

Son Kullanıcı Tabanlı Yazılım Geliştirme Aracı

Cengiz TOĞAY¹

Volkan AKKUŞ²

Ali Hikmet DOĞRU³

¹Netaş Telekomünikasyon A.Ş., İstanbul, Türkiye

²Yalın, Bursa

³Orta Doğu Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Ankara

¹ctogay@netas.com.tr, ²volkan.akkus@yalin.com.tr, ³dogru@ceng.metu.edu.tr

Özet. Son kullanıcılar sahip oldukları uygulamalara gereksinimleri doğrultusunda çeşitli düzeylerde üreticiden bağımsız olarak müdahale etme eğilimindedirler. Wordpress gibi uygulamalarda, veri içerik yönetimi kapsamında çok kısıtlı bir şekilde yapılan tanımlamalar aracılığı ile özelleştirilmiş (başlık isimleri, logolar vb.) ürün elde edilebilmektedir. Sunulan çalışma ise; son kullanıcıların web ortamında sunulan geliştirme ortamı aracılığı ile yeni ürün geliştirme, mevcut ürünü değiştirme ya da ürünü kullanan diğer kullanıcıların da faydasına olacak bir bileşeni geliştirme yeteneğini kazanmasını sağlamaktadır. Böylece, çalışmada sunulan projede tanımlı üç farklı tip son kullanıcı gereksinimleri doğrultusunda satın aldıkları ürüne müdahale edebilmektedirler. Son kullanıcıların etkide buldukları/tanımladıkları ürün tasarımı, sunulan motor aracılığı ile Web ve Microsoft Windows işletim sistemi olmak üzere aynı ürünün iki farklı çözümünün otomatik olarak üretilmesi sağlamaktadır. Web ortamında üretilen ürün bulut üzerinde çalıştırılabilir.

Anahtar Kelimeler: Son Kullanıcı, Yazılım Geliştirme Süreci, Yazılım Geliştirme Aracı, Nitelik Modeli

1 Giriş

Son kullanıcıların yazılım geliştirme süreçlerine dâhil olması giderek yaygınlaşmaktadır. Çevik yazılım geliştirme süreçleri bu yöntemlerden bir tanesidir. Ancak bu tip yaklaşımlarda son kullanıcı, isterlerin tespit edilmesi, ürünün doğrulanması ve istenilen ürünün üretilmesi için yer almaktadır. Yazılım ise mühendisler tarafından geliştirilmektedir. Ağırlıklı olarak, yazılım süreçlerinde son kullanıcıların isteklerini mühendislere sözlü, metinsel ya da belirli bir formatta aktarmalarının ardından sürece bağlı olarak zaman zaman kullanıcıların dâhil olması ile ürün geliştirilmektedir. Son kullanıcılar ile yazılım geliştiricilerin, farklı terimler kullanmaları nedeni ile ortaya çıkan iletişim problemlerine sunulan bir çözüm Nitelik modelidir. Nitelik modeli [1][2][3][4] ile ürün havuzundaki ürünler arasındaki benzerlik ve farklılıkların bir ağaç yapısında ifade edilmesi sağlanır. Dolayısı ile bu modelin kullanılması için mevcut ürün havuzunun olması ve yeni ürünün havuzdaki ürünü oluşturan bileşenlerin/varlıkların yeni bir kombinasyonu ya da değiştirilmesi ile elde edilmiş yeni bir ürün

şeklinde oluşturulması gerekmektedir. Sunulan çalışmada amacımız temelleri [20]'de atılan yöntem ile son kullanıcıların mümkün olduğunca fazla bir şekilde yazılım süreçlerine dâhil olmasının sağlanmasıdır. Bu amaçla detayları ikinci bölümde verilmele birlikte, üç tip kullanıcı tipi tespit edildi. Böylece, kullanıcıların yetkinlikleri düzeyinde yazılım süreçlerine dâhil olabilmesi sağlanmıştır. Birinci tip kullanıcıların nitelik modeli üzerinde yapacakları seçimler ışığında gereksinimlere uygun yazılımın otomatik olarak üretilmesi sağlanmaktadır. İkinci tip kullanıcılar, mevcut akış diyagramları üzerinde değişiklik yapmak, yeni akış diyagramları tanımlamak ve ürünün arayüzünde görünecek olan bilgileri eklemek ya da çıkartmak suretiyle uygulamaya doğrudan müdahale edebilmektedir. Son olarak üçüncü tip kullanıcılar, üründe kullanılmak üzere yeni bileşenlerin (örnek olarak bir SMS hizmetinin sağlanması) dll olarak geliştirilerek projeye dahil edilmesi ya da tamamen yeni bir alanda (domain) uygulama geliştirilebilmesi için veritabanının tasarlanması gibi görevleri yerine getirebilmektedirler.

1.1 Nitelik Modeli

Model, bir alan mühendisi ile birlikte yeni bir alanda kullanıcının bilgileri, kitaplar, mevcut farklı çözümler vb. kaynaklardan toplanarak oluşturulması sureti ile havuzda hiç bileşen olmasa da gerçekleştirilebilir. Bu durum için amaç, son kullanıcının isteklerinin en iyi şekilde algılanmasıdır. Ancak, ürün hattı mimarilerinde (product line architecture) nitelik modelleri yaygın şekilde kullanılmaktadır. Nitelik en geniş anlamı ile tüm paydaşlar (kullanıcı, müşteri, geliştirici, alan uzmanı, vb.) tarafından bilinen ve ürünler arasında benzerlik ve farkındalık yaratan bağlamlar (aspect) ile karakteristikler olarak bilinir [1][2][3][5]. Nitelik modelinde nitelikler genellikle bir ağaç yapısı ile ifade edilmektedir. Ancak, nitelikler arasındaki karmaşık (bir yaprak nitelik bir başka daldaki yaprak niteliğe 'gerektirir/hariç' ilişkisi ile bağlı olabilir) bağlantıların bulunması nedeni ile arka planda çizge şeklinde bir oluşum meydana gelebilmektedir. Anlaşılabilirliğin artırılması amacı ile ağaç üzerinde tekrarlı yapılar ile ifade edilebilir. Gösterim konusunda ihtilaflar bulunmakla birlikte Czarnecki'nin gösterimi [6] en çok tercih edilen gösterimlerdendir. Nitelik modelleri pek çok sahada hâlihazırda kullanılmaktadır; örnek olarak; "bulletin board system domain" [2], "the private branch exchange domain" [7], "web services domain" [8], "elevator control systems" [9] verilebilir [4]. Bunlarla birlikte yazılım mühendisliği kapsamında; tekrar kullanılabilir yazılım geliştirme [10], bağlam yönelimli programlama [11], üretimsel (generative) programlama [12], ürün hattı yazılım mühendisliği [2][13], nesneye yönelik yazılım mühendisliği [7], bileşen yönelimli sistemler [14][15][16][17] ve nitelik yönelimli programlama [18] örnek olarak gösterilebilir [4]. Nitelik modelinde yer alan niteliklerin kendi aralarındaki ve çözüm uzayındaki bileşenler ile bağlantılarının kural tabanlı sistemler aracılığı ile ifade edilmesi ve kontrol edilmesi mümkündür [19]. Bu şekilde gereksinim dünyasında yapılacak seçimlerin gerçek dünyada bir karşılığının olup olmadığı, uyumsuzluk yaratan niteliklerin tespit edilmesi sağlanabilmektedir.

Nitelik modeli, farklı kullanıcı tipleri (son kullanıcı, yazılım mühendisi, sistem mühendisi vb.) için farklı düzeyde (nitelik, işletim sistemi, donanım vb.) olarak hazır-

lanabilmektedir. Bu çalışmada Şekil 1’de gösterildiği üzere sadece son kullanıcı temelli nitelik modeli oluşturulmuş ve kullanılmıştır. Alan mühendisi tarafından hazırlanan model, bileşen havuzunda yer alan bileşenlerin bir araya getirilerek ürün üretmesinde farklılık yaratacak nitelikleri içermektedir. Birinci tip kullanıcı, yapacağı seçimler ile istediği ürünü tanımlama imkânına sahip olmaktadır. Çalışmada sunulan proje ile Şekil 1’de gösterilen “Araştırma Projeleri Yönetimi” niteliğinin seçilmesi durumunda üretilecek uygulamada ilgili nitelik ile bağlantılı tüm yapılar (veri tabanı, kod düzeyinde vb.) otomatik olarak üretilecektir. Dolayısı ile projede “Erasmus”, “Leonardo”, “Öğretim Üyesi Yetiştirme” ve “SanTez” ile ilgili varlıklar projede yer alacaktır. Nitelik modeli üzerinde çalışan kural tabanlı sistemler aracılığı ile hangi niteliklerin birlikte seçilebileceği/seçilemeyeceği, seçimler sırasında ayrıca önerilerin verilmesi, nitelik modeli üzerinde yapılan seçimlerin tutarlı olup olmadığı vb. sonuçlar otomatik olarak çıkarılabilmektedir.

Nitelik
Coklu Essiz / SANTEZ / Nitelik

Ad	Üst Nitelik	Açıklama	Ek
MRPII Tekstil Dokuma	N/A	N/A	
Üniversite Öğrenci İşleri Otomasyonu	N/A	N/A	
Dış Hekimliği	Üniversite Öğrenci İşleri Otomasyonu	N/A	
Doktora	Üniversite Öğrenci İşleri Otomasyonu	N/A	
Hazırlık	Üniversite Öğrenci İşleri Otomasyonu	N/A	
Lisans	Üniversite Öğrenci İşleri Otomasyonu	N/A	
Ön lisans	Üniversite Öğrenci İşleri Otomasyonu	N/A	
Akademik ve İdare Faaliyetler	Ön lisans	N/A	
Araştırma Projeleri Yönetimi	Ön lisans	N/A	
Erasmus	Araştırma Projeleri Yönetimi	N/A	
Leonardo	Araştırma Projeleri Yönetimi	N/A	
Öğretim Üyesi Yetiştirme Projeleri	Araştırma Projeleri Yönetimi	N/A	
SANTEZ	Araştırma Projeleri Yönetimi	N/A	
Bilim İnsanı - YÖKSİS	Ön lisans	N/A	
Destek Hizmetleri Yönetimi	Ön lisans	N/A	
Doküman Yönetimi	Ön lisans	N/A	
Döner Sermaye Yönetimi	Ön lisans	N/A	
İş Süreç Yönetimi - BPM	Ön lisans	N/A	
Öğrenci İşleri Yönetimi	Ön lisans	N/A	
Öğrenci Toplulukları Yönetimi	Ön lisans	N/A	

Page 1 of 2 (25 items) 1 2

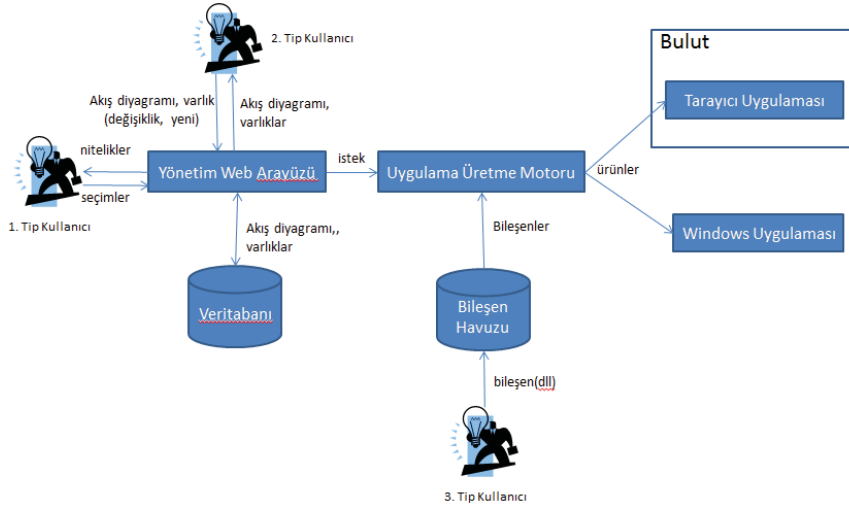
Şekil 1. Üniversite Öğrenci İşleri Otomasyonu Nitelik Modeli

Son kullanıcıların ürünleri teslim aldıktan sonra çeşitli düzeylerde üründe değişiklik ihtiyacı ortaya çıkmaktadır. Ürünün kaynak kodlarına sahip son kullanıcılar dahi üründe değişiklik yapılması için bakım ya da mühendisin şirket bünyesine dâhil edilmesi gibi ihtiyaçlar ortaya çıkmaktadır. Dolayısı ile son kullanıcılar, ürünün tüm yaşam döngülerinde yer alırken özellikle küçük müdahaleler şeklindeki istekler için bir yaklaşıma ihtiyaç bulunmaktadır. Sunulan yaklaşım ile son kullanıcıların nitelik mo-

deli ile ürünün niteliklerini seçme/belirleme, üründe akış diyagramları dâhil olmak üzere yenilikler veya değişiklikler yapmak ve yazılım geliştirme yetkinliği var ise geliştirilecek bileşenler aracılığı ile bileşen havuzunu genişleterek yeni bir ürünün oluşturulmasına katkı sağlaması hedeflenmektedir.

2 Önerilen Yaklaşım

Son kullanıcılar aldıkları eğitimler veya şirket içi pozisyonlarına bağlı olarak kullandıkları yazılımlara müdahale etmek eğilimindedirler. Bu müdahale uygulamanın satın alınması sırasında dahi olabilmektedir. Kullanıcı, ürünün tamamı yerine sadece belirli niteliklere sahip olan şeklini satın almak isteyebilmektedir. Ürün bazında hangi niteliğin hangi başka nitelikleri gerektirdiği ya da karşı olduğu gibi nitelikler arasındaki ilişki ağı, özellikle ürünü satmaya çalışan personel için dahi karmaşık bir hal alabilmektedir. Dolayısı ile bileşenlerin bir araya gelmesi ile çeşitlilik kazanan bir ürün havuzunun oluşturulması yerine, müşteri isteklerine göre otomatik olarak ürün üreten bir uygulama, farklı tip kullanıcı beklentilerini karşılamak üzere ürünleri otomatik olarak geliştirilebilir.



Şekil 2: Geneu Mimarisi

Mimarisi Şekil 2'de gösterilen "Geneu" isimli proje, kullanıcı tarafından tanımlanmış niteliklere uygun bir yazılımın otomatik olarak geliştirilmesini sağlamaktadır. Geneu ile web tarayıcıları ve Windows işletim sisteminde çalışacak uygulamanın aynı anda geliştirilmesi mümkün olmaktadır. Kullanıcı aynı yetkiler ile iki farklı uygulamada aynı özelliklere sahip olabilmektedir. Son kullanıcıların bu isteklerini karşılamaya yönelik olarak; Geneu ile üç farklı kullanıcı tipi için uygulamaya müdahale ve yeni bir uygulama geliştirilmesine olanak tanımaktayız:

- Birinci tip: Nitelik Modeli üzerinde ihtiyaçlarına yönelik seçimler ışığında istediği niteliklere sahip ürünü elde eder. Bu tip kullanıcı web ortamında istediği ürünü nitelik modeli üzerinde seçebilen bir kullanıcı ya da bir satış temsilcisi olabilir. Kullanıcılar, seçimleri sırasında mümkün olmayan seçimler konusunda sistem tarafından uyarılmak sureti ile yönlendirilirler. Sistem, nitelik modeli üzerinde çalışacak motorlar aracılığı ile daha akıllı yönlendirmelerin yapılması sağlayabilir. Yapılan seçimler, Geneu tarafından üretilecek kodun hangi bileşenlere ihtiyacı olduğunu belirler. Dolayısı ile son kullanıcının doğrudan ürünün özelliklerini belirleme yetkisi vardır. Bu tip kullanıcı bir satış temsilcisi de olabilir. Geneu'nun kod üretme yeteneği sayesinde satış temsilcilerinin müşteri ile gerçekleştirecekleri iletişim sonucunda doğrudan çözümün anlık üretilmesi ve müşteriye gösterilmesi imkânı bulunmaktadır. Şekil 2'de yer alan "SANTEZ" niteliğinin seçilmesi durumunda, Şekil 3'te gösterilmekte olan nitelik ile ilişkilendirilmiş varlıklar otomatik olarak ürünün üretilmesi safhasında veritabanına Şekil 4' de gösterilen özelliklerle birlikte hazırlanacaktır.

Ad: SANTEZ

Üst Nitelik: Araştırma Projeleri Yönetimi

Açıklama:

Ek:

Varlıklar	Parametreler	Alt Nitelikler	Projeler	Nitelik İlişkileri	Geçmiş	
<input type="checkbox"/>	<input type="button" value="Yeni"/>	<input type="button" value="Kopyala..."/>	<input type="button" value="Sil"/>	<input type="button" value="Edit"/>	<input type="button" value="Export to"/>	
Ad	Çoğul Ad	Anlam	Listede Görünen Özellik	Ek	Geçmiş Tut	
<input type="checkbox"/>	Başvuru	Başvurular	Tübitak'a yapılan başvurular	Başvuru No	N/A	<input type="checkbox"/>
<input type="checkbox"/>	Yazışma	Yazışmalar	Teydeb ile yapılan resmi yazışmalar	Belge no	N/A	<input type="checkbox"/>

Items per page: 20

Şekil 3: SANTEZ ile ilgili varlıklar

Varlık - Başvuru

Varlık

Ad: Ek:

Çoğul Ad: Geçmiş Tut

Anlam: Menü Grubu:

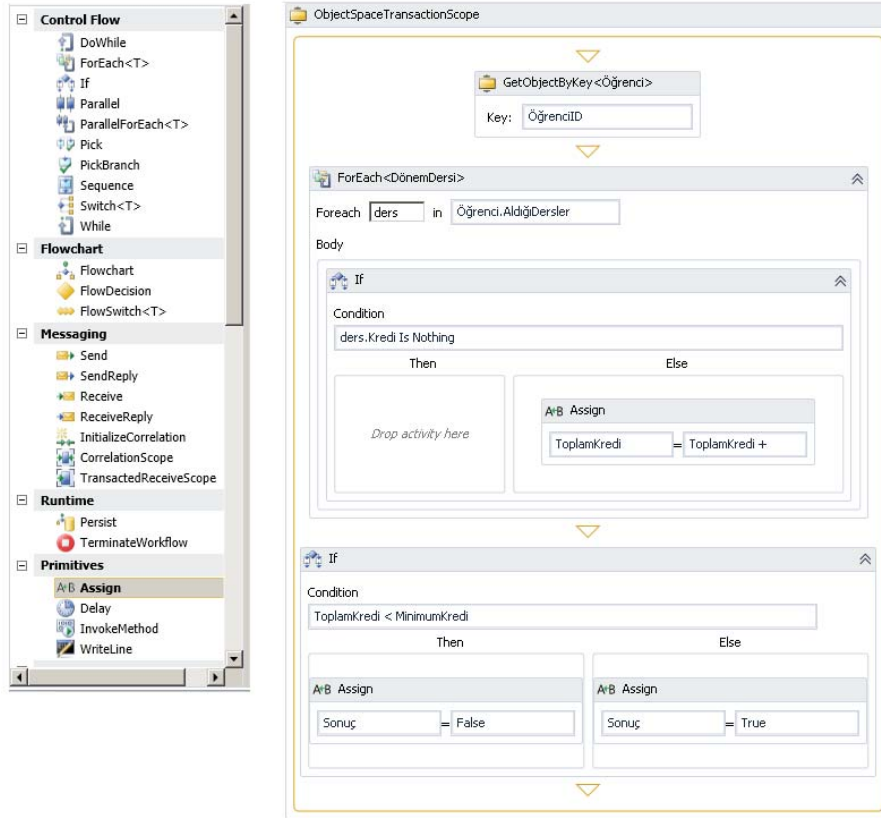
Listede Görünen Özellik: İkon:

Ozellikler	Hesaplanır Özellikler	Çoklu Eşsizler	Eylemler	Olaylar	Geçmiş
<input type="checkbox"/>	<input type="button" value="Yeni"/>	<input type="button" value="Kopyala..."/>	<input type="button" value="Sil"/>	<input type="button" value="Edit"/>	<input type="button" value="Export to"/>
Ad	Anlam				
<input type="checkbox"/>	Başvuru No	Teydeb'in atadığı numara			
<input type="checkbox"/>	Proje Sahibi	Proje Yöneticisi - Öğretim Görevlisi			
<input type="checkbox"/>	Tahmini Bütçe	Yaklaşık Maliyet (TL)			

Items per page: 20

Şekil 4: Başvuru varlığı ve özellikleri

- İkinci tip: Birinci tip kullanıcının tüm yeteneklerine sahip olmakla birlikte, bu tip kullanıcılar, yeni tablo oluşturma/değiřtirme ve sürükle bırak temelli iş akışı modeli oluşturma/değiřtirme ile ürüne yeni özellikler katma yeteneklerine de sahiptirler. Şekil 5'te bir öğrencinin aldığı toplam kredi sayısının mezuniyet için gerekli minimum değer ile kontrolü yapılmaktadır. Belirli bir rakamın aşılması durumunda ders alması engellenebilir. Bu tip kullanıcılar projede yeni varlıklar ve özellikleri tanımlamak sureti ile arka planda otomatik olarak oluşturulacak veri tabanı tablolarının oluşturulmasına yol açar. Yine arayüz üzerinde yapılan seçimler ile varlıklar arasında ilişkiler kurulur. Örnek olarak bir üniversitede yer alan öğrenciler, hocalar, dersler ve aralarındaki ilişkiler ele alındığında, bir öğrencinin alabileceği toplam ders sayısına kısıt getirme gibi temel özellikler Geneu arayüzü ile doğrudan gerçekleştirilebilir. Benzer şekilde, eğer böyle bir sistem var ise, kullanıcı, öğrenciye yeni bir özellik (öğrenci boyu/kilosu vb.) eklemek sureti ile sisteme katkıda bulunabilir. Bu kullanıcı tipinin bir miktar yazılım bilgisinin olması ya da kısa bir eğitim sürecine tabi tutulmasına ihtiyaç olabilir.



Şekil 5. Akış Diyagramı

- Üçüncü tip: İkinci tip kullanıcının tüm yeteneklerinin yanı sıra ürüne yeni bir bileşenin kodlanarak ekleme yeteneğine sahip olan kullanıcıdır. Örneğin, e-posta atabilme yetkinliği kazandırılması için hazırlanacak dinamik kütüphane (dll dosyası) Geneu bileşen havuzuna eklenmek sureti ile ilk iki tip kullanıcının hizmetine sunulur. Burada yapılan geliştirim faaliyetleri tamamen Geneu ortamının dışında meydana gelir. Mevcut sistemin, Microsoft .NET kütüphanelerine dayanması nedeni ile bu ortamda geliştirilmesi daha iyi sonuçlar verecektir. Ayrıca bu tip kullanıcıların yeni bir alanda uygulama geliştirilmesi için gerekli alt yapı çalışmalarını da gerçekleştirilmesi beklenmektedir.

3 Sonuç ve Tartışma

Son kullanıcıların sadece isterleri belirlemediği, doğrudan ürüne müdahale edebilme yeteneğinin kazandırılması çalışmaları her geçen gün önem kazanmaktadır. Web tabanlı kurulum arayüzü sonrasında, kurulum sırasında aldığı verilerden yola çıkarak kullanıcıya özel ürünün otomatik olarak oluşmasını sağlayan ürünler bulunmaktadır. Bu tip içerik belirleme sistemlerine örnek olarak Wordpress ya da Joomla gösterilebilir. Bu çalışmada ise içerik belirleme sistemlerinden farklı olarak son kullanıcının (özellikle ikinci ve üçüncü tip) bir adım daha öteye giderek, sadece ürünü özelleştirme değil; web tabanlı ortamda sürükle bırak tabanlı yeni uygulama geliştirme, uygulamaya eklentiler yapma ve ürünün davranışlarını değiştirme yeteneği kazandırılmış olmaktadır.

Gelinen noktada nitelik modeli editörü ve iş akış editörü birimleri ile birlikte bir tümleştirilmiş geliştirme ortamı hazırlanmıştır. Ortaya çıkan araç ile farklı kullanıcı tipleri ile test uygulamaları geliştirilmiştir. Uygulama aynı zamanda olgunlaşmış alanlarda hızlı uygulama geliştirmenin önünü açmaktadır. Ürün web ortamında tanımlanması nedeni ile build işlemi farklı bir makine de gerçekleştirilmektedir. Dolayısı ile özellik büyük projelerde geliştirici bilgisayarlarının yerine daha kapasiteli sunucular üzerinde günlük build işlemlerinin otomatik olarak gerçekleştirilmesi projenin doğasından kaynaklı olarak sunulmaktadır.

4 Teşekkürler

Bu çalışmada adı geçen Geneu projesi TÜBİTAK-Teknoloji ve Yenilik Destek Programları TEYDEB tarafından desteklenmiştir.

5 Kaynaklar

- [1] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study," Carnegie-Mellon University Software Engineering Institute, Nov. 1990.

- [2] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, and M. Huh, "FORM: A feature-oriented reuse method with domain-specific reference architectures," *Ann Softw Eng*, vol. 5, pp. 143–168, Jan. 1998.
- [3] K. C. Kang, J. Lee, and P. Donohoe, "Feature-oriented product line engineering," *Softw. IEEE*, vol. 19, no. 4, pp. 58 – 65, Aug. 2002.
- [4] Togay, Cengiz, "Nitelik Modeli Tabanlı Ürün Uygunluk Değerlendirme Yöntemi," presented at the Ulusal Yazılım Mühendisliği Kongresi 2012 (UYMK 2012), Izmir, 2012.
- [5] A. Deursen and P. Klint, "Domain-specific language design requires feature descriptions," *J. Comput. Inf. Technol.*, vol. 10, p. 2002, 2002.
- [6] K. Czarnecki, S. Helsen, and U. W. Eisenecker, "Formalizing cardinality-based feature models and their specialization.," *Softw. Process Improv. Pract.*, vol. 10, no. 1, pp. 7–29, 2005.
- [7] K. Lee, K. C. Kang, S. Kim, and J. Lee, "Feature-oriented engineering of PBX software," in *Software Engineering Conference, 1999. (APSEC '99) Proceedings. Sixth Asia Pacific*, 1999, pp. 394–403.
- [8] S. Robak and B. Franczyk, "Modeling Web Services Variability with Feature Diagrams," in *Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems*, London, UK, UK, 2003, pp. 120–128.
- [9] K. Lee, K. C. Kang, W. Chae, and B. W. Choi, "Featured-based approach to object-oriented engineering of applications for reuse," *Softw Pr. Exper*, vol. 30, no. 9, pp. 1025–1046, Jul. 2000.
- [10] M. L. Griss, J. Favaro, and M. d'Alessandro, "Integrating Feature Modeling with the RSEB," in *Proceedings of the 5th International Conference on Software Reuse*, Washington, DC, USA, 1998, p. 76–.
- [11] K. Lee, K. C. Kang, M. Kim, and S. Park, "Combining Feature-Oriented Analysis and Aspect-Oriented Programming for Product Line Asset Development," in *Proceedings of the 10th International on Software Product Line Conference*, Washington, DC, USA, 2006, pp. 103–112.
- [12] K. Czarnecki and U. W. Eisenecker, *Generative programming: methods, tools, and applications*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000.
- [13] P. C. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2001.
- [14] J. Kalaoja, E. Niemelä, and H. Perunka, "Feature Modelling of Component-Based Embedded Software," in *Proceedings of the 8th International Workshop on Software Technology and Engineering Practice (STEP '97) (including CASE '97)*, Washington, DC, USA, 1997, p. 444–.
- [15] X. Peng, Y. Wu, and W. Zhao, "A Feature-Oriented Adaptive Component Model for Dynamic Evolution," in *Proceedings of the 11th European Conference on Software Maintenance and Reengineering*, Washington, DC, USA, 2007, pp. 49–57.

- [16] C. Togay, "HLA Tabanlı Bileşenler ile Otomatik Uygulama Geliştirme," in *Proceedings of the Ulusal Yazılım Mühendisliği Sempozyumu*, Ankara, Turkey, 2005.
- [17] C. Togay and A. H. Dogru, "Federasyonların HLA Tabanlı Simülasyonlara Tümlleştirilme Otomasyonu için bir Mekanizma," in *26. Proceedings of the 1. Ulusal Savunma Uygulamaları Modelleme Simülasyon Konferansı*, Ankara, Turkey, 2005.
- [18] D. Batory, "Feature-Oriented Programming and the AHEAD Tool Suite," in *Proceedings of the 26th International Conference on Software Engineering*, Washington, DC, USA, 2004, pp. 702–703.
- [19] C. Togay, "Systematic Component-Oriented Development with Axiomatic Design," Dissertation, Middle East Technical University, Ankara, Turkey, 2008.
- [20] A. H. Dogru and Togay, Cengiz, "Son Kullanıcı Geliştirme Ortamı için Aksiyomatik Tasarım Esinli Mimari," presented at the UYMK 2012 Ulusal Yazılım Mimarisi Konferansı UYMK 2012, İzmir, 2012.