Sayısal Almaç Planlayıcı Bileşeninin Biçimsel Doğrulanması

Mustafa DURSUN, Özgür KIZILAY

REHİS EH Görev Yazılımları Müdürlüğü,
ASELSAN A.Ş., Ankara, Türkiye
{mdursun,ozgurk}@aselsan.com.tr

Özet Biçimsel doğrulama bir sistemin doğrulanmasının tasarım aşamasında yapılmasına izin veren ve son yıllarda kullanımı hızla artan bir yöntemdir. Bu bildiride almaç planlayıcı bileşeni tasarımının gerçek zaman gereksinimleri dahil edilerek biçimsel doğrulanması sunulmaktadır. Bu kapsamda sayısal almaç planlayıcı bileşeninin biçimsel modelinin ve gereksinimlerinin UPPAAL model denetim aracı ile belirtimi ve doğrulanması ele alınmaktadır.

Anahtar Kelimeler Model Denetim, Gerçek Zamanlı Yazılım, Biçimsel Doğrulama

1 GİRİŞ

Elektronik taarruz [1, 2] işlevi için taarruz uygulanacak yayınların parametrelerinin izlenmesi ve güncellenmesi gerekmektedir. Hedef izleme işlevinin doğru ve verimli yapılabilmesi, izlemeyi sağlayan planlamanın sistem kısıtlarına göre dinamik olarak yapılmasına ve yapılan planlamanın sayısal almaçta [1] doğru zamanlamayla uygulanmasına bağlıdır. Almaç kontrol yazılımının sayısal almaç planlayıcı bileşeni [3], hedef izleme işlevi kapsamında dinamik olarak almaç planlamanın oluşturulması ve oluşturulan planlama adımlarının uygulanmasını gerçekleyen gerçek-zamanlı [4] yazılım konfigürasyon birimidir. Almaç planlayıcı bileşeni sistemin işlevsel ve zamansal gereksinimlerini sağlaması amacı ile zaman tetikli özel bir durum ve görev tasarımına sahiptir. Sayısal almaç planlayıcı bileşeni, dinamik olarak tarama rejimi [3] oluşturmakta ve oluşturulan tarama rejimlerinin adımlarını sayısal almaçta uygulamaktadır. Sayısal almaç planlayıcı bileşeninin doğruluğu, tarama rejiminin sayısal almaçta doğru zamanlarda uygulanmasına ve sayısal almaç ile sayısal almaç planlayıcı bileşeninin durum tutarlılığının sağlanmasına bağlıdır. Sayısal almaç planlayıcı bileşeninin doğrulanması dinamik bir planlama yapmasından, gerçek zaman kısıtlarına sahip olmasından ve almaç kontrol yazılımındaki diğer görevler ile bağımlılığının bulunmasından dolayı yüksek bir test maliyetine sahiptir. Bir yazılımın tasarım aşamasında doğrulama yapılmasını, test maliyetini düşürecek bir etkendir. Yazılımların tasarım aşamasında doğrulanmasına imkan veren biçimsel doğrulama, yaygın olarak kullanılan etkin bir yöntemdir [9].

Bu bildiride sayısal almaç planlayıcı bileşeninin biçimsel doğrulanması sunulmaktadır. Bölüm 2'de biçimsel doğrulama ve model denetim anlatılmaktadır. Bölüm 3'te almaç planlayıcı tasarımı, sistem bileşeni modelleri ve sistem gerek-

sinimlerinin biçimsel belirtimi sunulmaktadır. Bölüm 4'te ise değerlendirme ve sonuç sunulmuştur.

2 BİÇİMSEL DOĞRULAMA VE MODEL DENETİMİ

Biçimsel doğrulama, sistem davranışının sistem gereksinimlerine göre matemetiksel modeller ve belirtimler kullanılarak doğrulanmasını sağlayan bir yöntemdir. Biçimsel doğrulamanın en büyük avantajı tasarımın erken aşamalarda doğrulanmasına olanak sağlamasıdır. Model denetimi [5], biçimsel doğrulamayı gerçekleştiren otomatik bir tekniktir. Model denetimi, sistemin sonlu durum modellerinin biçimsel özellikleri (gereksinimleri) ile tutarlılığını kontrol ederek biçimsel doğrulamayı gerçekleştirir [6]. Biçimsel doğrulamayı model denetim tekniği ile gerçekleştirmek için sistemin biçimsel modeline, sistem gereksinimlerinin biçimsel belirtimine ve bir model denetim aracına ihtiyaç vardır. Biçimsel doğrulamanın etkinliği sistem modeli ve belirtiminin doğruluğuna bağlıdır.

Zaman-kritik ve gerçek-zamanlı sistemlerin doğrulanmasında sistemin işlevsel doğruluğunun yanı sıra zamansal özelliklerinin doğrulanması da gerekmektedir. Dolayısıyla sistem davranışını belirten biçimsel modelde zamansal davranışların da belirtimi gerekmektedir. Zamanlanmış otomatlar [7] gerçek zamanlı sistemlerin davranışlarını zaman kavramı ile modellemek için kullanılan sonlu durum otomatlarıdır. Zamanlanmış otomatlar ile zaman kavramının modellenmesi için her bir otomata ait saat tanımlamak mümkümdür.

UPPAAL [8] zamanlanmış otomatları kullanarak biçimsel doğrulama sağlayan bir model denetim aracıdır. UPPAAL ile bir sistemin modellenmesi, haberleşen zamanlanmış otomatlar ile sağlanır. Her bir zamanlanmış otomat koşut zamanlı bir sistem bileşenini belirtir. Bu bileşenler gerçek zamanlı bir sistemin aynı işlemcide yürütülen görevleri olabileceği gibi farklı işlemci ve FPGA'lerde yürütülen sistem bileşenleri veya haberleşme protokolleri de olabilir. Zamanlanmış otomatlar arasındaki senkronizasyon kanallar kullanılarak, haberleşme ise paylaşılan değişkenler ile sağlanmaktadır. Buna ek olarak her bir otomat için özel değişkenler de tanımlanabilmektedir. Zamanlanmış otomatların haberleşmesi ve sistemde tanımlı saatlerin ilerleyişi senkron olarak gerçekleştirilir.

3 ALMAÇ PLANLAYICI BİLEŞENİ DOĞRULANMASI

3.1 Almaç Planlayıcı Bileşeni Tasarımı

Almaç planlayıcı bileşeni için işlevsel gereksinimler şu şekildedir: Almaç planlayıcı bileşeni;

1. Atanan yayınlar için tarama rejimini oluşturacaktır.
2. Tarama rejimi adımlarını almaçta uygulayacaktır.
3. Yayın listesinde değişiklik olduğu zaman mevcut tarama rejimini iptal edecek ve yeni bir tarama rejimi oluşturacaktır.

(a) Yayın listesi değişikliği tarama adımı uygulandığı sırada gerçekleşirse, yeni tarama rejimi uygulanan tarama adımı bittiğinde oluşturulacak ve uygulanmaya başlayacaktır.

4. Tarama rejiminde tarama adımları arasında en az anahtarlama süresi (400 μs) kadar boşluk olacaktır.
5. Sayısal almaçta bir tarama adımı uygulanırken yeni bir tarama adımı uygulanmayacaktır.

Tarama adımları, tarama rejimi dinamik olarak oluşturulduğundan ve tarama adımı periyotlarında gecikmeler olabileceğinden dolayı periyodik görevler ile tasarlanamamaktadır. Buna ek olarak işlemci kartlarında harici saat sayısı sınırlıdır. Bundan dolayı almaç planlayıcı bileşeni, tarama rejimi oluşturma ve uygulama işlevlerini bir görevin bağlamında dizisel olarak zaman tetikli bir tasarımla yürütmektedir. Almaç planlayıcı zamanı gelen bir tarama adımını uygulamak için önce sayısal almaca almaç ayarlama mesajını gönderir. Almacın ayarlandığına dair geribildirim mesajını aldığında ise kalış süresini göndererek tarama adımını başlatır. Tarama adımının zamanı geldiğinde tarama adımını uygulamadan önce adımın kalış süresi kadar saati kurar. Saat dolduğunda tarama rejiminde bulunan bir sonraki tarama adımının başlangıç zamanına kadar saati yeniden kurar. Eğer başlatılacak başka bir tarama adımı yoksa yeni bir tarama rejimini oluşturur ve yeni tarama rejmini yürütmeye başlar. Şekil 1 ardışık tarama adımlarının uygulanmasını göstermektedir.

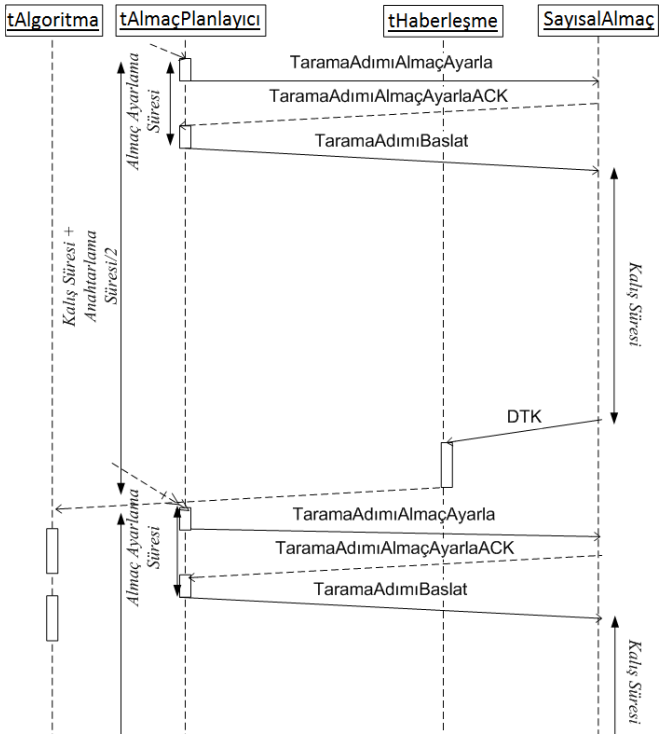
Tarama adımları arasındaki anahtarlama süresi boyunca sayısısal almacın ayarlanması ve sayısal almaçtan alınan DTK'ların yerel belleğe kopyalanması işlemleri gerçekleştirilir. Sayısal almaç, gönderilen parametreler ile gerekli ayarlamaları gerçekleştirir. Almaç ayarlama işlemi belirli bir süre almaktadır. Buna ek olararak DTKların yerel belleğe kopyalanması da DTK sayısına bağlı olarak değişen bir süreye sahiptir. Yapılan ölçümler doğrultusunda bu işlemler için 200 μs'lik süreler ayrılmıştır. Bu işlemlerden herhangi birisinin beklenen süreden uzun sürmesi ardışık iki tarama adımı uygulanırken ardıl adım için yukarıda belirtilen 5. işlevsel gereğin yerine getirilememesine yol açar.

3.2 Sistem Modeli Belirtimi

Almaç planlayıcı bileşeninin doğrulanması için sistem modeli almaç planlayıcı bileşeninin hem işlevsel hem de zamansal davranışını belirtmelidir. Bu kapsamda sistem modelinin sistemin işlevsel ve zamansal davranışlarını belirtebilmesi için sayısal almaç, almaç kontrol ve sistem kontrol bileşenlerinin modellerinden oluşması gerekmektedir. Denklem 1 sistemin bileşenlerini tanımlamaktadır.

$$S = C_{SayisalAlmac} || C_{AlmacKontrol} || C_{SistemKontrol} \quad (1)$$

Almaç kontrol bileşeni almaç planlayıcı, denetleyici, algoritma, sistem kontrol haberlesme (SistemKontHab) ve sayısal almaç haberleşme (SayAlmacHab) bileşenlerinin paralel bileşimidir. Denklem 2 almaç planlayıcının bileşenlerini gös-



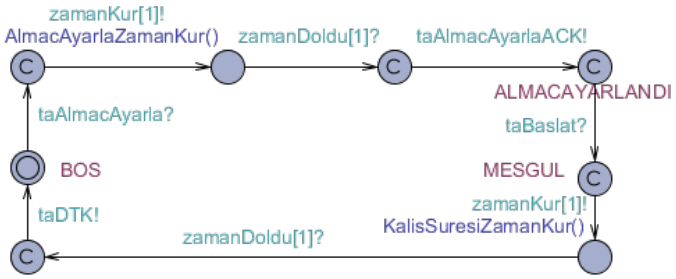
Şekil 1: Tarama Adımları Mesaj Akış Şeması

termektedir.

$$C_{AlmacKontrol} = C_{Denetleyici}||C_{AlmacPlanlayici} \\ ||C_{Algoritma}||C_{Zamanlayici}||C_{SayAlmacHab} \quad (2)$$

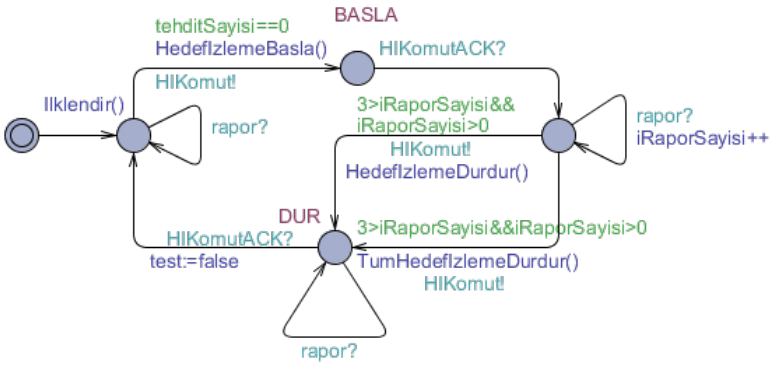
Sayısal Almaç Modeli Sayısal almaç modeli (Şekil 2) *taAlmacAyarla* ile tetiklendiğinde saati almaç ayarlama süresine kurar. Zaman dolduğunda *taAlmacAyarlaACK*'i tetikler. *taBasla* ile tetiklendiğinde ise saati almaç kontrol yazılımından gönderilen kalış süresine kurar. Kalış süresi zamanı dolduğunda almaç kontrol yazılımını *taDTK* ile tetikler.

Sistem Kontrol Modeli Sistem kontrol modeli (Şekil 3) sistem kontrol yazılımının soyutlanmış halidir. Sistem kontrol modeli *HIKomut* ile almaç kontrol bileşenini tetikleyerek en fazla iki yayın için hedef izleme başlatır. Bu iki yayın planlamaları arasında anahtarlama süresi olacak kadar tanımlanmıştır. *HIKomutACK* ile tetiklendikten sonra almaç kontrol bileşeninin ürettiği raporları almaya başlar. Daha sonra herhangi bir zamanda tüm yayınlar ya da bir yayın



Şekil 2: Sayısal Almaç Modeli

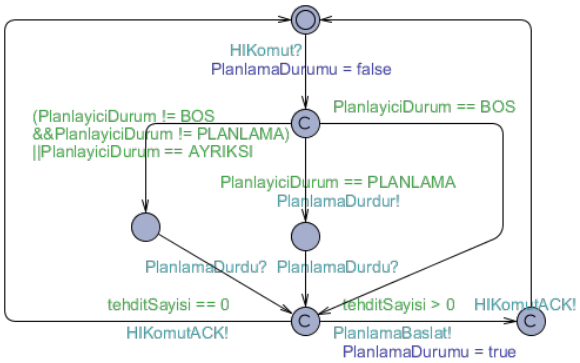
için *HIKomut* ile almaç kontrol bileşenini tetikleyerek almaç kontrol yazılımında mevcut yayınlar için tarama rejimi planlanır ve uygulanırken yayın listesi değişikliği gerçekleştirilir.



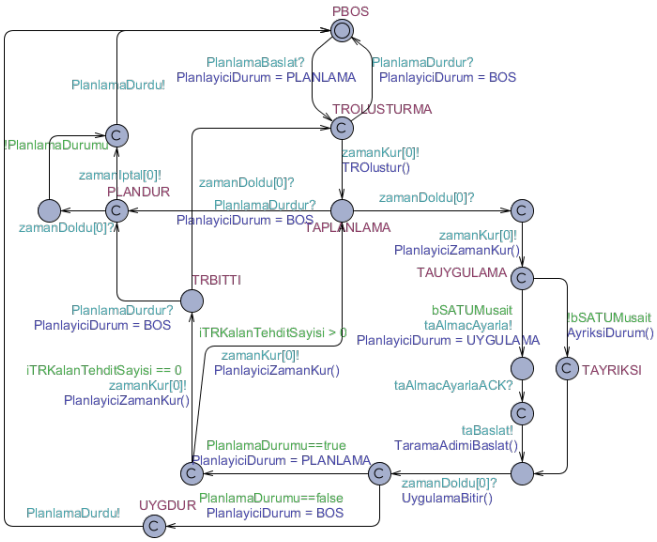
Şekil 3: Sistem Kontrol Bileşeni Modeli

Almaç Kontrol Denetleyici Modeli Almaç kontrol denetleyici modeli (Şekil 4) yayın listesinde herhangi bir değişiklik olduğu zaman sistem kontrol modelinden gönderilen *HIKomut* ile tetiklenerek almaç planlayıcı modelinin durumuna göre mevcut tarama rejiminin durdurulmasını sağlar. Almaç planlayıcının tarama rejimini durdurması ile gönderdiği *PlanlamaDurdu* ile tetiklendiğinde, eğer yayın listesinde halen izlenmesi gereken bir yayın varsa *PlanlamaBaslat* ile planlamayı tekrar başlatır ve sistem kontrol modelinde *HIKomutACK*'i tetikler.

Almaç Kontrol Almaç Planlayıcı Modeli Almaç planlayıcı modeli (Şekil 5) *PlanlamaBaslat* ile tetiklenerek tarama rejimi oluşturulması ve uygulanmasını başlatır. *TAPLANLAMA* durumuna geçerken mevcut yayınlar için tarama



Şekil 4: Denetleyici Bileşeni Modeli

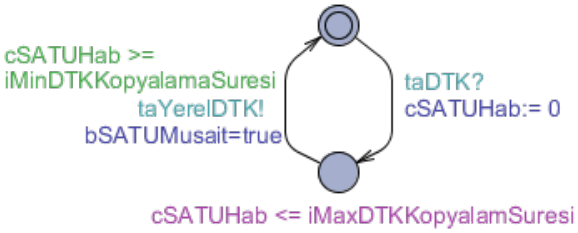


Şekil 5: Almaç Planlayıcı Bileşeni Modeli

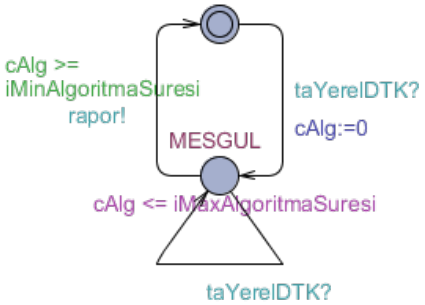
rejimini oluşturur ve zamanı ilk tarama adımının başlangıcına kurarak zamanın dolmasını bekler. Zaman dolduğunda planlanmış tarama adımının bitiş süresine zaman kurarak tarama adımını uygulamaya başlar. *TAUYGULAMA* durumuna geçtikten sonra sırası ile almacı ayarlar ve almaçta tarama adımının kalış süresi kadar uygulanacak tarama adımını başlatır. Almaç planlayıcı modeli tarama adımının planlanan bitiş zamanı geldiğinde sistem durumuna göre 3 farklı duruma geçebilir: *TRBITTI* durumuna tarama rejiminde uygulanacak başka bir adım kalmadığında, *TAPLANLAMA* durumuna tarama rejiminde uygulanacak başka adımlar olduğunda ve *UYGDUR* durumuna ise tarama adımı uygulanırken yayın listesi değişikliği gerçekleştiğinde geçiş yapılır. Almaç planlayıcı modelinde *PLANDUR* durumuna tarama rejimindeki bir adım uygulanmadığı esnada yayın

listesi değişikliği gerçekleştiği zaman geçilir ve 3. gereksinimi karşılamak amacı ile eğer yayın listesi boş değilse yeni bir tarama rejimi oluşturulur. 3.a gereksinimi karşılamak amacı ile tarama adımı bitişine dek herhangi başka bir durum geçişi yoktur. Almaç planlayıcı modelinde *AYRIKSI* durum 5. gereksinimi gerçeklemek amacı ile tanımlanmıştır. Eğer almaçta uygulanan bir tarama adımı mevcutken yeni bir tarama adımının başlatılması söz konusu ise model *AYRIKSI* durumuna geçerek uygulanmaya çalışılan adımı atlayarak tarama rejimindeki ardıl adıma geçer.

Almaç Kontrol Sayısal Almaç Haberleşme Bileşeni Modeli Sayısal almaç haberleşme modeli (Şekil 6) sayısal almaçtan DTK'lar alındığında DTK'ların yerel belleğe kopyalanması için gereken DTK sayısı ile değişen zamansal davranışı modeller. Model *taDTK* ile tetiklendikten sonra iMinDTKKopyalamaSuresi ve iMaxDTKKopyalamaSuresi arasında bir süre bekledikten sonra almacı meşgul durumundan çıkartarak *taYerelDTK*'ı tetikler.



Şekil 6: Almaç Haberleşme Bileşeni Modeli



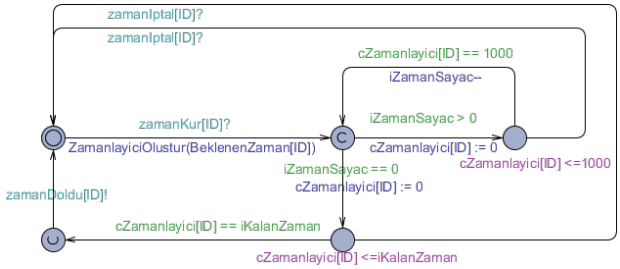
Şekil 7: Algoritma Bileşeni Modeli

Almaç Kontrol Algoritma Bileşeni Modeli Algoritma modeli (Şekil 7) sayısal almaçtan alınan DTK'lar ile çalıştırılan algoritmaları modeller. Model

taYerelDTK ile tetiklendikten sonra iMinAlgoritmaSuresi ve iMaxDTKAlgoritmaSuresi arasında bir süre bekledikten sonra *rapor* ile Sistem kontrol modelini tetikler. Algoritmaların yürütme süresi DTK sayısı ve DTK'ların içeriği ile değişebilmektedir. iMinAlgoritmaSuresi ve iMaxDTKAlgoritmaSuresi değişkenleri değişen bu süreyi belirtmektedir.

Zamanlayıcı Bileşeni Modeli Zamanlayıcı bileşeni modeli (Şekil 8) μs çözünürlükte bir zamanlayıcı modellemektedir. Zamanlayıcı modeli almaç planlayıcı ve sayısal almaç modelleri tarafından sürelerin kurulması için kullanılmaktadır. Biçimsel doğrulama için sistem konfigürasyonunda her iki modelin kullanımı için zamanlayıcı modelinin iki farklı örneği kullanılmaktadır.

UPPAAL aracında bir doğal sayı en fazla 32768 değerini alabildiğinden zaman planlayıcı bileşeninin istenilen çözünürlüğü elde etmek amacı ile farklı bir tasarıma ihtiyacı vardır. Bu amaçla Zamanlayıcı bileşeni $typedef\ struct\{int\ s; int\ ms; int\ us;\}zaman\_t;$ yapısını kullanır. Modelde zamanın s ve ms değerleri için soldaki döngü kullanılırken, μs değerler için sağdaki döngü kullanılır.



Şekil 8: Zamanlayıcı Bileşeni Modeli

3.3 Sistem Özellikleri Belirtimi

Model denetim ile biçimsel doğrulamada modellenen sistem davranışının sistem özelliklerini karşılayıp karşılamadığı denetlenir. Sistem davranışı sistem modelinde belirtilen eş zamanlı bileşenlerin durumlarının küresel bir durum uzayına çevrilmesi ile oluşturulur. Model denetimi ile küresel durum uzayındaki her bir durumun belirtilen sistem özelliğini sağlayıp sağlamadığı kontrol edilir. Bu kapsamda Bölüm 3.1'da tanımlı sistem gereksinimlerinin denetimi için bu gereksinimlerin belirtimi gerekmektedir. Bu belirtimler almaç planlayıcı bileşeninin doğrulanmasında iki kısma ayrılmıştır. Birinci kısım sistem davranışının herhangi bir kilitlenme olmadan işlediğinin denetlendiği 1. ve 2. sistem gereksinimlerinin belirtimini içermektedir. İkinci kısım ise almaç planlayıcı özelinde 3., 4. ve 5. sistem gereksinimlerinin belirtimini içermektedir. Sistem özellikleri belirtimi aşağıda gösterilmiştir.

1. (a) $A[]$ *not deadlock*
   (b) $E<>$ $pPlanlayici.TROLUSTURMA$
   (c) $E<>$ $pPlanlayici.TAPLANLAMA$
   (d) $E<>$ $pPlanlayici.TAUYGULAMA$
   (e) $E<>$ $pPlanlayici.TRBITTI$
2. (a) $A[]$ $PlanlayiciDurum!=AYRIKSI$
   (b) $pPlanlayici.UYGDUR->pSISKI.DUR$
   (c) $pPlanlayici.PLANDUR->pSISKI.DUR$

Model denetiminde 1.a özelliğinin sağlanması sistem davranışının herhangi bir kilitlenme olmadan çalışabildiğinin doğrulanmasını gösteririr. 1.b.'den 1.e.'ye kadar olan özellikler model denetimi esnasında almaç planlayıcı modelindeki *TROLUSTURMA*, *TAPLANLAMA*, *TAUYGULAMA* ve *TRBITTI* durumlarına geçişlerin olduğunu ve durumların ulaşılabilirliğinin doğrulandığını gösterir. 2.a özelliği ile *planlayici* bileşenin *PlanlayıcıDurum* değişkeninin hiçbir sistem durumda*AYRIKSI* değerini almadığını ve dolayısı ile *planlayici* bileşenin *TA-AYRIKSI* durumuna geçmediğini ifade eder. Bu özelliğin sağlanması anahtarlama süresinin yeterli uzunlukta olduğunun doğrulandığını gösterir. Bu özelliğin sağlanamaması sayısal almaçta mevcut bir tarama adımı uygulanırken yeni bir tarama adımının sayısal almaçta planlandığını ve 4. gereksinimin sistemin zamansal kısıtları ile uyumlu olmadığını gösterir. 2.b ve 2.c özellikleri *Planlayici* bileşeninin *UYGDUR* ve *PLANDUR* durumlarına eriştikten sonra *Sistem Kontrol* bileşeninin *DUR* durumuna geçeceğini ifade eder. Bu özelliklerin sağlanması ise 3. sistem gereksiniminin doğrulandığını gösterir.

4 DEĞERLENDİRME VE SONUÇ

Biçimsel doğrulama bir sistemin gereksinimlerine göre doğrulanmasına tasarım gibi erken aşamalarda olanak sağlayan matematiksel temele sahip bir doğrulama yöntemdir. Bu bildiride almaç kontrol yazılımının temel bileşeni olan almaç planlayıcı bileşeninin model denetim kullanılarak biçimsel doğrulanması sunulmuştur. Almaç planlayıcı bileşeni işlevsel ve zamansal gereksinimleri birlikte yerine getirmesi gereken gerçek-zamanlı bir yazılım bileşenidir. Almaç planlayıcı doğrulamasının yapılması için, almaç planlayıcı davranış modeli çevresel bileşen ve görevlerin davranışları ile birlikte modellenmiş, sistem gereksinimlerinin metematiksel özellikler ile belirtimi yapılmış ve model denetim uygulanmıştır. Zamansallığın modellenmesi için ise sistemin çözünürlük gereksinimleri göz önünde tutularak bir zamanlayıcı modeli oluşturulmuştur.

Model denetimi koşut zamanlı davranış modellerinin tüm durumlarının sistem özellikleri bağlamında denenmesine dayanır. Bundan dolayı model denetiminde çok sayıda koşut zamanlı model olması durum uzayı patlamasına yol açabilir. Almaç planlayıcı bileşeni doğrulanmasındaki zamansallığın modellenmesinde sistemin zamansal belirtimleri içinde mikrosaniye çözünürlüğün çok düşük olmasından dolayı bu sorun ile karşılaşılmakta ve doğrulama süreci uzun sürebilmektedir.

Sistem modelinde modellenmiş denetleyici, algoritma, sayısal almaç haberleşme ve almaç planlayıcı bileşenleri almaç kontrol yazılımında birer görev olarak tasarlanmaktadır. Bu görevler aynı işlemci kaynağını kullanmakta ve bir görevin işlemciyi kullanması diğer görevlerin yürütülmesinde gecikmeye yol açabilmektedir. Almaç planlayıcı bileşeni en yüksek öncelikli görev olarak tasarlandığından diğer görevlerin bu görevde herhangi bir gecikmeye yol açması mümkün değildir. Almaç planlayıcı bileşenin biçimsel doğrulanması açısından bu durumun yapılmış doğrulamaya olumsuz bir etkisi yoktur. Ancak algoritma süreleri ile ilgili gereksinimlerin de doğrulanmasının yapılması için görevlerin birbirleri ile olan etkileşimleri doğrulama sırasında dikkate alınmalıdır. Bu açıdan gelecekte işletim sistemi (vxWorks) ve geliştirme ortamı (UML-Rhapsody) davranışlarının belirtimlerine sistem modelinde yer verilmesi gerekmektedir.

Makalede sunulan sistem modeli ve özellikleri ile almaç planlayıcı bileşeninin biçimsel doğrulaması gerçeklenmiştir. Bu biçimsel doğrulama sürecinde görülen ancak testlerde görülemeyen 3. gereksinim ile ilgili bazı durumlar tasarıma eklenmiş ve ciddi bir fayda elde edilmiştir.

Kaynaklar

1. F. Neri, "Inroduction to Electronic Defense Systems", 2nd ed., MA, USA:Artech House, 2006.
2. Ç. Turan, G. Kahraman, C. Uzunoğlu, "RFKS Yazılım Mimarisi ve Arka Plan Yönetimi", TTD Konferansı,2012.
3. M. Dursun, Ö. Kızılay, "Zaman Tetikli Almaç Planlayıcı Yazılım Bileşeni Tasarımı", UYMS, 2014.
4. Jane W.S. Lui, Real-Time Systems,Prentice Hall,2000
5. E. M.Clarke, ,O. Jr. Grumberg, D. A. Peled, "Model Checking". s.l.: The MIT Press, 1999.
6. C. Baier, J. P. Katoen, "Principles of Model Checking", The MIT Press, 2008
7. R. Alur, D. L. Dill, " A Theory of Timed Automata", Theoritical Computer Science 126.2, 1994
8. J. Bengtsson, K.G. Larsen, F. Larsson, P. Pettersson, W. Yi. "UPPAAL - A Tool Suite for the Automatic Verification of Real-Time Systems", Hybrid Systems III, LNCS 1066, pages 232-243. Springer-Verlag, 1996
9. B. Bérard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen, "Systems and Software Verification: Model-Checking Techniques and Tools", Springer Publishing Company, Incorporated, 2010