PARALEL VE DAĞITIK AYRIK OLAYLI SİMÜLASYONLAR VE KARŞILIKLI ÇALIŞABİLİRLİK

Ahmet ZENGİN[1] **Hüseyin EKİZ[2]** **Cüneyd FIRAT[3]** **Raşit KÖKER[4]**

[1,2,4] Elektronik ve Bilgisayar Eğitimi Bölümü
Teknik Eğitim Fakültesi
Sakarya Üniversitesi, 54187, Esentepe/KAMPÜS SAKARYA
[3]Yazılım Sistemleri Bölümü
Bilişim Teknolojileri Araştırma Enstitüsü
TÜBİTAK Marmara Araştırma Merkezi, Gebze, KOCAELİ

[1]e-posta: azengin@sakarya.edu.tr [2]e-posta: ekiz@sakarya.edu.tr
[3]e-posta: cfirat@mam.gov.tr [4]e-posta: rkoker@sakarya.edu.tr

Anahtar sözcükler: Modelleme ve Simülasyon(M&S), Paralel ve Dağıtık Simülasyon, Senkronizasyon, HLA

ABSTRACT

Parallel discrete event simulation can be defined as executing of simulation programs on a multi-processor computer platform. Distributed simulation is concerned with the execution of simulations on geographically distributed computers interconnected via a local area and/or wide area network. In both cases the execution of a single simulation model, perhaps composed of several simulation programs, is distributed over multiple computers.

In this work, in addition to briefly introduction of the parallel and distributed discrete event simulation, synchronization issue in parallel discrete event simulation and historically development of interoperability concept are also considered. Also, HLA technology that brings together systems built for separate purposes, technologies from different eras, products from various vendors, and platforms from various services and permits them to interoperate in a synthetic environment is summarized.

1 GİRİŞ

Paralel ayrık olaylı simülasyon; simülasyon programlarının çok işlemcili bir bilgisayar platformu üzerinde çalıştırılması olarak ifade edilebilir. *Dağıtık simülasyon ise;* simülasyonların yerel alanlı (LAN) ve/veya geniş alanlı (WAN) bir ağ aracılığıyla birbirine bağlı coğrafik olarak dağıtılmış bilgisayarlar üzerinde çalıştırılması olarak açıklanabilir. Çok işlemcili birden çok bilgisayar üzerinde simülasyonu çalıştırma ise paralel ve dağıtık simülasyon olarak tanımlanır.

Bu çalışmada, paralel ve dağıtık simülasyon konusu hakkında bilgi verilmesi yanında, paralel ayrık olaylı simülasyonlardaki senkronizasyon ve karşılıklı çalışabilirliğin tarihsel gelişimi özetlenmiştir. Ayrıca, bir simülasyon uygulamasının paralel ve dağıtık platformlar üzerinde çalıştırmanın altında yatan teknikler ve bunların tarihsel gelişimi incelenmektedir. Ek olarak, farklı amaçlar için geliştirilen simülasyon sistemlerinde farklı dönemlerde kullanılan teknolojiler ve bu teknolojilerde kullanılan değişik ürünler ile çeşitli servis platformlarını bir araya getiren ve bunların yapay bir ortamda karşılıklı olarak çalışmalarına olanak tanıyan HLA teknolojisi tanıtılmaktadır.

Ayrık olaylı simülasyon algoritmaları, ayrık olaylı modelleri çalıştırmak için yapılması gereken işlemi tanımladıkları için *soyut simülatörler* olarak adlandırılırlar. Ayrık olaylı simülatörler bir sonraki olay zamanlarını içeren olay listeli algoritmalar kullanılarak tek işlemcili sistemlerde çalıştırılabilirler. Ancak, ayrık olaylı bir simülasyonu birden fazla işlemci üzerinde çalıştırmak çeşitli uygulamalarda iyi bir alternatif olabilir [1].

Geniş kapsamlı, yüksek kararlı simülatörler tarafından talep edilen işlemlerin karmaşıklığı ve büyüklüğü, tek işlemcili mimarilerin kullanılması yerine, çok işlemcili mimarilerin kullanılmasını gerektirmektedir. Çok işlemcili mimariler, çalışma esnasındaki koordinasyonu ve veri paylaşımını sağlayan bir iletişim ağı aracılığıyla birbirine bağlanmış birden çok işlemciden oluşur.

Bir simülatörü tanımlamak için "paralel" ve / veya "dağıtık" terimlerinin kullanılıp kullanılmaması büyük ölçüde simülatörün tasarlanmasında göz önünde tutulan hedeflerle ilgilidir. Bu hedefler / amaçlar aşağıda sıralanmıştır [2-3]:

i. *Simülatörün çalışma zamanını düşürme:* Büyük bir simülasyon işlemini eş zamanlı olarak çalışabilen birden fazla alt-işleme bölerek ve her bir alt-işlemi farklı bir işlemciye yaptırarak, kullanılan işlemci sayısı ile orantılı olarak işlem zamanı azaltılabilir. Simülasyon işleminin uzun bir süre aldığı uygulamalarda bu çalışma şekli önemli faydalar sağlayabilir. Örneğin; tek bir işlemci ile günler hatta haftalar alabilen çok sayıda düğümü barındıran bir iletişim ağının simülasyonu çok işlemcili bir platformda çalıştırılarak çok daha kısa sürede bitirilebilir.
ii. *Coğrafi dağıtım:* Coğrafik olarak dağıtılmış bir takım bilgisayarlar üzerinde simülasyon programını çalıştırmak, fiziksel olarak farklı yerlerde konumlanmış bir çok katılımcıya sahip sanal ortamlar oluşturmamıza olanak tanır. Bu yöntem, farklı alanlardaki katılımcıları gerektiren müşterek uygulamalarda, katılımcıların seyahat harcamalarını büyük oranda düşürür.
iii. *Farklı üreticilerin simülatörlerini bütünleştirme:* Farklı üreticiler tarafından üretilen ve her biri farklı bir işlem için uygun olan simülatörlerin birbirleri ile bütünleşik olarak çalıştırılmasına olanak tanır. Örneğin; farklı uçak türleri için tasarlanmış değişik uçuş simülatörlerini tek bir bilgisayara taşımak yerine, sanal bir ortam oluşturarak her birini farklı bir bilgisayar üzerinde çalıştırıp, kullanılan simülatörleri "birbirlerine bağlamak" daha mantıklı ve verimli bir yöntem olabilir.
iv. *Hata toleransı:* Birden fazla işlemci kullanmanın bir başka üstünlüğü, hata toleransının artmasıdır. İşlemcilerden birisinin başarısız olması veya işlemcilerin yetersiz kalması durumunda, diğer işlemcilerin hatalı işlemcilere bağlı olmadan simülasyonu sürdürmesi mümkün olabilir.
v. *Kapasitenin / model büyüklüğünün artması ve kaynakların paylaşımı:* Birden çok işlemci kullanılması durumunda, işlemcilerin bellek kapasitelerinin toplamı kullanılabilir. Ayrıca, bir simülasyon ağındaki özel düğümler tarafından sağlanan daha büyük veri işleme ve grafik kapasitesini kullanmak bu şekilde mümkün olabilir.

2 MODELLEME VE SİMÜLASYON KONUSUYLA İLGİLİ TEMEL TERİMLER

Modelleme ve simülasyon hakkında bilgi edinmede ilk adım, bu konuyla ilgili terimleri / kavramları anlamak olacaktır. Aşağıda, modelleme ve simülasyon ile ilgili uluslararası kurumlar tarafından kabul gören terimler / kavramlar kısaca açıklanmakta ve bu bildiride kullanılan kısaltmaların anlamları verilmektedir [4]:

Model ; Bir sistemin, varlığın, olayın veya işlemin fiziksel, matematiksel veya mantıksal gösterimidir.

Simülasyon ; Personeli eğitmek veya daha uygun bir şekilde bilgi elde etmek için, benzer bir model, durum veya bir alet vasıtasıyla, bir durum veya sistemin (ekonomik, mekanik, vs.) davranışını taklit etme tekniği olarak ifade edilebilir.

Modelleme ve simülasyon (M&S) ; Yönetimsel veya teknik kararlar verebilmek maksadıyla temel olarak kullanılabilecek veriyi geliştirmek için, statik veya dinamik emülatörleri, prototipleri, simülatörleri, uyarıcıları (stimulators) kapsayan modellerin kullanımı ile ilgilidir.

Simülatör ; Simülasyonu yerine getiren bir aygıt, bilgisayar programı veya sistemidir.

Sanal (Virtual) simülasyon ; Gerçek insanların simüle edilen sistemle etkileşim halinde olduğu simülasyondur.

Canlı (Live) simülasyon ; Gerçek insanların gerçek sistemlerle çalışmasını kapsayan simülasyondur.

Yapısal (Constructive) simülasyon ; Simüle edilen insanların simüle edilen sistemlerde çalışmasını kapsayan simülasyon demektir. Gerçek insan bu simülasyon için uyarıcı vazifesi görür (girişleri üretir) ancak çıkışları belirleyemez.

Analitik simülasyonlar ; Sistemlerin davranışını nicel olarak analiz etmek için kullanılırlar. Örneğin; bir iletişim ağı içerisinde dosyaları indirmedeki ortalama gecikmeyi belirlemek, bir fabrika montaj hattında dar boğazları saptamak, vb işlemlerin analizi. Bu simülasyonlar genellikle hızlı çalışır. Analitik simülasyonlar insan müdahalesi olmadan "grup" programlar olarak çalışabilirler veya modellenen sistemin çalışmasını görüntüleyen bir animasyonu barındırabilirler [2].

Sanal ortamlar ; Simülasyon uygulamalarının yeni bir sınıfıdır. Sanal simülasyonlar, insanların eğitim veya eğlence için içerisine katılabildiği sanal dünyalar oluşturmak için kullanılır. Örneğin; askeri tatbikatlar, tank ve uçuş simülatörlerinin bir birine bağlanması yoluyla prova edilebilir ve çok oyunculu video oyunları farklı mekanlardaki oyuncuların bir birleriyle ve simüle edilen rakiplerle etkileşim yapmalarına olanak tanır. Bu simülasyonlar gerçek zamanlı olarak çalışmak zorundadır.

3 PARALEL AYRIK OLAYLI SİMÜLASYON VE SENKRONİZASYON

Doğal olarak paralel bir şekilde çalışan gerçek hayattaki sistemlerde bilgi paylaşımı, modellenmesi çok zor olan gerçek hayattaki sistemlere özgü paralellik kullanılarak gerçekleştirilir. Gerçek dünya sistemlerindeki paralelliğin modellenmesinin zor olmasının nedeni; model bileşenleri arasında mevcut olan sağlam *nedensel bağımlılıklardır (causal dependencies)*. Bir bileşen içinde meydana gelen herhangi bir olay başka bir bileşendeki olayları doğrudan veya dolaylı etkilediğinde, burada olayların *nedensel bağımlılıklarından* söz edilir [1].

Model bileşenleri, ayrı işlemciler üzerinde simüle edildiği zaman, bu bağımlılıklar çalışmanın doğruluğunu garanti etmek için dikkatli bir senkronizasyonu gerektirir. Bu nedenle paralel ayrık olaylı simülasyonlardaki senkronizasyon konusu son derece önemlidir.

Ayrık olaylı simülasyonların senkronizasyonu, merkezi bir problem olarak çok erken fark edilmiş ve bu konu bir çok araştırmacının ilgisini çekmiştir. Tarihsel olarak, çözümsel uygulamalar için paralel ve dağıtık simülasyonlarla çalışma, büyük ölçüde yüksek performanslı bilgisayarlar üzerine araştırma yapan topluluklardan çıkmıştır. Analitik simülasyonların çok işlemcili bilgisayarlar üzerinde çalıştırılması ile ilgili çalışmaların çoğu, senkronizasyon ile ilgilidir. Konu ile ilgili ilk algoritmalar 1970'lerin sonunda geliştirilmiş ve senkronizasyon ile ilgili sorunlar üzerindeki araştırmalar günümüze kadar süregelmiştir. Bu alanda geliştirilen ilk algoritmalar Bryant (1977), Chandy ve Misra (1978) tarafından ortaya atılmıştır [5-17].

Senkronizasyon algoritmasının amacı, simüle edilen sistemdeki sebep-sonuç ilişkilerinin simülasyon programı içinde doğru bir şekilde oluşturulmasını garanti etmektir. Bu amaca yönelik, *koruyucu (conservative)* ve *iyimser (optimistic)* senkronizasyon mekanizmaları geliştirilmiştir. Bu mekanizmalar, genellikle, simülasyonun zaman-işaretli (time-stamped) mesajlar ve olayların karşılıklı değişilmesi yoluyla birbiriyle haberleşen bir takım mantıksal işlemlerden (logical processes - LP) oluştuğunu varsayar. Senkronizasyon mekanizmasının hedefi; her LP'nin olayları zaman işareti sırasına göre işlemesini garanti etmektir ve bu şart, '*yerel nedensellik sınırlaması'* olarak adlandırılır. LP'nin yerel nedensellik sınırlamasına bağlı kalması durumunda, simülasyon programının paralel bilgisayarlar üzerinde çalışması, tek bir bilgisayar üzerindeki çalışmasıyla tamamen aynı sonuçlar verecektir. Bu, simülasyon çalışmasının tekrar edebilir (repeatable) olması sonucunu ortaya çıkarır. Diğer bir değişle, aynı giriş verisi ve parametreler kullanılarak simülasyon programının çalıştırılması durumunda daima aynı sonuçlar üretilecektir [10].

Paralel bir uygulamada her bir mantıksal işlem seri bir ayrık olaylı simülasyon olarak görülebilir. Yani, her bir LP, kendi yerel durumunu ve kendisi için programlanmış ancak henüz işlenmemiş olan zaman işaretli olaylar listesini işler. Henüz işlenmemiş (askıda olan) olaylar listesi, aynı zamanda, diğer LP'lerden herhangi bir LP'ye gönderilen olayları da kapsar. LP'nin ana çalışma döngüsü, en küçük zaman işaretli olayı listeden kaldırır ve onu işlemeye başlar. Böylece, LP tarafından yapılan işlemler, olay tabanlı işlemler dizisi olarak görülebilir. Bir olayın işlenmesi LP içerisinde sıfır veya daha fazla durum değişkeninin değişebileceği ve LP'nin kendisi veya diğer LP'ler için ilave olayları programlayabileceği anlamına gelir. Her LP, kendisi tarafından işlenmiş en son olayın zaman işaretini gösteren bir simülasyon saatini kullanır. Bir LP tarafından programlanmış herhangi bir olay, olay programlandığında en azından LP'nin simülasyon zamanı saati kadar büyük bir zaman işaretine sahip olmalıdır [7-8].

3.1 Koruyucu Senkronizasyon

Geliştirilen, ilk senkronizasyon algoritmaları, koruyucu yöntemlere dayanmaktaydı [6]. Senkronizasyon algoritmasının *yerel nedensellik sınırlamasının* ihlalinden kaçınmak için önlem alması, '*koruyuculuk'* olarak tanımlanır. Bunun anlamı, herhangi bir koruyucu protokolün başlıca görevinin, bir olayın işlenmesinin ne zaman güvenli olacağının belirlemesi, yani daha küçük bir zaman-işaretine sahip bir olayın LP tarafından daha sonra alınmayacağının garanti edilmesidir.

3.2 İyimser Senkronizasyon

İyimser senkronizasyon, nedenselliği ihlal edebilen riskler alabilmesi gibi bir takım farklılıklarla koruyucu senkronizasyondan ayrılır. Olayların güvenle çalıştırılmasının garanti edilemediği durumlarda bile simülatör olayları işlemeye devam eder. Optimistik yöntemler nedensellik ihlallerine izin verir, ancak, bu tür olayları daha sonra tarayarak onları bulup düzeltebilir.

4 SANAL ORTAMLARIN KARŞILIKLI ÇALIŞILABİLİRLİĞİNİN GELİŞİM SÜRECİ

Çeşitli simülasyonlar ve simülatörler, özel hedefleri gerçekleştirmek maksadıyla, yönelinen alana ilk zamanlarda oldukça güç bir şekilde uygulanmıştır. Belirli bir eğitim yönteminin / uygulamasının hedeflerini tek bir sistem yerine getiremezse, yeni bir sistem geliştirilir veya hedefler mevcut yeteneklere uydurulmak üzere değiştirilir. 70'li yılların ikinci yarısında, simülatör olarak adlandırılan eğitim aygıtlarının birbirine bağlanması fikri biçimlenmeye başlamıştır. Simülatör teknolojisi, ilk olarak personel eğitiminde en etkili ve ucuz araçlar olarak askeri kurumlar tarafından benimsenmiştir.

Dağıtık sanal ortamlardaki ilk çalışmalar iki ayrı grup tarafından yapılmıştır. Bunlar askeri kurumlar ve bilgisayar oyun firmalarıdır. Aşağıda bu grupların karşılıklı çalışabilirlik üzerindeki çalışmaları kısaca özetlenmiştir.

Askeri simülatörleri karşılıklı çalıştırmak için ilk somut çalışma, 1983'te DARPA (Defence Advanced Research Projects Agency) tarafından başlatılmıştır. 1990 yılına kadar süren SIMNET (SIMulator NETworking) projesi, askerleri eğitmek için sanal dünyalar oluşturmak üzere dağıtık simülasyonların kullanımının pratikliğini göstermiştir. SIMNET, kollektif takım eğitimini desteklemek üzere tank eğitimcilerini karşılıklı çalıştıran bir çalışmaydı ve görüntü üretim teknolojileri, düşük maliyetli simülatör tasarımı, ağ teknolojileri, vb. alanlarda bir takım ilerlemeler göstermiştir.

Dağıtık sanal ortamlardaki diğer bir araştırma ve geliştirme çalışmalarını yapan grup internet ve bilgisayar oyunları konusunda çalışan ticari oyun firmalarıdır. Bu grubun yaptığı çalışmalar, zindanlar ve ejderhalar olarak adlandırılan bir oyundan 1970'lerde geliştirilmiş '*Adventure'* isimli bilgisayar oyununa kadar devam etmiştir. 1980'lerde MultiUser Dungeon (MUD) oyunlarının geliştirilmesi ile devam eden çalışmalar video oyun endüstrisinin gelişmesine önemli ölçüde yardımcı olmuştur.

SIMNET'in ve yukarıda bahsi geçen diğer alanlardaki çalışmaların başarısı, ağ tabanlı eğitim uygulamalarını destekleyen genel amaçlı protokollerin tanımlanması sonucunu doğurmuştur. Bu sonuç, aşağıda kısaca tanımlanacak olan iki farklı protokolün ortaya çıkmasıyla sonuçlanmıştır: Dağıtık İnteraktif Simülasyon (DIS) ve Yüksek Seviyeli Yapı (HLA).

4.1 DAĞITIK İNTERAKTİF SİMÜLASYON (DIS)

Birbirinden ayrı olarak geliştirilmiş simülatörler arasında karşılıklı çalışılabilirliği tesis etmek maksadıyla bir takım standartlar geliştirilmiştir. Bu standartlardan biri; simülatörleri birbirine bağlamak amacıyla kullanılan *Dağıtık İnteraktif Simülasyon (Distributed Interactive Simulation - DIS)* standartlarıdır. Dağıtık İnteraktif Simülasyon (DIS) protokolleri 1993 yılında IEEE standardı durumuna gelmiştir (IEEE Std 1278.1-1995). DIS kavramının anlamı, uyumlu mimari, modelleme, protokoller ve standartlar kullanılarak çeşitli konumlarda bulunan ağ haline getirilmiş simülatör(ler) aracılığıyla insanların etkileşim yapabildiği yapay bir ortamdır. DIS standardının içeriğini analitik simülasyonları da kapsayacak şekilde geliştiren *Yüksek Seviyeli Yapı (High Level Architecture - HLA)* günümüzde DIS standardının yerini almış durumdadır [9].

DIS protokolleri ve operasyonları kavramı SIMNET'inkileriyle yakın bir şekilde bağlanmıştır. Genel olarak DIS simülasyonları varlık-düzeyli ve gerçek-zamanlı simülasyonlardır. Bir model çalıştırma açışından bakıldığında, DIS uygulaması her biri kendi sahip olduğu perspektiften savaş alanının kendisine özgü tasvirini üreten, sanal (insanlı eğitim simülatörleri), canlı (fiziksel ekipman) ve yapısal (savaş oyunu simulatörleri ve diğer çözümsel araçlar) simülatörlerin otonom bir koleksiyonu olarak görünebilir [15]. Her bir simülatör, kendi durumundaki değişimler bir başka simülatörü bir şekilde etkileyecek olduğunda *protokol veri parçaları (protocol data units – PDUs)* olarak adlandırılan mesajları gönderir. Tipik PDUs yeni bir mevkiye hareketi, bir başka simüle edilen varlığa ateş açmayı ve diğer simülatörlere göre görünümdeki değişimleri içerir (bir tankın taretinin dönmesi gibi).

4.2 YÜKSEK SEVİYELİ YAPI (HLA)

Yüksek Seviyeli Yapı, U.S. Savunma Bakanlığı kapsamındaki simülasyonların yeniden kullanımını ve karşılıklı çalışmasını desteklemek üzere ortaya atılmış ve geliştirilmiştir. HLA, DIS'nin bir genellemesi ve uzantısı olarak ortaya çıkmıştır. HLA üç adet bileşen tarafından tanımlanmıştır [11-12]:

- Ortak bir model tanımı ve tanımlama formalizmi,
- HLA çalışma zamanı ortamını tanımlayan bir takım servisler,
- Mimariyle uyumu yöneten bir takım kurallar [13, 14].

HLA, savunma simülasyon uygulamalarının tümü karşısında geniş bir uygulanabilirliğe sahip olacak şekilde tasarlanmıştır. Bunlar; eğitim, analiz, görev provası ve kazanmayı kapsar. HLA yüksek derece esnekliğe sahip olacak şekilde tasarlanmıştır.

HLA'nın temelinde, federasyon kavramı yatmaktadır. Bir federasyon, yapı tarafından tarif edilmiş protokolleri kullanarak karşılıklı etkileşen federeler (simülasyonlar ve diğer sistemler) topluluğudur.

Yüksek seviyeli yapı; interaktif simülasyon modelleri ve ortamlarının gelişmesine katkıda bulunacak nesneye-yönelik bir yaklaşımdır. HLA, herkesi her şeyi bilmek için zorlamaktan ziyade, bilgiyi sadece ihtiyacı olan varlıklara veya ortamlara yönlendirerek simülasyon verisini ve ağ trafiğini iyi bir şekilde kontrol etmeye yarayan bir yöntemdir. HLA, birbirlerine bağlanabilme ve işbirliği yapabilme yeteneğine sahip simülasyonların karşılıklı çalışmasını sağlayan bir çerçevedir. HLA, genellikle uygulanabilir bir takım kurallar yoluyla karşılıklı çalışabilirliği (Interoperability) ve yeniden kullanılabilirliği (Reusability) tesis eder [16].

HLA içerisindeki tipik bir federasyon uygulamasında, federe federasyona katılır, kendi çalışma parametrelerini gösterir (örneğin; federenin federasyona sağlayacağı ve federasyondan kabul edip alacağı bilgi) ve daha sonra federe federasyondan ayrılana veya simülasyon sona erene kadar federasyon durumunun evrimine katılır.

5 SONUÇLAR

Simülasyonların çok işlemcili ve dağıtık bilgisayar sistemlerinde çalışmasıyla ilgili olan paralel ve dağıtık simülasyon teknolojileri 1970'lerden günümüze kadar aktif bir araştırma alanı olmuştur. Bilgisayar ve ağ teknolojilerinin gelişmesine paralel olarak bu teknolojilerde yeni metodolojiler ve protokollerle gelişimini sürdürmektedir.

Bu çalışmada, paralel ve dağıtık simülasyon teknolojilerini kısaca özetledik. Dağıtık sanal ortamlardaki merkezi konular senkronizasyon ve verinin verimli bir şekilde dağıtılmasıyla ilgilidir ve kendisinden önceki standartlara göre daha verimli çalışan ve kaynakları daha etkili kullanan HLA yapısı günümüzün bu konu üzerinde çalışan araştırmacılar tarafından en çok ilgi alan teknolojilerinden birisi konumuna gelmiştir. HLA, esnekliği ve tutarlılığı sayesinde bu alandaki üstünlüğünü kabul ettirmiştir. HLA standardı kendisinden önceki standartların (DIS, ALSP, vb.) yerini almalıdır. Mevcut veya geliştirilmekte olan tüm dağıtık modelleme ve simülasyon (M&S) uygulamaları HLA ile uyumlu olmalıdır.

KAYNAKLAR

[1] Zeigler, B. P., T. G. Kim, and H. Praehofer. 2000. *Theory of Modeling and Simulation*. New York, NY, Academic Press.

[2] Fujimoto, R. M. Parallel and Distributed Simulation. Proceedings of the 1999 Winter Simulation Conference**:** 122-131.

[3] Page, Ernest H., Introduction to Military Taining Simulation, Proceedings of the 1998 Winter Simulation Conference**:** 53-60.

[4] U.S. Department of Defense. 1997b. DoD modeling and simulation (M&S) glossary, DoD 5000.59-M,December.

[5] Chandy, K. M. and J. Misra (1978). "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs." IEEE Transactions on Software Engineering **SE-5**(5): 440-452.

[6] Chandy, K. M. and J. Misra (1981). Asynchronous Distributed Simulation via a Sequence of Parallel Computations." Communications of the ACM **24**(4): 198-205.

[7] Fujimoto, R. M. (1999). Exploiting Temporal Uncertainty in Parallel and Distributed Simulations. Proceedings of the 13th Workshop on Parallel and Distributed Simulation**:** 46-53.

[8] Fujimoto, R. M. (1999). Parallel and Distributed Simulation Systems, Wiley Interscience.

[9] IEEE Std 1278.1-1995 (1995). IEEE Standard for Distributed Interactive Simulation -- Application Protocols. New York, NY, Institute of Electrical and Electronics Engineers, Inc.

[10] Fujimoto, R.M. 1997. Zero lookahead and repeatability in the High Level Architecture. In *Proceedings of the 1997 Spring Simulation Interoperability Workshop,* Orlando, FL, 3-7 March.

[11] U.S. Department of Defense. 1996. DoD high level architecture (HLA) for simulations, Memorandum signed by USD(A&T), September.

[12] U.S. Department of Defense. 1998a. High level architecture object model template, Version 1.3, February.

[13] U.S. Department of Defense. 1998b. High level architecture rules, Version 1.3 (Draft 2) February.

[14] U.S. Department of Defense. 1998c. High level architecture interface specification, Version 1.3 (Draft 9) February.

[15] Zengin, A., Ekiz, H., Firat, C., The High Level Architecture (HLA) For Distributed Simulations, 3rd International Symposium on Intelligent Manufacturing Systems, August 30 – 31, 2001, Sakarya

[16] Zengin, A., Ekiz, H., Dağıtık Simülasyon Uygulamaları ve Yüksek Seviyeli Yapı (HLA), Elektrik – Elektronik Bilgisayar Mühendisliği 9. Ulusal Kongre ve Sergisi, 19 – 23 Eylül 2001, Kocaeli

[17] Bryant, R. E. (1977). Simulation of Packet Communication Architecture Computer Systems. Computer Science Laboratory. Cambridge, Massachusetts, Massachusetts Institute of Technology.