



TÜBİTAK

2008-341
√

TÜRKİYE BİLİMSEL VE TEKNOLOJİK ARAŞTIRMA KURUMU
THE SCIENTIFIC AND TECHNOLOGICAL RESEARCH COUNCIL OF TURKEY

Elektrik, Elektronik ve Enformatik Araştırma Grubu
Electrical, Electronical and Informatics Research Group

135
9/ 96086

**Koşutzamanlı ve Yaklaşım Dayalı Programlar için
Otomatik Doğrulamaya Yönelik Tasarım**

Proje No: 106E032

Yard. Doç.Dr. Aysu Betin Can

Munara Tolubaeva
Mustafa İspir

ARALIK 2008
ANKARA

ÖNSÖZ

TÜBİTAK tarafından desteklenen “Koşutzamanlı ve Yaklaşım Dayalı Programlar için Otomatik Doğrulamaya Yönelik Tasarım” başlıklı bu projede yakın zamanda oluşan ve popülerliği artan yaklaşıma/ilgiye dayalı programlama ile doğrulanması zor ve ilgi çekici olan koşutzamanlı programlama paradigmaları üzerinde çalışılmıştır. Yaklaşım dayalı programlar için arayüz tabanlı modüler otomatik doğrulama teknikleri geliştirilmiş ve bir doğrulamaya yönelik tasarım çerçevesi oluşturma yönüne gidilmiştir. Buna paralel olarak, koşutzamanlı programların doğrulamaya yönelik tasarım yöntemi içerisindeki otomasyonu için gerekli gizli kalan ortak nesnelerin otomatik bulan bir araç geliştirilmiş, ayrıca iş parçacıklarının yalıtımı için otomatik veri bağımsızlığı analizi ve otomatik yalıtım metotlarına entegrasyonu gerçekleştirilmiştir.

İÇİNDEKİLER

ÖNSÖZ.....	1
1. Giriş.....	5
2. Genel Bilgiler.....	7
2.1. Yaklaşım/İlgiye Dayalı Programlama için Doğrulama Teknikleri.....	7
2.2. Koşutzamanlı Programlama için Doğrulamaya Yönelik Tasarım	8
3. Gereç ve Yöntem	10
4. Bulgular.....	13
5. Tartışma/Sonuç.....	14
Kaynakça.....	16

ÖZET

Bu projede yaklaşıma dayalı programlar için otomatik modüler geçerleme yöntemleri geliştirilmiş ve koşutzamanlı programlar için doğrulamaya yönelik tasarım yöntemi eksiklikleri üzerine çalışılmıştır. Çalışmanın her iki yönünde de biçimsel metotlara yabancı programcıların güvenilir yazılım geliştirmelerini sağlamak ve yazılım geliştirme ile otomatik yazılım doğrulama arasında bir köprü oluşturmaya odaklanılmıştır.

Yaklaşımlar, çapraz kesen olguları bir araya getiren ve temel yazılımın davranışını üzerinde büyük etkisi olabilen bir modülerlik birimidir. Geliştirdiğimiz teknikler yaklaşımların istenen etkiyi sağladıklarını ve temel programın ise yaklaşımın varsayımlarını yerine getirdiğini ayrık olarak geçerlenmektedir. Ayrıca var olan bir yaklaşımın bulunduğu sistemden çıkarılarak yeni sistemde kullanımında gerekli koşulların çıkarılması ve sınanması üzerine de çalışma yapılmıştır. Sözü edilen teknikler popüler bir yaklaşım dili olan AspectJ ile yazılmış programlar üzerine uygulanmıştır.

Gerçekleştirimi ve geçerlenmesi zor olan koşutzamanlı programlar üzerine yapılan çalışmada iş parçacıklarının ortak kullandığı ve korunması gereken nesnelerin otomatik bulunması için bir araç geliştirilmiş; iş parçacıklarının yalıtımı için otomatik veri bağımsızlığı analizi ve otomatik yalıtım metotlarına entegrasyonu gerçekleştirilmiştir. Bu iki çalışma koşutzamanlı programların doğrulamaya yönelik tasarım çerçevesine entegre edilmiştir.

Anahtar kelimeler: Otomatik doğrulama, koşutzamanlı programlama, yaklaşıma/ilgiye dayalı programlama (AOP), arayüzler, model denetimi

ABSTRACT

This project we worked on developing automated modular verification techniques for aspect oriented programs and filling the gaps of design for verification of concurrent programs. We aim to enable the programmers, who have no formal method background, to develop reliable concurrent or aspect oriented software and to bridge a gap between software development and automated software verification.

Aspects are the modularization units to encapsulate crosscutting concerns and have very powerful features whose effects can drastically change software behavior. The verification techniques we have developed checks to questions modularly: "does the aspect have the provisioned effect?" and "does the base program satisfy the assumptions made by the aspect?" In addition, we have developed a semi-automated system that takes an aspect oriented program, extracts the contract of the aspect, and checks the contract on the new environment of the aspect. We have applied all of these techniques on programs implemented with a popular aspect language called AspectJ.

The other part of the project is on concurrent programs, which are challenging in both implementation and verification. In this part we developed a tool that automatically extracts the object that are shared by multiple threads and that needs to be synchronized. Also, we have proposed a data independence analysis and integrated the analysis into thread isolation techniques. Both of these works have contributed to the design for verification framework for concurrent programs.

Keywords: Automated verification, concurrent programming, aspect oriented programming, interfaces, model checking

1. Giriş

Yazılım sistemlerinin günümüz modern teknolojilerinde yaygın olarak kullanımı, yazılımların güvenilirliklerini (güvenlik, kesintisiz çalışma ve doğruluğu gibi) teyit ve garanti edecek tekniklerin gereksinimini öne çıkartmıştır. Yazılımın güvenilirliği özellikle görevsel olarak kritik sistemlerde önemli bir sorun ve bileşendir. Bu güvenilirliği teyit etmek üzere geliştirilen model denetimi kullanan otomatik doğrulama teknikleri günümüzde oldukça önemli ilerlemeler kaydetmiştir. Ancak tam otomatik doğrulama teknikleri ölçeklenilirliliği yeterli olmayıp, ölçeklenebilir teknikler ise oldukça fazla insan müdahalesine gereksinim duymaktır. *Doğrulamaya yönelik tasarım* (DYT) adı verilen, otomatik doğrulamaya uygun yazılım geliştirme yollarını araştırmadan çıkacak sonuçlar bu problemlerin çözümünde etkili ve yararlı olacaktır.

Bu projede DYT yöntemi iki farklı programlama paradigması üzerinde uygulanmıştır. Son yıllarda ortaya çıkan ve popülerliği artan yaklaşıma/ilgiye dayalı programlama ile doğrulanması zor ve ilgi çekici olan koşutzamanlı programlama paradigmaları üzerinde çalışılmıştır. Yaklaşıma dayalı programlar için arayüz tabanlı modüler geçerleme teknikleri geliştirilmiş ve bir doğrulamaya yönelik tasarım çerçevesi oluşturma yönüne gidilmiştir (Raporun geri kalanında *yaklaşım* yerine daha doğru bir terim olan *ilgi* terimi kullanılacaktır). Buna paralel olarak, koşutzamanlı programların DYT yöntemi içerisindeki otomasyonu için gereken eksikliklerin giderilmesi üzerine çalışma yapılmıştır. Her iki kısımda da projenin ikincil amacı olarak yazılım geliştirme ve otomatik yazılım doğrulama arasında bir köprü oluşturmaya yoğunlaşmıştır. Biçimsel/formal metotlara yabancı programcıların güvenilir yazılım geliştirmelerinde yardımcı olacak sistemler amaçlanmıştır.

İlgiye dayalı programlar için geliştirilen geçerleme yöntemi bir varsayım-garanti yöntemi olup iki aşamadan oluşmaktadır. Kullanıcının kuramsal modellerden ve modelleme-geçerleme detaylarından soyutlanması için tekniğimiz ilgi tanımlarını, ifade şekli karmaşık olmayan arayüzler ile birlikte verilmesini gerektirmektedir. Bir ilgi arayüzü durum makineleri ve ön/art koşullardan oluşmaktadır. Bu arayüz verilen ilginin en geniş ortamını belirtmektedir. Verilen bir ilgi ve arayüzünden kapalı bir ilgi modeli oluşturma algoritması geliştirilmiş ve gerçekleştirdiğimiz bir yazılım sayesinde otomatik oluşturulan bu model bir sonlu durum model denetleyicisine verilerek ilginin davranışı sorgulanmaktadır. Bu yöntemin matematiksel modelleri hazırlanmıştır. Geçerlemenin ikinci aşamasında ise verilen herhangi bir programın ilgiye uyumluluğu sorgulanmaktadır. Bu aşamada ilgi arayüzleri kullanılarak sınıyıcı ilgiler oluşturularak ve direk program üzerinden uyumluluk kontrolü otomatik olarak yapılmaktadır. Özetle kullanıcı sadece arayüz ve ilgi tanımı yaptıktan sonra geliştirdiğimiz teknikler ile dokunmuş yazılımın istenen ilgi davranışını gösterdiğini geçerleyebilmektedir.

Yukarıda açıklanan tekniğin kullanıcı tarafını kolaylaştırabilmek için yöntemin girdilerini sistemimize verebilecek bir yazılım şablonu (idiom) oluşturulmuştur. Bu şablonlar sayesinde kullanıcı ilgi kodunu yazarken aynı zamanda arayüzünü de belirtmiş olmaktadır.

İlgiler üzerine yapılan bir diğer çalışma ise ilgilerin tekrar kullanımı üzerine olmuştur. Geliştirdiğimiz bir yarı otomatik araç ile verilen dokunmuş bir ilgi kodundan ön/art koşullar ve değişmezler otomatik olarak elde edilmekte ve ilginin tekrar kullanılacağı programı sınamak üzere denetleyici ilgiler üretilmektedir.

Koşut zamanlı programlar için daha önce geliştirilmiş olan DYT çerçevesindeki iki eksiklik giderilmiştir. Bunların birincisi olarak gizli kalan ortak nesnelere otomatik bulan bir yazılım geliştirilmiştir. Bu araç için kaçış analizi (escape analysis) teknikleri irdelenmiştir (Choi vd., 1999). Kaçış analizi tekniklerinin amacı bir uygulama eriminden (scope) dışarı kaçan nesnelere belirlemektir. Ancak her kaçan nesne ortak kullanılmayabilir. Örneğin bir nesne birçok iş parçacığı tarafından okunup hiç bir zaman içeriğini değiştirilmeyebilir. Bu çeşit nesnelere somut örnek olarak global sabit nesnelere verilebilir. Diğer bir grup örnek ise bir çok iş parçacığı tarafından görülebilmemesine rağmen hiç bir zaman kullanılmayan nesnelere. Bu örneklerde olduğu gibi gerçekte ortak olarak kullanılmayan ancak kendini yaratan iş parçacıklarından kaçan nesnelere ayıklanması gerekmektedir. Aksi takdirde, mevcut kaçış analizi araçlarında (Bogda 2001; Indus) olduğu gibi, pek çok nesne ortak kullanım içerisinde belirlenecektir. Bu projede gerçekleştirilen yazılımda bu sorunları ele alınarak genel kaçış analizi tekniklerinden farklı bir yol izlenmiş ve doğrulama çerçevesi içerisinde yerleştirilmiştir.

Projenin katkıda bulunduğu diğer sorun ise iş parçacıklarının yalıtımıdır. Ölçeklenebilirlik için her iş parçacığının yalıtılması büyük önem taşımaktadır. Bu izolasyon parçacıkların ortamını oluşturma problemi içine girer. Var olan çerçeve, diğer tekniklerin aksine, senkronizasyon bileşenlerinin ortamını oluşturma sorunu kaldırırsa da iş parçacıklarının otomatik izolasyonu yönünden gelişmelere açıktır. Bu konuda özellikle otomatik veri bağımsızlığı analizi gereklidir. Veri bağımsızlığı analizi Wolper tarafından modellenmiştir (Wolper, 1986) ancak bu analiz amacımızın isteklerini karşılamamaktadır. Projede gerçekleştirilen yeni bir analiz ile senkronizasyonu etkilemeyecek parametreler ve/veya parametre değerler bulmakta ve bu eksikliği doldurmaktadır. Ortamın modellenmesinde bu parametrelere sabit değerler verilerek durum uzayı indirgenmesi önemlidir. Bu analiz için gerekli algoritma geliştirilmiş ve otomatik ortam oluşturma araçlarına entegre edilmiştir.

Proje süresince bir adet makale uluslararası hakemli dergide yayımlanmış; uluslararası hakemli konferanslarda bir adet kısa bildiri ve ulusal bir adet tam bildiri sunulmuş ve yayımlanmış; bir adet bildiri uluslararası hakemli konferansta sunulmak ve yayımlanmak üzere kabul edilmiştir. Ayrıca bu proje sonucunda bir tezli yüksek lisans tezi ve beş tezsiz yüksek lisans programı bitirme projesi oluşturulmuştur ve desteklenmiştir. Bitirme projelerinde oluşan teknik raporlar dergilere gönderilmek üzere hazırlanmaktadır. Halen yürütülmekte olan bir doktora tezi bu proje ile kısmen desteklenmiştir ve çalışmanın ilk bulguları bir makale olarak dergiye gönderilmek üzere.

2. Genel Bilgiler

Yazılım güvenilirliği teyit etmek üzere geliştirilen model denetimi kullanan otomatik doğrulama teknikleri günümüzde oldukça önemli ilerlemeler kaydetmiştir. Model denetiminin büyük ölçekli yazılım sistemlerinde etkili olabilmeleri, yazılımın yapısının ve modelinin çıkarılması ile orantılıdır. Bu sebeple yazılımın yapısı hakkında bilgi toplama işleminin, geriye dönük mühendislik (reverse engineering) teknikleri yerine, tasarım aşamasında yapılması daha verimli olacaktır. Tasarım kalıpları (design patterns) (Gamma vd., 1994) bu amaç için kullanılabilir. Bu kalıplarla tasarımı ve gerçekleştirimi yapılmış yazılım bileşenleri, hem yürütüm sırasında görev alır hem de bileşenin modelini belirtir. Bu yöntem model çıkarımı sorununa çözüm getirebilecektir. Ayrıca tasarım kalıplarının bileşenlerin davranışsal arayüzlerini belirtmesi gerekmektedir. Bu arayüzler bileşenlerin kontratlarıdır. Bu yöntem, kontrat tabanlı tasarım (design by contract) (Meyer, 1992) kavramının bir uygulaması olarak düşünülebilir. Davranışsal arayüzler sayesinde sistemin geri kalanı modellenenebilir. Bu sayede modüler denetleme için büyük olanak sağlanmaktadır. Model denetleme tekniklerinin ölçeklendirilmesi için modülerlik gereklidir.

Doğrulamaya yönelik tasarım (DYT) kavramı yazılım mühendisliği alanında yeni oluşmaktadır. Holzmann ve arkadaşları (Holzmann vd., 2008) model denetleme alanının geleceğinin programları doğrulamaya uygun şekilde yapılandırmanın olduğunu belirtmektedirler. Bu yeni açılım DYT kavramının gelecek araştırmadaki önemini vurgulamaktadır. Bu alandaki çalışmaların bazıları şunlardır. Mehlitz ve Phenix (Mehlitz ve Phenix, 2003) tasarım kalıplarının kullanımıyla doğrulamanın verimliliğini geliştirilmesini öne sürmüştür. Tasarım kalıpları büyük programlarda bağımsız olarak doğrulanabilir bileşenlere ayırmada kullanılmıştır. Bu çalışma henüz olgunlaşmadığından, kalıplar için kullanım kuralları olmasını öne sürmesine rağmen, bu kuralların tanımlama şekli kesin olarak belirtilememiştir. Bir diğer çalışma, Sharygina ve ekibi (Sharygina vd., 2001) tarafından DYT kavramı ile UML modellerinin doğrulanması üzerinedir. Adından da anlaşılacağı gibi, bu çalışmada programın kodu göz önüne alınmamıştır. Yazılımın yanı sıra, DYT kavramı donanım sistemleri ve devre tasarımında da ilgi görmüştür (Sforza vd. 2001; Krasniewski, 1990).

2.1. Yaklaşım/İlgiye Dayalı Programlama için Doğrulama Teknikleri

Programlama paradigmalarına yeni eklenen *ilgiye dayalı programlama* (aspect-oriented programming) (Kiczales, 1997), yazılımlara yeni bir tür modülerlik eklemiştir. Bu yeni paradigma ile programcılar, *ilgi* (aspect) olarak adlandırılan, nesnelere çapraz kesen olguları (crosscutting concerns) yazılımın diğer tüm yapısal olgularından ayırık olarak ele almaktadırlar. Örneğin, nesneye dayalı programlarda kayıt tutma yazılım içinde bir çok nesnenin içine yayılırken, ilgiye dayalı programlamada kayıt tutma bir ilgi olarak diğer nesnelere ayrı gerçekleştirilir. Bir başka ilgi örneği ise koşut zamanlı programlarda senkronizasyon kurallarıdır.

Bu yeni paradigmanın popülerlik kazanması nedeniyle otomatik doğrulama alanında ilgiye dayalı programların güvenilirliğini (reliability) teyit edecek yeni teknik ve araçlara ihtiyaç doğmuştur. İlgilerin getirdiği ifade gücü, potansiyel hata sayısını arttırmıştır. Bu yönde çeşitli çalışmalar yürütülmektedir. Krishnamurthi ve ekibi, ilgileri önceden belirtilmiş olan kesim noktalarına göre doğrulayabilen modüler bir yöntem geliştirmiştir (Krishnamurthi vd., 2004). Ancak burada hatasız gerçekleştirimi güç olan koşutzaman ilgileri açık bırakılmıştır. Denaro ve Monga ilgileri ortamlarından izole bir biçimde denetlenmektedir (Denaro ve Monga, 2001). Ancak doğrulama için model çıkarımı otomatik olmayıp, model ve programın arasında uyumsuzluğa meyillidir. Ayrıca sadece ilgilerin değil, yeni hatalara sebep olabilecek olan, bu ilgilerin dokunmaları (weaving) ve ana programa etkisi de incelenmelidir. İlgilerin hiç birisinde hata olmadığı halde, dokunmaları sırasında etkileşimler nedeniyle yeni hatalar ortaya çıkabilmektedir (Ubayashi ve Tamai, 2002). Bu sebeple dokunmuş kodun otomatik doğrulanması için Ubayashi ve Tamai, doğruluk kriterlerini birer ilgi olarak kaynak koda eklemiş ve Java PathFinder (Visser vd., 2003) ile otomatik incelemiştir. Ancak bu çalışmanın ölçeklenebilirliği şüphe götürmektedir. Durum uzayı böyle bir model denetleyicisinin kolayca kapasitesini aşacak büyüklüktedir. Weston ve ekibi kontrol akış ve veri analizine dayalı bir modüler doğrulama yöntemi sunmuştur (Weston vd., 2005). Bu yöntem ile ancak sabit kesim noktalarına göre analiz yapılabilmektedir. Dokumanın nasıl olacağını önceden bilinmesi gerekmektedir.

Çalışmamızın bu bölümünde ilgilerin model denetleme teknikleri ile doğrulanması için yöntemler ve bir kuramsal model oluşturulmuştur. Ölçeklenebilirliği sağlamak açısından arayüz tabanlı bir varsayım-garanti çerçevesi geliştirilmiştir. İlgilerin kuramsal olarak doğrulanabilecek şekilde adaptasyonunu ve tekrar kullanımını sağlayan yöntemler araştırılmış ve bu konuda bilgi birikimi elde edilmiştir. Çalışma sırasında yaşanan zorluk ise gerçek hayatta kullanılan yeterli büyüklükte bir ilgiye dayalı program bulmak olmuştur. Bu sorun ilgilerin nispeten yeni olmasından kaynaklanmaktadır. Bu konuda kısa dönemli olarak dört yüksek lisans öğrencisinin bitirme projeleri tamamlanmış (Uzunoğlu, 2008) (Karaca, 2008) (Özgür, E, 2008) (Özgür, A, 2008); bir doktora öğrencisi ile yapılan çalışma uluslararası bir konferansta kısa bildiri olarak (İspir ve BetinCan, ASE 2008) ve bir ulusal konferansta bildiri olarak (İspir ve BetinCan, UYMK 2008) sunulmuş ve yayımlanmıştır. Doktora öğrencisi şu anda konu üzerine çalışmalarına devam etmekte ve bir dergi yayını hazırlama aşamasındadır.

2.2. Koşutzamanlı Programlama için Doğrulamaya Yönelik Tasarım

Koşutzamanlı programların yürütümü sırasında ortaya çıkan hataların çoğunluğu senkronizasyon birimlerinin (Java'da bulunan Hoare monitörleri gibi) yanlış kullanımından kaynaklanmaktadır. Sinama teknikleri gibi geleneksel geçerlilik yöntemleri bu tarz programlarda yetersiz kalmaktadır. Bu yetersizlik, durum uzayının eşzamanlı çalışan işlemlerin sayısı ile üstel olarak artmasından ileri gelmektedir. Buna karşın, yakın zamanda koşutzamanlı programlar için model denetlemeye dayalı otomatik yazılım doğrulama teknikleri önemli aşamalar kaydetmiştir (Corbett vd., 2000; Chaki vd., 2003; Dwyer vd. 2001; Alur vd., 1998, Magee ve Kramer, 1999). Bu çalışmalarda ölçekliliği artırabilmek için çeşitli modüler denetleme teknikleri ve kısmi-

sıra indirgeme teknikleri kullanılmıştır. Bu teknikler durum uzayını indirgese de ölçeklenebilirlik sorunu aşılamamıştır.

Koşut zamanlı programlama için doğrulamaya yönelik tasarım alanında yakın zamanda çalışmalar yapılmıştır (Betin-Can ve Bultan 2004; Betin-Can vd., 2005). Bu çalışmalarda ortak kullanılan nesnelere ve bunları korumakla görevli senkronizasyon kısımlarını yazılım içerisindeki kullanıcı bileşenlerden ayırıştırılmıştır. Senkronizasyon bileşenleri eylemlerle (action), bu bileşenlerin kullanım kuralları ise arayüzlerle tanımlanması için tasarım kalıpları geliştirilmiştir. Böylelikle senkronizasyon hataları ve kullanım hataları ayrı ayrı modüller olarak otomatik denetlenebilmektedir. Bu modülerlikle, durum uzayı çok büyük olan bir program yerine, durum uzayı küçük bir çok bileşen denetlenmiş olmaktadır.

Ancak bu çalışmalarda açık kalan önemli bir kaç konu bulunmaktadır. Bunlardan en önemlisi, iş parçacıklarının (thread) ortak eriştiği nesnelere otomatik bulunmasıdır. Şu anki çerçevede ortak nesnelere belirlenmesi tamamen programcıya bırakılmış bir durumdur. Programcı ortak nesnelere ve bunlara uygun senkronizasyonu tasarım aşamasında belirleyip, gerçekleştirimi koşut zaman denetim kalıbı (*concurrency controller pattern*) ile yapmaktadır. Oysa yazılımda programcının tasarım aşamasında göremediği gizli ortak nesnelere olabilmektedir ve var olan çerçevenin geçerliliği tüm ortak nesnelere biliniyor olmasına dayanmaktadır. Proje kapsamında bu sorun üzerine gidilmiş ve verilen bir koşut zamanlı programın kaynak kodu üzerinden ortak kullanımda olan ve erişimi sorun yaratacak olan nesnelere otomatik olarak raporlayan bir araç geliştirilmiştir. Bu araç yazılımı bir yüksek lisans tezini oluşturmakta ve bir uluslararası konferansta sunulmak üzere kabul edilmiştir (Tolubaeva ve BetinCan, 2008).

Sözü geçen DYT çerçevesinin diğer açık kalan konusu ise iş parçacıklarının yalıtımı bölümündedir. Yalıtım sırasında iş parçacıkları arası bilgi taşıyan parametrelerin alabileceği değerlerin kısıtlanması/soyutlanması çerçeve içerisinde ad-hoc olarak yapılmıştır. Projede gerçekleştirilen yeni bir analiz ile senkronizasyonu etkilemeyecek parametreler ve/veya parametre değerler bulmakta ve bu eksikliği doldurmaktadır. Ortamın modellenmesinde bu parametrelere sabit değerler verilerek durum uzayı indirgenmesi önemlidir. Bu analiz için gerekli algoritma geliştirilmiş ve otomatik ortam oluşturma araçlarına entegre edilmiştir. Bir uluslararası dergi makalesi ile bulgular yayımlanmıştır (BetinCan vd., 2007).

3. Gereç ve Yöntem

Projede gerçekleştirilenler aşağıda kısaca özetlenmiştir

- İş parçacıklarının yalıtımı için bir veri bağımsızlığı analizi oluşturulmuştur. Bu analiz durağan zaman analizi (static analysis) olup bağlam (context) duyarlı değildir. Analiz sırasında program bağımlılık çizgesinin tümünün oluşturulması yerine kontrol ve veri akışları anında (on the fly) hesaplanmıştır. Analiz gerçekleştirilecek iş parçacığı yalıtımı yazılımında sürücü ve taklit nesnelere oluşturulmasına entegre edilmiştir. Oluşan sistem bir hava trafiği kontrolü yazılımı üzerinde sınanmış ve bulgular Automated Software Engineering Journal dergisinde yayımlanmıştır. Analiz hakkında detaylı bilgi bu bildirinin 5.3 bölümünde yer almaktadır.
- Dağıtımli sistemler için bir DYT uygulaması geliştirilmiş, özellikle bozukluk işleme (failure handling) için bir tasarım önerilmiştir (Şimşek, 2007). Bu çalışma bir Özlem Şimşek'in tezsiz yüksek lisans bitirme projesini oluşturmuştur.
- Koşutzamanlı Java programları için ortak kullanım nesnelere otomatik belirlenmesi için kaynak kod üzerinde analiz yapan bir araç geliştirilmiştir. Aracımız diğer yöntemlerin aksine her iş parçacığının eriminden (scope) dışarı çıkan nesneyi raporlamamaktadır. Bunları sadece potansiyel ortak nesne olarak görmektedir ve eğer bir potansiyel ortak nesnenin durumu herhangi bir iş parçacığı tarafından değiştirilirse aracımız bu nesneyi ortak kullanımda olarak göstermektedir. Örneğin sadece sorgu metotları çağrılan bir nesne aracımız tarafından ortak kullanımda olarak gösterilmez. Bu açıdan diğer kaçış analizlerinden farklı olup daha çok programcıya yardımcı olmaya yöneliktir. Aracımız sayesinde programcı senkronizasyonu düşünmeden yazılımını yapabilir, daha sonra aracımızın bulunduğu ortak nesnelere uygun bir senkronizasyon ile koruyabilir. Bu şekilde zor olan senkronizasyon gerçekleştirimi en sona bırakılır ve iş mantığı üzerine yoğunlaşılır. Ayrıca daha önce adı geçen DYT yöntemi bu son aşamada oldukça verimli çalışacak ve güvenilir bir yazılım ortaya çıkacaktır. Aracımızın bir önemli özelliği ise uzak metot çağırımı (RMI) ve kullanıcı arayüzleri (GUI) için de ortak nesnelere bulabilmesidir. Analiz bağlam duyarlı olup tepeden aşağı ve alttan yukarı iki yöntemi birleştirmektedir. Sınıf yapısından dolayı gizli kalan alan nesnelere kaçış bilgisini de bulabilmektedir. Bu analiz ve araç olarak gerçekleştirimi Munara Tolubaeva'nın yüksek lisans tezini oluşturmaktadır. Detaylı bilgiler uluslararası bir konferansta sunulmak ve yayımlanmak üzere kabul edilen bildiride bulunmaktadır.
- İlgilerin doğrulanması için bir varsayım-garanti metodolojisi ve ilgilerin birimsel doğrulanması üzerine alan araştırması çalışmaları Mustafa İspir adlı doktora öğrencisi ile yapılmış ve bulgular ulusal ve uluslararası birer bildiri ile yayımlanmıştır. Bu çalışma İspir'in doktora tezinin temellerini oluşturmaktadır. Yapılan alan araştırması ile edinilen bilgiler çalışmanın devamında

kullanılmaktadır. Sunulan doğrulama yöntemi iki aşamadan oluşmaktadır. Kullanıcının kuramsal modellerden ve modelleme-geçerleme detaylarından soyutlanması için tekniğimiz ilgi tanımlarını, ifade şekli karmaşık olmayan arayüzler ile birlikte verilmesini gerektirmektedir. Bir ilgi arayüzü durum makineleri ve ön/art koşullardan oluşmaktadır. Bu arayüz sayesinde ilginin ortamı en geniş anlamda modellenmiş olarak bir sonlu durum model denetleyicisi olan SPIN aracına verilerek ilginin davranışı sorgulanmaktadır. Bu yöntemin matematiksel modelleri hazırlanmıştır. İkinci aşamada ise verilen herhangi bir programın ilgiye uyumluluğu sorgulanmaktadır. Bu aşamada ilgi arayüzleri kullanılarak otomatik denetleyici ilgiler oluşturularak ve direk program üzerinden uyumluluk kontrolü önerilmiştir. Bu tekniğin doğruluk (soundness) ispatı üzerinde çalışmalar devam etmekte ve bilgiler bir dergiye gönderilmek üzere düzenlenmektedir.

- İlgi davranışlarının otomatik doğrulaması için PROMELA modeline çevrimi otomasyonu geliştirilmiştir. Bu araç popüler bir model denetleyicisi olan SPIN (Holzmann, 1997) kullanarak ilgilerin zamansal özelliklerini doğrulamaktadır. Geliştirilen yazılım, SPIN aracının girdisi olacak bir PROMELA model çıkarımı otomatik olarak yapmaktadır. İlgiler için yeni bir model çıkarımı algoritması oluşturulmuş ve Emre Uzunoğlu tarafından hayata geçirilmiştir. Bu çalışma Uzunoğlu'nun tezsiz yüksek lisans projesini oluşturmuştur. İlgilerin aktive olmaları *channel* yapısı ile modellenmiş ve her bir öneri için bir işlem birimi olan *proctype* tanımlanmıştır. Grubumuz bu çalışmadaki araç ve teknik rapor, projedeki doktora öğrencisinin çerçevesi ile birleştirilerek bir dergi yayını hazırlığı içerisinde.
- İlgilerin doğrulanmasına yönelik bir tasarım şablonu oluşturmuştur (Karaca, 2008). Bu şablon programcılarının bir ilgi gerçekleştirirken otomatik model çıkarımını kolaylaştırıcı bir program yapısı oluşturmalarını sağlamaktadır. Bu şekilde ilgi modülü denetleyici için gerekli bilgiler bu şablonun kullanımından otomatik olarak elde edilmekte ve tasarım ile doğrulama arasında bir köprü kurulmuştur. İrfan Karaca tarafından oluşturulan şablon bu öğrencinin tezsiz yüksek lisans bitirme projesidir. Oluşturulan şablon kullanıcının genişleteceği ana ilgi modülleri ve arayüz tanımlamak için gerekli sonlu durum makinası yapısından oluşmaktadır. Bu şablon yukarıda adı geçen Uzunoğlu'nun aracına model bilgisi sağlamak için kullanılabilir.
- İlgilerin yeniden kullanımında davranış özelliklerinin korunması üzerine bir çalışma yapılmıştır (Özgür,E 2008; Özgür,A 2008). Yapılan çalışmada verilen bir ilgi kodundan ön/art koşullar ve değişmezler otomatik olarak elde edilmekte ve ilginin tekrar kullanılacağı programı denemek üzere denetleyici ilgiler üretilmektedir. Bir programcının daha önce çalışmakta olan bir yazılımdaki bir ilgiyi tekrar kullanmak istediğinde o ilginin yeni örüleceği sistem üzerine olan koşulları çıkarıp kontrol edebileceği bir teknik sunuyoruz. Daha önceki yazılım bir prototip olabilir ve bu teknik ile programcı test ettiği prototip ortamındaki ilgiyi asıl yazılımına taşıyabilir. Gerçekleştirilen yöntem bu taşıma sırasında yardımcı olacaktır. Yöntem, iki yüksek lisans öğrencisi tarafından oluşturulmuş ve yarı otomatik bir araç geliştirilmiştir. Bu denetlemeyi geçen yazılımlarda

ilginin davranış özelliklerinin korunacağı öngörülmüştür. Bu çalışmanın biçimsel ispatı daha sonraki çalışmalara bırakılmıştır. Yukarıda açıklanan arayüz belirtimi ve ilginin varsayımlarını kullanıcıdan almak yerine, bu çalışmada ana program üzerinden arayüz kontratları çıkarılmaktadır. Çalışma sonucu iki teknik rapor olarak sunulmuştur. Proje sonucunda oluşan teknik raporlar bir dergiye gönderilmek üzere düzenlenmektedir.

4. Bulgular

Proje bir kaç adımdan oluştuğundan her biri için şu bulgular elde edilmiştir. Koşut zamanlı programlar için veri bağımsızlığı analizinin ve iş parçacığı yalıtım yapılan durum çalışmasında oldukça önemli bir etkide bulunmuştur. Durum uzayının küçültülmesinde etkin doğrulama sonuçları alınmasında büyük katkıları görülmüştür. Ortak nesnelerin bulunması programcı için bir kolaylık sağlamakta ve programcının gözden kaçırabileceği senkronizasyon gereken alanları bulmasında yardımcı olmaktadır.

İlgiye dayalı programlar üzerine yapılan çalışmalarda bir otomatik doğrulamaya ihtiyaç duyulduğunu ve dolayısıyla kullanımında çekincelerin bulunduğunu gözlemledik. Bu sebep ile durum çalışması için gerçek hayatta kullanılan yeterli büyüklükte ve karmaşıklıkta bir ilgiye dayalı program bulmak zorluğu yaşanmıştır. Ancak çalışmalarımız sonucu geliştirilen yöntemler bu güven eksikliğini giderme açısından etkili olacağını düşünmekteyiz. Doğrulama çalışmalarının ve araçlarının kullanılabilmesi için programcıyı biçimsel yöntemlerin detaylarından soyutlayarak yazımı kolay arayüzler ve şablonlar sunmanın etkili olacağı ve bu alana yoğunlaşılması gerektiği sonucuna varmış bulunmaktayız. İlgilerin temel yazılım üzerindeki programcının düşünemeyeceği etkileri bulmanın ve ya istenen etkiyi sağladığı göstermenin ilgiye dayalı programların kullanım alanlarının genişleteceğini öngörmekteyiz.

5. Tartışma/Sonuç

Bu proje sırasında Munara Tolubaeva adlı yüksek lisans öğrencisi tam desteklenmiştir. Proje sırasında öğrenci bulma sorunları yaşansa da çeşitli dönemlerde olmak üzere beş tezsiz yüksek lisans öğrencisi dışarıdan katkıda bulunmuşlardır. Son dönemde Mustafa İspir adlı doktora öğrencisi proje ekibine katılmış ve konferans sunumları desteklenmiştir. Kuramsal modellerin oluşumunda Dicle Düvencioğlu son dönem katkıda bulunmuş ve desteklenmiştir. Bu aksaklıklara rağmen projede aşağıda belirtilen yayımlar yapılmıştır.

Bu proje sonucunda bir tezli yüksek lisans tezi ve beş tezsiz yüksek lisans programı bitirme projesi oluşturulmuştur ve desteklenmiştir. Ayrıca halen yürütülmekte olan bir doktora tezi bu proje ile kısmen desteklenmiştir.

1. Altan Özgür, "A practical approach for the verification of aspect reusability", Teknik rapor ODTÜ/II-TR-2008-14, Orta Doğu Teknik Üniversitesi, 2008.
2. Ertan Özgür, "Automated aspect code generation for the verification of ESC/Java annotated programs", Teknik rapor ODTÜ/II-TR-2008-15, Orta Doğu Teknik Üniversitesi, 2008.
3. Özlem Şimşek, "Failure Handling in Distributed Systems", Teknik rapor ODTÜ/II-TR-2007-13, Orta Doğu Teknik Üniversitesi, 2007.
4. İrfan Nuri Karaca, "A Design Template for Development of Reusable Aspects", Teknik rapor ODTÜ/II-TR-2008-6, Orta Doğu Teknik Üniversitesi, 2008.
5. Emre Uzunoğlu, "Translation of Aspect Behaviors to PROMELA Model for Automated Verification", Teknik rapor ODTÜ/II-TR-2008-16, Orta Doğu Teknik Üniversitesi, 2008.
6. Munara Tolubaeva, "DoSSO: Automatic Detector of Shared Objects in Multithreaded Java Programs" yazım aşamasındadır.

Proje çerçevesinde, 1 adet makale uluslararası hakemli dergide yayımlanmış, uluslararası hakemli konferanslarda 1 adet kısa bildiri ve ulusal 1 adet tam bildiri sunulmuş ve yayımlanmıştır. Ayrıca 1 adet bildiri uluslararası hakemli konferansta sunulmak ve yayımlanmak üzere kabul edilmiştir. 1 makale de dergilere gönderilmek üzere hazırlanmaktadır. Tüm bunlarla ilgili detaylı bilgi aşağıda verilmektedir.

1. Aysu Betin-Can, Tevfik Bultan, Mikael Lindvall, Benjamin Lux, Stefan Topp. "Eliminating synchronization faults in air traffic control software via design for verification with concurrency controllers". Automated Software Engineering Journal. 14(2): 129-178 (2007)
2. Mustafa İspir ve Aysu Betin Can. "An Assume Guarantee Verification Methodology for Aspect Oriented Programming", Proceedings of the 23rd

IEEE/ACM International Conference on Automated Software Engineering (2008) sayfa 391-394

3. Mustafa İspir ve Aysu Betin Can "İlgilerin Birimsel Doğrulanması Üzerine Alan Araştırması", 2. Ulusal Yazılım Mühendisliği Konferansı Bildiri Kitapçığı, 2008
4. Munara Tolubaeva ve Aysu Betin Can. "Automatic Detection of Shared Objects in Multithreaded Java Programs", Proceedings of the International Conference on Innovation in Software Engineering (2008) kitapçığında yayımlanmak üzere kabul edildi.
5. Mustafa İspir ve Aysu Betin Can, "Modular Verification of AspectJ Programs". Bir dergiye gönderilmek üzere.

Kaynakça

- ALUR, R., Henzinger, T. A., Mang, F.Y.C., Qadeer, S., Rajamani, S. K., Tasiran, S., Mocha: Modularity in Model Checking, Proceedings of the 10th International Conference on Computer-Aided Verification (CAV), Vancouver, Kanada, (1998) pp: 521-525.
- BETİN-CAN, A., Bultan, T., Verifiable concurrent programming using concurrency controllers. Proceedings of the 19th IEEE International Conference on Automated Software Engineering, Linz, Austria (2004) pp: 248-257.
- BETİN-CAN, A., Bultan, T., Lindvall, M., Lux, B., Topp, S., Application of design for verification with concurrency controllers to air traffic control software. Proceedings of the 20th IEEE International Conference on Automated Software Engineering, Long Beach, California, ABD (2005) pp:14-23.
- BETİN-CAN, A., Bultan, T., Lindvall, M., Lux, B., Topp, S., Eliminating synchronization faults in air traffic control software via design for verification with concurrency controllers, *Automated Software Engineering Journal*, 14(2), 129-178 (2007)
- BOGDA, J. G., Program Analysis Alleviates Java Synchronization, (Ph.D. tezi) University of California, Santa Barbara, (2001).
- CHAKI, S., Clarke, E., Groce, A., Jha, S., Veith, H., Modular verification of software components in C. Proceedings of the 25th International Conference on Software Engineering (ICSE), Portland, Oregon, ABD, (2003) pp: 385–395.
- CHOI, J. D., Gupta, M., Serrano, M., Sreedhar, V. C., Midkiff, S., Escape analysis for Java, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA), volume 34, Denver, Colorado, ABD, (1999) pp:1-19.
- CORBETT, J. C., Dwyer, M. B., Hatcliff, J., Laubach, S., Pasarenau, C. S., Robby, Zheng, H., Bandera: Extracting finite-state models from java source code, Proceedings of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland (2000) pp: 439–448.
- DENARO, G., Monga, M., An Experience on Verification of Aspect Properties, Proceedings of the International Workshop on Principles of Software Evolution (IWPE),Viyena, Avusturya , (2001) pp:186-189.
- DWYER, M. B., Hatcliff, J., Joehanes, R., Laubach, S., Pasareanu, C. S., Robby, Visser, W., Zheng, H., Tool-supported program abstraction for finite-state verification, Proceedings of the 23rd International Conference on Software Engineering (ICSE), Toronto, Ontario, Kanada, (2001) pp: 177–187.
- GAMMA, E. , Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, Massachusetts, (1994).
- HOLZMANN, G. J., The Model Checker SPIN, *Software Engineering*, 23(5), 279-295, (1997)
- HOLZMANN, G. J., Joshi, R., Groce, A., New Challenges in Model Checking, 25 Years of Model Checking: History, Achievements, Perspectives, Springer-Verlag, Berlin, Heidelberg, (2008) pp: 65–76.
- Indus. <http://indus.projects.cis.ksu.edu>.
- İSPIR, M. ,Betin Can A. ,An Assume Guarantee Verification Methodology for Aspect Oriented Programming, Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering, L'aquila, Italya (2008) pp: 391-394.

İSPİR, M., Betin Can A., İlgilerin Birimsel Doğrulanması Üzerine Alan Araştırması, 2. Ulusal Yazılım Mühendisliği Konferansı Bildiri Kitapçığı, İzmir, Türkiye (2008).

KARACA, İ. N. , A Design Template for Development of Reusable Aspects, (Tezsiz MS Bitirme projesi) Teknik rapor ODTÜ/II-TR-2008-6, Orta Doğu Teknik Üniversitesi, (2008).

KICZALES, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C. V., Loingtier, J.-M., Irwin, J., Aspect-oriented programming, Proceedings of the European Conference on Object-Oriented Programming, Jyväskylä, Finlandiya (1997) pp:220-242.

KRASNIEWSKI, A., Design for verification testability, Proceedings of the conference on European design automation (EDAC), Glasgow, UK, (1990) pp: 644–648.

KRISHNAMURTI, S., Fidler, K., Greenberg, M., Verifying aspect advice modularity, ACM SIGSoft International Symposium on the Foundations of Software Engineering, Newport Beach, California, ABD (2004) pp:137-146.

MAGEE, J., Kramer, J., Concurrency: State Model and Java Programs, John Wiley & Sons, Inc. New York, ABD, (1999).

MEHLITZ, P., Penix, J., Design for verification using design patterns to build reliable systems, Proceedings of 6th Workshop on Component-Based Software Engineering, Portland, Oregon, ABD, (2003).

MEYER, B., Applying 'Design by Contract', *IEEE Computer*, 25(10), 40-51, (1992).

ÖZGÜR, A. , A practical approach for the verification of aspect reusability, (Tezsiz MS Bitirme projesi) Teknik rapor ODTÜ/II-TR-2008-14, Orta Doğu Teknik Üniversitesi (2008).

ÖZGÜR, E. , Automated aspect code generation for the verification of ESC/Java annotated programs", (Tezsiz MS Bitirme projesi) Teknik rapor ODTÜ/II-TR-2008-15, Orta Doğu Teknik Üniversitesi, (2008).

SFORZA, F., Battu, L., Brunelli, M., Castelnovo, A., Magnaghi, M., A design for verification methodology, International Symposium on Quality Electronic Design (ISQED), San Jose, California, ABD (2001) pp:50-55.

SHARYGINA, N., Browne, J. C., Kurshan, R. P., A formal object-oriented analysis for software reliability: Design for verification, Proceedings of the 4th International Conference on Fundamental Approaches to Software Engineering (FASE), Genova, Italya, (2001) pp:318-332.

ŞİMŞEK, Ö. , Failure Handling in Distributed Systems, (Tezsiz MS Bitirme projesi) Teknik rapor ODTÜ/II-TR-2007-13, Orta Doğu Teknik Üniversitesi, (2007).

TOLUBAEVA, M., Betin Can A., Automatic Detection of Shared Objects in Multithreaded Java Programs, International Conference on Innovation in Software Engineering (2008) kitapçığında yayımlanmak üzere kabul edildi.

UBAYASHI, N., Tamai, T., Aspect-Oriented Programming with Model Checking, Proceedings of the 1st International conference on Aspect-oriented software development, Enschede, Netherlands, (2002) pp:148-154.

UZUNOĞLU, E. , Translation of Aspect Behaviors to PROMELA Model for Automated Verification, (Tezsiz MS Bitirme projesi) Teknik rapor ODTÜ/II-TR-2008-16, Orta Doğu Teknik Üniversitesi, (2008).

VISSER, W., Havelund, K., Brat, G., Park, S., Model Checking Programs, *Automated Software Engineering Journal*, 10(2), 203-232, (2003).

WESTON, N., Taiani, F., Rashid, A., Modular Aspect Verification for Safer Aspect-Based Evolution, 2nd ECOOP Workshop on Reflection, AOP and Meta-Data for Software Evolution, Glasgow, Scotland, (2005).

WOLPER, P., Expressing interesting properties of programs in propositional temporal logic, Proceedings of the 13th Annual ACM Symposium on Principles of Programming Languages (POPL), St. Petersburg Beach, Florida, ABD (1986) pp: 184-193.

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

Proje No: 106E032
Proje Başlığı: Koşutzamanlı ve Yaklaşım Dayalı Programlar için Otomatik Doğrulamaya Yönelik Tasarım
Proje Yürütücüsü ve Araştırmacılar: Dr. Aysu Betin Can Araştırmacılar: Munara Tolubaeva, Mustafa İspir
Projenin Yürütüldüğü Kuruluş ve Adresi: Orta Doğu Teknik Üniversitesi, İnönü Bulvarı, 06531, Ankara
Destekleyen Kuruluş(ların) Adı ve Adresi: Türkiye Bilimsel ve Teknolojik Araştırma Kurumu(TÜBİTAK) Tunus Caddesi No:80 06100 Kavaklıdere / Ankara
Projenin Başlangıç ve Bitiş Tarihleri: 01.07.2006 – 01.10.2008
Öz (en çok 70 kelime) Bu projede yakın zamanda popülerliği artan yaklaşıma/ilgiye dayalı programlar ile koşutzamanlı programlar üzerinde çalışılmıştır. Yaklaşıma dayalı programlar için arayüz tabanlı modüler otomatik doğrulama teknikleri geliştirilmiş ve bir doğrulamaya yönelik tasarım çerçevesi oluşturma yönüne gidilmiştir. Buna paralel olarak, koşutzamanlı programların doğrulamaya yönelik tasarım yöntemi içerisindeki otomasyonu için gerekli gizli kalan ortak nesnelerin otomatik bulan bir araç geliştirilmiş, ayrıca iş parçacıklarının yalıtımı için otomatik veri bağımsızlığı analizi ve otomatik yalıtım metotlarına entegrasyonu gerçekleştirilmiştir.
Anahtar Kelimeler: Otomatik doğrulama, koşutzamanlı programlama, yaklaşıma/ilgiye dayalı programlama (AOP), arayüzler, model denetimi
Projeden Yapılan Yayınlar: Aysu Betin-Can, Tefvik Bultan, Mikael Lindvall, Benjamin Lux, Stefan Topp. "Eliminating synchronization faults in air traffic control software via design for verification with concurrency controllers". Automated Software Engineering Journal. 14(2): 129-178 (2007) Mustafa İspir ve Aysu Betin Can. " An Assume Guarantee Verification Methodology for Aspect Oriented Programming", Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering (2008) sayfa 391-394 Munara Tolubaeva ve Aysu Betin Can. "Automatic Detection of Shared Objects in Multithreaded Java Programs", Proceedings of the International Conference on Innovation in Software Engineering (2008) kitapçığında yayımlanmak üzere kabul edildi. Mustafa İspir ve Aysu Betin Can "İlgilerin Birimsel Doğrulanması Üzerine Alan Araştırması", 2. Ulusal Yazılım Mühendisliği Konferansı Bildiri Kitapçığı, 2008

Munara Tolubaeva, "DoSSO: Automatic Detector of Shared Objects in Multithreaded Java Programs" yazım aşamasındadır.

Mustafa İspir ve Aysu Betin Can, "Modular Verification of AspectJ Programs". Bir dergiye gönderilmek üzere.

Özlem Şimşek, "Failure Handling in Distributed Systems", Teknik rapor ODTÜ/II-TR-2007-13, Orta Doğu Teknik Üniversitesi, 2007.

Altan Özgür, "A practical approach for the verification of aspect reusability", Teknik rapor ODTÜ/II-TR-2008-14, Orta Doğu Teknik Üniversitesi, 2008.

Ertan Özgür, "Automated aspect code generation for the verification of ESC/Java annotated programs", Teknik rapor ODTÜ/II-TR-2008-15, Orta Doğu Teknik Üniversitesi, 2008.

İrfan Nuri Karaca, "A Design Template for Development of Reusable Aspects ", Teknik rapor ODTÜ/II-TR-2008-6, Orta Doğu Teknik Üniversitesi, 2008.

Emre Uzunoğlu, "Translation of Aspect Behaviors to PROMELA Model for Automated Verification", Teknik rapor ODTÜ/II-TR-2008-16, Orta Doğu Teknik Üniversitesi, 2008.