

## İŞLETİM ZAMANLI KOŞUM ANALİZİ

**M. Fatih Hocaođlu<sup>(a, b)</sup>, Erdem Reşber<sup>(b)</sup>**

<sup>(a)</sup>İstanbul Medeniyet Üniversitesi, Mühendislik ve Mimarlık Fakültesi, İstanbul  
mfatih.hocaoglu@medeniyet.edu.tr

<sup>(b)</sup>Agena Bilişim ve Savunma Teknolojileri, 34844 İstanbul Teknopark /İstanbul  
erdem.resber@agenabst.com

### ÖZ

Çalışmada, koşum zamanlı analiz, koşum yineleme ve enstantane kayıt yönetimi için kural tabanlı ve kullanıcı tanımlı bir çözüm geliştirilmiştir. İşletim zamanlı analiz çözümü, iki temel hedefe sahiptir. Bunlardan birincisi koşum sonrası analizlerin işletim zamanına taşınmasıdır. Burada amaç, tekrarlı koşum-analiz çevrimlerinden kurtularak, istatistik sonuçlara dayalı yürütülecek seçimlerin koşum zamanlı olarak gerçekleştirilmesi ve senaryo işletiminin yönetilmesi amaçlanır. Benzer şekilde, doğru istatistiklerin hesaplanmasına yetecek koşum uzunluğunun (ilişkili olarak küme sayılarının) hesaplanması koşum zamanına taşınacaktır. İkinci hedef ise, tanımlı şartların sağlanması durumunda veya kullanıcı tanımlı zaman çizelgelerine uygun olarak, koşuma ait enstantane kayıtların alınması ve ilgili zaman noktalarının yeni senaryo düzenlemeleri için hazırlanmasıdır.

Çalışma ile koşum sonrası analizler ve bu analizlere dayalı karar süreci, koşum zamanına taşınmış, elde edilen analiz sonuçları ile koşumun yönlendirilmesi sağlanmış ve koşum sonrası analizler ile tekrarlanan koşum çevrimleri otomatikleştirilmiştir. Çözüm Etmen tabanlı Simülasyon Sistemi (EtSiS) tabanlı olarak geliştirilmiştir ve HLA dağıtık koşum için özel bir uyarlama gerektirmemektedir.

**Anahtar Kelimeler:** Dinamik koşum yönetimi, Enstantane kayıt, EtSiS, Koşum yineleme, Kural tabanlı işletim, Yığın koşum

## EXECUTION TIME ANALYSIS

### ABSTRACT

In this study, a user defined generic rule based solution is developed for run time simulation analysis and execution replication management and snapshot management. The run time analysis has two fundamental purposes. The first one is to carry post-data analysis into run time phase. By discarding necessity of execution-analysis cycle being repeated, taking actions and driving simulation execution based on statistical results in run time is the decisive aim to do that. Similarly, computation of execution length as enough as to calculate accurate statistics and number of bathes are carried into run time. The second purpose is to get simulation snapshots at scheduled time points and/or in the cases that user defined conditions are satisfied and to prepare the snapshots for new scenario definitions.

By this study, a novel approach is proposed to move post-data analysis and decision process based on the analysis results into simulation run time and to manage simulation execution regarding with the analysis results, and to automate simulation execution and analysis cycles. The solution is developed based on Agent driven Simulation Framework (AdSiF) and it does not need any customization for HLA based distributed simulation.

**Keywords:** AdSiF, Batch execution, Dynamic execution management, Replication, Rule based execution, Snapshot record

### 1. GİRİŞ

Etmen tabanlı simülasyon sistemi EtSiS'in simülasyon ve etmen programlama için sunduğu esnek geliştirme ve işletim ortamı, simülasyon modelleri ile işletim zamanlı etkileşimlerle senaryo yönetimine imkan sağlamaktadır.

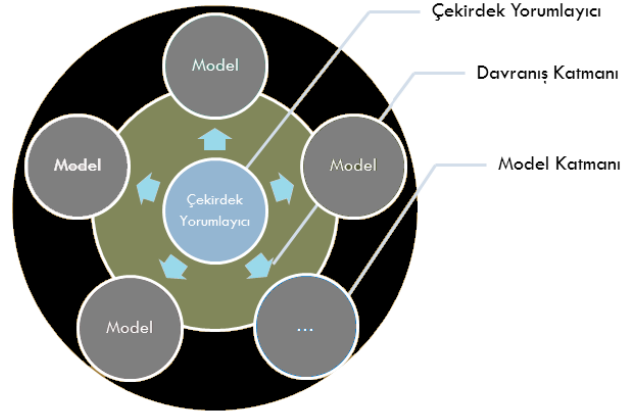
Bu makaleye konu olan çalışmada, modeller arası mesaj etkileşimleri ve modellerin öznitelikleri arasında kurulan abonelik mekanizması ile koşum yineleme karalarının verilmesi ve yinelemelerin gerçekleşmesi, enstantane kayıtların alınması, yığın ve enstantane koşumlar için makro tanımlama imkanlarının sağlanması ve koşum zamanlı istatistik analizlerin yürütülerek koşumun yönlendirilmesi için bir çözüm geliştirilmiştir. Literatürde işletim zamanlı analizleri, simülasyonda yürütülen fonksiyonların gerçek zamanlılık kısıtını sağlayıp sağlayamayacağını test etmek amacıyla işletim zamanını ölçmek amaçlı olarak kullanıldığını görmekteyiz. Bunun iyi örneklerinden biri Matlab/Simulink için geliştirilen ve simülasyon davranış işlemci zamanlarını ölçen En kötü durum koşum zamanı uygulamasıdır (worst-case execution time (WCET) analysis) [1]. WCET'den farklı olarak, simülasyon işletiminde hafıza kullanımının ölçülmesi ve analizine dönük daha genel amaçlı çalışmalar literatürde yer almaktadır [2]. Simülasyon tasarımlarının durum tabanlı sembolik işletimleri ile analiz

edilmesinin yine işletim zamanlı analiz kavramı içerisinde ele alındığı görülmektedir [3].

Bölüm 2’de simülasyon işletiminin modeller ile yönetilmesi kavramını ele alınmış olup, Bölüm 3’de koşum yineleme, enstantane kayıtların modellerin güdümüyle nasıl alındığı ve koşum zamanlı analiz açıklanmıştır. Bölüm 4’de yığın koşumlar ve enstantane kayıttan başlatılan koşumlar için makro tanımlamaları ele alınmıştır. Makale özelliklerin sağladığı avantajların tartışıldığı sonuç bölümü ile sonlandırılmıştır.

## **2. MODEL GÜDÜMLÜ SİMÜLASYON YÖNETİMİ**

EtSiS mimari yapısı merkezinde bir yorumlayıcı bulunan ve üst katmanında davranışlar ile simülasyon ve etmen yönetim davranış yapılarının modellendiği bir yapıdadır [4]–[6]. Şekil 1’de yapı ana hatları ile görülmektedir. Şekilde çekirdek yorumlayıcıyı saran katman simülasyon yönetim algoritmalarının yine EtSiS’in kendi dil yapısı ile tanımlandığı katmandır. Sistemin merkezi bir yorumlayıcı ile davranış yönetimi, geliştirilen uygulamaların iş mantığına dayanmayan, jenerik bir dil ile yönetilmelerini sağlar. Bu simülasyon işletim algoritmalarının (kesikli veya sürekli), senkronizasyon algoritmalarının, olay yönetiminin EtSiS belirtimsel betik dili ile yönetilmesini sağlar ve EtSiS’i bir simülasyon ortamından hesaplama ortamı olmaya taşır. EtSiS kesikli olay zaman yönetiminde DEVS [7]’i kuşatan ve farklı olarak merkezi koordinasyona ihtiyaç duymadan paralel simülasyon işletimi sağlayan bir çözüme sahiptir. Paralel işletimde senkronizasyon için iyimser ve kötümser senkronizasyon algoritmalarının işletimine imkan sağlar ve kullanıcı tanımlı algoritmalara açık bir mimari sunar. Sürekli olay işletimlerinde sürekli zaman akışı kesikli bir uzaya resmedilir ve zaman adımları koşum boyunca değişken yapıda kullanılır. DEVS’den bir diğer önemli farklılık, modelleme ve işletim süreci durum tabanlı programlama paradigması (DtPP) kavramı üzerinde, bir dil formunda toplanmış ve çoklu programlama desteği DtPP bünyesinde birleştirilmiştir. Model katmanında yer alan modeller ise, simülasyon işletim davranış yapıları tarafından yönetilen, betik dil ile modellenmiş tasarımlardır ve altyapı, gerek çekirdek yorumlayıcı, gerekse de çekirdek yapı üzerinde bulunan davranış katmanı, ile modeller arasında herhangi bir bağımlılık kurmaz.

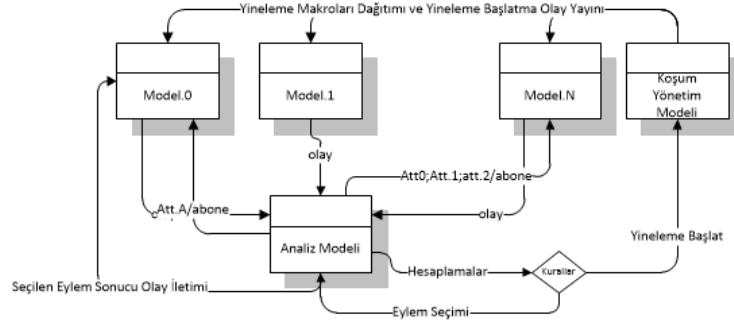


**Şekil 1. EtSiS Mimari Gösterim**

Anılan bu özellikler, simülasyon modellerinin ve etmenlerin farklı işletim algoritmaları ile işletiminin sağlanabileceğini ve bunun yalnızca davranış tasarımları ile çekirdek yorumlayıcının değiştirmeden yapılabileceğini gösterir. Örnek olarak, kuyrukta bekleyen olayların seçim kuralı, davranış işletim öncelikleri, dağıtık simülasyon uygulamalarında senkronizasyon algoritma seçimleri model iş mantıklarından bağımsız olarak yürütülür. Koşum yineleme kararları ve işletimleri ile enstantane kayıtları bu özelliğe büyük ölçüde dayanan fonksiyonel kullanımlardır.

### 3. ÇEVİRİM İÇİ ANALİZ VE KOŞUM YİNELEME

Çevrim içi analiz modeli simülasyon içerisinde ilişkilendirilen tüm modellerin paylaştığı koşum verilerini işletim zamanlı olarak analiz eden bir etmendir. Veri paylaşımı olay iletimleri ve öznitelik aboneliği ile sağlanır. Olay gönderimleri ilgili değer değişimini gerçekleştiren durum geçişinde, duruma ilişkilendirilmiş olay ile iletilirken, öznitelik aboneliğinde iki farklı yöntem izlenir. Bunlardan ilki, tanımlı frekanslarda özniteliğin değerinin alınması, diğeri ise, özniteliğin değer değişimlerinde yeni değeri ileten bir olayın çizelgelenmesidir. Şekil 2’de görüldüğü şekilde modeller olay iletimleri ile hesapladıkları parametreleri iletirken, analiz modeli modellerin üzerinde hesaplama yürüteceği özniteliklerine abone olarak erişim sağlar. Analiz modeli içerdiği ve elde ettiği verileri analiz ettiği fonksiyonları yazılım eklentisi (plugin) olarak işletim zamanlı olarak yükler. Bu durum modelin farklı analizleri yürütebileceği fonksiyonların yüklenebileceği bir esneklik sağlar. Analiz modelinin işlemlerini e-tablo (elektronik tablo) tabanlı olarak yürütmesi ve MVC (Model, Viewer, Controller) tabanlı tasarımı uygulama sahasından bağımsızlığı destekler.



**Şekil 2. Çevrim içi Analiz Modeli Etkileşimleri**

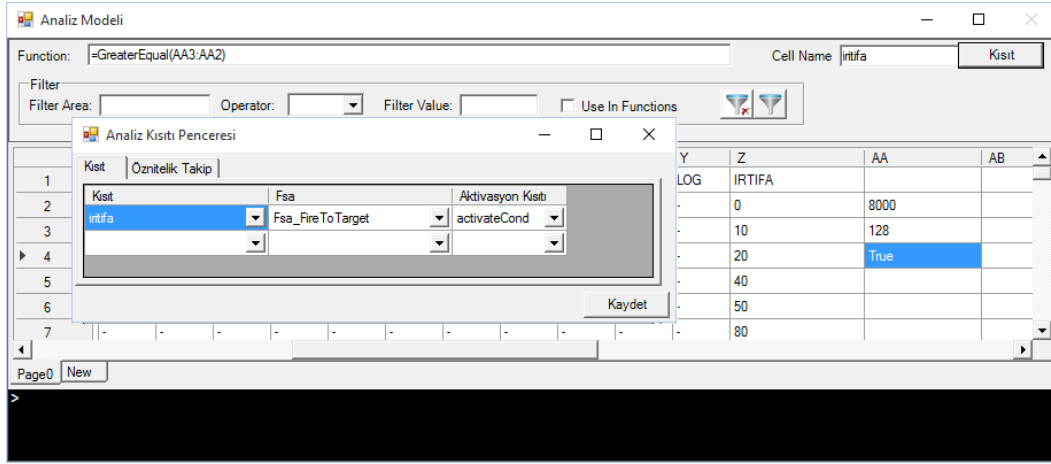
Çevrim içi analiz yeteneği koşum yineleme ve enstantane kayıtları alma yeteneklerini içerse de, koşum yineleme çizelgeleri ve makro tanımlamaları için ayrıca koşum yönetim modeli tanımlanmıştır. Koşum yönetim modeli koşum yinelemelerine ait makro tanımlamalarını, ilgili modellere dağıtımlarını, koşum yineleme olayının tüm modellere dağıtımını ve simülasyon koşum hızı ayarlama görevlerini üstlenir. Şekil 2’de görüldüğü gibi analiz modeli koşum yönetim modeline bir yineleme başlatma emri verir. Koşum yönetim modeli ilgili yineleme makrolarını modellere dağıtır ve koşumu yeniden başlatacak olayı yayımlar. Olayı alan tüm modeller Şekil 1’de görülen ve çekirdek yeteneklere erişen çekirdek üzerinde işletilen davranışları işleterek koşum yönetimini sağlarlar. Analiz modeli dahil simülasyon modellerinin tümü lokal koşumda tek bir proses olarak işletilir. Dağıtık koşumda ise kullanıcının dağıtımına bağlı olarak farklı makinelerde işletilirler.

### 3.1. Çevrim İçi Analiz

Çevrimiçi analiz koşum esnasında simülasyon senaryosunda yer alan modellerin ürettikleri ve kullanıcı tarafından seçilen değerlerin işletim zamanında toplanması ve tanımlı analizlere tabi tutulması ile gerçekleştirilir. Analizin üç aşaması vardır;

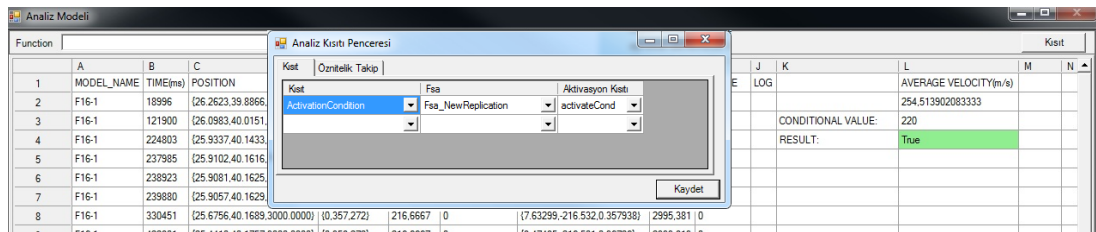
1. Üretilen verilerin seçimi,
2. Analiz hesaplamalarının tanımlanması,
3. Karar yordamlarının belirlenmesi,

Bir olay ile gönderilen ve analiz modeli tarafından alınan koşum verisi Şekil 3’de görüldüğü gibi adreslenen hücrelere yazılır. Şekilde bir uçağa ait yükseklik, pozisyon, hız, hasar seviyesi ve yakıt bilgileri görülmektedir. Koşum boyunca üretilen her yeni değer analiz modeline iletilir ve model arayüzünde yer alan e-tabloya yerleştirilir.



**Şekil 3:Analiz Modeli E-Tablo**

Tanımlamanın ikinci aşamasında tablo üzerinde seçilen hücrelere analiz fonksiyonları tanımlanır. Şekil 3’de irtifa verilerinin yazıldığı kolonun maksimumunun alındığı bir tanımlama görülmektedir (bkz. E-tablo hücre AA3 fonksiyon =Max(Z2:end)). Bu fonksiyon koşum esnasında e-tabloya eklenen her bir irtifa verisi için tekrar çalıştırılmakta ve güncel fonksiyon çıktısı karar yapılarında kullanılmaktadır. Benzer şekilde AA4 hücresinde tanımlanan kısıt ile uçağın AA2 hücresinde tanımlı irtifayı geçtiğinde hedefe atış edilme davranışı işletimi görülmektedir. Karar yapıları belirli kuralların tanımı sonucu bir doğruluk değeri döndüren fonksiyon olarak tanımlanır. Kuralların davranış yapıları ile ilişkilendirilmeleri, davranışlara bir güdüm şartı olarak atanmaları ile sağlanır. Şekil 3’de hız ortalama değerlerinin L3 hücresindeki değerden büyük olması durumu için bir matematiksel kıyaslama operatörü tanımlanmıştır (Hücre L4). Tanımlanan operatör ilişkili davranışı yöneten bir kısıt olarak seçilmiş ve isim verilmiştir (*ActivationCondition* isim olarak atanmıştır). Şekil 4’de yeni bir koşum yineleme başlatılması için gerekli tanımlama yapılmıştır. Tanımlamada seçilen şartın sağlanması ile koşum yineleme davranış işletilecektir. Kısıt tanımlamada herhangi sınırlama bulunmamaktadır. İç içe yuvalanmış (nested) olarak mantıksal operatörler ve matematiksel operatörler kullanılabilirler.



**Şekil 4:Analiz Modeli Kural Tanımlama**

### 3.2. Koşum Yineleme

Koşum yineleme simülasyonda örnekleme hacmini artırarak, rastsal sayı kullanımından doğan dalgalanmaları elimine etmeye yetecek örnek hacmine ulaşmak için yürütülür [8], [9]. Yineleme sayısı koşum esnasında elde edilen istatistikler kullanılarak işletim zamanlı olarak hesaplanmıştır. Tanımlı koşum zamanı sonuna ulaşılması veya koşum sonlanma şartı sağlanması durumunda tüm modellere simülasyon yönetimi görevi üstlenen model tarafından yineleme mesajı gönderilir. Simülasyon yönetimi herhangi bir simülasyon modeli tarafından üstlenebileceği gibi bu amaç için geliştirilmiş bir model tarafından da yürütülebilen bir görevdir. Uygulamamızda yapısal olarak standart bir model tasarımına sahip bir simülasyon yönetim modeli geliştirilmiş ve tüm simülasyon uygulamalarında kullanılmak üzere kavramsal model kütüphanesine eklenmiştir.

Id	Rastsal Sayı Kökü	Çizelgeler	Model Özellikleri Dosyası
1	345124	Çizelge1;Çizelge2	D:\Project\AdSIF\models\E
2	1237	Çizelge3	D:\Project\AdSIF\models\E

Repikasyon Özellikleri  
Çizelgeler  
Çizelge3

Model Özellikleri Dosyası  
D:\Project\AdSIF\models\Export  
Gözet

Rastsal Sayı Kökü  
1237  
 Sistem Zamanı

Repikasyonu Kayıt Et  
Repikasyonu Güncelle  
Repikasyonu Sil

Çizelge Güncelle

Repikasyonları Dosyadan Yükle  
Repikasyonları Dosyaya Çıkar  
Tamam

**Şekil 5. Örnek Bir Yineleme Tanım Arayüzü**

Yineleme işlemi ile simülasyon saati başlangıç zamanına çekilir, modellerin ajandasında yürüttüğü davranışlar sonlandırılır ve başlangıç durumu davranışları aktive edilir ayrıca model öznitelikleri başlangıç değerlerine getirilir ve modelin ilgili yineleme numarası ile ilişkilendirilmiş betikleri işletilir. Betik içeriğinde aşağıdaki eylemler yer alır;

- Olay çizelgeleri: Koşum yinelemesinde uygulanacak olay çizelgesi tanımlanır. Olay çizelgelemede amaç, tanımlanan zamanlarda, tanımlanan zaman aralıkları ile senaryo varlıklarına olay gönderimlerinin çizelgelenmesidir. Olay çizelgeleri, senaryoya koşum esnasında eklenecek varlıkların, eklenme zamanları ile çizelgelenmelerini de içerir.
- Dağılım ve rastsal sayı seçimleri: Koşum yinelemesinde kullanılacak rastsal sayı üretim tekniği ve davranışlara ait durumlar için kullanılacak, teorik dağılım ve parametre seçimleri tanımlanır.
- Model öznitelik tanımlamaları; Modellerin hangi yinelemede hangi özniteliklerinin hangi değerleri alacağı belirtilir.

İlgili tanımlamalar Şekil 5’de görülen arayüz kullanılarak gerçekleştirilir. Uygulamada simülasyona dahil edilmiş olan tüm modellere, koşum yineleme olayı gönderilir ve alınan olaya karşı bir davranış işletimi gerçekleştirilir. Yinelemelerin olay-davranış mekanizması ile yürütülüyor olması, tasarımcıya, farklı davranış tasarımları ile yineleme işlevlerini farklı bir semantik akış içerisinde tasarlayabilme imkanı verir.

	A	B	C	D	E	F	G	H	I	J
1	Yineleme-1	Sayfa1	=Avg(A1:end)@B1							
2	Yineleme-2	Sayfa2								
3	Yineleme-3	Sayfa3								
4										

**Şekil 6: Koşum Yineleme Analizleri**

Analiz modeli koşum esnasında her bir koşum yinelemesi için yeni bir e-tablo sayfası açar ve koşuma ait tüm koşum verilerini ve hesaplamaları ilgili sayfada tutar. Tüm koşum yinelemelerini içeren istatistiksel hesaplamalar ise bir ana sayfa üzerinde koşum boyunca hesaplanarak görselleştirilir ve istatistiklere dayalı olarak tanımlanmış kurallar işletilerek ilişkilendirilen davranışlar yönetilir. İlişkilendirilen davranışlar yeni bir koşum yinelemesinin başlatılması, koşumun sonlandırılması, senaryo kaynaklarının değiştirilmesi (varlık ekleme, çıkarma), senaryo varlıkları öznitelik değerleri değiştirme, tanımlı olayların çizelgelenmesi ve kullanıcılar tarafından tanımlanmış herhangi bir davranış kümesi olabilir. Şekil 6’de koşum yineleme analizleri görülmektedir. Ana sayfadan da görülebileceği gibi, hangi yinelemenin hangi sayfada yer aldığı yazılmakta ve ilişkilendirilmiş sayfa değerleri kullanılarak analizler yürütülmektedir.

Standart uygulamalarda koşum sonrası analizler kapsamında ele alınan ve yürütülen koşumlardan elde edilen istatistiklere göre hesaplanan örneklem hacmine göre veya uzun koşumların kümeleme ortalamaları (batch-means) analizleri ile yürütülen kararların işletim zamanında gerçekleşmesi sağlanır.

#### **4. ENSTANTANE KAYIT KOŞUMLARI VE YIĞIN KOŞUMLAR**

Enstantane koşum kayıtları simülasyonun işletimi esnasında alınan ve farklı bir zamanda kayıt noktasından simülasyonu başlatmaya yetecek detayda bilgileri içeren kayıtlardır. Enstantane kaydı, yineleme uygulamalarında olduğu gibi, tanımlı zaman aralıklarında, tanımlı şartların sağlanması durumunda bir modelin yayınladığı ve senaryodaki tüm modeller tarafından alınan olay ile yürütülür. Olayın tetiklediği davranış ile her model ajandasında aktif olarak yürütmekte olduğu davranışlar ve model öznitelikleri otomatik olarak kaydedilir ve kullanıcı tanımlı özel veri kayıtları için ilgili yazılım arayüzleri işletilir.



Enstantane kayıtlar üzerinde senaryo tanımlama imkanı sağlayarak, işletimin şartları tanımlanmış özel bir anında alınan bir enstantane kaydının değiştirilerek, daha detaylı analizler yapılabilecek, “Olsaydı ne olurdu (What-if)” sorgulamalarının daha geniş bir yelpazede tasarımlar yapılmasına imkan sunar.

Yığın koşum birden fazla senaryonun ardışık olarak işletimini ifade eder. Her bir senaryonun işletimi kendi içerisinde yinelemeler yürütebilir. Yinelemelerin model güdümünde, işletim zamanlı alınan kararlarla yürütülebiliyor olması, yığın koşumlarda senaryo yinelemeleri için özel tanımlamalar yapılmasını gerektirmez. Yineleme ve yığın koşumların her bir senaryo koşumu için rastsal sayı seçimleri belirtilir.

Enstantane kayıtlarının yüklenmesi, kayıtların alınmasına benzer bir yapıda, olay etkileşimi ile gerçekleştirilmektedir. Modeller tarafından veya kullanıcı etkileşimi ile gönderilen enstantane yükleme olayı ile olayda belirtilen enstantane kaydı yüklenir ve işletilmekte olan senaryo ilgili zaman noktasına geçerek işleme enstantane zaman noktasından devam edilir.

Modellerin enstantane kayıt alma ve yükleme ve yineleme koşumları olay gönderimleri, tanımlı şartların gerçekleşmesi ve seçilen davranışların durum geçişleri ile sağlanır. Örnek olarak, bir hava savunma simülasyonunda lider uçağın vurulması bir yineleme kısıtı olarak tanımlanabilir ve bu durumda uçağın vurulması durumunda olay tüm modellere gönderilir ve senaryo başlangıç durumundan yeni bir yineleme olarak koşuma başlar.

## 5. SONUÇ

Çalışmada sunulan çevrim içi analiz modeli çözüm yaklaşımı ile, geleneksel olarak koşum sonrası gerçekleştirilen analizlerin çevrim içerisinde gerçekleştirilmesi sağlanmıştır. Önceden veya işletim esnasında tanımlanan kurallar ile koşumun yönlendirilmesi, senaryo kaynaklarının değiştirilmesi, işletim esnasında varlıklara olay enjekte edilmesi ve iyi tanımlı karar yordamları ile senaryo yönetimi sağlanmıştır. Koşum yineleme sayısı, senaryo yinelemelerinde farklı rastsal sayı köklerinin kullanımı ve betik tanımları ile yineleme konfigürasyonlarının hazırlanması koşum zamanlı ve otomatikleştirilmiş bir formda gerçekleştirilebilmiştir.

Yığın koşumlar ile birden fazla senaryonun tek bir oturumda ardışık olarak yürütülmesi ve kullanıcı müdahalesine ihtiyaç duyulmadan yürütülmesi sağlanmıştır. Uzun senaryo koşum süreleri, özellikle analiz simülasyonlarında, müdahalesiz koşuma olan ihtiyacı artırmaktadır. Çözüm ile ihtiyacı çözmeye yönelik esnek ve kullanıcı tarafından genişletilebilir bir çözüm üretilmiştir.

Enstantane kayıtlar üzerinde senaryo tasarım değişiklikleri yapılarak yeni bir senaryo olarak işletilebilmeleri sağlanmıştır. Bu avantaj, simülasyon uygulamalarında sıklıkla beklenen “Olursa ne olur (what-if)” sorusunun cevabını aramada daha ileri bir noktaya,

belirli simülasyon durumlarında ve zaman noktalarında alınan kararların etkisini görmeye daha etkili bir uygulama olmaktadır. Bu amaca hizmet etmek için tanımlı şartlar ve zamanlarda alınan enstantane kayıtları, senaryo tasarımına girdi olarak kullanılabilir.

## 6. KAYNAKÇA

- [1] R. Kirner, R. Lang, G. Freiberger, and P. Puschner, "Fully automatic worst-case execution time analysis for MATLAB/Simulink models," in *Proceedings - Euromicro Conference on Real-Time Systems*, 2002, pp. 31–40.
- [2] J. Weidendorfer, M. Kowarschik, and C. Trinitis, "A Tool Suite for Simulation Based Analysis of Memory Access Behavior," in *Proceedings of the 4th International Conference on Computational Science ICCS*, 2004, vol. 3, pp. 440–447.
- [3] T. Schule and K. Schneider, "Exact runtime analysis using automata-based symbolic simulation," in *First ACM and IEEE International Conference on Formal Methods and Models for Co-Design, 2003. MEMOCODE '03. Proceedings.*, 2003.
- [4] M. F. Hocaoglu, "AdSiF: Agent Driven Simulation Framework," *Hunsv. Simul. Conf. - HSC2005*, 2005.
- [5] Mehmet F. HOCAOGLU, "Agent based Simulation: Lecturer Notes," AUZEF, Istanbul University, Istanbul, 2014.
- [6] M. F. Hocaoglu, "AdSiF: Agent driven Simulation Framework," in *USMOS 2011- National Defense Application and Modeling & Simulation Conference -(in Turkish)*, 2011.
- [7] T. G. K. B. P. Zeigler, H. Praehofer, *Theory of Modeling and Simulation*. Florida: Academic Press, 2000.
- [8] D. M. N. Jerry Banks, John S. Carson II, Barry L. Nelson, *Discrete-Event System Simulation*, 5th ed. Prentice Hall, 2009.
- [9] M. F. Hocaoglu, "Artificial Neural Network Embedded in Simulation: An Example on Logging operations in Forestry." 1997.