AJIT-e: Online Academic Journal of Information Technology
2015 Spring/Bahar – Cilt/Vol: 6 - Sayı/Num: 19
DOI: 10.5824/1309-1581.2015.2.002.x
http://www.ajit-e.org/?menu=pages&p=details_of_article&id=157

Etkileşimli Hikaye Anlatma Sistemlerinde Oyuncu Profilleri: Örnek C++ Örüntü Tanıma Profil Çıkartıcı

Barbaros BOSTAN, *Assoc. Prof. Dr., Yeditepe University, Faculty of Commerce, Department of Information Systems and Technologies, İstanbul, Turkey, bbostan@yeditepe.edu.tr*

Gökhan ŞAHİN, *Asst. Prof. Dr., Yeditepe University, Faculty of Commerce, Department of Information Systems and Technologies, İstanbul, Turkey, sahin@yeditepe.edu.tr*

ÖZET *Oyuncu profili çıkartma bilgisayar oyunları ile ilgili popüler bir araştırma sahasıdır ama Etkileşimli Hikaye Anlatma Sistemleri için çok önemlidir. Bu çalışmada amacımız: (1) oyuncu eylemlerini sürekli gözlemleyerek gerçek zamanlı profiller çıkartmak, (2) profil çıkartabilmek için örüntü (oyuncu eylemleri dizisi) tiplerini tanımlamak, (3) örüntüleri hızlı bir biçimde eşleştirmek, ve (4) eşleşen örüntüler ile oyuncu profilleri arasındaki ilişkiyi örüntü-motivasyon eşleştirmeleri ile ortaya koymaktır. Bu bağlamda, bir Etkileşimli Hikaye Anlatma projesinde kullanılmak üzere karmaşık örüntü örnekleri tanımladık ve C++ dilinde bir profil çıkartıcı geliştirdik.*

Anahtar Kelimeler: *oyuncu profilleri, oyuncu türleri, örüntü tanıma*

Player Profiling for Interactive Storytelling Systems: A C++ Pattern Matching Profiler

ABSTRACT *Player profiling is a popular research area in computer gaming but it is especially important for interactive storytelling (IS) systems. In this article our aim is: (1) to real-time profile players by constantly monitoring player actions, (2) to define pattern (sequence of player actions) types for profiling, (3) to match patterns rapidly, and (4) to define the relationship between matched patterns and player profiles by pattern-motivation pairings. We defined complex pattern samples for profiling and developed a C++ profiler for an Interactive Storytelling project that matches user actions to these patterns.*

Keywords ***:*** *player profiles, player types, pattern matching*

DOI: 10.5824/1309-1581.2015.2.002.x

1. Giriş

Kullanıcı profili çıkarma her kullanıcıya kendi kişisel ilgisine göre kişiselleştirilmiş içerik sunmayı amaçlar. İçerik kişiselleştirmesi ve bire bir pazarlamanın kullanımı e-ticaret gibi yükselen alanlarda başat rol almaktadır. Kullanıcı profilleri, duygusal ve zihinsel durumları, amaçları, motivasyonları, karar oluşturma süreçlerini ve kullanıcı yeteneklerini temsil eder. Kullanıcı hakkında hiç bir bilginin olmadığı durumlarda sistem bütün kullanıcılar için tam olarak aynı şekilde davranır. Kullanıcı profilleri seyahat tercihlerini belirleme (Waszkiewicz et al, 1999), kişiye uyarlanabilir elektronik öğrenim sistemleri sistemleri ve uyarlanabilir öğrenim ve öğrenenim programları tasarımı (Nebel et al, 2003; Froschl, 2005), telefon konuşmaları analizi (Fawcett ve Provost, 1996), akıllı temsilci tasarımı (Godoy ve Amandi, 2003), ağ gezinme verisinden kullanıcı İnternet tercihlerini belirleme (Esposito et al, 2004; Krulwich, 1997), bilgisayar temelli edimlerde kullanıcıya destek sağlayan tavsiye sistemleri tasarlama (Schiaffino ve Amandi, 2000; Semeraro et al, 2007), çalışan seçimi veya değerlendirmesi desteği gibi çeşitli bağlamlarda kullanılmaktadır. Her ne kadar profil oluşturmanın amacı ve profil oluşturma teknikleri farklılık gösterse de ana fikir bir çevredeki kullanıcının seçim/eylemlerini gözlemleme ve bu seçimleri varsayımsal profiller ile karşılaştırıp/işleyerek kullanıcı tercihlerini anlamaktır.

1.1. Etkileşimli Hikaye Anlatma Sistemleri

Kullanıcı profili oluşturma bilgisayar oyunlarında önemli bir araştırma konusu olsa da asıl olarak etkileşimli hikaye anlatma sistemlerinde öncelik taşır. Etkileşimli hikaye anlatma sistemleri oyuncunun eylem ve seçimlerine göre oluşturulmuş kişisel içerik sunma amacını taşır; bunu gerçekleştirmek için sisteme oyuncunun tercihlerine göre oyunu şekillendirme beceresi sağlayan kullanıcı modelleri kullanılır. Etkileşimli hikaye anlatım sistemleri genellikle bir "Drama Yöneticisi" ve bir "Kullanıcı Modeli" içerir. Drama Yöneticisi hikaye senaryolarını araştırma ve uygulama (Bates, 1992; Kelso, et al, 1993), beyan edilmiş bilgiye dayanarak hikayenin belirleyici olaylarını sıralama (Mateas, 1997; Mateas, 2000; Mateas ve Stern, 2001), sanal aktörlerle olan ilişkiler ve kullanıcı seçimlerine dayanarak hikaye oluşturma (Cavazza et al, 2002), hikayenin olaylarını özelleştirme ve detaylandırma (Thue et al, 2007), kullanıcı modeline dayanarak hikayenin belirleyici olaylarını seçme (El-Nasr, 2007), uyumlu bir hikaye örüntüsü içinde çelişkileri etkileşim noktaları olarak bağlama (Barber ve Kudenko, 2008), soyut hikaye olay noktalarının kısmi sıralaması (Magerko, 2005), karakterlerin eylemlerini en değerliden en az değerliye doğru sıralama (Szilas, 2003) ile yükümlüdür. Kullanıcı etkileşimlerini ve/veya seçimlerini takip eden modül veya bileşen ise Kullanıcı Modeli veya Kullanıcı Profil Çıkartıcı olarak adlandırılır. Etkileşimli Hikaye Anlatma (EHA) sistemlerinin çoğu yukarıda bahsi geçen hikaye modifikasyonlarına, sistemin oyuncu profillerini nasıl tanımladığına ve oyun süresi içinde oyuncunun eylemlerinin nasıl takip edildiğine odaklanır ve aynı zamanda Drama Yöneticisi üzerine yoğunlaşır.

1.1.1. Oyuncu Tipleri

Oyuncu profili oluşturma ve kişiye özel profil çıkartma için genel profillerin oluşturulması teknikleri büyük çeşitlilik sergilerler. Bu bağlamda araştırmacılar genellikle daha önceki çalışmalarda tanımlanmış oyuncu stilleri ve oyuncu tiplerine bel bağlar. Bartle (2004) ünlü dört oyuncu stilini şöyle tanımlar: sosyalleşenler, başarı peşinde koşanlar, katiller ve kaşifler; Salen ve Zimmerman (2003) ise beş oyuncu tipi tanımlamıştır: standart oyuncu, adanmış oyuncu, sportmenlik dışı oyuncu, hilebaz ve oyun bozan; Mulligan ve Patrovsky (2003) ise oyuncular arası etkileşime dayanan bir gruplandırma getirir: genel oyuncular, barbarlar, kabile üyeleri ve vatandaşlar; Pohjola (2004) canlı rol yapma oyunlarından yola çıkarak dört kategori tanımlar: kendini kaptıranlar, dramacılar, oyuncular ve simulasyoncular; Dena (2008) ise oyuncular için üç katman tanımlar: bulmaca oyuncuları, hikaye oyuncuları ve gerçek dünya oyuncuları. Oyuncu profili oluşturma konusunda Bartle'ın (2004) meşhur oyun sitilleri bile ampirik olarak dört oyuncu tipinin (sosyalleşenler, başarı peşinde koşanlar, katiller ve kaşifler) birbirinden bağımsız olup olmadığı konusunda test edilmemiştir. Eğer profil tanımları çakışıyorsa oyuncu davranışlarını ayırt etmek çok zor hale gelir. Örneğin, bir bilgisayar oyununda oyuncu geniş bir alanı tarıyorsa kendisinin kaşif olduğu varsayılır oysa ki temel motivasyonu tüm görevleri bitirmek, tüm oyun içi ekipmanları ele geçirmek, istisnasız bütün düşmanları öldürmek veya hepsi olabilir.

1.1.2. Etkileşimli Hikaye Anlatma Sistemlerinde Kullanıcı Profili Oluşturma

Yukarıda bahsi geçen oyuncu tipleri genel bilgisayar oyuncusu kategorileridir, oyuncu profili oluşturmak amacıyla geliştirilmemişlerdir ama araştırmacılar etkileşimli hikaye anlatım sistemleri için kendi profil oluşturma tekniklerinin yanı sıra bunları da kullanmaktadır. Örneğin *PASSAGE*, Peinado ve Gervas'a (2007) ait olan oyuncu tiplerini kullanır: savaşçılar, güç oyuncuları, taktisyenler, hikaye anlatıcaları ve metod aktörleri (Thue et al, 2007) ; *IDA* içsel olasılık kurallarına dayanan bir model kullanır (Magerko, 2005); ve *Mirage* ise bir karakter stereotipini tanımlayan kişilik özelliklerini kullanır (El-Nasr, 2007). IDA'nın olasılık kuralı tabanlı modeli oyuncunun hikaye olaylarının ön koşullarını ihlal etmesi olasılığı üzerine odaklanır; model kullanıcı hakkında bir profil oluşturmak için açıkça kullanmaz. *Mirage* gönülsüz kullanıcı, vahşi, benmerkezcil, korkak, gerçeği arayan vb. gibi kişilik özelliklerini kullanır ama El-Nasr (2007) bu modelin aşırı basitleştirici olduğunu söyler. Bu bağlamda bu makale bir oyuncunun profilini çıkarmak için güç arzusu, ilişki kurma ihtiyacı, saldırganlık vb. gibi oyuncu motivasyonları üzerinde yoğunlaşır. Kullanıcının niyetini anlamak için motivasyonların kullanılmasının sebebi oyun oynamanın amaca yönelik bir davranış olması ve amaçların elde edilmesi veya ulaşılmaz amaçlardan vazgeçilmesinin oyuncunun motivasyonları tarafından belirlenmesidir (Heckhausen ve Heckhausen, 2005). Motivasyon ayrıca ödül örnekleri ("yemleme") ve ödül ipuçları (Berridge, 2001) ile farklı tipteki oyunculara kişiselleştirilmiş oyun deneyimi sağlamak için ortaya çıkartılabilir.

DOI: 10.5824/1309-1581.2015.2.002.x

Oyuncu deneyiminin kişiselleştirilmesi sanal dünyadaki oyuncu eylemlerinin sürekli gözlemlenmesini gerektirir ve bu eylemlerden sağlanan bilginin oyuncu profiline işlenmesi için çeşitli teknikler kullanılır. *PASSAGE* bir oyuncu modelini beş oyuncu tipine göre seçilmiş ağırlıklar üzerinden öğrenir; eylem planları oyun başlamasından önce tanımlanır ve ağırlık deltaları ile genişletilir (Thue et al, 2007). Oyun dünyasından her hangi bir eylem geldiğinde profile ağırlık değeri kullanılarak bir ekleme veya çıkarma yapılır. *Mirage* her kişilik özelliği vektörü için hikaye bağlamı, kullanıcı eylemi ve önceki vektör değerlerini kullanarak yeni bir değer hesaplayan kural temelli bir sistem kullanır. Sistem "eğer kullanıcı silahsız karakterlere saldırmak üzere ilerlerse oyuncunun şiddet ölçeğini bir adım yükselt" gibi çok basit kurallar kullanır (El-Nasr, 2007). Bu teknikler, her bir eylemin oyuncunun profiline katkısı olduğunu varsaymalarından dolayı profil oluşturmada yeterli olmayabilirler. Bilgisayar oyunlarında oyuncu eylemleri karmaşık örüntüler sergileyebildiği için her bir eylem bir profile katkıda bulunmak zorunda değildir. Örneğin bir oyuncu altınlarını elde etmek için canavarları öldürebilir ve sonrasında bu altınları bir prensesin kalbini kazanmak amacı ile pahalı bir kolye almak için kullanabilir. Eğer profil oluşturma için tek tek eylemler kullanılıyorsa sistem oyuncunun savaşmayı ve öldürmeyi sevdiği sonucuna varabilir oysa ki oyuncunun gerçek motivasyonu kolyeyi aldıktan sonra anlaşılacaktır. Buna göre bir hedefe ulaşmadaki motivasyon oyuncunun an itibari ile geçerli eylemi tarafından belirlenebileceği gibi oyuncunun önceki veya sonrasında ki eylemleri tarafından da belirlenebilir.

1.2. Diğer Bağlamlarda Oyuncu Profili Oluşturma

Etkileşimli hikaye anlatımı dışında kalan bağlamlarda kullanılan farklı profil oluşturma teknikleri de bulunmaktadır. Sahte cep telefonu aramalarını belirlemek amacı ile çok sayıda cep telefonu araması belirli kurallar ve şablon kümeleri ile veri madenciliği yaklaşımı ile incelenmekte ve bulunan örüntüler müşteri profillerini çıkarmak için kurallar ve şablonlar kümesi ile işlenmektedir (Fawcett ve Provost, 1996). Fakat oyuncu profilini oluşturmak için kullanılan veri miktarı veri madenciliği tekniklerinin bu veri üzerinde kullanımına izin verecek ölçüde çok değildir. Kullanıcı tercihlerini ve beğenilerini belirlemek için kullanılan tekniklerden biri de işbirliğine dayalı filtrelemedir (Lashkari et al, 1994). Bu teknik bir kullanıcıdan alınan veriyi diğer kullanıcılardan alınan veri ile ilgiler arasında kesişme bulmak amacı ile karşılaştırır ve kesişme bulması durumunda kullanıcıya diğer kullanıcı verilerine dayanarak yeni ögeler tavsiye edilir. Bu açıdan bilgisayar oyunlarında kullanıcı profili oluşturma kullanıcılar arası karşılaştırma ile ilgilen(e)mez ama her kullanıcı üzerinde biricik veri kaynağı olarak odaklanır. Kullanıcı modelleme için kullanılan bir başka model ise Durum Tabanlı Çıkarsama'dır (DTÇ). DTÇ'de sistem eldeki benzer durumların tarihini hatırlar ve bu bilgiyi yeni problemleri çözmek için kullanır. Böylece profiller problem uzayında yakında bulunan örnekler ile tanımlanır. Örneğin Waszkiewicz'in (1999) Kişisel

Seyahat Asistanı durum olarak kullanıcı eylemlerini saklar ve DTÇ tekniği ile kullanıcı davranışını profillerle eşleştirir.

Kullanıcı profili oluşturma için kullanılan daha karmaşık çözümler Bayes çıkarsama, Bayes ağları gibi teknikler ilgili elemanları olasılık dağılımlarına göre sınıflandırıp bu olasılıklar hakkında çıkarsama yaparak optimal kararlar alan teknikler de içerir. DTÇ ve Bayes ağlarını birleştirip kullanıcı profilleri oluşturan melez teknikler de vardır (Schiaffino ve Amandi, 2000). Uyarlamalı öğrenme ve öğretme programları kullanıcı profilini oluşturan veriyi atıf değer ikilileri halinde saklar ve bu ikililer kullanıcının o anki bilgi durumunu ve kişisel karakteristiklerini, tercihlerini vb. temsil eder (Nebel et al, 2003). Bilgisayar destekli akıllı öğrenim sistemlerinde öğrenim konusunun alanı çok iyi analiz edilip yapılandırılır ama bilgisayar oyunlarında çalışma alanı çok daha az tanımlı ve takip edilecek kullanıcı davranışlarının erim ve tipi çok daha az öngörülebilirdir (Beal et al, 2002). İnternet'te kullanıcı profili oluşturmak için sık kullanılan iki teknik vardır : bilgi temelli profil oluşturma ve davranış bazlı profil oluşturma. Bilgi temelli modeller veri toplamak için genelde anket ve mülakatları kullanır ve sonrasında istatistiksel modelleri kullanıcı verilerini en yakın modelle eşleştirir. Davranış temelli yaklaşımlar kullanıcı davranışını model olarak kullanır, genelde işe yarar örüntüleri keşfetmek için yapay zeka öğrenim tekniklerini devreye sokar, aynı zamanda DTÇ sistemleri de ağ kullanıcılarının profilini oluşturmak için kullanılır (Bradley et al, 2000).

2. Profil Oluşturma İçin Örüntü Eşleştirme

Oyuncuların profilini çıkarmak gerçek zamanlı bir görevdir bundan dolayı yukarıda bahsedilen, analiz edilecek işlem başlamadan önce elde edilen veriye dayanan sistemler uygun değildir. Aynı zamanda bilgisayar oyunlarındaki oyuncu profilleri kümesi yapay zeka öğrenim ve DTÇ tekniklerini kullanacak kadar karmaşık ve nicelik olarak yeterli miktarda değildir. Bu çalışmada oyun içi profil oluşturma sırasında eylem-katkı eşleşmesi problemini çözmek için oyuncunun niyet ve isteklerinin tasvir eden örüntüleri kullanıyoruz, bu bağlamda örüntü belirli bir motivasyonu sağlayan oyuncu eylemleri dizisi olarak tanımlanmıştır. Böylece kullanıcı profilleme için oyuncuların bütün olası örüntülerin önceden tanımlanmış olması gerekliliği ortadan kalkmaktadır. Eğer kullanıcının olası eylemlerinin ana kümesi ufak ise oyuncu davranışlarının uyacağı profillerin sayısı sınırlı olacaktır, böylece bütün olası örüntüleri tanımlamak bir sorun teşkil etmeyecektir. Bilginin edinimi anlık olacak ve oyuncunun oyun içi eylemlerini gözlemlemeye dayanacaktır.

2.1. Olay Tanımları

Halihazırda bulunan etkileşimli hikaye anlatımı sistemlerinde oyun dünyası sürekli gözlemlenir ve oyuncu eylemleri Profil Çıkartıcı tarafından oyuncu profilleri oluşturmak üzere alınır. Önerdiğimiz profil gerçek bir oyunun içinde kullanılmadığından dolayı bir

DOI: 10.5824/1309-1581.2015.2.002.x

Oyun Dünyası simülasyonu için bir Olaylar sınıfı kullanacağız. Bu sınıfın işlevi kullanıcı eylemlerini sürekli olarak profil çıkartıcıya aktarmaktır. Bu sınıfın *ReadEvents* adı altında olayların listesini bir metin dosyasından okuyup bu bilgiyi bir boyutlu vektörlere tutmakta olan tek bir metodu vardır. Bu olaylar sonrasında oyun dünyasının simülasyonu için tek tek profil çıkartıcıya yollanacaktır. Örüntüleri bir olay örgüsü olarak tanımlıyoruz. Oyun dünyasında ki her olay bir fiil (*verbID*), bir nesne (*objectID*) ve bir hedeften (*targetID*)'den oluşmaktadır, örneğin bir "kılıçla" bir "köylüye" "saldırmak", bir fiili (saldırmak), bir nesneyi (kılıç) ve bir hedefi (köylü) içeren bir olay oluşturur. Olaylara ayrıca bir zaman değeri iliştirilmiştir, böylece örneğin 3-7-4-11 örüntüsü *verbID*'si (fiil) 3, *objectID*'si (nesne) 7 ve *targetID*'si (hedef) 4 olan olay 11. saniyede vuku bulan bir olayı temsil eder. Bu bilgi *Event* sınıfında aşağıda belirtilen dört tane vektörde tutulur:

```
vector<int> verbIDEF;

vector<int> objectIDEF;

vector<int> targetIDEF;

vector<int> eventTimeEF;
```

2.2. Örüntü Tanımları

Her olay örüntüsünün biricik bir kimlik numarası vardır ve her örüntü her biri bir *eventID* ile tanımlı çeşitli sayıda olaylardan oluşur. Örneğin eğer bir saldırının *verbID*'si 3, nesnesi "kılıç' (7 sayılı kimlik numarası ile tanımlı) hedefi de köylü (kimlik numarası 4) ise bu olayın biricik kimlik numarası 374 olur. Bir diziyi oluşturan olayların doğasını belirmek için işaretçiler kullanılmıştır. Örneğin olaylar süreklilik sergiliyor olabilir; bir eylemin her zaman bir başka eylemi takip ettiği kati bir diziye sahip olabilir veya örüntüler çok daha akışkan olay dizileri ile tanımlanabilir; sabit olan iki olay arasına çeşitli başka olaylar girebilir. Bütün bu varyasyonlar bir işaretleyici ile belirtilir. Örneğin 1 değerini taşıyan bir işaretçi mutlak bir olaylar dizesini tanımlarken -1 değeri değilleme işareti tanımlar (değeri -1 olan olay olay dizisi içinde yer almamalıdır).

PATTERNID	FLAG	EVENTID	FLAG	EVENTID	...	END FLAG

Şekil 1: Genel Örüntü Yapısı

Örnek olarak 2.1.374.1.404.-2 dizisi ile tanımlanan olay örüntüsünü ele alalım: Burada 2 (*PatternID*) örüntünün kimlik numarasını, 1 değeri işaretçi değerini (*Flag1*), 3 ilk fiili (*VerbID1*), 7 ilk nesneyi (*ObjectID1*), 4 hedefi (*TargetID1*), 1 ikinci bir işaretçi değerini , 4 ikinci fiili ,0 ikinci nesneyi, 4 ikinci hedefi ve -2 ise örüntü tanımı sonunu ifade eder. Bu 2 değerli kimlik numarasına sahip örüntünün ilk olayının 1 işaretçisi ile işaretlenmiş 374 olayı, ikinci olayının ise 404 kimlik numaralı 1 işaretçisi ile belirtilmiş olduğu anlamına gelir. İşaretçiler her bir örüntünün yapısı ve tanımını belirler. Bu örnekte kullanıcı bir köylüye kılıçla saldırıyorsa oyuncunun silahsız köylülere saldıracak kadar saldırgan olduğunu varsayabiliriz. Fakat bir sonraki olayda köylüden (hedef: *TargetID* 4 değeri) özür dileyen bir konuşma seçeneğini seçerek (fiil kimlik numarası 4, nesne kullanılmadığı için numarası 0) bize oyuncunun köylüye kaza ile saldırdığı bilgisini verir ki bu durumda saldırı bir saldırganlık belirteci olarak sınıflandırılamaz. Bütün bu örüntü tanımları bir veri dosyasında saklanır ve sonrasında 2 boyutlu vektörlere transfer edilir. Böylece iki indeks numarası ile bu vektörden istenen her bilgi çağrılabilir. Örneğin Vektör 2,5 2. örüntünün 2. işaretçisini gösterir. Bahsi geçen vektör yapısı aşağıda ki gibidir.

		COLUMN																
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ROW	0	0	1	3	7	4	1	3	7	4	-2	0	0	0	0	0	0	0
	1	1	1	3	7	4	-2	0	0	0	0	0	0	0	0	0	0	0
	2	2	1	3	7	4	-1	3	7	4	1	3	7	4	-2	0	0	0
	3	3	1	3	6	8	1	3	7	4	-2	0	0	0	0	0	0	0
	4	4	1	2	6	7	-1	2	5	7	1	3	7	4	-2	0	0	0
	5	5	1	2	5	7	-3	0	0	0	1	3	7	4	-2	0	0	0
	6	6	1	2	9	7	-4	0	0	0	1	3	7	4	-2	0	0	0
	7	7	1	2	9	7	-4	0	0	0	1	3	7	4	-2	65	0	0
	8	8	1	2	9	7	-4	0	0	0	1	3	7	4	-2	68	88	0

Şekil 2: Örüntü Vektör Yapısı

2.2.1. Sürekli Örüntüler

Sürekli örüntüler eylemlerin zamanda ardışık olmasını gerektiren örüntülerdir. En basit örneği bir oyuncunun kombo dizisini gerçekleştirmek için yapmak zorunda olduğu tuşlara basma olaylarıdır. X ardından Y hemen ardından A tuşuna basma olaylar dizisi oyun dünyasından beklediğimiz kombo dizisi olabilir.

4	1	370	1	380	1	390	-2

Şekil 3: Sürekli örüntüler

Dört numaralı örüntünün 1 işaretçisi ile (tam eşleşme) işaretli son olayı 370'dir (A tuşuna basılması) , ondan önceki olay ise 1 ile işaretli 380 (Y tuşuna basılması) ve ilk olay ise 1 ile

DOI: 10.5824/1309-1581.2015.2.002.x

işaretli 390'dır (X tuşuna basılması). Bu durumda algoritma önce 370 ile biten örüntülere bakacak, bulduğunda hemen öncesinde 380 numaralı olaya bakacak ve sonrasında 390 ile biten olayları tarayacaktır. -2 örüntünün bittiğini gösteren işaretçidir.

2.2.2. Değillenmiş Sürekli Örüntüler

Değillenmiş sürekli örüntüler burada kendisinden hemen sonra gelmemesi gereken eylemler içeren örüntüler olarak tanımlanmıştır. Mesela bir kombo hareket dizisini başlatmak için kullanıcının bastığı klavye tuşları dizisi; X tuşuna bastıktan sonra Y tuşuna basma hareketini içermeyen ve takibinde A tuşuna basmayı içeren bir örüntü oyun dünyasında gözlemlediğimiz bir kombo dizisini gösterebilir.

5	1	370	-1	380	1	390	-2

Şekil 4: Değillenmiş Sürekli Örüntüler

Örüntü numarası 5'in son olayı 370 numaralı eylemdir (A tuşuna basılması) ve 1 işaretçisi ile gösterilmiştir (tam uyum) ve son eylemden önceki eylem 380 numaralı eylem değildir (Y tuşuna basılması) ve -1 ile işaretlenmiştir, ayrıca ilk eylemin kimlik numarası 390'dır (X tuşuna basılması) işaretçisi ise 1. Bu durumda önerdiğimiz algoritma sonu 370 ile biten örüntüleri inceler ve bulması durumunda bir önceki eylemin 380 olmamasını kontrol eder, bu koşulun sağlanması halinde 390 numaralı olayı arar -2 işaretçi sonu ifadesidir.

2.2.3. Son Eşleşmeli Kesintili Örüntüler

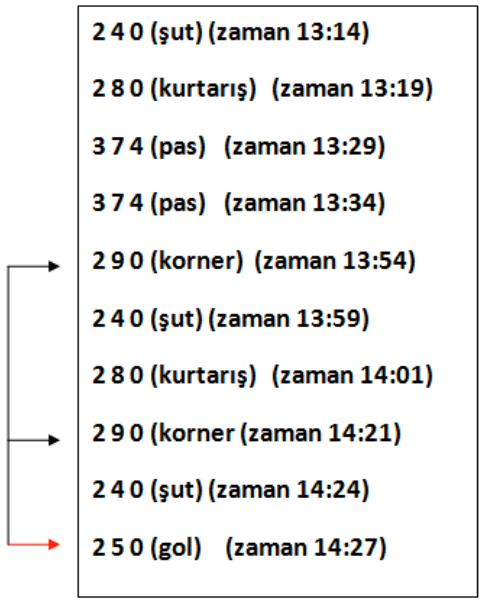
Bu örüntülerde bir eylemin hemen bir başka eylemi takip etmesini gerektirmeyen örüntülerdir. Aynı zamanda eylem dizesindeki en yakın eşleşmeye bakarlar. Buna örnek olarak bir futbol oyununun oyuncu profillerini anlamak için sürekli gözlemlenen oyuncu istatistikleri ele alınabilir: hangi oyuncunun korner atışları daha etkili sorusunu cevaplamak için gol ile sonuçlanan korner atışlarına bakılabilir. Bu örüntüde korner atışı ile gol arasında çok sayıda farklı olay yer alabilir, örneğin Y sut çeker, X şutu kurtarır, arada Z de X'e faul yapar vb. Bu durumda korner atışı ile gol arasında ne olduğu ile değil korner ile başlayıp arada envai çeşit eylemi içeren ve gol ile biten örüntülerle ilgileniriz.

7	1	250	-3	000	1	290	-2

Şekil 5: Son Eşleşmeli Kesintili Örüntüler

Yedi numaralı örüntünün son olayı 250 numara (gol atılması) ve 1 değeri taşıyan işaretçi ile gösterilen bir olaydır. Son olaydan önce her bir olay (000 ile temsil edilen) -3 işaretçisi ile işaretlenmiştir ve ilk olay ise işaretçisi 1 olan (tam eşleşme), 290 numaralı olaydır (korner atışı). Bu durumda algoritma 250 ile biten örüntüleri tarar , bulması durumunda sonunda

290 numaralı olayı bulduğu sürece geri kalanları ihmal eder. Aşağıdaki gibi bir eylemler dizisi olduğunu var sayalım:



Şekil 6: Son Eşleşmeli Kesintili Örüntü örneği

Oyun dünyasından 250 gelmesi durumunda algoritma 250 (gol) ile biten örüntüleri tarar. Bunlardan biri yukarıda örüntü numarası 7 ile temsil edilen örüntüdür. Algoritma bu örüntüyü bulduktan sonra örüntü içinde köşe atışı olaylarını arar ve diğerlerini ihmal eder. 14:21'de bir korner atışı ve 13:54'de bir başka korner atışı bulacaktır. Bu durumda hangi köşe atışını kullanmak için "sonuncuyu kullan" anlamına gelen -3 değerli işaretçiyi kullanarak sonuncuyu kullanacaktır. Bu durumda eşleşmiş örüntü 14:21'de başlar ve 14:27'de biter. İstenirse bu örüntü eşleşmeleri zaman içinde sınırlanabilir öyle ki algoritma sadece (*örüntü sonu - örüntü başlangıcı)* < *belirtilen zaman* uzunluğu koşuluna uygun olanları bulur. Ayrıca istenirse minimum zaman ve maksimum zaman dilimleri de tanımlanabilir böylece örüntü araması *minimum*< *(örüntü sonu- örüntü başlangıcı)* < *maksimum* koşulu ile çift taraftan da sınırlandırılabilir.

2.2.4. Son Eşleşme ve Değillenmiş Kesikli Örüntüler

Bu örüntüler sürekli değildir, eylemler hemen başka bir eylem tarafından takip edilmek zorunda değildir. Aynı zamanda algoritma eylem dizisinde ki en yakın eşleşmeye bakacaktır, daha önce ki eşleşmeler kaale alınmaz. Yine futbol oyunundan örnek verecek olursak; var sayalım gene gol ile sonuçlanan eylem dizilerine bakmak istemekteyiz ama bu sefer araya bir ofsayt atışının girmesinin artık korner ile başlayıp gol ile biten bir örüntü olma koşulunu ihlal ettiği kuralını da işletelim (korner ofsayt gol sıralamasında ofsayt olması bu örüntüyü kornerden gol ile değil ofsayttan gol örüntüsüne sokar). Bu örüntü de gol ile korner atışı arasında istenilen sayıda eylem olabilir bunun tek istisnası ise arada ofsayt

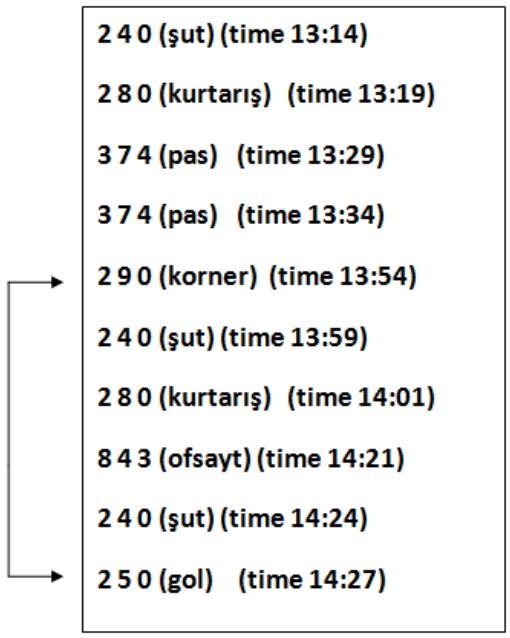
DOI: 10.5824/1309-1581.2015.2.002.x

eyleminin olmasıdır. Önerdiğimiz örüntü yapısı oyun içinde algoritmanın bu tip aramalar yapmasına izin verir.

8	1	250	-5	843	1	290	-2

Şekil 7: Son Eşleşme ve Değillenmiş Kesikli Örüntüler.

Sekiz numaralı örüntünün son olayı 1 işaret değerli bir 250 'dir (gol atma) , son olaydan önce olaylar -5 işaretçili bir 843 (ofsayt) olmamak şartı ile her hangi bir olay olabilir ve ilk olay 1 işaretçi değerine sahip bir 290 (köşe atışı olmalıdır). Bu durumda algoritma 250 ile biten ve 290 ile başlayan ve arasında 843 içermeyen bütün örüntüleri tarayacaktır. Aşağıda ki eylem dizisine sahip olduğumuzu varsayalım:



Şekil 8: Son Eşleşme ve Değillenmiş Kesikli Örüntü örneği

Oyun dünyasından bir 250 (gol) olayı gelmesi durumunda algoritma bu değerle biten örüntüleri bulmaya çalışır. Yukarıdaki 8 numaralı örüntü buna bir örnek teşkil eder. Bu örüntü için 250 tam bir eşleşme olduğundan arada ki olayları 843 numarasını taşımamak kaydıyla ihmal ederek köşe atışı aramaya başlar. Elimiz de ki örnekte önce 240 numaralı olayı bulacak ve bunu ihmal edecek ama sonrasında 843 kimlik numaralı olayı bulacak ve örüntü eşleştirme algoritmasını durduracaktır. Bu durumda 13:54'deki korner atışı araya ofsayt girdiği için gol ile sonuçlanan korner atışı örüntüsüne girmeyecektir. Bu durumda 8 numaralı örüntü istendiği gibi örüntü eşleşmesi olarak kaydedilmeyecektir.

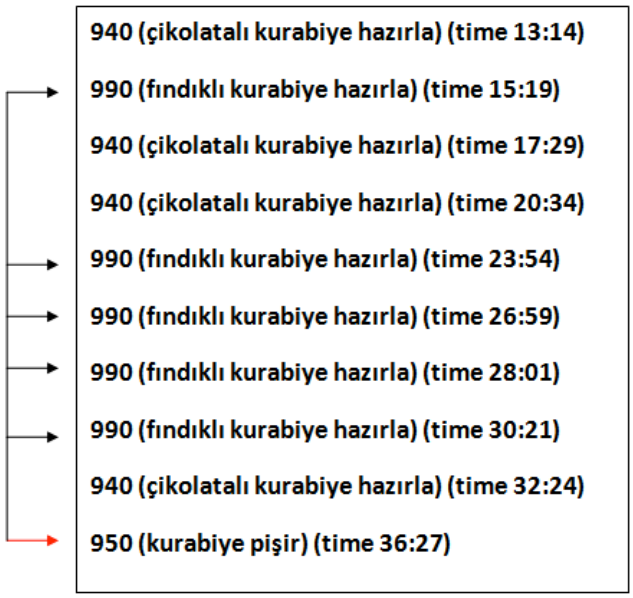
2.2.5. Çoklu Kesikli Örüntüler

Bu örüntüde de eylemler ardışık olmak zorunda değildir, aynı zamanda bir eylem dizisindeki bütün eşleşmeleri bulmaya çabalar. Örnek olarak bir oyunda oyuncu tarafından yapılan fındıklı kurabiyelerin sayısını tutmak gösterilebilir. Aradığımız örüntünün fındıklı kurabiye yapımı olduğunu varsayarak bu örüntü iki bölümden oluşacaktır, gerekli malzemelerin toplanıp kurabiyelerin hazırlanması ve kurabiyelerin fırında pişirilmesi. Bu durumda oyuncun iki farklı izlek izleyebilir: kurabiyeleri teker teker pişirebilir veya hepsini birden fırına koyabilir.

9	1	950	-4	000	1	990	-2

Şekil 9: Çoklu Kesikli Örüntüler

Dokuz numaralı örüntünün son olayı 1 işaretçi değerine sahip 950 numaralı eylemdir (kurabiye pişirmek) , son eylemden önce ki eylemler -4 işaretli eylemlerin her biri olabilir (000 ile temsil edilen) ve ilk eylemse 1 işaretçili 990 eylemidir (fındıklı kurabiye hazırlama). Bu durumda algoritma 950 ile biten örüntüleri aramaya başlayacak, bulması durumunda sonu 990 ile biten bütün eylem dizilerini aradaki eylemlere aldırmadan bakacaktır. Aşağıda ki eylem dizisinin verildiğinin var sayalım:



Şekil 10: Çoklu Kesikli Örüntü örneği

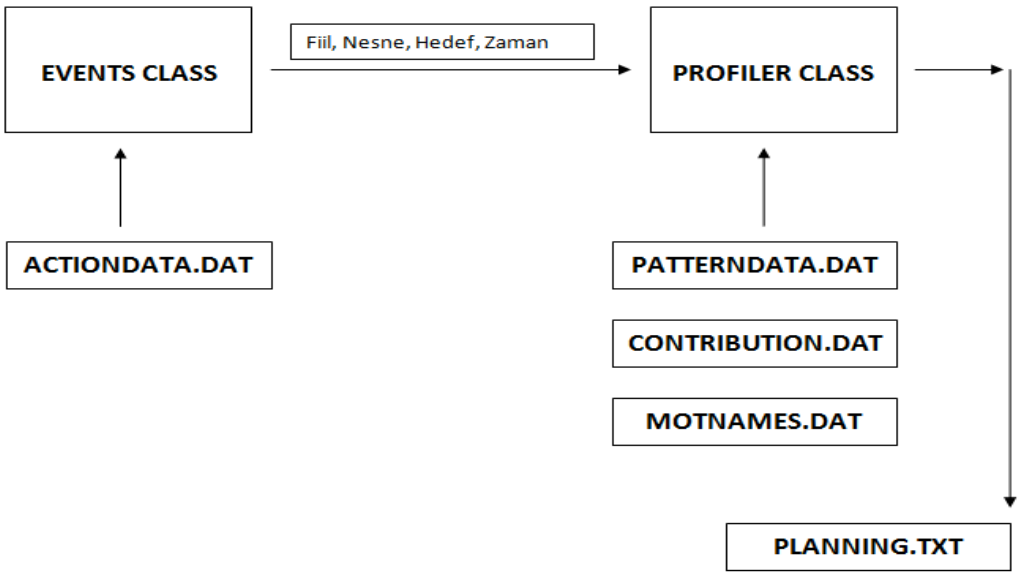
Oyun dünyasından 950 (hazırlanmış olan kekleri pişir) numaralı eylemi aldığımızda algoritma 9 numaralı örüntüyü 990 numaralı olaylarla eşleştirmeye çalışacaktır. Bu örüntü için eylem dizisini taramaya başlayacak ve bütün 940'ları ihmal edecek ve bulabileceği bütün

DOI: 10.5824/1309-1581.2015.2.002.x

990ları arayacaktır. Sonunda 36:27'de 950 alındığında bu 9. örüntü için 5 eşleşme anlamına gelir. Bu algoritma bütün örüntüleri tarar ve geride bulunmamış hiç bir 9 numaralı örüntüyü parça olarak bile bırakmaz. Bu örüntülerde zamanda ne kadar geriye gidileceğini sınırlandırmak mümkündür.

2.3. Profil Çıkartıcı Yapısı

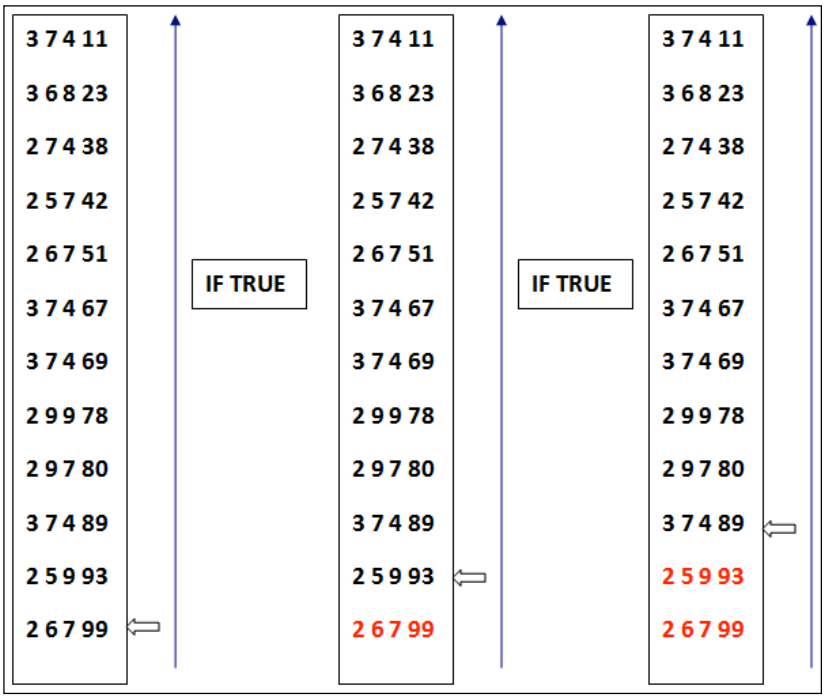
Profil Çıkartıcı Sınıfı tanımlı eylem örüntülerini okumakla yükümlüdür ve kayıtlı eylem örüntülerini oyun içindeki örüntüler ile eşleştirmek için kayıtlı bilgiyi kullanır (*PatternData.dat*). Aynı zamanda her bir örüntünün katkısını da bir dosyadan okumak ve bir başka dosyadan motivasyon isimlerini okumak ile yükümlüdür. Tasarlanmış olan sistemde bulunan her eylem örüntüsü bir motivasyon değişkene katkı sağlar. Katkı dosyası formatı şöyledir: ilk değişken örüntünün kimlik numarası, sonraki motivasyonun kimlik numarası sonuncusu ise katkıdır ; örneğin 1-2-1 girdisini ele alalım: Bu 1 numaralı örüntünün 2 numaralı motivasyona artı bir katkı yaptığını ifade eder. Burada motivasyonun ismi ve tanımı kullanıcıya ekran bildirimi verecek şekilde tutulmaz. Bu bilgi ayrı bir motivasyon isimleri dosyasında tutulur. Bu dosyanın formatı ilk girdi motivasyon numarası ikinci girdi ilgili motivasyonun adı şeklindedir. İlk aşamada bütün dosyalar Profil Çıkartıcı sınıf tarafından okunur . Sonra örüntüler işlenir. İşleme tamamlandıktan sonra eşleşen profil bilgisi tipik olarak bir Drama Yöneticisine yönlendirilir. Bu çalışmada bu işlemi simüle etmek için gerekli bilgiler bir başka metin dosyasında tutulmuştur. Kaydedilen bilgiler hem motivasyon değişkenlerine yapılan son katkıları (örneğin 'ilişki kurma ihtiyacı' 2 birim arttı) hem de değişmiş olan motivasyon değişkeninin güncel toplam değerini (Örneğin: an itibarı ile 'ilişki kurma ihtiyacı' 8'dir) içerir. Önerilen Profil analizcisinin genel mimarisi aşağıda ki gibidir:



Şekil 11: Profil Çıkartıcı Yapısı

2.3.1. Örüntü İşleme

Örüntüleri işlemek veya gelin veri akışı ile tanımlı örüntüleri eşleştirmek için örüntü eşleme algoritması kullanılmıştır. Bir örüntüyü eşleştirmek için son koşul örüntüye ait son eylem verisini almaktır. Örneğin 374-368-374 sürekli örüntüsünü veri akışında bulmak için önce bir 374 kodlu olayın bilgisi gelmeli, sonrasında bunu 368 takip etmelidir fakat örüntünün bulunması için en son olarak bir başka 374 olayı bilgisinin gelmesi gerekmektedir. Bu mantığı kullanarak algoritma oyun dünyasından yeni bir eylem aldığında önce bu olay ile biten örüntüleri tarar. Sonrasında bu biçimde bulduğu örüntülerde son alınan olaydan bir önce ki olayı kontrol eder. Aşağıdaki olayların veri akışının oyun dünyasından geldiğini varsayalım: En son gelen olay 99'da 267dir (mavi oklar zamanın akışının tersine işleyen işleme algoritmasının yönünü gösterir). Bu olay geldiğinde algoritma 267 ile biten bütün örüntülere bakar. İlk önce 0 numaralı örüntünün son olayı olan vektör(0,2)'ye bakar ve eşleşmediğini bulur, sonra 1 numaralı örüntünün son olayı olan vektör(1,2)'ye geçer eşleşme bulamaz ve devam eder. Kıyaslamaları bitirdiğinde bu örnekte örüntü kriterlerine uyan tek bir örüntü bulacaktır; 267 numaralı olayla biten 4 numaralı örüntü.

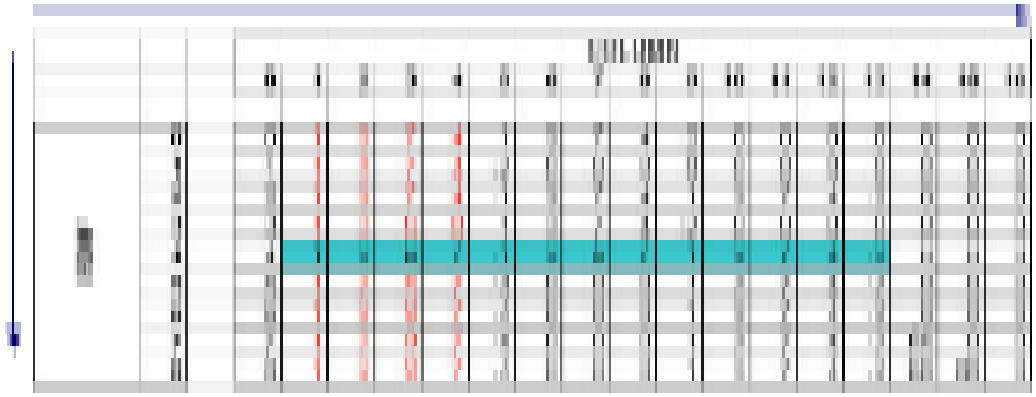


Şekil 12: Örüntü İşleme Mantığı

Son eylemin eşleştiğini bulduktan sonra örüntü tanımının işaretçisine geçer ve veri akışında bir eylem geri gider ki bu örnekte 259 değerini taşır. 4 numaralı örüntünün tanımı bu değerin 257 olmaması gerektiğini ön koşul olarak sunar. Bu kritere göre 259 257 olmadığı için bu testi geçer ve bir sonra ki işaretçiye bakar ve tekrar bir eylem geri gider, bu örnekte kendisi 374'tür. 4 numaralı örüntünün tanımı da bu olayın 374 olduğunu söylediği için bu test de geçilir ve bir sonraki işaretçiye bakılır. Bu -2 değerli örüntünün sonuna geldiğini gösteren işaretçidir. Bütün eylemler örüntü tanımı ile eşleştiğinden dolayı olay akışının 89 ve 99 zaman etiketleri arasında ki olaylar 4 numaralı örüntü ile eşleşmiş olur. Aşağıdaki mavi çizgiler örüntü matriksindeki işleme yönleridir. İşleme ilk satır ilk işaretçi ile başlar, eşleşme bulursa ilk sıra ikinci işaretçiye bakar ve devam eder. Hiç bir koşulun uymaması durumunda bir satır aşağı iner. Bu şekilde bütün 2 boyutlu vektörü eşleşen bir örüntü

DOI: 10.5824/1309-1581.2015.2.002.x

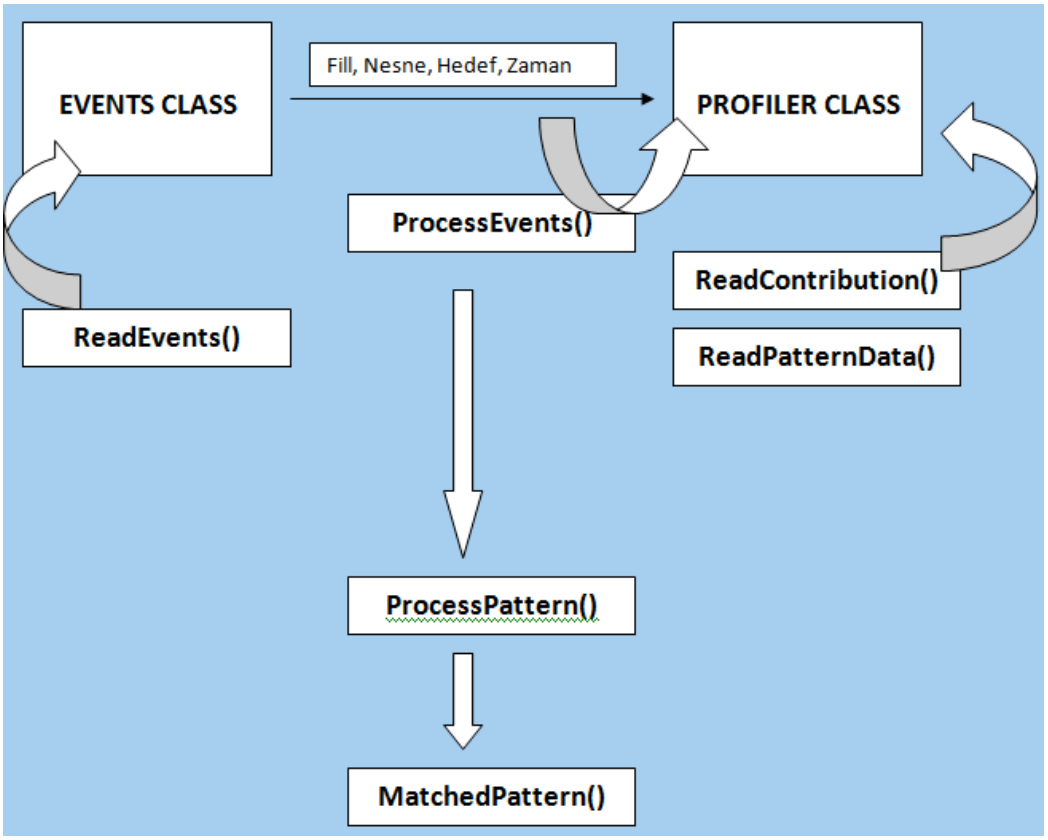
bulmak için taramış olur. Aşağıda ki örnekte mavi ile gösterilen hücreler verilen eylem alışında eşleşen örüntüleri göstermektedir.



Şekil 13: Örüntü İşleme Süreci

2.3.2. Sınıf Diyagramı

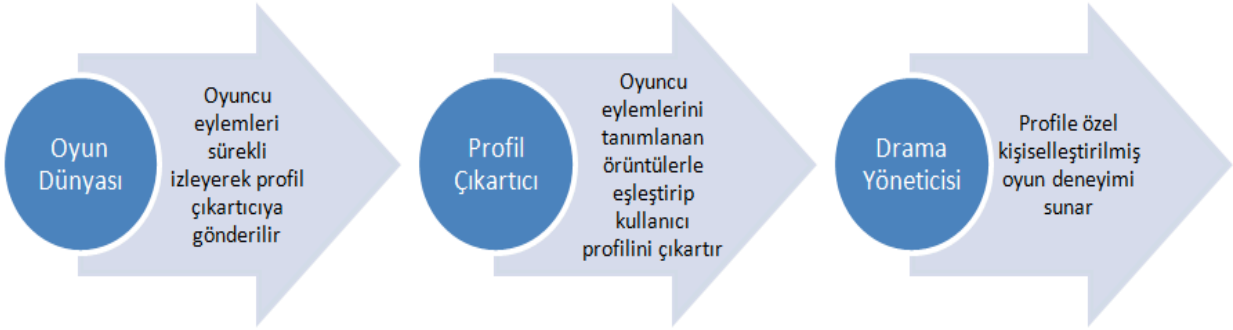
Tasarlanan Profil Çıkartıcı'nın sınıf diyagramı aşağıdaki gibidir:



Şekil 14: Sınıf Diyagramı

3. Sonuç

Bilgisayar oyunları dışındaki sahalarda kullanılan profil çıkartma teknikleri oyuncu analizine uygun değildir. Örneğin, veri madenciliği yapabilmek için oyuncuya ilişkin bilgileri bir veritabanında saklayıp bunların üzerinde sorgular yapmak gerekmektedir ancak bunu gerçek zamanlı olarak yapmak hem çok hızlı olmayacak hem de işlemci yükünü arttıracağı için oyun performansını etkileyecektir. Ayrıca, kullanıcı bilgilerini karşılaştırmak için bu veritabanını bir sunuca tutmak gerekecektir ki ağ bağlantısı olmayan veya donanımları yetersiz kalacak bilgisayarlarda böyle bir işlemi gerçekleştirmek mümkün olmayacaktır. Durum Tabanlı Çıkarsama veya Bayes ağları gibi yöntemler için oyuncu eylemleri oyundan oyuna değişiklik gösterebilen ve az tanımlı bir sahadır, aynı zamanda oyuncu eylemlerinin öngörülebilirliği de sorun olmaktadır. Geliştirdiğimiz sistemin esas amacı Oyun Dünyası ile Drama Yöneticisi arasındaki Profil Çıkartıcı'nın mümkün olduğunca hızlı ve efektif çalışmasıdır.



Şekil 15: Etkileşimli Hikaye Anlatma Sistemi Yapısı

Literatürdeki oyuncu profilleri oyuncu eylemleri dizilerini incelemekte yetersiz kaldığı için eylem örgüleri ve bunların motivasyon odaklı profillere katkısı tercih edilmiştir. Oyun oynamak amaca yönelik bir davranıştır ve oyuncuların eylemlerini motivasyonları belirler. Ayrıca, her oyuncu eyleminin bir profile katkıda bulunacağı hipotezi bu çalışmada reddedilmiştir çünkü oyuncu eylemleri bir eylemler örgüsü içerisinde anlamlıdır ve geçmişteki eylemlerin yanı sıra gelecekteki eylemler ile bir araya gelerek farklı bir motivasyona hizmet eder, bir profile katkıda bulunur. Eylemlerin birbirinden bağımsız olarak incelenmesi ve her bir eylemin bir profile işaret etmesi çok karmaşık bir süreci aşırı basitleştirmekten başka birşey değildir.

Mevcut etkileşimli hikaye anlatma sistemleri genelde drama yöneticisi tasarımını ele almakta profilin nasıl oluşturulduğu ile ilgilenmemektedirler. Bu çalışmada eylem çeşitliliği çok geniş olmayan oyunlar için oyuncunun gerçek zamanlı eylemlerini daha önceden tanımlanabilen eylem örüntüleri ile eşleştirip buna dayalı olarak motivasyon değişkenlerini değiştiren, bunu yaparken de ağır bilgisayar işlemi gerektirmeyen bir çerçeve ve ilgili C++

DOI: 10.5824/1309-1581.2015.2.002.x

yazılımı sunulmuştur. Böylece profillere göre farklı bir içerik sunacak drama yöneticileri için basit, tek bir olaya bakmaktan ziyade eylem dizilerinin kullanıcının karakteristiğini belirttiğini varsayan dinamik bir profil çıkartıcı hazırlanmıştır. Profil oluşturma kullanılacak olan eylem dizilerinin ana karakteristiği beş madde olarak sınıflandırılmıştır: Çoklu Kesikli Örüntüler, Son Eşleşme ve Değillenmiş Kesikli Örüntüler, Son Eşleşmeli Kesikli Örüntüler, Değillenmiş Sürekli Örüntüler, Sürekli Örüntüler. Bu sınıflandırma profil tanımlayan örüntüleri oluşturmada bütün olası eylem dizilerinin sınıflandırılması için yeterlidir.

4. References

Barber, H., Kudenko, D. (2008). Generation of Dilemma-based Interactive Narratives with a Changeable Story Goal, in: *Proceedings of Second International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)*.

Bartle, R.A. (2004). *Designing Virtual Worlds*, New Riders Publishing.

Bates, J. (1992). Virtual Reality, Art, and Entertainment. *Presence: The Journal of Tele-operators and Virtual Environments, 1* (1), 133-138.

Beal, C. R., Beck, J., Westbrook, D., Atkin, M., Cohen, P. (2002). Intelligent modeling of the user in interactive entertainment, paper presented at the *AAAI Spring Symposium*.

Berridge, K.C. (2001). Reward Learning: Reinforcement, Incentives and Expectations, in: Medin, D. L. (ed.), *Psychology of Learning and Motivation, 40*, pp. 223-278.

Bradley, K., Rafter, R., Smyth, B. (2000). Case-Based User Profiling for Content Personalisation, In: *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2000)*.

Dena, C. (2008). Emerging Participatory Culture Practices: Player-Created Tiers in Alternate Reality Games. *Convergence: The International Journal of Research into New Media Technologies*, 14, 41-57.

Cavazza, M., Charles, F., Mead, S.J. (2002). Interacting with Virtual Characters in Interactive Storytelling, in: *Proceedings of the First ACM Joint Conference on Autonomous Agents and MultiAgent Systems*, Bologna, Italy, pp. 318-325.

El-Nasr, M.S. (2007). Interaction, Narrative, and Drama Creating an Adaptive Interactive Narrative using Performance Arts Theories, *Interaction Studies, 8* (2), 209-240.

Esposito, F., Semeraro, G., Ferilli, S., Degemmis, M., Di Mauro, N., Basile, T.M.A., Lops, P. (2004) Evaluation and Validation of two Approaches to User Profiling, in: Berendt, B., Hotho, A., Mladenic, D., van Someren, M., Spiliopoulou, M. and Stumme, G. (Eds.), *1st European Web Mining Forum, Lecture Notes in Artificial Intelligence*, Springer:Berlin.

Fawcett, T., Provost, F. (1996) Combining Data Mining and Machine Learning for Effective User Profiling, in: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*.

Froschl, C. (2005) *User Modeling and User Profiling in Adaptive E-learning Systems*, Master Thesis, University of Technology, Graz, Austria..

Godoy, D., Amandi, A. (2003) A User Profiling Architecture for Textual-Based Agents, Revista Iberoamericana de Inteligencia Artificial, 7 (21), pp. 27-36.

Heckhausen, J., Heckhausen, H. (2005). *Motivation and Action*, Cambridge University Press.

Kelso, M., Weyhrauch, P., Bates, J. (1993). Dramatic Presence. *Presence: The Journal of Teleoperators and Virtual Environments, 2* (1), 1-15.

Krulwich, B. (1997) Lifestyle finder: Intelligent user profiling using large-scale demographic data, Artificial Intelligence Magazine, 18, 37-45.

Lashkari, Y., Metral, M., Maes, P. (1994). Collaborative Interface Agents. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence*.

Magerko, B. (2005) Story Representation and Interactive Drama, in: *Proceedings of the 1st Conf. On Artificial Intelligence and Interactive Digital Entertainment*.

Mateas, M. (1997). An Oz-Centric Review of Interactive Drama and Believable Agents, in: *AI Today: Recent Trends and Developments. Lecture Notes in Artificial Intelligence 1600*, pp. 297-328.

Mateas, M. (2000). A Neo-Aristotelian Theory of Interactive Drama, in *Proceedings of AAAI Spring Symposium on AI and Interactive Entertainment*.

Mateas, M., Stern, A. (2001). Interactive Drama. A Thesis Proposal, Ph.D. thesis, Pittsburgh, Carnegie Mellon University.

Mulligan, J., Patrovsky, B. (2003). *Developing Online Games. An Insider's Guide*. Indianapolis, IN: New Riders.

Nebel, I., Smith, B., Paschke, R. (2003) A user profiling component with the aid of user ontologies, in: *Workshop Learning - Teaching - Knowledge - Adaptivity (LLWA 03)*, Karlsruhe.

Peinado, F., Gervás, P. (2007). Automatic Direction of Automatic Storytelling: Formalizing the Game Master Paradigm, in: *Proceedings of the 4th International Conference on Virtual Storytelling: Using Virtual Reality Technologies for Storytelling (ICVS)*.

Pohjola, M. (2004). Autonomous Identities: Immersion as a Tool for Exploring, Empowering and Emancipating Identities, in: Montola, M., Stenros, J. (Eds.), *Beyond Role and Play: tools, toys and theory for harnessing the imagination*. Ropeconry, Helsinki, pp. 81-96.

Salen, K., Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*, Cambridge, MA: The MIT Press.

DOI: 10.5824/1309-1581.2015.2.002.x

Schiaffino, S.N., Amandi, A. (2000) User profiling with Case-Based Reasoning and Bayesian Networks, *IBERAMIA-SBIA 2000 Open Discussion Track*, pp. 12-21.

Semeraro, G., Degemmis, M., Lops, P., Basile, P. (2007) Combining Learning and Word Sense Disambiguation for Intelligent User Profiling, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 2856-2861.

Szilas, N. (2003). IDTension: A Narrative Engine for Interactive Drama, in: *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, 187-203.

Thue, D., Bulitko, V., Spetch, M., Wasylishen, E. (2007). Interactive Storytelling: A Player Modelling Approach, *in: Proceedings of Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE)*, Stanford, California, 43-48.

Waszkiewicz, P., Cunningham, P., Byrne, C. (1999) Case-based User Profiling in a Personal Travel Assistant, *User Modeling: Proceedings of the 7th International Conference*, pp. 323-325.