

Karar Tablosu Destekli Olay Sıra Çizgeleri Temelli Sınama Durum Üretim Aracı

Fevzi Belli¹ Michael Linschulte² Tuğkan Tuğlular³

^{1,3}İzmir Yüksek Teknoloji Enstitüsü, Urla, İzmir, Türkiye

²Andagon GmbH, Köln, Almanya

¹fevzibelli@iyte.edu.tr ²m.linschulte@andagon.com

³tuğkantuglular@iyte.edu.tr

Özet. Model temelli sınama alanında otomatik sınama durum üretimi önemli konuların başında gelir. Otomatik sınama durum üretimi için formel bir modelleme yöntemi ile üretim için algoritmalar gereklidir. Bu bildirinin ilk yazarının geliştirdiği Olay Sıra Çizgeleri (*Ing. Event Sequence Graphs*) formel modelleme yöntemi ve sınama durum üretim algoritmaları, bu bildirinin diğer yazarları tarafından karar tabloları destekli şekilde genişletilmiştir. Ortaya konan modelleme yöntemi ve algoritmalar kodlanarak bir sınama durum üretim aracı 2008 yılından beri geliştirilmektedir. Anılan araç birçok çalışmada kullanılmış ve bu çalışmaların anlatıldığı yayınlarda bu araçtan söz edilmiştir. Bu bildiride karar tabloları destekli olay sıra çizgeleri temelli sınama durum aracının son durumu tanıtılmış ve anılan araç ile bir örnek çalışma sergilenmiştir.

Anahtar Kelimeler: olay sıra çizgeleri, karar tablosu, model-temelli sınama, otomatik sınama durum üretimi, sınama durum üretim aracı.

1 Giriş

Model temelli sınamada “sınama altındaki bir sisteme ait bir model kümesinden otomatik ve sistematik olarak sınama durumları üretilir” [1]. Model temelli sınama yaklaşımının önemli avantajı modellerin formel modeller olabilmesi ve bu sayede sınama durumlarını üretecek algoritmaların bu modelleri kullanacak şekilde geliştirilebilmesidir. Sınama durumları; sınama girdisi ve sınama çıktısı olarak tanımlanır ve “sınama girdilerine karşılık gelen beklenen çıktıları üretirse sistem sınamayı geçer; aksi takdirde sistem sınamadan kalır” [2]. Geçme/kalma kararını kahin (*Ing. oracle*) verir. Model temelli sınama yaklaşımında formel modellerin kullanımı kahin problemini çözmeye yarar.

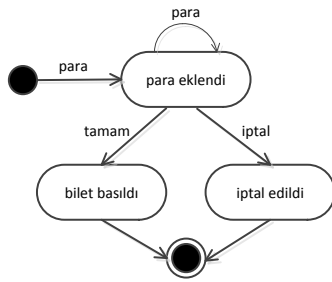
Olay Sıra Çizgeleri (*Ing. Event Sequence Graphs*) çizge temelli bir formel modelleme yöntemidir [3]. Olay Sıra Çizgelerinden sınama durumu üreten değişik algoritmalar geliştirilmiştir [4,5,6]. Bu bildirinin amacı Olay Sıra Çizgeleri ile modellemeyi mümkün kılan ve geliştirilen modeli kullanarak sınama durumu üretim algoritmalarını çalıştırıp sınama durumlarını listeleme aracı tanıtmaktır. Bunu yaparken çok bilinen

bir bilet otomatu örneği üzerinden ilerleyerek bu araç tanıtılacaktır. Model temelli sınaama yaklaşımının endüstride hak ettiği yeri bulamadığı gözlemi [1] bildiri yazarları tarafından da paylaşılmakta ve araç yokluğunun nedenlerden biri olduğu düşüncesi ile bu bildiri hazırlanmıştır.

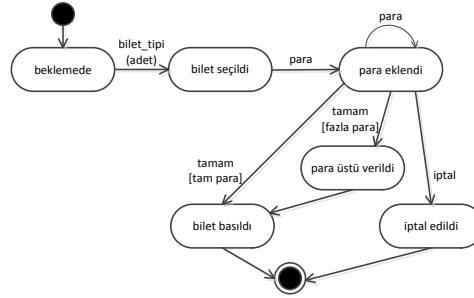
2. Bölümde çalışmada kullanılan örnek irdelenmiştir. 3. Bölümden itibaren izlenen yöntemin çalıştırıldığı araç tanıtılacaktır. Bu aracın sadece Olay Sıra Çizgesi ile çalışan kısmı 3. Bölümde, Karar Tablosu Destekli Olay Sıra Çizgesi ile çalışan kısmı 4. Bölümde açıklanmıştır. Sonuç bölümünde böyle bir aracın önemi vurgulanmıştır.

2 Çalışmada Kullanılan Örnek

Model temelli sınaamada kullanılan modellere; sonlu durum makinaları ile UML durum diyagramları örnek olarak verilebilir. Bu bildiride örnek olarak bir bilet otomatu kullanılmıştır. Bilet otomatu, hem tam para bekleyen ve tek tip bilet alınabilen basit bilet otomatu olarak hem de para üstü verebilen ve bilet türünün seçilebildiği gelişmiş bilet otomatu olarak kurgulanmıştır. Basit bilet otomatu durum diyagramı Şekil 1-a'da ve gelişmiş bilet otomatu durum diyagramı Şekil 1-b'de verilmiştir.



Şekil 1-a. Basit bilet otomatu durum diyagramı [7]



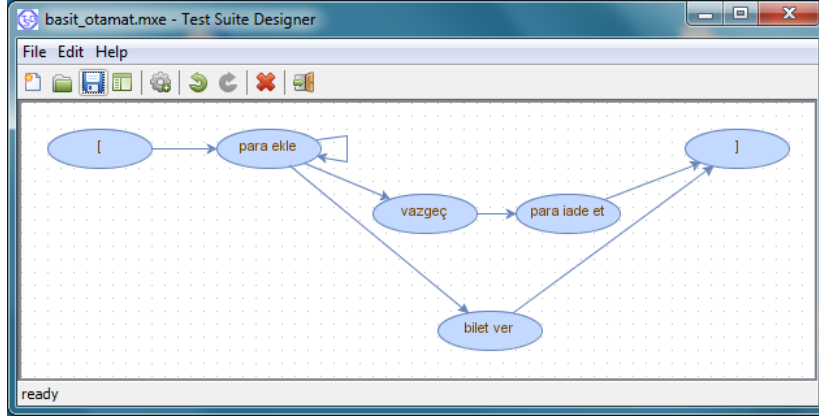
Şekil 1-b. Gelişmiş bilet otomatu durum diyagramı [8]

Basit bilet otomatu şu şekilde çalışır: Para eklendikçe toplam parayı gösterir. Beklenen tutara gelene kadar ya para eklenir ya da vazgeçilir. Beklenen tutara ulaşıldığında ya bilet basılır ya da vazgeçilir. Vazgeçilmiş ise eklenen para geri verilir. Gelişmiş bilet otomatu şu şekilde çalışır: Önce bilet türü ve kaç adet bilet alınacağı seçilir. Sonra para eklenmeye başlanır. Eklenen para tutara eşit veya tutardan fazla olana kadar ya para eklenmesi ya da vazgeçilmesi beklenir. Vazgeçilmiş ise eklenen para geri verilir. Eklenen para tutardan büyük ise önce üstü verilir ve sonra bilet basılır, eşit ise sadece bilet basılır.

3 Olay Sıra Çizgeleri ile Modelleme Aracı

Bir olay sıra çizgesi, basit olarak açıklamak gerekirse, durum temelli modelleme yaklaşımından farklı olarak olay temelli modelleme yaklaşımı benimser. Durum

makinaları ve diyagramlarında durumlar arası geçişler söz konusu iken olay sıra çizgesinde olaylar arası geçişler söz konusudur. Olay Sıra Çizgeleri formel tanımı [3] içinde detaylı olarak anlatılmıştır ve bu bildiriye tekrarlanmayacaktır. Basit bilet otomatına ilişkin olay sıra çizgesi sözü edilen araç ile çizilmiş ve ekran görüntüsü Şekil 2’de verilmiştir.



Şekil 2. Basit bilet otomatına ait olay sıra çizgesi

Şekil 2’deki olay sıra çizgesi, daha önce sözü edildiği üzere olayların akışı üzerine kurgulanmıştır ve geçerli olası olay akışlarını göstermektedir. [4,5,6] içinde verilen sına durum üretim algoritmaları ilgili araç içinde kodlanmış ve araçtaki çark simgesi tıklanıp çalıştırıldığında geçerli olay akış sıralarını (bakınız Şekil 3) içeren sına durumlarını üretmektedir. CES (*Complete Event Sequence*) başlığı altında yer alan her satır bir sına durumuna karşılık gelmekte ve her sına durumu içinde kaç olay olduğu sına durumunun başında belirtilmektedir.

```

Solution
*****
Full Resolution Approach
*****
Länge: 2
# Nodes: 6
# Edges: 7

# Nodes (CPP): 6
# Edges (CPP): 7

CES:
3: [, para ekle, para ekle, bilet ver, ],
3: [, para ekle, vazgeç, para iade et, ],
No. of CES: 2
No. of Events: 6

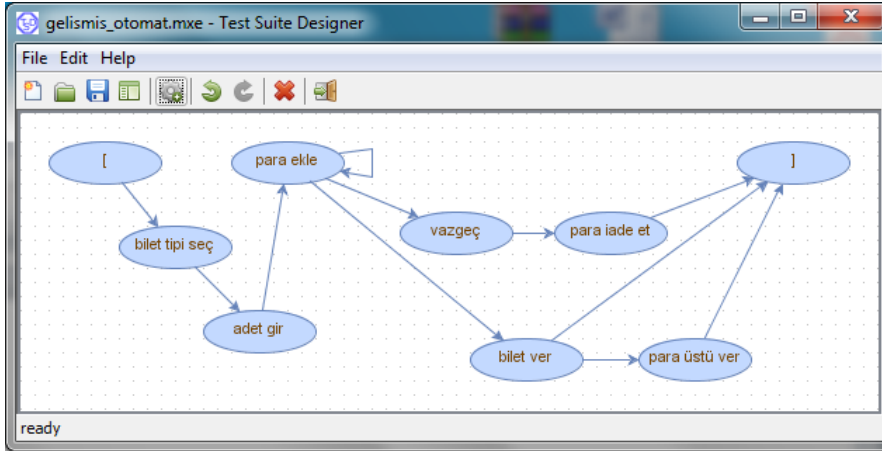
FCES: not generated

Time: 78ms
  
```

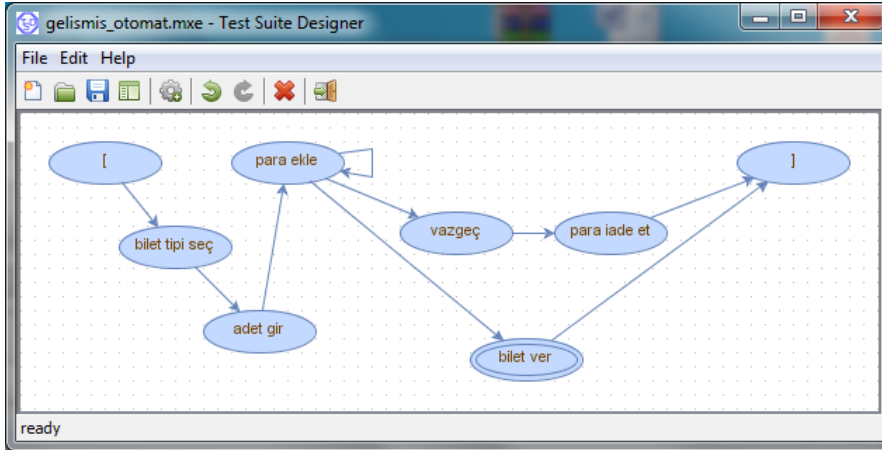
Şekil 3. Basit bilet otomatına ait geçerli olay akış sıralarını içeren sına durumları

4 Karar Tablosu Destekli Olay Sıra Çizgeleri ile Gelişmiş Modelleme Aracı

Olay sıra çizgeleri karar tablosu desteği ile genişletilmiştir. Karar Tablosu Destekli Olay Sıra Çizgelerinin formel tanımı [9] içinde detaylı olarak anlatılmıştır ve bu bildiride tekrarlanmayacaktır. Gelişmiş bilet otomatına ilişkin olay sıra çizgesi ekran görüntüsü Şekil 4’te ve gelişmiş bilet otomatına ilişkin karar tablosu destekli olay sıra çizgesi ekran görüntüsü Şekil 5’te verilmiştir.



Şekil 4. Gelişmiş bilet otomatına ait olay sıra çizgesi

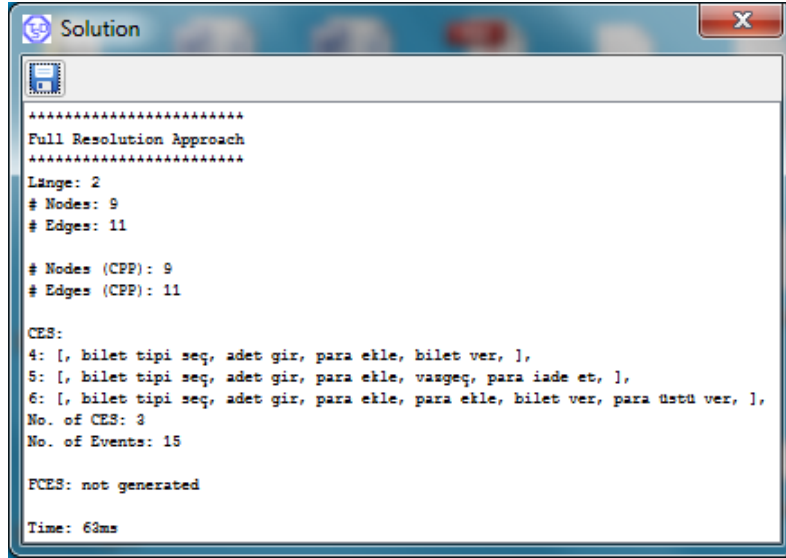


Şekil 5. Gelişmiş bilet otomatına ait karar tablosu destekli olay sıra çizgesi

Şekil 5, Şekil 4 ile karşılaştırıldığında “*bilet ver*” olayından sonraki akışa ait detaylar çift halkalı düğüm içine gömülmüş ve bu düğüm ile ilgili işleyiş bir karar tablosu olarak ifade edilmiştir. Bu karar tablosunda para üstü var koşulu ile “*para üstü ver*” olayı tetiklenirken ve para üstü yok ise “*birşey yapma*” işletilmektedir. Karar tablosu

desteđi ile olay akıř çizgesindeki karmařıklık azaltılmakta ve deđiřik veri tiplerindeki deđiřikler üzerinde tanımlanan kořullar ile olay akıř sıra çizgeleri geniřletilmektedir.

Araçtaki çark simgesi tıklanıp çalıřtırıldıđında geçerli olay akıř sıralarını ieren sına ma durumlarını retilmiř ve Őekil 6’da verilmiřtir.



```
Solution
*****
Full Resolution Approach
*****
Länge: 2
# Nodes: 9
# Edges: 11

# Nodes (CPP): 9
# Edges (CPP): 11

CES:
4: [, bilet tipi se, adet gir, para ekle, bilet ver, ],
5: [, bilet tipi se, adet gir, para ekle, vasge, para iade et, ],
6: [, bilet tipi se, adet gir, para ekle, para ekle, bilet ver, para st ver, ],
No. of CES: 3
No. of Events: 15

FCES: not generated

Time: 69ms
```

Őekil 6. Geliřmiř bilet otomatına ait geçerli olay akıř sıralarını ieren sına ma durumları

Söz konusu araç aynı zamanda geçersiz olay akıř sıralarını ieren sına ma durumlarını da retmektedir. Geçersiz olay akıř sırası ile “modellenen yazılım bu Őekilde çalıřmıyor olmalı” gösterilmektedir ve bu sına ma durumları modellenen yazılıma iin retilmiř negatif sına ma durumlarıdır. Örneđin, geliřmiř bilet otomatı iin [, bilet tipi se, para ekle,] iki uzunluđunda bir negatif sına ma durumudur. Negatif sına ma durumları çizge üzerinde düđmler arasında olmayan kenarlar ile retilmektedir. Detaylı algoritma ve örnekler [6, 10]’da verilmiřtir.

Söz konusu araç; basit bilet otomatı iin 15 negatif sına ma durumu ve geliřmiř bilet otomatı iin 48 negatif sına ma durumu retmiřtir. Bu araç, 2.70 GHz Intel i7 iřlemciye ve 8GB RAM belleđe sahip Windows 7 bilgisayar üzerinde çalıřtırılmıř ve Őekillerde görlen srelerde sına ma durumları retilmiřtir.

5 Sonu

Bu bildiride, model temelli sına ma yaklařımlarından biri olan olay sıra çizgeleri ile karar tablosu destekli olay sıra çizgeleri iin geliřtirilmif araç tanıtılmıř ve iřleyiřiok bilinen bir bilet otomatı örneđi üzerinde gösterilmifdir. Bu bildiride tanıtılan

aracın önceki sürümleri ile gerçekleştirilen büyük örnekler literatürde [9, 10, 11, 12] bulunabilir. Aracın güncel sürümü <http://download.ivknet.de/> adresinden indirilebilir.

Kaynakça

- [1] I. Schieferdecker, "Model-Based Testing", *IEEE Software*, vol. 29, no .1, pp. 14-18, 2012.
- [2] F. Belli ve M. Beyazıt, "Olay Tabanlı Sınama İçin Mutant Seçimi," 2013 Ulusal Yazılım Mühendisliği Sempozyumu, İzmir, 2013.
- [3] F. Belli, "Finite state testing and analysis of graphical user interfaces," in *Proceedings of 12th International Symposium on Software Reliability Engineering*, 2001, pp. 34–43.
- [4] F. Belli, C. J. Budnik and L. White, "Event based modelling, analysis and testing of user interactions: Approach and case study," *Software Testing, Verification and Reliability*, vol. 16, no. 1, 2006, pp. 3–32.
- [5] F. Belli and C. J. Budnik, "Test minimization for human-computer interaction," *Applied Intelligence*, vol. 26, no. 2, 2007, pp. 161–174.
- [6] F. Belli and M. Linschulte, "On negative tests of web applications," *Annals of Mathematics, Computing & Teleinformatics*, vol. 1, no. 5, 2008, pp. 44–56.
- [7] M. Hübner, I. Philippow, and M. Riebisch, "Statistical usage testing based on UML," *Proceedings of the 7th World Multiconferences on Systemics, Cybernetics and Informatics*, Orlando, FL, USA, Jul 27, 2003.
- [8] R. K. Swain, P. K. Behera, and D. P. Mohapatra, "Minimal TestCase Generation for Object-Oriented Software with State Charts," *arXiv preprint*, arXiv:1208.2265, 2012.
- [9] T. Tuglular, F. Belli, and M. Linschulte, "Input Contract Testing of Graphical User Interfaces," *International Journal of Software Engineering and Knowledge Engineering*, vol. 26, no. 02, 2016, pp. 183-215.
- [10] F. Belli, "A holistic view for finite-state modeling and testing of user interactions." *Proceedings of the 1st South-East European Workshop on Formal Methods*. 2003.
- [11] F. Belli, A.T. Endo, M. Linschulte, and A. Simao, "A Holistic Approach to Model-Based Testing of Web Service Compositions," *Software: Practice and Experience*, vol.44, no.2, pp. 201–23, 2014.
- [12] F. Belli, N. Güler, M. Linschulte, "Does 'Depth' Really Matter? On the Role of Model Refinement for Testing and Reliability", *Proc. 35th Annual IEEE Computer Software and Applications Conference (COMPSAC)*, pp.630-639, 2011.