mesajlar Eğitmen Konsolu'nda üretilmektedir ve sistemin diğer bileşenlerine gönderilmektedir. Simülasyon'un yaratılması, duraklatılması, snapshot alınması, gibi temel simülasyon komutlarını içermektedir. Ayrıca simülasyondaki sanal saha nesnelerinin durumlarına müdahale etmek amacıyla bir takım özel müdahale komutları tanımlıdır.

2) *Request Message:* RequestMessage mesaj türü Kontrol Merkezi Client (KMC) modülünün Kontrol Merkezi Sunucu (KMS) modülüne gönderilen komutları taşımaktadır. KMC ve KMS arasındaki protokol "client-server" mimarisi olup RequestMessage komutları istemci tarafın sunucu tarafından hizmet isteği olarak değerlendirilmektedir. RequestMessage komutları Tablo x e listelenmektedir.

3) *State Message:* StateMessage mesajlar simülasyon çalışması esnasında sanal sahaların tanımlı elemanlarının durum bilgilerinin iletilmesi için kullanılmaktadır. Kontrol Merkezi Sunucu (KMS) modülü StateMessage kullanılarak Kontrol Merkezi Client (KMC) modülüne saha bilgilerini göndermektedir. Kontrol Merkezi Sunucu ayriyetten saha görüntüleme ihtiyacı duyan Geniş Ekran Konsolu (GEK) ve Eğitmen Saha Müdahale (ESM) modüllerine de StateMessage vasıtasiyla bilgi göndermektedir.

4) *System Message:* SystemMessage mesajlar Öğrenci Kontrol Yönetimi (ÖKY) ile KMS modüller arası kullanılan mesajlaşmadır. Bu mesaj türü öğrenci olarak eğitim sisteme dahil olmak isteyen kullanıcıların giriş (login) yapmasını sağlamakta kullanılmaktadır.

C. Modbus

Modbus otomasyon sanayi çevrelerinde kabul görmüş bir ham veri iletişim protokoludur. Bu protokol “client-server” mantığında çalışıp sistemler arası bit dizilerin sorgulanması ve iletilmesinde kullanılmaktadır. RAYTES projesinde Modbus paketler UDP/IP üzerinden yazılım modülünden diğer modüllere iletilmektedir. KMS modülü, Anlaşman Simülatör (AS) modülü, Saha Simülatör (SS) modülü ve Tren Simülatör (TS) modüller arası komut ve bilgi transferi Modbus paketler ile gerçekleşmektedir.

D. SSY Simülasyon Sunucu Yönetim Modülü

SSY yani Simülasyon Sunucu Yönetim modülü simülasyon sistemi içerisindeki tüm aktif modüllerin bağlı olduğu haberleşme altyapısının bulunduğu ve koordinasyonun sağlandığı modüldür. Şekil 2'de sequence diagram ve Şekil 3'de arayüzü gösterilmiştir. Simülasyon akışı içerisinde doğrudan bir görevle sorumlu tüm aktif modüller bu module ya doğrudan ya da altında çalıştığı ana modül üzerinden dolaylı olarak bağlı olmak zorundadır. Simülasyonun çalışması ve hazırlanması esnasında gerçekleşen tüm mesajlaşmalar bu modül üzerinden gerçekleşmektedir. SSY ayrıca sistemdeki aktif konsollar gibi kendi içerisindeki bilgileri uygun mesajlara karşılık olarak diğer konsollarla yada uygulamalarla paylaşabilmektedir. SSY mesajlaşma amacıyla tüm sistem elemanları gibi SD yani Simülasyon Destek ve Mesajlaşama Modülü'nü kullanmaktadır. Bu birim tüm giriş bilgilerini SD üzerinden almaktak olup yine çıkış bilgilerini de SD üzerinden sisteme dağıtmaktadır. SD bu SSY üzerinde Mesaj Sunucu görevi ile tanımlanmış olarak kullanılmaktadır. SSY üzerine gelen her bir mesaj bir ön kontrolden geçirilerek mesaj ile ilgili yapılması gereken işleme karar verilir. SSY'nin bir mesaj üzerine yapacağı 4 işlem vardır;

- Mesajı doğrudan hedef ya da hedeflerine göndermek
- Mesajı tüm ağa yayılmamak
- Mesajın içeriği SSY ile ilgili ise SSY'den doğrudan cevap mesajı yollamak
- Mesajın hiyerarşik olarak gösterilmesi gerekiyorsa bu hiyerarşinin sağlanması

Hiyerarşik mesaj gönderimi bir mesajın, örneğin simülasyon yükleme mesajının öncelik sırasına göre her bir birime yollanması anlamına gelmektedir. Sıra ile yollanan her bir mesaj sonrasında olumlu cevap gelmesi durumunda bir sonraki ilgili birime mesaj iletilir. Bu zincir bir noktada kırılırsa sistem mesaj gönderen ilk birime olumsuz mesajı döndürür. Eğer mesaj zinciri tamamlanırsa olumlu mesajı iletilir.

E. SM Simülasyon Motor Modülü

1) *Sınıf Yapısı:* Simülasyon Motoru (SM) modülündeki Şekil 4'de gösterilen ana sınıf SimMotor sınıfıdır. Bu sınıf Config, XmlUtility, SahaEventList, FTSimManager::FTSimMgr ve Communicator::Client sınıfları bünyesinde yaratıp kullanmaktadır. FTSimMgr Saha Trafik Simülasyon Yönetimi (FTS) modülünde tanımlıdır. SimMotor sınıfı FTSimMgr sınıfının sağladığı arayüz üzerinden anlaşman, saha ve trafik simülasyon modellerine

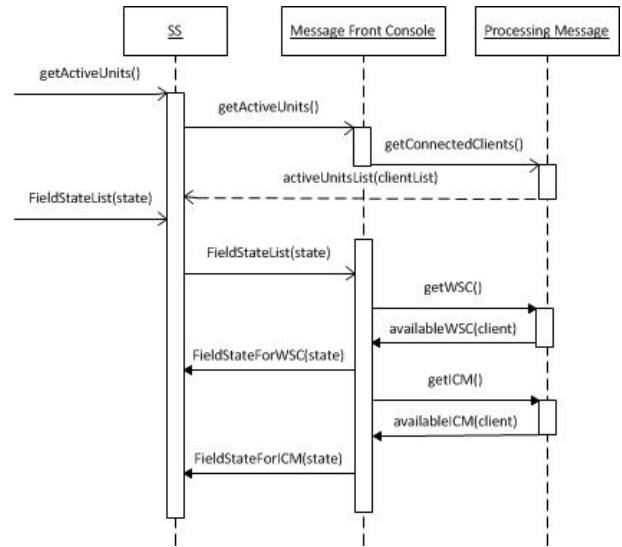


Fig. 2: Representation of SSY sequence diagram

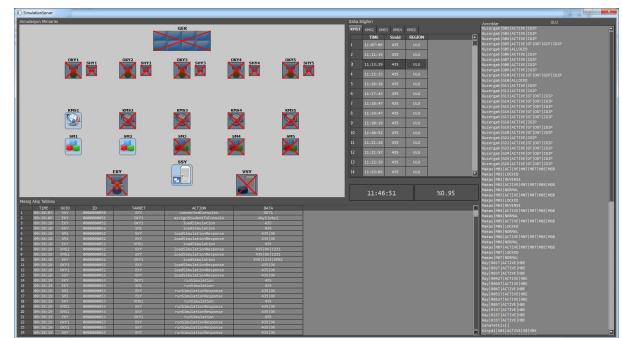


Fig. 3: Representation of SSY UI

erişmektedir. Communicator modüllünde tanımlı Client sınıfı ise SimMotor sınıfına SimulationMessage mesajlaşma arayüz sağlamaktadır. SimMotorGui, SimMotor sınıfına debug ve test maksatlı grafik pencere arayüzü sağlamaktadır.

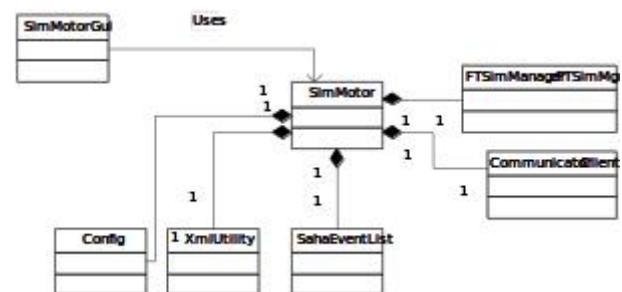


Fig. 4: Representation of SM class diagram

2) *Mesajlaşma:* SSY ilk başlığında 5 adet bağımsız SM proses yaratır. Bu SM proses'ler sayesinde 5 bağımsız simülasyon eşzamanlı olarak SSY tarafından kontrol edilmesine olanak sağlar. SSY SimulasyonMessage tür mesajını SM ye aktarmaktadır. Bu tür mesajlar, simülasyon yükle, simülasyon başlat tarzında üst seviye kontrol mesajlarıdır. SM

bu mesajları Saha Trafik Yönetimi (FTS) simülasyon modülüne iletmektedir. Simülasyon modülü bu mesajların gerektirdiği işlemleri tamamladığında SM'ye yanıt (olumlu veya olumsuz) vermektedir ve bu netice SSY ye geri bildirilmektedir. LoadSimulation mesajı genel örnek olarak ele alınırsa; bu mesaj SM tarafından alındığında iki işlem tetiklemektedir. Birincisi, senaryo da oynatılması gereken önceden programlanmış olaylar veritabanından okunuyor ve Saha Event List (SEL) listesine yüklenmektedir. Bu olayların her biri bir Saha Event (SE) olarak tanımlanmıştır. İkinci işlem ise FTS ve onun bünyesindeki Saha Simülasyon modeli, Anlaşman Simülasyon modeli ve Trafik Simülasyon modelini ilgili senaryonun teknik parametrelerini veritabanından yüklemesi ve simülasyon başlatma durumu için hazırlamasıdır. Simülasyon Running durumundayken SEL deki sıralanmış SE olaylar simülasyon zamanına göre FTS modülüne işlenmek için aktarılmaktadır. Bu mekanizma bir senaryonun müdahalesiz olarak işlenmesini sağlamaktadır. Ancak bir simülasyona ayrıca eğitmen tarafından müdahale edilmesine olanak vermektedir. Eğitimen, Eğitmen Saha Müdahale (ESM) aracı kullanarak manuel Saha Event yaratabilir. Bu manuel olarak yaratılan SE olaylar SM'ye anlık olarak iletilerinde SE olayı SEL listesinin başına eklemektedir ve böylelikle simülasyon'un sonraki çevriminde hemen işlenmesi sağlanmaktadır. Bütün bu akış Şekil 5'deki diagramda gösterilmiştir.

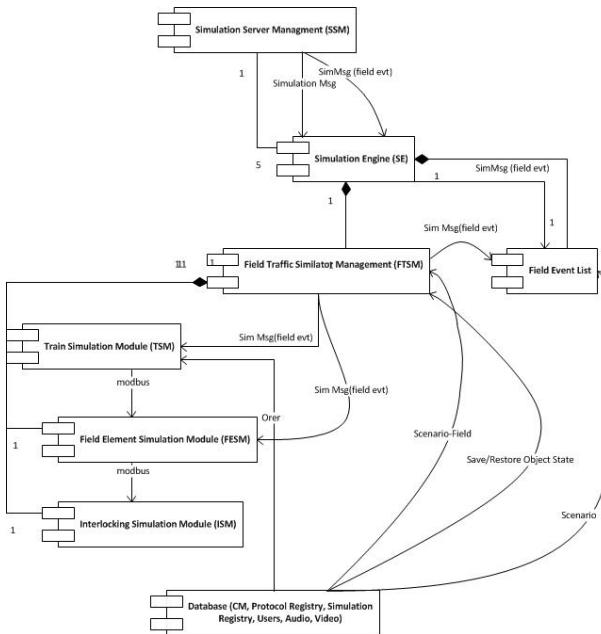


Fig. 5: Representation of SM Communication diagram

3) SD Simülasyon Destek ve Mesajlaşma Modülü: Şekil 6'de görüldüğü gibi Simülasyon Destek ve Mesajlaşma modülü proje içerisinde birbirinden fiziksel olarak bağımsızlar fakat haberleşme ve etkileşim içinde olan birimlerin birbiri ile olan haberleşmesini birlikte sağlamaktadır. Temel sunucu istemci mimarisini üzerine kurulmuş haberleşme modülü projeye özgü bağlantı kopukluğu kontrolü, tekil bağlantı özelliği, XML tabanlı mesaj kontrolü gibi özellikler eklenerek

projeye özgü protokol kuralları çerçevesinde tasarlanmıştır. Mesajlaşma ve iletişimini sağlayacak olan ana modül iki modda çalışabilmektedir. Bu modalardan ilki Server yani sunucu mod iken diğer mod ise Client yani istemci modüldür. Sunucu modda çalıştırıldığında belirli bir bağlantı portu üzerinden haberleşmenin sağlandığı bu sistem diğer yani İstemci modda diğer sistemlerin bağlanması beklenmektedir.

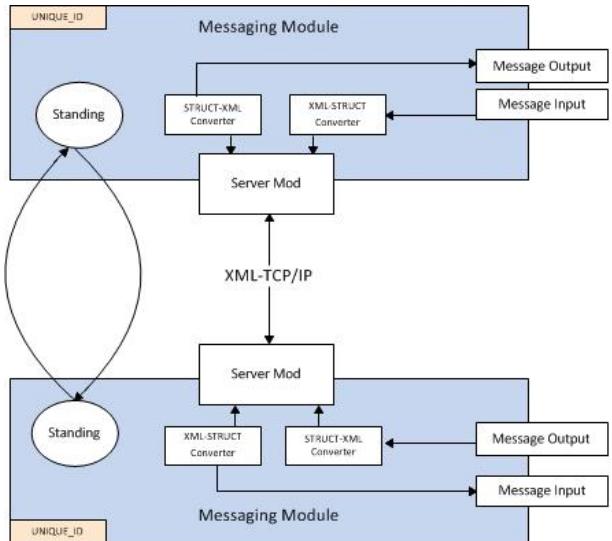


Fig. 6: Representation of SD diagram

İstemci mod bağlantı esnasında hedef sunucu bilgilerini ve kendisini tanıtan tekil eşsiz bir ID ile erişimi sağlayacaktır. Bu bağlantıları eşsiz hale getirmek için IP numaraları kullanılmayacaktır. Çünkü aynı makina üzerindeki farklı uygulamalar birbirleri ile sunucu üzerinden haberleşme yeteneğine sahip olmalıdır. Aynı ID ile birden fazla erişim denemesi yapıldığında sistem 2. sistemin bağlantı denemesini “Kullanımdaki bir ID” uyarısıyla reddedecektir. Şekil 7'de görüldüğü gibi yine bir sunucuya bağlı olan tüm istemciler o sunucuya periyodik olarak sunucuya bağlı oldukları yani ayakta oldukları bilgisini yollarlar. Ters açıdan bakıldığından sunucu da kendisine bağlı olan tüm istemcilere çalışır yani ayakta olduğu bilgisini periyodik olarak bildirmektedir. Bu sayede sistemdeki tüm elemanların bağlantı kontrolü gerçekleştirebilecektir.

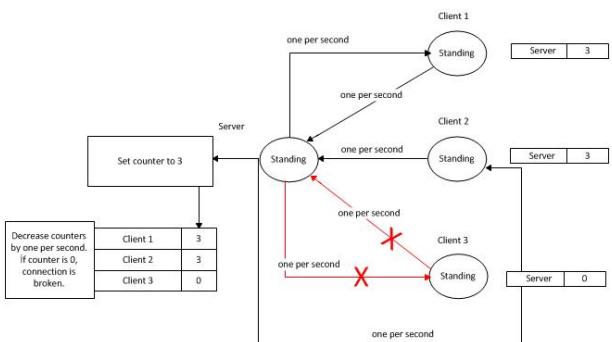


Fig. 7: Representation of SD diagram

IV. TRAINING ENVIRONMENT

A. ÖK Öğrenci Konsolu

Öğrenci Konsolu (ÖK) bir Öğrenci Bilgisayar Ünitesi (ÖBÜ) ve bir Sesli Haberleşme Cihazı (SHC) alt donanım bileşenlerini içermektedir. ÖK'nin yapısı Şekil ??'de gösterilmiştir. Öğrenci Konsol Yönetimi (ÖKY) yazılım modülü üst seviye modül olup öğrenci konsolunun genel işlevlerinden sorumludur. Dispeçer arayüz uygulaması KMC modülü tarafından gerçekleştirilmekte olup ekranların video görüntü kayıtları Video Kayıt Altyapı (VKA) modülü tarafından gerçekleştirilmektedir.



Fig. 8: Representation of Dispatcher Console

B. GEK Geniş Ekran Konsolu

Geniş Ekran Konsolu (GEK) bir Geniş Ekran Bilgisayar Ünitesi (ÖBÜ) ve bir Geniş Ekran Duvar Ünitesi (GDÜ) donanım bileşenlerinden oluşmaktadır. GEK'in yapısı Şekil 3 de gösterilmiştir. Geniş Ekran Bilgisayar Ünitesi (GBÜ) çok çıkışlı ekran kartı ile donatılmış olup, her çıkış Geniş Ekran Duvar ünitesinin birer ekranına bağlıdır. GDÜ iki satır dört sütün düzeneğe göre yerleştirilmiş 8 LCD ekranlık görüntüleme duvarı şeklinde tasarlanmıştır. Geniş Ekran Konsol Yönetimi (GKY) yazılımı dispeçer arayüz uygulaması olan KM Client (KMC), Video Kayıt Altyapı (VKA) ve Video Geri Oynatma Aracı (VGO) modüllerinin çalıştırılması ve kontrol edilmesinden sorumludur.

Şekil 13'de geniş ekrana ait deney sonucunun ekran görüntüsü görülmektedir.

C. Eğitmen Konsolu

Eğitmen Konsolu (EK) simülasyon başlatıp simülasyona öğrenci ataması yapabilen ve senaryo atayan konsoldur. Simülasyon anında simülasyona müdahale edebilmektedir. Simülasyonu durdurabilir, devam ettirilebilir ve sonlandırabilir. Öğrenciden gelen taleplere onay verebilir. Snapshot alabilir. Sınav modunda anlık öğrenciden düşen geri dönüşleri notlandıracı. Şekil 12'de eğitmen paneli ile yapılan deneye ait ekran görüntüsü yer almaktadır.

D. SSÜ Simülasyon Sunucu Ünitesi

RAYTES projesinde tüm mesajlaşma koordinasyonu sağlamak amacıyla Simülasyon Sunucu Ünitesi uygulaması kullanılmaktadır. Endüstriyel bir sunucu bilgisayarda koşan ve proje ekibi tarafından geliştirilmiş olan uygulamanın temel işlevleri simülasyon bileşenleri arasında mesajlaşmanın senkronizasyonunu ve koordinasyonu gerçekleştirmek, sistemin çalışmasını tüm mesaj akışlarını görüntülemek, simülasyon altyapısını oluşturan diğer sunucu tarafı bileşenlerin ayağa kaldırılmasını sağlamak, simülasyon bileşenlerin bağlantı bilgilerini tutmak ve bunları diğer bileşenlerle paylaşmak gibi görevlerdir. Simülasyon sisteminin tüm aktif bileşenleri Simülasyon sunucusuna doğrudan ya da dolaylı olarak bağlı bulunacaktır.

E. Editör Analiz Konsolu

Editör Analiz Konsolu üzerinde Senaryo Editör Aracı (SED), Performans Analiz Aracı (PA), Kullanıcı Yönetim Aracı (KYA), Öğrenci Eğitim Kayıtları Aracı (ÖEK) ve Video Geri Oynatma Aracı (VGO) yazılım modüller çalışmaktadır. EAK ile simülasyon öncesi senaryoların hazırlanması, kullanıcıların tanımlanması ve simülasyon sonrası eğitim performans analizi ve öğrenci eğitim kayıtlarının incelenmesi mümkündür. Dolayısıyla bu konsol simülasyon eğitimi esnasında değil "off-line" kullanılmaya yönelikir.

F. Train Graph

Raytes Projesi kapsamında geliştirilen Trengraf bileşeni, tren hareket kayıtlarının (TrainEvent) sorgulanarak grafik üzerinde çeşitli seçenekler ile gösterimini sağlayan bir veri analiz aracıdır. Yapılan sorgulamalar sonucunda trenlerin bloklardan geçiş hareketleri rahatlıkla izlenebilmektedir. 74 de örnek bir ekran görüntüsü verilmiştir.

- Eğrinin solundaki çizgi ile işaretlenmiş noktalar trenin bloklara girişini,
- Eğrinin sağındaki çizgi ile işaretlenmiş noktalar trenin bloklardan çıkışını,
- İki çizgi arasında kalan transparan boyalı alan ise giriş-çıkış arasındaki zaman farkını, yani trenin o blokta ne kadar kaldığını ifade etmektedir.

Bu seçeneklerin hangilerinin gösterileceği kullanıcı tarafından seçilebilmektedir. Şekil 11'de tren grafa ait çalıştırılan deney sonucundaki ekran görüntüsü görülmektedir.

V. EXPERIMENT RESULT

Geliştirdiğimiz dağıtık yapılı simülasyon RAYTES projemiz 5 öğrencinin aynı anda eğitim alabildiği durumu 2 farklı şekilde testini gerçekleştirdik. Bunlardan ilki 5 trenin bulunduğu bir senaryoda tek simülasyon oturumunda 5 kullanıcının yetkisi dahilinde 1 saatlik bir eğitim olarak planlandı. Yapılan eğitim sonucunda ilk testten elde edilen toplam RAM ve CPU kullanımı Şekil 9'de gösterilmiştir

İkinci testimizi aynı senaryoyu kullanarak 1'er saatlik 5 ayrı simülasyon oturumunda her simülasyona birer öğrenci atanacak şekilde planlandı. Yapılan eğitim sonucunda elde

	RAM(MB)	CPU(%)
Instructure	115	3
Simulation Server	34	1
Simulation Motor	134	3

Fig. 9: Tek simülasyon ve 5 farklı öğrenci ile yapılan eğitime ait modüllerin kaynak kullanımını göstermektedir

	RAM(MB)	CPU(%)
Instructure	364	3
Simulation Server	104	1
Simulation Motor	135*5=675	3*5=15

Fig. 10: 5 farklı simülasyon ve 5 farklı öğrenci ile yapılan eğitime ait modüllerin kaynak kullanımını göstermektedir

edilen toplam RAM ve CPU kullanımı Şekil 10'de gösterilmiştir

Deneysel sonucunda elde edilen tren hareketlerini gösteren tren graf şeklindeki gibi olmaktadır.

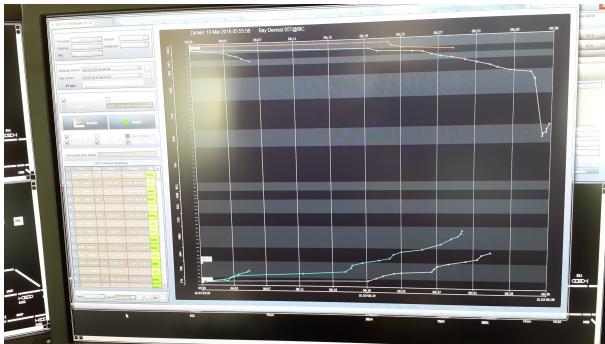


Fig. 11: Tren hareketlerinin gösterimi



Fig. 12: 5 farklı simülasyon ve 5 farklı öğrenci ile yapılan eğitemin eğitmen konsolunda gösterimi

VI. CONCLUSION

Computer Simulations can be considered as a powerful tools for learning such as analysing, designing, and interacting. Especially in the vital criticality level it has become more important tools such as train traffic simulation. The most



Fig. 13: 4 farklı kullanıcıya ait ekran görüntülerinin geniş ekranda gösterimi

important purpose of the train control system to prevent train collisions with other trains, keeping them in safe range.

The purpose of this study is to provide train traffic control in a distributed simulation system. The system consists of an instructor support five students and a scenario-editor. The system use real train route model located in Turkey. During the simulation, dispatchers console can controls train traffic which have different size and speed in system. Success in educational outcomes can be measured. Instructor console make decisions about the organization of teaching and learning experiences, classroom management, and responses to individual students. The user is able to monitor and track the progress of five targeted students throughout the course of the simulation.

In future work, we will find the optimal solution of train re-scheduling problems, autonomous path planning for train. Dispatcher will only monitor.

ACKNOWLEDGMENT

This work has been conducted within Rail Transit systems Simulation Research Lab- project (project number 3920-S513000) for Turkish State Railways, which is part of the Rail Transit Systems research program funded by The National Research Institute of Electronics and Cryptology (TUBITAK BILGEM). We thank all project partners for their work and contributions to the project.

REFERENCES

- [1] A. D. Middelkoop and L. Loeve, "Simulation of traffic management with FRISO," 2006, vol. 1, pp. 501–509.
- [2] M. Baohua, J. Wenzheng, C. Shaokuan, and L. Jianfeng, "A computer-aided multi-train simulator for rail traffic," in IEEE International Conference on Vehicular Electronics and Safety, 2007. ICVES, 2007, pp. 1–5.
- [3] I. Sahin, "Railway traffic control and train scheduling based on inter-train conflict management," Transportation Research Part B: Methodological, vol. 33, no. 7, pp. 511–534, Sep. 1999.
- [4] M. Boccia, C. Mannino, and I. Vasilyev, "The dispatching problem on multitrack territories: Heuristic approaches based on mixed integer linear programming," NETWORKS, vol. 62, no. 4, pp. 315–326, Dec. 2013.