Avrupa Bilim ve Teknoloji Dergisi
Sayı 20, S. 85-92, Aralık 2020
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi



www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 20, pp. 85-92, December 2020
Copyright © 2020 EJOSAT
Research Article

Federelerin Yüksek Seviye Mimari Standart Uyumluluğunun Denetlenmesi İçin Çözüm Önerisi

Arzum Berrak Arıbal[1*], Tülin Erçelebi Ayyıldız[2]

[1*] Başkent Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye, (ORCID: 0000-0002-9834-5802), arzumberrakaribal@gmail.com
[2] Başkent Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye (ORCID: 0000-0002-7372-0223), ercelebi@baskent.edu.tr

(İlk Geliş Tarihi 12 Haziran 2020 ve Kabul Tarihi 9 Ekim 2020)

(**DOI:** 10.31590/ejosat.749583)

ATIF/REFERENCE: Arıbal, A.B. & Ayyıldız, T.E., (2020). Federelerin Yüksek Seviye Mimari Standart Uyumluluğunun Denetlenmesi için Çözüm Önerisi. *Avrupa Bilim ve Teknoloji Dergisi*, (20), 85-92.

Öz

Bu çalışma bir federasyona dahil olan federelerin Yüksek Seviye Mimari (High Level Architectre, HLA) uyumluluk sınamasının gerçekleştirilmesi için bir temel teşkil etmektedir. Federelerin HLA uyumunun denetlenmesi için mevcut yöntemler analiz edilmiş ve bu analiz sonrası HLA uyumluluğu ile beraber birlikte çalışabilirlik problemlerinin çözümü için bir yöntem önerilmiştir. Bu yöntem geliştirilen araç seti ile desteklenerek geliştirilmesi devam eden bir Koşum Zaman Altyapısı (Run-Time Infrastructure, RTI) ürünü ile birlikte denenmiş: birlikte çalışabilirlik problemleri en aza indirilmiş ve simülasyonların birlikte çalışabilirliği için gerekli olan HLA uyumluluğu test edilmiştir. Araştırma sonucunda birlikte çalışabilirlik faaliyetleri kapsamında karşılaşılan uyumluluk problemlerine bir çözüm üretilmiştir. Gerçekleştirilen birçok projede HLA kurallarının sadece belirli bir kısmı uygulanmakta olup bu yaklaşım simülasyon sistemlerinin birlikte çalışabilirliğini sağlamak açısından çoğu zaman yetersiz kalmaktadır. Bu nedenle federelerin HLA uyumlu tasarlanması büyük önem taşımaktadır. Özellikle askeri amaçlı olmak üzere farklı kuruluşlar tarafından farklı teknolojiler kullanılarak geliştirilmiş HLA uyumlu simülasyonlar mevcuttur. Müşterek harekât konseptinin gelişmesine paralel olarak farklı simülasyonların entegre bir şekilde çalışması ihtiyacı doğmuştur. Bununla birlikte HLA uyumlu simülasyonların entegrasyonu ve birlikte çalışabilirliği için standart uyumluluğu gözetmek kaçınılmaz hale gelmiştir.

Anahtar Kelimeler: Yüksek Seviye Mimari, Modelleme ve Simülasyon, Birlikte Çalışabilirlik.

Solution Proposal to Check High Level Architectural Standard Compliance of Federate's

Abstract

This study is the basis for performing the High Level Architecture (HLA) compatibility test of federates included in a federation. Existing methods were analyzed to check the HLA compliance of federates, and after this analysis, a method was proposed for the solution of interoperability problems with HLA compatibility. This method has been tested with a Run-Time Infrastructure (RTI) product that is still being developed by supporting it with the developed toolkit: interoperability problems are minimized and the HLA compatibility required for simulations interoperability has been tested. As a result of the research, a solution has been produced for compatibility problems encountered within the scope of interoperability activities. Only a certain part of the HLA rules are implemented in many developed projects, and this approach is often insufficient to ensure the interoperability of simulation systems. For this reason, it is very important to design the federates in compliance with HLA. There are HLA compatible simulations developed by different organizations using different technologies, especially for military purposes. In parallel with the development of the joint operation concept, the need for different simulations to work integratedly. However, it has become inevitable to observe standard compatibility for the integration and interoperability of HLA compatible simulations.

Keywords: High Level Architecture, Modelling and Simulation, Interoperability.

[*] Sorumlu Yazar: arzumberrakaribal@gmail.com

1. Giriş

Modelleme ve Simülasyon (Modelling & Simulation, MODSIM) sistemleri için birlikte çalışabilirlik, birden fazla benzer sistemin fiziksel olarak birbirine bağlı olması, birbirlerine mesaj, veri veya hizmet sunması ve sunulan bu mesaj, veri ve hizmetlerin bir arada çalışmaları için etkin bir biçimde kullanılması olarak tanımlanabilir. Böylece farklı simülasyon çözümlerinin güçlü yanları birleştirilerek, bu alanda kullanıcı ihtiyaçlarının daha iyi karşılanması hedeflenmektedir. Birlikte çalışabilirlik; "bir sistemin ya da sürecin, ortak standartlar çerçevesinde diğer bir sistemin ya da sürecin bilgisini ve/veya işlevlerini kullanabilme yeteneği" olarak da tanımlanabilir (IEEE Standard Computer Dictionary, 1990) (Tolk ve Muguira, 2003). Birlikte çalışabilirlik için sadece veri değişimi değil, simülasyonlar arası senkronizasyonda önem taşmaktadır. Bu amaçla birçok uluslararası standart tanımlanmıştır ve geliştirilen standartlardan en önemlisi Yüksek Seviye Mimari (High Level Architecture, HLA)'dir. HLA, 1995 yılında Amerikan Savunma Bakanlığı (Department of Defense, DoD) Savunma Modelleme ve Simülasyon Ofisi (Defense Modeling and Simulation Office, DMSO) tarafından askeri projelerin ihtiyaçlarına uygun olarak önerilmiştir. Amacı, DIS, ALSP gibi önceki standartlardan öğrenilmiş dersleri uygulayarak, çoklu donanım, yazılım ve ağ mimarilerini destekleyecek fonksiyonel bir mimariyi tanımlayarak mevcut simülasyon sistemlerinin birlikte çalışabilirliği ve yeniden kullanabilirliğini sağlamak için genel bir simülasyon çerçevesi oluşturmaktır. HLA üç versiyona sahiptir: 1998'de DMSO tarafından yayınlanan HLA 1.3, 2000 yılında IEEE (Institute of Electrical and Electronics) tarafından yayınlanan HLA1516-2000 ve 2010 yılında IEEE tarafından yayınlanan ve HLA Evolved olarak bilinen HLA1516-2010 (IEEE Standard for Modeling and Simulation High Level Architecture, 2000). Başlangıçta tamamen askeri uygulamaların ihtiyaçlarına uygun olarak geliştirilmiştir. Fakat günümüzde simülasyon sistemlerinin kullanıldığı bir çok farklı alanda uygulanmaktadır (Dahmann, Kuhl ve Weatherly, 1998) (Yılmaz, Durak, Taylan ve Oğuztüzün, 2014).

Bir simülasyon sisteminin HLA standardı ile uyumluluğu, "RTI Uyumluluğu" ve "Federe Uyumluluğu" olmak üzere iki farklı adım ile test edilmektedir. RTI Uyumluluğu, DMSO tarafından HLA standardı ile uyumluluk için 2000 civarında test ile gerçekleştirilir. Federe Uyumluluğu ise HLA servislerinin testi yapılarak sağlanabilir. Uyumluluk (Sertifikasyon) için temel adımlar şu şekildedir;

- Web üzerinden test uygulamasının çalıştırılması
- Federe geliştiricisinin web üzerinden FUT (Federate Under Test) için bir uygunluk çizelgesi göndermesi
- Arayüz çevre verilerinin sunulması
- Arayüz dokümantasyonu ve raporlamanın sunulması

Öte yandan DoD Ocak 2015'te bütçe kısıtları nedeniyle RTI sertifikasyon faaliyetlerini durdurduğunu açıklamıştır. Bu tarihten sonra simülasyon alanında gerçekleştirilen ürünler standartta tanımlanan servisleri gerçekleştirmesine rağmen sertifikalanmış «certified» veya doğrulanmış «verified» olarak tanımlanamamakta, ancak bu standartla uyumlu «compliant» olduğundan bahsedilebilmektedir. Bu nedenle de günümüzde HLA Evolved (IEEE 1516-2010) standardı ile sertifikalanmış veya doğrulanmış herhangi bir ürün bulunmamakta, geliştiriciler ürünlerini bu standarla uyumlu olarak geliştirmektedir.

Simülasyonun çok uzun bir geçmişi olmamasına rağmen, son yıllarda dünyada ve ülkemizde yaygın bir şekilde kullanılmaya başlanan modelleme ve simülasyon teknolojilerinin artan önemine bağlı olarak, ülkemizde özellikle savunma sanayiinde bu konuda ciddi çalışmalar yapılmaktadır ve dünyada bu işle uğraşan diğer kurum ve kuruluşlarla rekabet edecek HLA uyumlu birçok taktik, operatif ve stratejik seviye simülasyon sistemi geliştirilmiştir ve geliştirilmeye devam etmektedir. Geliştirilen bu modelleme ve simülasyon projelerinde de standartlara uyumluluk konusuna önem verilmekte, bu amaçla geliştirilen sistem ve projelerin uluslararası standartlara uyumluluk kapsamında bazı hazır çözüm yazılım ve altyapılar kullandıkları görülmektedir. Bununla birlikte literatürde yapılan araştırmalar doğrultusunda bilindiği kadarıyla, bir simülasyonun HLA standardına uyumluluğunu test eden bir çalışmaya rastlanamamıştır.

Bu çalışmada simülasyon projelerinde yer alan federelerin HLA uyumluluğunun sınanması için mevcut yönetemleri analiz edilmiştir. Yöntemler, HLA Standardı'nda yer alan federe kurallarına dayanarak analiz edilmiştir. Giriş bölümünde HLA'yı odakta tutan Modelleme ve Simülasyon temel kavramlarına genel bakıştan bahsedilmiştir. Bu makalenin geri kalan kısmı şu şekilde düzenlenmiştir. Bölüm 2'de Yüksek Seviye Mimari ile ilgili konulara genel bir bakış sağlayarak, federenin uyumluluğunun denetlenmesi ile ilgili yöntemler listelenmiş, Bölüm 3'te, stadarda uygun federe geliştirimi ile ilgili mevcut çalışmalar incelenmiş ve Bölüm 4'te ise elde edilen sonuçlar tartışılmıştır.

2. Materyal ve Metot

2.1. Yüksek Seviye Mimari (High Level Architeture, HLA) Bileşenleri

HLA'yı Tablo 1 üzerinde görüldüğü gibi (i) HLA Çerçeve ve Kurallar (IEEE Std 1516-2000); (ii) Federe Arayüz Spesifikasyonu (IEEE Std 1516.1-2000); (iii) Nesne Modeli Şablonu (OMT) (IEEE Std 1516.2-2000); (iv) Federasyon Geliştirme ve Çalıştırma Süreci (IEEE Std 1516.3-2003) kavramları oluşturmaktadır (IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA), 2000).

HLA Çerçeve ve Kurallar, federasyon iletimi sırasında, federelerin doğru şekilde etkileşimi için gerekli prensiplerdir. Kurallar, federelerin ve federasyonu tasarlayan kişilerin sorumluluklarını tanımlamaktadır.

Federe Arayüz Spesifikasyonu, federeler ve RTI arasındaki ara yüzü tanımlamaktadır. Simülasyonların birbirleriyle nasıl etkileştiğini açıklar.

Nesne Modeli Şablonu (Object Model Template, OMT), simülasyonda nesneleri, nitelikleri ve aralarındaki iletişim biçimini tanımaktadır (Falcone and Garro, 2019). FOM'un yapısını belirler. Federasyon Nesne Modeli (Federation Object Model, FOM) ve Simülasyon Nesne Modeli (Simulation Object Model, SOM) olmak üzere iki farklı modeldir.

Federasyon Geliştirme ve Çalıştırma Süreci, federasyonun geliştirilmesinde ve işletilmesinde uygulanması önerilen süreç adımlarını tanımlamaktadır.

Tablo 1. Yüksek Seviye Mimari (High Level Architecture, HLA)

Standart Tanımı	
IEEE 1516–2000	IEEE Standard for Modeling and Simulation (M&S) HLA Framework and Rules, approved by IEEE on September 21st, 2000.
IEEE 1516.1–2000	IEEE Standard for M&S HLA Federate Interface Specification, approved by IEEE on September 21st, 2000.
IEEE 1516.2–2000	IEEE Standard for M&S HLA Object Model Template (OMT) Specification, approved by IEEE on September 21st, 2000.
IEEE 1516.3–2003	IEEE Recommended Practice for HLA Federation Development and Execution Process, approved by IEEE on April 23rd, 2003.

2.2. Modelleme ve Simülasyon Terimleri

Bu bölümde HLA standardına genel bir bakış verilmektedir.

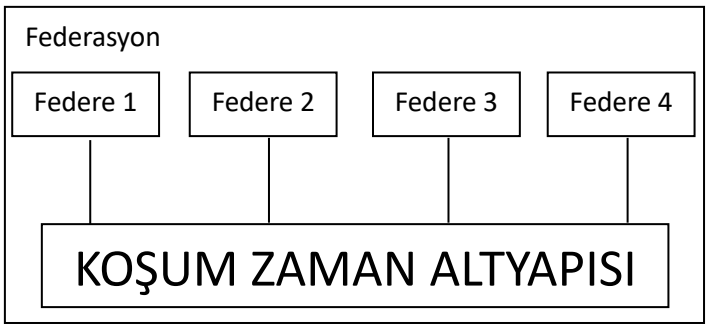
2.2.1. Federasyon

HLA standardında birlikte çalışan simülasyonların bütünü federasyon olarak ifade edilmektedir. Federasyon üç bileşenden oluşmaktadır (Gerlach, Durak and Knüppel, 2016).

Federe, federasyonun bir üyesidir ve tek noktadan RTI'a bağlıdır.

FOM, federasyon geliştirici tarafından tanımlanan, federelerin kendi aralarında değiş tokuş yaptığı veriler arasındaki ilişkileri tanımlamaktadır. Bir kısmı RTI tarafından okunan federeler için tanımlanmış bir anlaşmadır.

Koşum Zaman Altyapısı (Run-Time Infrastructure, RTI), federasyonun işletilmesini sağlamaktadır. İleri ki bölümlerde detaylandırılacaktır (Şekil 1).



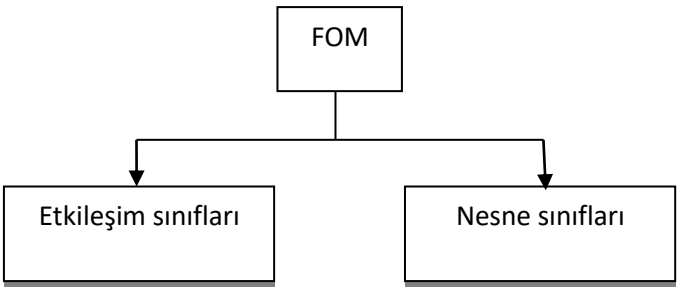
Şekil 1 – Koşum Zaman Altyapısı (Run-Time Infrastructure, RTI)

2.2.2. Federe

HLA standardında her bir simülasyon ise federe olarak isimlendirilmektedir. Federeler federasyon içerisinde işletilir (Federation Execution). Federe tek bir gemiyi, bir filoyu, bir ülkenin ordusunu temsil eden bir simülasyon vb. şeklinde olabilir.

2.2.3. Nesne Modeli Şablonu (Object Model Template, OMT)

Önceki bölümde de belirtildiği gibi, OMT tüm FOM'lar için bir yapı tanımlamaktadır Her bir federasyon, o federasyon için tanımlanmış FOM'u kullanmaktadır. FOM federeye özel bilgiler bulundurmaz; sadece, bir federasyon işletimi sırasında RTI üzerinden federelerin alıp vereceği veriyi tanımlamaktadır. Federasyon işletiminin başında FOM, RTI tarafından okunmaktadır. FOM, nesne sınıfları ve etkileşim sınıfları olarak iki temel bileşenden oluşmaktadır (Şekil 2).



Şekil 2 – Federasyon Nesne Modeli (Federation Object Model, FOM)

2.2.4. Etkileşim Sınıfları (Interaction Class)

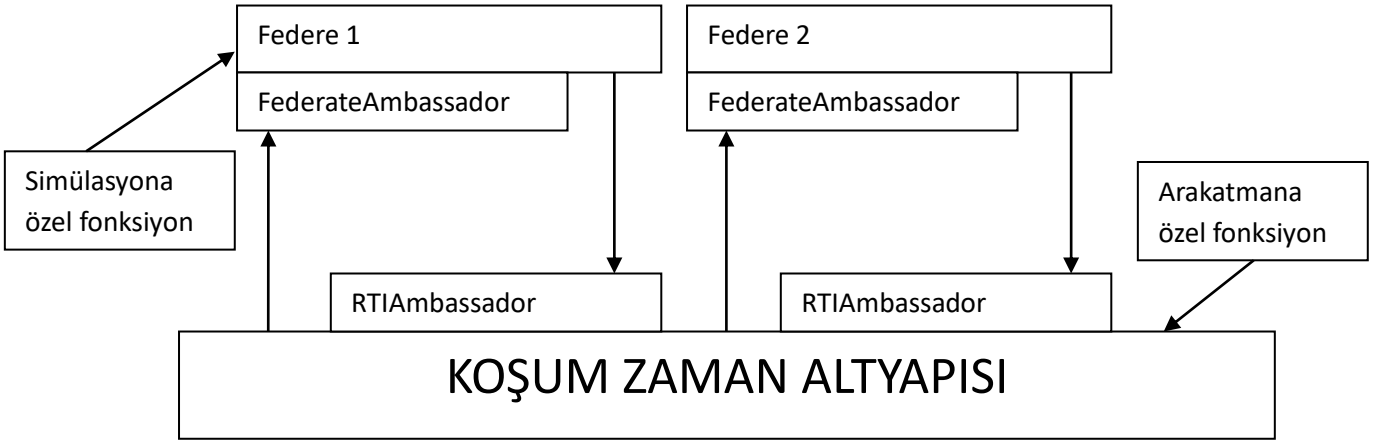
Etkileşim sınıfları, belirli bir anda bir bir federeden diğer federelere RTI üzerinden gönderilen veri bütününe verilen isimlerdir. Etkileşim sınıfları, simülasyonda oluşan ve diğer simülasyonların (federelerin) ilgilendiği bir oluşum ya da olaydır. Örneğin patlama olayı etkileşim kullanılarak diğer federelere iletilebilir. Bir federe etkileşimi gönderir (Send Interaction) ve ilgilenen diğer federeler etkileşimi alır (Receive Interaction). Etkileşim sınıflarında olay başlar ve biter, süreklilik içermez. Etkileşim sınıfları parametrelerden oluşmaktadır.

2.2.5. Nesne Sınıfları (Object Class)

Nesne sınıfları, simüle edilmiş varlıklardır. Birden fazla federenin ilgisi dahilinde tanımlanırlar. Etkileşim sınıflarından farklı olarak anlık değil, belirli bir süre boyunca kalıcılıklarını sürdürürler. OMT, nesne sınıflarını tanımlamaktadır. Etkileşim sınıflarının parametreleri içermesi gibi, nesne sınıfları da öznitellikleri (attribute) içerir. Federeler, nesnelerin örneklerini (instance of objects) yaratabilirler. Her bir örnek farklı bir kimliğe (identity) sahiptir. Özniteliklerde o örnek için farklı kimliğe sahiptir. Federeler, bir nesne örneğinin durumunu, öznitelikleri güncelleyerek değiştirirler. Federeler birbirleriyle, RTI üzerinden etkileşim ve nesneler aracılığıyla haberleşirler. Her bir federe kendi içinde tuttuğu simüle varlıkları, FOM içinde tanımlanmış nesnelere dönüştürmek zorundadır.

2.2.6. Koşum Zaman Altyapısı (Run-Time Infrastructure, RTI)

RTI, bir federasyonda tüm federelere iletişim hizmetleri sağlayan bir ara yazılımdır. **Hata! Başvuru kaynağı bulunamadı.** üzerinde görüldüğü gibi federeler birbirleri ile doğrudan konuşmazlar. RTI her bir federe için RTIambassador arayüzünü sunmaktadır ve federe bu arayüzü kullanarak RTI servislerini çağırmaktadır. Bu servisler federe tarafından başlatılan (federate-initiated) servisler olarak ifade edilmektedir. Her bir federe FederateAmbassador



Şekil 3 – FederateAmbassador & RTIambassador Arayüzü

arayüzünü sunar. RTI bu arayüzü kullanarak federe ile olan etkileşimini yürütür. Bu servislerde, RTI tarafından başlatılan (RTI-initiated) servisler olarak ifade edilir. FederateAmbassador ve RTIambassador kullanılarak çağrılan tüm servisler standartta tanımlanmıştır. FederateAmbassador soyut sınıftır ve federe bu soyut sınıftan türetilmek zorundadır.

2.3. Federasyon & Federe Kuralları

HLA standardı federeler için 5 kural ve federasyonlar için de yine 5 kural olmak üzere, toplam 10 kural tanımlamaktadır. Bir simülasyon modelinin HLA uyumlu olduğunun söylenebilmesi için HLA standardında yer alan on kuralın hepsine uyması yeterli bir koşul olacaktır. Başka bir deyişle eğer simülasyon modeli bu on kuralın hepsini sağlıyorsa HLA uyumlu olduğu kabul edilmektedir.

Federasyon kuralları aşağıdaki gibidir;

Kural 1: Federasyonlar, OMT'ye göre tanımlanmış FOM'a sahip olmalıdır. OMT, RTI'ın ihtiyaç duyduğu bilgilerden daha fazlasını içerir. Bu ekstra bilgi, federasyon için tutarlılığı sağlamak için mühendisler tarafından kullanılır.

Kural 2: Bir federasyonda nesne örnekleri RTI yerine federelerde tutulmalıdır. Bu RTI'ın jenerik kalmasını sağlar. Bu beraberinde bir kısıtı da getirir. RTI hiçbir nesnenin özniteliğinin değerini saklamamalıdır.

Kural 3: Federasyon işletimi sırasında, federeler arasında alıp verilen ve FOM'da tanımlanan veri RTI aracılığı ile federelere ulaştırılmalıdır. Başka bir deyişle federeler birbirleri ile doğrudan haberleşemez. Bu kuralın amacı tekrar kullanılabilir simülasyon bileşenleri oluşturmaktır.

Kural 4: Federasyon işletimi sırasında federeler RTI ile Federe Arayüz Spesifikasyonu dokümanında belirtildiği üzere haberleşmelidir. Bu kuralın amacı farklı RTI implementasyonlarını kullanmanın yolunu açmaktır.

Kural 5: Federasyon işletimi sırasında bir nesnenin özniteliğine sadece tek bir federe sahip olabilir.

Federe Kuralları aşağıdaki gibidir.

Kural 6: Federeler, HLA Nesne Modeli Şablonu (OMT) dokümanına uygun olarak oluşturulmuş bir SOM'a sahip olmalıdır.

Kural 7: Federeler, Simülasyon Nesne Modeli (SOM)'da belirtilen nesne özniteliklerini almalı ya da vermelidir. Benzer şekilde SOM'da belirtilen etkileşimi almalı ya da göndermelidir. Bu kurala uyulduğu takdirde, federe veri açısından HLA standardı ile uyumlu olur.

Kural 8: Federeler dinamik olarak bir özniteliğin sahipliğini alıp verebilirler. Bu bilgi SOM'da bulunmak zorundadır.

Kural 9: Federeler öznitelikleri yayınlarken koşulları (örneğin threshold) değiştirebilirler. Bu bilgi SOM'da belirtilmek zorundadır.

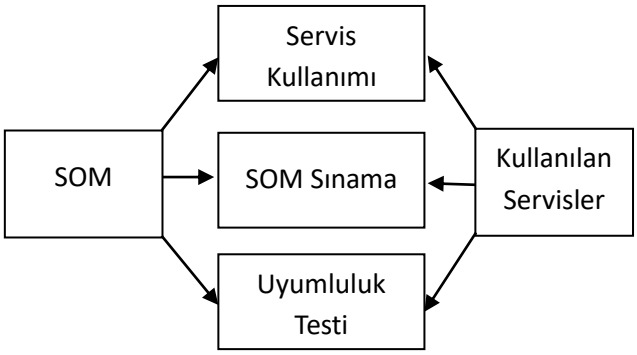
Kural 10: Federeler kendi lokal zamanlarını yönetebilirler.

2.4. Uyumluluğun Sınanması

Federenin HLA uyumluluğunu sınanması için "Test Simülasyonu" olarak adlandırılan bir test aracı geliştirilmiştir. Geliştirilen test aracı yardımı ile test senaryoları hazırlanmaktadır. Bu araç simülasyon sistemleri birlikte çalışabilirliği için CEY Savunma bünyesinde geliştirilmesi devam eden HLA uyumlu, yüksek performanslı, yerli ve milli bir RTI yazılım çözümü testlerinde de kullanılmaktadır. Bu kapsamda federelerin HLA uyumluluğu HLA 1516 2000.1, 2000.2 ve 2000.3 standartına uygun olarak gerçekleştirilmiştir. HLA standardında federelerin uyulması gereken kurallar belirtilmiştir. Federelerin SOM ile bu kurallara uyması beklenir. Bu kurallara uymadığı durumda federe amaçları bilinmediği için birlikte çalışabilirlik faaliyetlerini yürütmek zorlaşmaktadır (Özkaymak, Zorba, Arıbal and Sayın, 2019).

Federe kurallarında görüldüğü gibi Kural 10 dışındaki bütün kurallarda SOM'dan bahsedilmektedir fakat bir simülasyon altyapısı için SOM kullanımı zorunlu değildir. Zorunlu olmaması nedeniyle kullanılmaması durumunda federeler HLA uyumlu olamamaktadırlar. Bir federenin HLA uyumlu olmaması ise HLA standardı tarafından vurgulanan birlikte çalışabilirlik ve tekrar kullanılabilirlik hedeflerinin gerçekleştirilmediği anlamına gelmektedir. Bu nedenle federenin HLA uyumluluğunu test etmek için SOM'un kullanılması gerekmektedir. Uyumluluğun sınanması için gerekli adımlar **Hata! Başvuru kaynağı bulunamadı.** üzerinde görülmektedir. Örneğin X Federesinin HLA uyumluğunun sınanması için birtakım girdilere ihtiyaç vardır. Bunlardan ilki SOM girdisidir. SOM'a sahip federenin Gerçek Zamanlı Platform Referans FOM (Real-time Platform Reference FOM, RPR-FOM) uyumlu olması gerekmektedir.

SOM verisi simülasyonda mevcut ise bu sınama için ilk girdi sağlanmış olmaktadır. İkinci test girdisi olan Kullanılan Servisler, federenin kullandığı HLA servislerinin listesidir. Bu iki veri girdisi sağlanırsa sınama adımları başarılı şekilde işletilir.



Şekil 4 – Federe Uyumluluğunun Sınanması

2.4.1. SOM (Simulation Object Model) Sınama

İlk sınama adımı federenin 2000.1 HLA kuralları ve 2000.3 OMT kurallarına uygunluğunun sınanması ile gerçekleştirilir. Bunun için SOM sınaması yapılır. Bu sınama adımı statiktir. Örnek SOM verisi okunarak SOM dosyasının içeriğinin OMT standartındaki kurallara uygunluğu test edilir (Şekil 5).

SOM sınama işlemi için, Şekil 6 üzerinde yer alan adımlar uygulanır. 1 numaralı adımda yer alan SOM dosyası örneği Şekil 5 üzerinde yer almaktadır. SOM dosyası, federeler arasındaki etkileşimi/iletişimi açıklayan geçerli bir XML dosyası olmalıdır. Bölüm 2.2.4'te SOM dosyasında kullanılan etkileşimlerle ilgili, Bölüm 2.2.5'te ise nesne sınıfları ile ilgili bilgiler verilmiştir. Geçerli bir SOM dosyası mevcut ise 2 numaralı adım kontrol edilmelidir. Şekil 5'te bulunan SOM dosyası birinci satırı üzerinde bir HLA nesne modeli (objectModel) yer almaktadır. HLA nesne modeli, model ismi, model tipi, model versiyonu, model referansı vb. gibi bilgiler barındırmalıdır. Eğer HLA nesne modeli içinde bulanan bir bilgi uygun değilse nesne modelinin ilgili bilgisine "NA" girilmelidir. 2 numaralı adımda yer alan nesne modeli kontrolü bu şekilde sağlanır. 3, 4, 5 ve 6 numaralı adımlarda, SOM dosyası üzerinde ikinci satırda ve daha sonraki satırlarda yer alan nesne sınıfları ve onların özniteliklerinin yapısı, etkileşim sınıfları ve onların parametrelerinin yapısı kontrol edilir. Nesne/Etkileşim/Öznitelik/Parametre isimlendirmeleri XML adlandırma kurallarına uygun olmalıdır. XML adlandırma kurallarında iki nokta üst üste ":" ve "hla" dizesiyle başlayan isimler veya (("H|h") ("L|l") ("A|a")) ile eşleşen dizeler dahil edilmemelidir. 7 numaralı adımda lexicon kontrolü gerçekleştirilir. Bu adımda SOM dosyasında bulunan nesne sınıfları, etkileşim sınıfları ve parametre sınıflarının her birinin içerisinde "semantic" özelliği kontrol edilir. Eğer adımlar sonucunda HLA standardına uygun şekilde sonuçlar elde edilirse sınama işlemi başarılı kabul edilir (Şekil 6).

2.4.2. Servis Kullanımı

İkinci sınama adımı için geliştirilen test aracı kullanılarak bir test senaryosu hazırlanır. Test senaryosu federenin kullandığını beyan ettiği servislerden oluşmaktadır. Sınaması yapılan federe ve test federesi ile RPR-FOM verisi kullanılarak test senaryosu işletilir. Bu test sonrasında sınaması yapılan federenin beyan ettiği tüm servisler kullanılmış ise sınama adımı başarılı kabul edilir (Şekil 7).

Gerçekleştirilen bu sınama adımı aşağıdaki sorular ile yönetilir.

- Zorunlu olan servisler çağırılmış mı?
  - Federasyonu Yarat (Create Federation Execution)
  - Federasyonu Yoket (Destroy Federation Execution)
  - Federasyona Katıl (Join Federation Execution)
  - Federasyondan Ayrıl (Resign Federation Execution)
- Federenin kullandığını beyan ettiği diğer servisler çağırılmış mı?
  - Nesne Sınıfına Abone Ol (Subscribe Object Class)
  - Nesne Sınıfını Yayınla (Publish Object Class)

2.4.3. Uyumluluk Testi

Üçüncü sınama adımı ise SOM'da federe tarafından verilen bilgilerin sınanmasının denetlenmesidir. Bu sınama adımı 2. sınama adımı işetilirken RTI yazılımı vasıtasıyla toplanan veriler ile gerçekleştirilir (Şekil 8).

- Yayınlamayı ve Abone Olmayı beyan ettiği sınıfları yayınlayıp abone olmuş mu?
- Sahipliği devredebileceğini beyan etmiş ise öznitelikleri devredebiliyor mu?
- Göndereceğini beyan ettiği etkileşimleri göndermiş mi?
- Güncelleyeceğini beyan ettiği öznitelikleri güncellemiş mi, nesneler yaratılmış mı?
- Abone olacağını beyan ettiği öznitelikleri almış mı, nesneleri keşfetmiş mi?
- Boyut (dimension) tanımı yaptığı öznitelikleri bölgelerle ilişkilendirmiş mi?

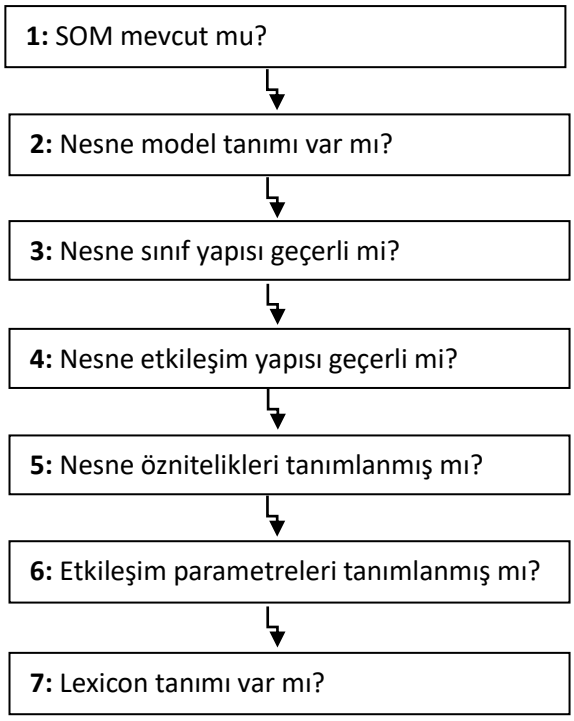
Uyumluluk testlerinde birkaç farklı adım gerçekleşmektedir. Standart servis grupları için ön koşullar sunmaktadır. Bu ön koşulları sağlayarak uyumluluk testleri işletilmiştir.

Nesne testleri için iki federe ve aralarındaki iletişimi sağlamak için RTI kullanılmaktadır. Test edilecek federe bir nesne yarattığı zaman (registerObjectInstance) eğer test uygulaması federesi yaratıldığı belirtilen nesneye abone işlemi gerçekleştirmiş ise RTI test uygulaması federesine keşfetme işlemi gerçekleştirmektedir (discoverObjectInstance).

HLA standardına göre federeler birbirlerine doğrudan veri gönderemez ancak veriyi federasyon için erişilebilir hale getirebilirler. RTI bu veriyi ilgilenen federelere ulaştırmakla görevlidir. Bu işlem deklarasyon yönetimi servisleri ile gerçekleştirilmektedir. Bu servisleri kullanarak federeler ilgilerini deklare edebilirler ve veriyi üretebilirler (publish) ya da veriyi tüketebilirler (subscribe to). RTI bu deklarasyonları kullanarak veriyi yönlendirir, değiştirir ve ilgi yönetimi yapar. Yönlendirme, RTI'nın abonelik (subscription) bilgilerini kullanarak hangi federelerin veriyi alacağına karar vermesi ile gerçekleştirilir. Değiştirme, verinin abonelik durumuna göre tekrar adlandırılması ile gerçekleştirilir. İlgi yönetimi, veriyi üreten (publish) federeler için yapılır. RTI, veriyi üreten federeye o veriye herhangi bir federenin abone olup olmadığını söyler. Aboneliğin olmadığı durumda federe veriyi üretmeyebilir. Nesne

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE objectModel SYSTEM "hla.dtd"><objectModel DTDversion="1516.2" name="New FOM.xml" type="FOM" vers
<objects>
<objectClass name="HLAobjectRoot" sharing="Neither">
<attribute name="HLAprivilegeToDeleteObject" dataType="NA" updateType="NA" updateCondition="NA" ownership=
<objectClass name="GriddedData" sharing="PublishSubscribe" semantics="Depict global, spatially varying env
<attribute name="TotalValues" dataType="UnsignedLong3" updateType="Conditional" updateCondition="On change
<attribute name="RecordTotal" dataType="UnsignedShort4" updateType="Conditional" ownership="NoTransfer" sh
<attribute name="CoordinateSystem" dataType="EnvironmentDataCoordinateSystemEnum16" updateType="Conditiona
<attribute name="GridDataInfo" dataType="GridDataStructArray1" updateType="Conditional" updateCondition="C
<attribute name="SampleTime" dataType="UnsignedLongLong1" updateType="Conditional" updateCondition="On cha
<attribute name="RecordNumber" dataType="UnsignedShort4" updateType="Conditional" ownership="NoTransfer" s
<attribute name="GridAxisInfo" dataType="GridAxisStructArray1" updateType="Conditional" updateCondition="C
<attribute name="FieldNumber" dataType="UnsignedShort4" updateType="Conditional" ownership="NoTransfer" sh
<attribute name="ConstantGrid" dataType="EnvironmentGridTypeEnum8" updateType="Conditional" updateConditic
<attribute name="VectorDimension" dataType="HLAoctet" updateType="Conditional" updateCondition="On change'
<attribute name="GridID" dataType="EntityIdentifierStruct" updateType="Static" updateCondition="N/A" owner
<attribute name="Orientation" dataType="OrientationStruct" updateType="Conditional" updateCondition="On ch
<attribute name="EnvironmentType" dataType="EnvironmentTypeStruct" updateType="Static" updateCondition="N/
<attribute name="NumberOfGridAxes" dataType="HLAoctet" updateType="Conditional" updateCondition="On change
</objectClass>
<objectClass name="EmitterBeam" sharing="Subscribe" semantics="A sector of concentrated energy from a devi
<attribute name="BeamElevationCenter" dataType="Float1" updateType="Conditional" updateCondition="&gt; EE
<attribute name="PulseRepetitionFrequency" dataType="Float12" updateType="Conditional" updateCondition="&g
<attribute name="BeamIdentifier" dataType="HLAoctet" updateType="Static" updateCondition="N/A" ownership="
<attribute name="EventIdentifier" nameNotes="27" dataType="EventIdentifierStruct" updateType="Conditional"
<attribute name="EmissionFrequency" dataType="Float12" updateType="Conditional" updateCondition="&gt; EE E
<attribute name="PulseWidth" dataType="Float13" updateType="Conditional" updateCondition="&gt; EE PW THRSH
```

Şekil 5 – SOM Dosyası



Şekil 6 – SOM Sınama Adımları

ile veri göndermenin ilk koşulu, bir federenin yeni bir nesne örneği yaratmasıdır (register). Nesne örneği yaratılması RTI'ya yeni bir unsurun federasyona girdiğini anlatmaktadır. Bazı özniteliklere ya da özniteliklerin tamamına abone (subscribe) olan federe, yeni bir unsur yarattığında, nesneyi keşfeder (discover). Aynı zamanda etkileşim (interaction) gönderme ve alma yine bu servis grubunda yer alan servislerdir. Tüm bu servislerin davranışı Deklarasyon Yönetimi servis grubunda bulunan yayınlama (publication) ve abonelik (subscription) servisleri tarafından kontrol edilir. Bu yöntemle simülasyon bileşenlerinin birbirlerine gereksiz mesaj göndermeleri engellenebilir. Etkileşim testleri de SOM dosyasında yer alan nesnelerin/etkileşimlerin abone ol/yayınla mekanizması ile uygulanan testler olarak gerçekleştirilmiştir.

Sahiplik testlerinde sahipliği devretmenin üç farklı yolu vardır. İlk olarak test edilecek federe FOM'da yer alan özniteliğinin sahipliğini hemen elden çıkarabilir. Bunun için başka bir federe nesneyi alacak mı diye beklemesine gerek yoktur. Federe bu sahipliği elden çıkarır ve artık bu öznitelik sahipsiz olur (Divest Attribute Ownership Unconditionally). İkinci bir seçenek olarak test edilecek federe FOM'daki özniteliğinin sahipliğini elden çıkarmak istediğini RTI' a bildirir. RTI bu isteği diğer federelere haber verir. Diğer federeler tarafından bu özniteliğe sahip olmak isteyen bir federe olursa RTI devir işlemini tamamlar ve tüm federelere haber verir (Divest Attribute Ownership By Negotiation). Son olarak test edilecek federe özniteliğini elden çıkarmak istediğini diğer federelere bildirir almak isteyen olursa devri gerçekleştirir (Divest Attribute Ownership If Wanted). Burada sahipliği devretmek için test edilen adımlar şunlardır:

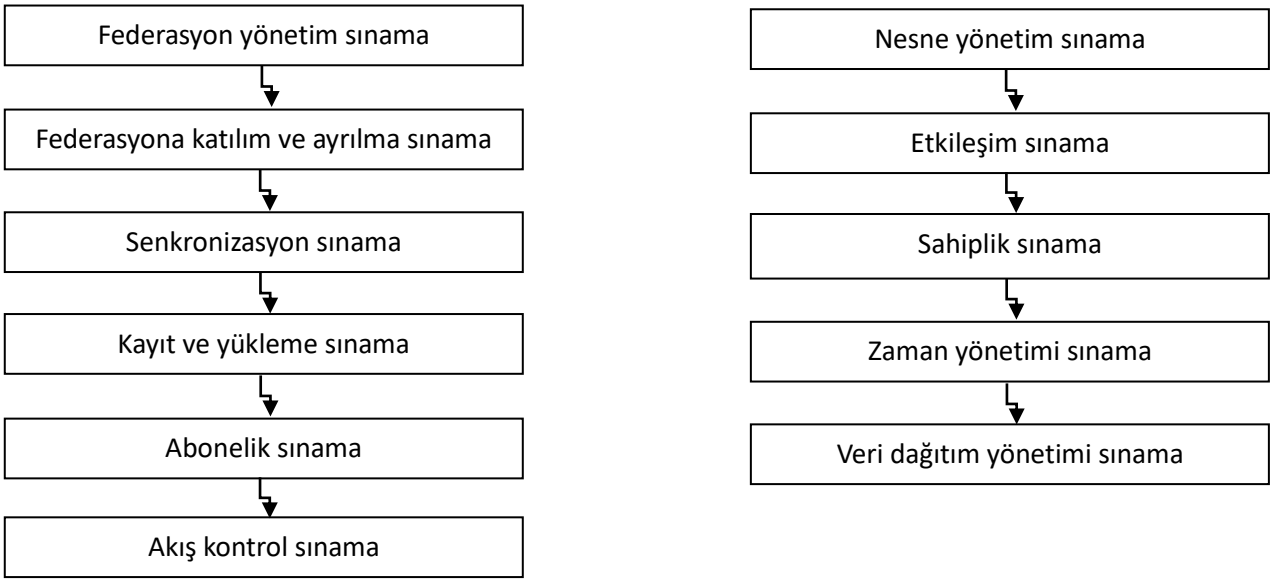
Her test için;

-Öznitelik yayınlanmış mı?

-Nesne yaratma (register) işlemi yapılmış mı? (Bu adımda register eden federe özniteliğin sahipliğini almış demektir)

-Sahiplik, devretmek isteyen federede mi?

-Her notification servisi için geri dönüş var mı bakılır.



Şekil 7 – Servis Kullanım Aracı



Şekil 8 – Uyumluluk Testi Adımları

Sahiplik almak için test edilecek durumlar:

Her test için;

-Öznitelik yayınlanmış mı?

-Nesne yaratma (register) işlemi yapılmış mı?

-Sahipliği devralmak isteyen federe, almak istediği özniteliklere sahip olmamalı kontrolü (Olan bir öznitelik tekrar alınamaz)

-Her notification servisi için geri dönüş var mı bakılmalıdır.

Veri Dağıtım testleri verilerin tanımlanan belirli bir alan için paylaşılması ile gerçekleşmektedir. Burada güncelleme bölgesi (registrationRegion) ve abonelik bölgesi (subscriptionRegion) isimli iki ayrı bölge yaratılır. Bölgelerin alt ve üst limitlerinin ne olduğu RTI'a bildirilir. Bu test üzerinde güncelleme bölgesinin alt limit 2, üst limit ise 4 olarak verilir. Abonelik bölgesinin ise alt limiti 5, üst limiti 7'dir. Güncelleme ve abonelik bölgeleri kesişmediği için test uygulaması federesinin yaptığı güncellemeler test edilecek federe tarafından alınmamaktadır. Daha sonra güncelleme bölgesinin alt limiti 2, üst limit 6 olarak güncellenerek RTI'a bildirilir. Bu güncelleme sonucunda güncelleme bölgesi ve abonelik bölgesi 5 ve 6 aralığında kesiştiği için test uygulaması federesinin yaptığı bu bölge içinde bulunan güncellemelerin test edilecek federe tarafından alındığı görülür.

Zaman Yönetimi testleri federelerin kendi kontrol zincirlerinde yürüttükleri, federeler arasındaki olayların doğru sıralanmasının gerektiği önemli bir sorundur. HLA'da olayların sıralaması "mantıksal zaman" olarak ifade edilir. Mantıksal zaman soyut bir kavramdır; herhangi bir temsile veya zaman birimine bağlı olması gerekmez. Zaman yönetimi hizmetleri iki şey için kullanılır:

• Her federenin diğer federelerle koordineli olarak mantıksal zamanını ilerletmesine izin verir.

• Zaman damgalı olayların dağıtımını kontrol eder, böylece federenin "geçmişinde" diğer federelerden hiçbir zaman, yani mantıksal zamanından daha az mantıklı zamanı olan olayları almaması gerekir.

RTI, bir federenin zaman yönetimine katılacağı zamanı seçmesine izin verir. Bir federe zaman kısıtlı olabilir, bu durumda mantıksal zaman ilerlemesi diğer federeler tarafından kısıtlanır. Bir federe zaman düzenleyici olabilir, bu durumda mantıksal zaman ilerlemesi diğer federeleri düzenler. Bir federe zaman kısıtlamalı olabilir ve zaman düzenleyici olabilir veya hiçbiri olmayabilir. Federasyonun amaçlarına ve federenin ihtiyaçlarına bağlı olarak farklı seçimler yapılır. Hem düzenleyici hem de kısıtlı olan federelerin yerel saatlerini, federenin değeri belirli federelerin yerel saatinden daha az olan zaman damgalı bir olay alabilecek şekilde ilerletmesine izin verilir.

3. Araştırma Sonuçları ve Tartışma

3.1. Standarda Uygun Federe Geliştirimi

Bölüm 2.4'te gerçekleştirilen test adımları işletilerek Federe Uyumluluğu sınanmıştır. Bu kapsamda gerçekleştirilen bu sınama işlemleri bir federenin HLA uyumluluğunun test edilmesi için yeterli adımlardır. Yine aynı bölümde verilen testler gözden geçirilerek bahsedilen ve gerçekleştirimi sağlanan testlerde olduğu gibi HLA federe kurallarına uyulduğu durumda federelerin ihtiyaç duyduğu ve dışarıya servis ettiği veriler bilinmektedir. Federelerin Uyumluluk Testi bölümünde bahsedildiği gibi sahiplik yönetimi ile ilgili bilgi almasını ya da Abone Ol/Yayınla mekanizmasında bahsedildiği gibi federelerin bilgi değişimini nasıl yaptıklarını anlamak mümkündür.

Bahsedilen bölümde gerçekleştirilen sınama işlemleri gerçekleştirildikten sonra tanımlanan SOM ve Federe Uyumluluk verisi yeniden kullanılabilirlik açısından çok değerlidir. Yine aynı veriler ile başka bir simülasyon sistemi için şablon olarak kullanılarak birlikte çalışabilirlik sağlanabilir.

4. Sonuç

Bu makalede, federenin HLA uyumluluğunun denetlenmesi konusunda bir çalışma sunulmuştur. Federenin HLA uyumluluğunun sınanması ile ilgili adımlar ayrıntılı olarak açıklanmıştır. Federenin birlikte çalışabilirliğini sağlamak; teknik, sözdizimsel, anlamsal ve pragmatik düzeyde çaba ve standardizasyon gerektirmektedir. Modelleme ve Simülasyon sektöründe ülkemizde simülasyonların birlikte çalışabilirliği için gerekli olan HLA uyumluluğuna duyulan ihtiyaç, bu çalışmada geliştirilen araç setleri ile CEY Savunma şirketi bünyesinde geliştirilmekte olan RTI ürünü ile test edilmiştir. Bu testin gerçekleştirilmesi, birlikte çalışabilirlik sorunlarının bir kısmının giderilmesine yardımcı olmuştur. Bu sayede HLA uyumlu federeyi oluşturan simülasyon bileşeninin tekrar kullanılması durumunda oluşabilecek birlikte çalışabilirlik problemleri en aza indirilmiştir.

Kaynakça

- IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. (1990). New York.
- Tolk, A. Muguira, J. (2003). The Levels of Conceptual Interoperability Model, *Fall Simulation Interoperability Workshop*. Paper 03F-SIW-007.
- IEEE Standard for Modeling and Simulation HighLevel Architecture (HLA)-- Framework andRules. (2000). IEEE Std 1516-2010, 1-38.
- Dahmann, J.S., Kuhl, F., Weatherly, R. (1998). Standards for Simulation: As Simple As Possible But Not Simpler The High Level Architecture For Simulation, *SIMULATION*, 71, 378-387.
- Yılmaz, F. Durak, U. Taylan, K. Oğuztüzün, H. (2014). Adapting functional mockupunitsfor HLA-compliant distributed simulation. *Proceedings of the 10th International Modelica Conference* (247-257) Lund, Sweden.
- IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA): 1516–2000 (Framework and Rules); 1516.1-2000(Federate Interface Specification); 1516.2–2000 (Object Model Template (OMT) Specification).
- Falcone, A., Garro, A., (2019). Distributed Co-Simulation of Complex Engineered Systems by Combining the High Level Architecture and Functional Mock-up Interface, S*imulation Modelling Practice and Theory*, 97.
- Gerlach, T. Durak, U. Knüppel, A. (2016, June). *Running High Level Architecture in Real Time for Flight Simulator Integration*. Paper presented at the AIAA Modeling and Simulation Technologies Conference. https://doi.org/10.2514/6.2016-4130
- Özkaymak, S. Zorba, H.O. Arıbal, A.B. Sayın, M.B. (2019) Birlikte Çalışabilirlik Sorunları ve ARTI ile Çözüm Yaklaşımı, *Ulusal Savunma Uygulamaları Modelleme ve Simülasyon Konferansı (USMOS)*, ODTÜ,Ankara.