

İşsizlik Sigortası Sisteminin Bileşen Yönelimli Modellemesi

Hüseyin Cahit Ekiz¹, Ali Hikmet Doğru², Halit Oğuztüzün³

^{1,2,3}Bilgisayar Mühendisliği Bölümü
Orta Doğu Teknik Üniversitesi
06531 Ankara

¹cahitekiz@ada.anet.tr ²dogru@metu.edu.tr, ³oguztuzn@metu.edu.tr

Özet. Bu tebliğde COSEML modelleme dili kullanılarak İşsizlik Sigortası Sisteminin Bileşen Yönelimli Modellemesi içermektedir. 1999 yılı Eylül ayında çıkan İşsizlik Sigortası Kanununun 2000 yılı Haziran ayında başlayan uygulamasının otomasyonu için geliştirilmiş bir yazılım sisteminin, grafik araç ile modellemesi sunulmaktadır. COSEML ve işsizlik sigortası konuları özetlendikten sonra modelin değişik görünüşlerine yer verilmektedir.

1 Giriş

Bu çalışma İşsizlik Sigortası uygulamalarını destekleyen bir yazılım sisteminin COSEML (Component Oriented Software Engineering Modeling Language) ile bileşen (component) tabanlı olarak geliştirilen modelini sunmaktadır. COSEML henüz yeni geliştirilmiş bir dildir [1]. Bu dil aslında daha geniş bir perspektiften bakılacak olursa, kod yazma yerine ‘tümleştirme yolu ile sistem geliştirme’ anlayışını desteklemektedir. Bileşen tabanlı teknolojiler olgunlaşmış, ama bunların gizledikleri kabiliyetlerinden yararlanacak mühendislik yaklaşımları henüz yeterli düzeye ulaşmamıştır [2]. Bu olgu, nesne yönelimli (Object Oriented) dillerin önce ortaya çıkması ve nesne yönelimli metodolojilerin daha sonra bulunması gibidir. COSEML, bileşen teknolojilerinin kolaylaştıracağı tümleştirmeye yönelik yaklaşımı gerçekleştirecek bir metodoloji çalışmasının ara ürünüdür.

Diğer taraftan, İşsizlik Sigortası devlet tarafından tesis edilmiş bir sigorta sistemidir. İşini kaybeden bir işçinin ya da ailesinin kısmen de olsa gelir edinebilmesi amacıyla ortaya çıkarılmıştır. Bir mukavele ile kanunun kapsadığı bir işyerinde çalışan bir işçi, çalışmaya isteği, kabiliyeti ve elverişli sıhhatine rağmen kendi isteği dışında işini kaybederse ve çalıştığı sürece işsizlik sigortası ödemelerini de yapmışsa, devlete çalışma isteği ile başvurduğunda sözkonusu gelir için hak kazanmaktadır.

Yazılımın üç ana birimi vardır: Kayıt, Uygunluk ve Ödeme. Ayrıca daha önceden mevcut olan ve İnsan Kaynakları sistemi ile İş Müracaat ve Yönlendirme sistemlerinin parçası olan ‘İnsan Kaynakları’ ve ‘Müracaatçı’ birimleri de geliştirilen İşsizlik Sigortası yazılımının bir parçası gibi çalışacaklardır [3]. Bu yazılımın grafiksel modellemesi, akademik prototip düzeyinde geliştirilmiş COSECASE aracı [4] ile yapılmıştır.

2 Bileşen Yönelimi

Yazılım sahası, değişik tekniklerin önce ortaya çıktığı ve bu teknikleri kullanan mühendislik yaklaşımlarının sonradan geliştiği bir tarihçe sergilemektedir. Nesne yönelimli dillerin yaygınlaşmasını izleyen bir kaç yıl içinde nesne yönelimli metodolojiler ortaya çıkmıştır. Günümüzde bileşen teknolojileri yaygınca kullanılır hale gelmiş, ancak metodolojileri henüz olgunlaşmamıştır. Bileşenler günümüzde değişik araçların desteğini almış ve çoğu ortam onları en azından nesne yönelimli metodolojiler içerisinde uygunca temsil edilebilecekleri bir şekilde uyarlamıştır. “Yönelimli” ya da “tabanlı” nitelemesiyle araç ve yöntemler, bileşen teknolojisini bir şekilde mühendislik düzeyinde kullanma çabasıdadırlar. Çoğu yaklaşım da bu yeni teknolojinin temelinde gizli olan büyük gücü göremeyip bileşenleri ancak dışlamamak üzere yapılandırılmıştır. Bazı nesne yönelimli ortamların (diller ve derleyiciler) tam yönelimleri olmadığı halde bu isimlendirmeyi kullanmaya çalıştıkları olmuştur, ancak nesne tabanlı isimlendirmesi bu ortamlar için daha uygun bulunmuştur. Günümüzde bileşen yönelimli yerine bileşen tabanlı, hatta bileşen uyumlu denebilecek ortamlar söz konusudur.

Aslında yeni teknolojinin sunduğu görünen yararların yanında, temelden bir anlayış değişimini destekleme olanağı da ortaya çıkmıştır. Artık sahanın özellikle büyük projelerde alışılmış olan başarısızlıklarla dolu tarihi, belki de sıfırdan geliştirme külfetinden kurtularak aşılabilecektir. Geleneksel mühendislik disiplinlerinin çoktan farkettiği bileşen temelli yaklaşımların yazılım sahasında da olgunlaşması beklenmekteydi. Saha bileşen yönelimi kazandıkça, mevcut kod parçaları bileşen mimarilerine uyarlanacak ve değişik sahaların bileşen kümeleri standartlaşıp tanınır duruma gelecektir.

İstenen yönelim, yaşam döngüsünün tümü boyunca kod geliştirmeyi hedeflemeyen, ancak sistemi mevcut bileşenleri bulup birleştirerek ortaya çıkarmayı amaçlayan görevlerle edinilebilecektir. Tümlleştirme zor bir çaba türüdür. Ancak sözügeçen ‘tümlleştirme yolu ile geliştirme’ anlayışı (paradigması) bu konuda ek bir zorluk getirmemektedir. Aksine yönelim uygun olduğu için daha ayrıştırma evresinde tümlleştirmenin belirtimi yapılmalı, arayüz (interface) tanımları ortaya çıkmalıdır. Bu şekilde yapılaşdırılmış bir ayrıştırma ve sonunda birleştirme, tümlleştirme karmaşıklığını azaltabilecektir. Ayrıca, risk taşıyan yeni kod geliştirme çabasından hemen hemen tamamı ile kurtulunmak hedeflenmektedir. Doğal olarak her zaman tam istenen bir bileşenin hazır olarak bulunamayacağı durumlar ortaya çıkacaktır: bu bileşenler yine de kod geliştirerek edinilebilir. Bu tür geliştirmeler için günümüze kadar gelmiş herhangi bir yaklaşım kullanılabilir.

Bileşen tabanlı yazılım mühendisliği konuları genellikle bileşenlerin oluşturulması ve sonra da birleştirilmesi şeklinde iki ayrı çaba türü altında sınıflandırılabilir. Bu tebliğde savunulan yaklaşım, bileşenlerin nasıl geliştirildiği ile ilgili değildir. Esas zorluğu oluşturan büyük bir sistem tanımı verildiğinde çözümün üretilmesidir. Bu yüzden önerilen yaklaşım tümlleştirme ile ilgilidir. Alt-çözüm parçacıklarının daha önce üretildiği, üretilmemişse bile yeniden geliştirilmesinin esas problem yanında basit olacağı varsayılmaktadır.

2.1 COSEML

Önerilen geliştirme yaklaşımı [5], bu amaçla oluşturulmuş bileşen yönelimli yazılım mühendisliği modelleme dili olan COSEML ile desteklenmektedir. Bu grafik dil, soyut seviyede sistem öğelerini tanımlar ve bütün/parça ilişkilerini belirtir. Sistemin sıradüzensel yapısını görsel olarak ön planda tutmayı hedefler. Temelde bir yapı tasarımı aracı olarak algılanabilecek bu ortamda sistem tepeden aşağıya doğru bir sırada parçalarının tanımı ile belirtilir. Öğeler arası arayüzler de tanımlanır. Ayırıştırma, yaklaşımın temel etkinliğidir; her yöntemde tasarımcının yükleneceği yaratıcılık bu etkinlikte kendisini gösterir. Etkin bir ayırıştırma yeteneği deneyimle kazanılacaktır: mantıksal olarak öğelerin sınırlarına karar verirken ileride bu öğelere karşı düşecek bileşenlerin mevcut olması da gözetilmelidir.

Soyut düzeyde öge, veri, işlev ve kontrol (package, data, function, control) elemanları tanımlanmıştır. Öge birimleri ile sistemin yapısal parçalanması tanımlanır. Bütün sistem bir ögedir. Her ögenin alt öğeleri olabilir. Aslında öğeler, temsil edebilecekleri bileşenlerin soyut belirtilmeleri olarak kullanılabilirler. Onlara soyut ya da mantıksal bileşenler de denebilir. Bir öge, diğer öğelere ayrıştırılabileceği gibi, veri, işlev ve kontrol elemanları da içerebilir.

Ayırıştırma şeklinde başlayıp gelişen sistem belirtimi, daha sonra alt düzeylerdeki mantıksal elemanların gerçek bileşenlere denk düşürülmesi ile tasarımın ileri aşamalarına ulaşır. Bileşenler ve arayüzleri, COSEML'in uygulama düzeyindeki grafik elemanlarıdır. Bir bileşenin birden çok arayüzü olabilir. Sonuç olarak alt düzeyde yeniden kullanılabilir somut bileşenler ve üst düzeyde bunların mantıksal çözümlenişi ortaya çıkmaktadır. Modelin üst düzeyi de etkin olarak yapılandırılabilmişse, tasarım örüntülerinde (Design Patterns) olduğu gibi tekrar kullanılacak soyut çözüm parçacıklarına karşılık gelecektir.

Bağlantılar (connector) soyut elemanları olduğu gibi uygulama düzeyindeki elemanları da birbirine bağlar. Mesajlar bu bağlantı elemanları içerisinde taşınacaklardır. Bir bağlantı, birden çok mesaj belirtimi içerebilir. Mesajlar işlev çağırır ya da olay bildirirler.

Özet olarak yapı boyutunu öne çıkaran bir ayırıştırma ile sistem tanımının modellenmesi istenmektedir. Nesne yönelimli yaklaşımlarda ana boyut 'veri' dir; aslında nesneler, ulaşım metodları ile desteklenmiş veri yapılarıdır. Geleneksel yaklaşımlarda ise modelleme çalışması veri ya da işlev ekseninde başlar, ama sonunda bir işlevler hiyerarşisi şeklinde bütün sistem modellenir. Esas boyut işlevdir. Tasarımın üç temel boyutu olan veri, işlev ve yapı böylece paylaşılmış olur. Yeni gelişen bileşen yönelimli yaklaşımlar, temel boyut olarak yapıya önem verirler. COSEML dilinin geliştirilme nedeni, hem sıradüzen hem de grafik belirtim önceliklerine sahip bileşen yönelimli modelleme dillerinin bulunmamasıdır. Genelde nesneye yönelik bir yaklaşım içerisinde bileşenlerin de temsil edilmesine yönelik bileşen 'tabanlı' ("yönelimli" dense bile) teknikler mevcuttur.

3 İşsizlik Sigortası

İşçilerin ve işyerlerinin, işsizlik sigortası kapsamındaki hak ve sorumlulukları, 2000 yılı itibari ile kanunen başlamıştır. İşyerleri iş kaybı formlarını 3 nüsha olarak hazırlamakla yükümlüdürler. İşveren, 15 gün içerisinde bir nüshayı işçiye, diğer bir kopyayı da İş ve İşçi Bulma Kurumuna göndermelidir. Sigortalı işçi, iş kaybı formu ile Kuruma işin sonlandırılmasından en geç bir ay süre içinde başvurmalıdır. Ödeme süreleri 600 günlük ödenmiş sigorta süresi için 180 gün, 900 gün için 240 gün ve 1080 gün için 300 gündür.

Günlük ödemeler, çalışılan son 4 ayda sigorta aidatlarına temel oluşturan gelir üzerinden hesaplanır ve gün başına bu gelirin yarısı olarak ödenir. Her ayın sonunda yapılan ödemeler, uygunluğun olduğu ayın sonunda başlatılır. Bu esnada kurum işçiye eski işine benzer işler bulmaya çalışır ve iş eğitimleri temin eder. Ancak yeni bir işin bulunması ya da kurumun önerdiği işin reddedilmesi gibi nedenlerden uygunluk, geçici ya da süresiz olarak ortadan kalkabilir. Sosyal Sigortalar Kurumu'na ödeme süresince işsizlik sigortasından para aktarımı da sözkonusudur.

4 İşsizlik Sigortası Sistemi Modellemesi

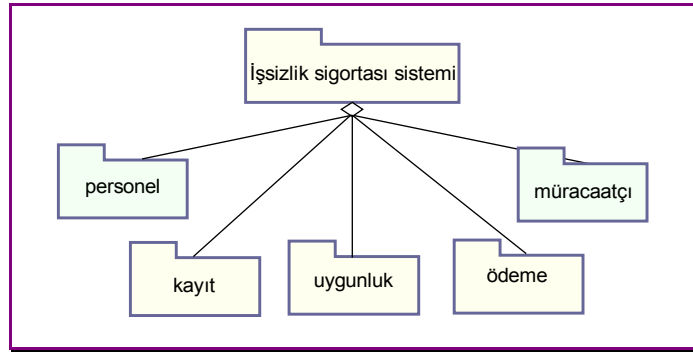
Bu bölümde daha önceden mevcut olan tasarım modellerinden alıntılar ile sistemin tanımı desteklenmektedir. Aslında Kayıt, Uygunluk ve Ödeme şeklinde sistemin 3 ana kabiliyeti vardır. Bu kabiliyetler altındaki 23 sistem işlevi, 8 veri tabanı tablosu ile ilişkilendirilmiştir. Tablo 1, bu veri tabanı tabloları ile 5 sistem işlevini ilişkilendirmektedir.

Tablo 1. Veritabanı tabloları ile sistem işlevlerinin ilişkileri

Veritabanı Tablosu	Kayıt			Uygunluk	Ödeme
	işsizlik	işyeri	iş kaybı bildirimi		
Müracaatçılar	X		X		
İşyerleri		X	X		
Faal formlar			X	X	X
Bildirim formları			X	X	
Sigorta kesintileri				X	
Pasif formlar				X	X
Ödemeler					X
SSK ödemeleri					X

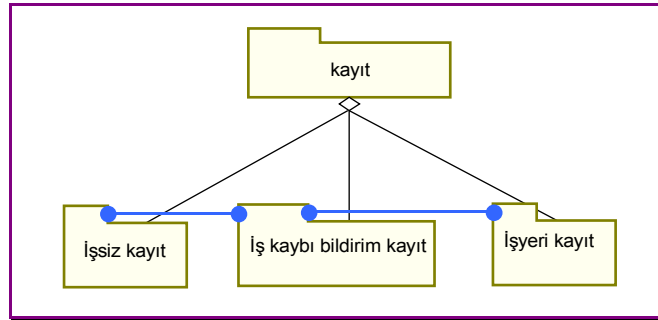
COSEML modelinde sistem kabiliyetleri alt sistemler olarak (en üst düzeydeki ögeler), üst düzey sistem işlevleri ise bunların altındaki ögeler olarak modellenmiştir. Ayrıca veritabanı tabloları veri elemanları olarak ve daha alt düzeydeki sistem işlevleri ise işlev

elemanları olarak modellenmiştir. Şekil 1, en üst düzeydeki ayrıştırmayı göstermektedir.



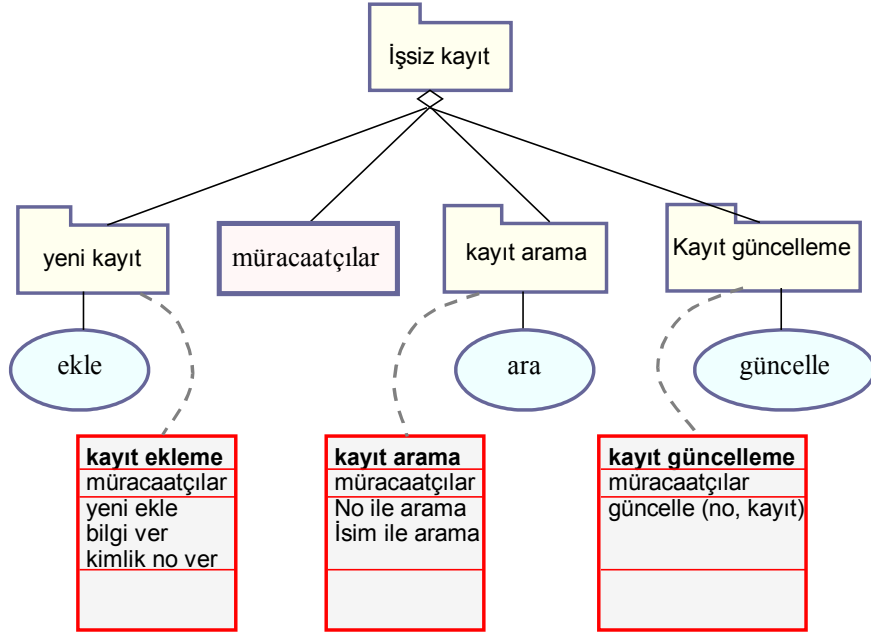
Şekil 1. İşsizlik sigortası sisteminin ilk ayrıştması

İlk ayrıştırmadan sonra geliştirilen bölümü oluşturan Kayıt, Uygunluk ve Ödeme birimlerinin ayrıştırılması farklı şekillerde verilecektir. COSEML ile oluşturulan modellerde tek bir diyagram olması gerekmektedir. Şekil 2 Kayıt ögesinin ayrıştırılmasını göstermektedir. Bu şekilde alt birimler arası bağlantılar da içerilmiştir.



Şekil 2. Kayıt biriminin parçaları

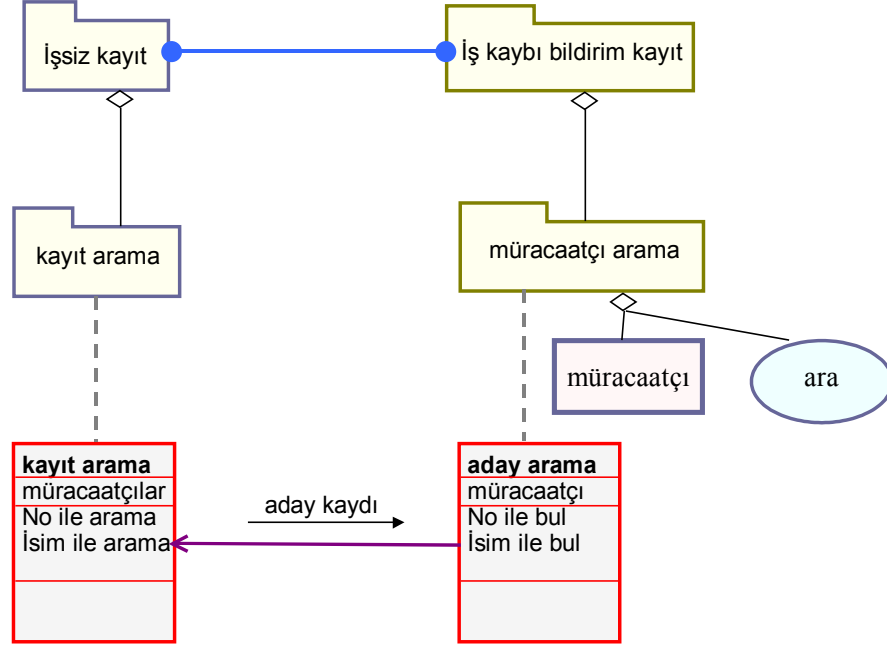
Modelin bu makalede yansıtılan kısmı, detay düzeylerinde de örnek içerilmek üzere Şekil 2'deki 'işsiz kayıt' ögesinin açılması ile devam etmektedir. Bu açılım Şekil 3'te gösterilmektedir.



Şekil 3. Kayıt biriminin parçaları

Şekil 3, daha önce tanımlanan mantıksal birimlerin olası bileşenlere denk düşürülmesini de göstermektedir. Kesikli çizgiler, bir mantıksal ögenin tanımladığı işi üstlenen bileşenleri bu öğelere bağlamaktadır. COSEML uyarınca bileşenler dikdörtgen kutular şeklinde çizilip değişik arayüzleri ayrıca gösterilebilmektedir. Ancak tek arayüzü olan bileşenlerin içerisine arayüz ayrıntıları konulabilmektedir. Şekil 3'deki 3 adet bileşen de bu kurala uymaktadır. Birer adet arayüzleri olduğundan arayüzde belirtilmesi gereken özellik, işlev ve olaylar (şekil 3 olay içermemektedir) bileşenlerin içerisine çizilmiş yatay çizgilerle ayrılmış olarak gösterilmektedir.

Son olarak, 'iş kaybı bildirim kayıt' paketi ile müracaatçılar arasında bir etkileşimi göstermek üzere iki birimden de elemanları içeren bir diyagram şekil 4'te sunulmaktadır. Değişik bileşenler arası bilgi alışverişini mesajlarla gerçekleştirmek için bu bileşenleri tanımlayan mantıksal ögeler arasında da bağlantılar olması gerekmektedir. Şekil 4 hem bu soyut bağlantıyı hem de gereken mesaj bağlantısını göstermektedir.



Şekil 4. İşsiz kayıt ve işyeri kayıt birimleri arasında etkileşim örneği

Modelin yapısal ayrıştırılması ve birimler arasındaki etkileşime örnek oluşturacak alıntılar bunlardır. Modeldeki elemanlar kullanılarak UML (Unified Modeling Language) diyagramlarından biri olan işbirliği diyagramı da çizilebilir [6]. Nitekim Şekil 4'teki tek mesaj bağlantısında UML mesajlarının biçimine uygun olarak mesajın taşıdığı sonuç parametresi, mesaj oku ile ilişkili ayrı bir ok üzerinde yazılıdır. UML mesajlarının tersine, mesajın adı okun üstünde değil, okun ucunun gösterdiği arayüz işlevinde yazılıdır. Bu mesaj ismi, bir işlev ya da olay ismi olmak durumundadır.

5 Sonuç

İşsizlik sigortası sisteminin COSEML dili ile modellemesini açıklayacak nitelikte bu modelin bir alt kümesi sunulmuştur. Yapısal ayrıştırma tabanında model, mantıksal ve fiziksel öğeleri sıradüzen sunumunda düzenlemiş ve birimler arası etkileşimi sağlayan bağlantıları içermiştir. Bu makalede bileşen yönelimli yaklaşım ile karmaşık sistemlerin modellenmesinde anlaşılabilirlik öne çıkarılmaya çalışılmıştır. Yapılan çalışmanın tümü burada gösterilmemekle birlikte, benzer çalışmalarda olumlu tasarımcı izlenimleri kaydedilmiştir. Karşılaşılan zorluklar aracın mevcut versiyonunun yetersizliklerinden kaynaklanmıştır. İşsizlik Sigortası sisteminin daha önce nesne yönelimli modelinin yapılmış olması bileşen yönelimli modeli kurmada tasarımcıya avantaj sağlamış olabilir.

Bu ve benzeri çalışmalar, mevcut nesne yönelimli tasarımı bulunan projeler için

yapılmıştır. Ayrıca üniversite öğrencileri tarafından ders projeleri COSEML ile yapılmış ve nesne yönelimli modelleri ile bir karşılaştırma çalışması da tamamlanmıştır [7]. Bu çalışma sonucunda yeni yaklaşımın modelleme gücünden ödün vermeden, az da olsa daha kolay anlaşılır olduğu gibi gözlemler edinilmiştir. Bundan sonra gerçek sanayi projelerinden oluşacak pilot çalışmaların COSEML ile *başından planlanarak* geliştirilmesi ve gerçek uygulama deneyimleri edinilip değerlendirmeleri yapılmalıdır. Ayrıca COSEML dilini öneren bileşen yönelimli yaklaşım için metodolojik düzeylerde daha fazla yönlendirici bilgilerin de oluşturulması, gerçek projelerde kullanım için yararlı olacaktır.

Kaynakça

1. Doğru, A.H., “Component-Oriented Software Engineering Modeling Language: COSEML”, 1999 *Technical Report TR-99-3*, Computer Engineering Department, Middle East Technical University, Ankara Turkey.
2. Szyperski, C. Component Software: Beyond Object-Oriented Programming, 1998, Addison Wesley, New York.
3. Kara A. A Graphical Editor for Component Oriented Modeling, Master’s Thesis, Department of Computer Engineering, METU, Ankara, Nisan 2001.
4. Ekiz, H.C. Component-oriented modeling of the Unemployment Insurance System using COSEML, Master’s Thesis, Department of Computer Engineering, METU, Ankara, Aralık 2001.
5. Dogru A.H. and Tanik M.M. “A Process Model for Component-Oriented Software Engineering”, *IEEE Software*, Mart-Nisan 2003, Cilt. 20 No. 2, s. 34-41.
6. Booch, G., Rumbaugh, J., Jacobson, I. The Unified Modeling Language User Guide, 1999, Addison-Wesley.
7. Altıntaş, I. A Comparative Study for Component Oriented Design Modeling. Master’s Thesis, Computer Engineering Department, Middle East Technical University, Ankara Mayıs, 2001.