Kosutzamanli ve Yaklasima Dayali Programlar için Otomatik Dogrulamaya Yönelik Tasarim

Proje No: 106E032

Yard. Doç.Dr. Aysu Betin Can

Munara Tolubaeva
Mustafa Ispir

ARALIK 2008
ANKARA

ÖNSÖZ

TÜBITAK tarafindan desteklenen “Kosutzamanli ve Yaklasima Dayali Programlar için Otomatik Dogrulamaya Yönelik Tasarim” baslikli bu projede yakin zamanda olusan ve popülerligi artan yaklasima/ilgiye dayali programlama ile dogrulanmasi zor ve ilgi çekici olan kosutzamanli programlama paradigmalari üzerinde çalisilmistir. Yaklasima dayali programlar için arayüz tabanli modüler otomatik dogrulama teknikleri gelistirilmis ve bir dogrulamaya yönelik tasarim çerçevesi olusturma yönüne gidilmistir. Buna paralel olarak, kosutzamanli programlarin dogrulamaya yönelik tasarim yöntemi içerisindeki otomasyonu için gerekli gizli kalan ortak nesnelerin otomatik bulan bir araç gelistirilmis, ayrica is parçaciklarinin yalitimi için otomatik veri bagimsizligi analizi ve otomatik yalitim metotlarina entegrasyonu gerçeklestirilmistir.

IÇINDEKILER

ÖNSÖZ.....1
ÖZET.....3
1. Giris.....5
2. Genel Bilgiler.....7
2.1. Yaklasima/Ilgiye Dayali Programlama için Dogrulama Teknikleri.....7
2.2. Kosutzamanli Programlama için Dogrulamaya Yönelik Tasarim.....8
3. Gereç ve Yöntem.....10
4. Bulgular.....13
5. Tartisma/Sonuç.....14
Kaynakça.....16

ÖZET

Bu projede yaklasima dayali programlar için otomatik modüler geçerleme yöntemleri gelistirilmis ve kosutzamanli programlar için dogrulamaya yönelik tasarim yöntemi eksiklikleri üzerine çalisilmistir. Çalismanin her iki yönünde de biçimsel metotlara yabanci programcilarin güvenilir yazilim gelistirmelerini saglamak ve yazilim gelistirme ile otomatik yazilim dogrulama arasinda bir köprü olusturmaya odaklanilmistir.

Yaklasimlar, çapraz kesen olgulari bir araya getiren ve temel yazilimin davranisini üzerinde büyük etkisi olabilen bir modülerlik birimidir. Gelistirdigimiz teknikler yaklasimlarin istenen etkiyi sagladiklarini ve temel programin ise yaklasimin varsayimlarini yerine getirdigini ayrik olarak geçerlenmektedir. Ayrica var olan bir yaklasimin bulundugu sistemden çikarilarak yeni sistemde kullaniminda gerekli kosullarin çikarilmasi ve sinanmasi üzerine de çalisma yapilmistir. Sözü edilen teknikler popular bir yaklasim dili olan AspectJ ile yazilmis programlar üzerine uygulanmistir.

Gerçeklestirimi ve geçerlenmesi zor olan kosutzamanli programlar üzerine yapilan çalismada is parçaciklarinin ortak kullandigi ve korunmasi gereken nesnelerin otomatik bulunmasi için bir araç gelistirilmis; is parçaciklarinin yalitimi için otomatik veri bagimsizligi analizi ve otomatik yalitim metotlarina entegrasyonu gerçeklestirilmistir. Bu iki çalisma kosutzamanli programlarin dogrulamaya yönelik tasarim çerçevesine entegre edilmistir.

Anahtar kelimeler: Otomatik dogrulama, kosutzamanli programlama, yaklasima/ilgiye dayali 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. Giris

Yazilim sistemlerinin günümüz modern teknolojilerinde yaygin olarak kullanimi, yazilimlarin güvenilirliklerini (güvenlik, kesintisiz çalisma ve dogrulugu gibi) teyit ve garanti edecek tekniklerin gereksinimini öne çikartmistir. Yazilimin güvenilirligi özellikle görevsel olarak kritik sistemlerde önemli bir sorun ve bilesendir. Bu güvenilirligi teyit etmek üzere gelistirilen model denetimi kullanan otomatik dogrulama teknikleri günümüzde oldukça önemli ilerlemeler kaydetmistir. Ancak tam otomatik dogrulama teknikleri ölçeklenilirligi yeterli olmayip, ölçeklenebilir teknikler ise oldukça fazla insan müdahalesine gereksinim duymaktir. *Dogrulamaya yönelik tasarim* (DYT) adi verilen, otomatik dogrulamaya uygun yazilim gelistirme yollarini arastirmadan çikacak sonuçlar bu problemlerin çözümünde etkili ve yararli olacaktir.

Bu projede DYT yöntemi iki farkli programlama paradigmasi üzerinde uygulanmistir. Son yillarda ortaya çikan ve popülerligi artan yaklasima/ilgiye dayali programlama ile dogrulanmasi zor ve ilgi çekici olan kosutzamanli programlama paradigmalari üzerinde çalisilmistir. Yaklasima dayali programlar için arayüz tabanli modüler geçerleme teknikleri gelistirilmis ve bir dogrulamaya yönelik tasarim çerçevesi olusturma yönüne gidilmistir (Raporun geri kalaninda *yaklasim* yerine daha dogru bir terim olan *ilgi* terimi kullanilacaktir). Buna paralel olarak, kosutzamanli programlarin DYT yöntemi içerisindeki otomasyonu için gereken eksikliklerin giderilmesi üzerine çalisma yapilmistir. Her iki kisimda da projenin ikincil amaci olarak yazilim gelistirme ve otomatik yazilim dogrulama arasinda bir köprü olusturmaya yogunlasilmistir. Biçimsel/formal metotlara yabanci programcilarin güvenilir yazilim gelistirmelerinde yardimci olacak sistemler amaçlanmistir.

Ilgiye dayali programlar için gelistirilen geçerleme yöntemi bir varsayim-garanti yöntemi olup iki asamadan olusmaktadir. Kullanicinin kuramsal modellerden ve modelleme-geçerleme detaylarindan soyutlanmasi için teknigimiz ilgi tanimlarini, ifade sekli karmasik olmayan arayüzler ile birlikte verilmesini gerektirmektedir. Bir ilgi arayüzü durum makineleri ve ön/art kosullardan olusmaktadir. Bu arayüz verilen ilginin en genis ortamini belirtmektedir. Verilen bir ilgi ve arayüzünden kapali bir ilgi modeli olusturma algoritmasi gelistirilmis ve gerçeklestirdigimiz bir yazilim sayesinde otomatik olusturulan bu model bir sonlu durum model denetleyicisine verilerek ilginin davranisi sorgulanmaktadir. Bu yöntemin matematiksel modelleri hazirlanmistir. Geçerlemenin ikinci asamasinda ise verilen herhangi bir programin ilgiye uyumlulugu sorgulanmaktadir. Bu asamada ilgi arayüzleri kullanilarak sinayici ilgiler olusturularak ve direk program üzerinden uyumluluk kontrolü otomatik olarak yapilmaktadir. Özetle kullanici sadece arayüz ve ilgi tanimi yaptiktan sonra gelistirdigimiz teknikler ile dokunmus yazilimin istenen ilgi davranisini gösterdigini geçerleyebilmektedir.

Yukarida açiklanan teknigin kullanici tarafini kolaylastirabilmek için yöntemin girdilerini sistemimize verebilecek bir yazilim sablonu (idiom) olusturulmustur. Bu sablonlar sayesinde kullanici ilgi kodunu yazarken ayni zamanda arayüzünü de belirtmis olmaktadir. Ilerleyen çalismalarda sablondan çok tasarim kalibi gelistirme arzu etmekteyiz.

Ilgiler üzerine yapilan bir diger çalisma ise ilgilerin tekrar kullanimi üzerine olmustur. Gelistirdigimiz bir yari otomatik araç ile verilen dokunmus bir ilgi kodundan ön/art kosullar ve degismezler otomatik olarak elde edilmekte ve ilginin tekrar kullanilacagi programi sinamak üzere denetleyici ilgiler üretilmektedir.

Kosutzamanli programlar için daha önce gelistirilmis olan DYT çerçevesindeki iki eksiklik giderilmistir. Bunlarin birincisi olarak gizli kalan ortak nesnelerin otomatik bulan bir yazilim gelistirilmistir. Bu araç için kaçis analizi (escape analysis) teknikleri irdelenmistir (Choi vd., 1999). Kaçis analizi tekniklerinin amaci bir uygulama eriminden (scope) disari kaçan nesneleri belirlemektir. Ancak her kaçan nesne ortak kullanilmayabilir. Örnegin bir nesne birçok is parçacigi tarafindan okunup hiç bir zaman içerigini degistirilmeyebilir. Bu çesit nesnelere somut örnek olarak global sabit nesneler verilebilir. Diger bir grup örnek ise bir çok is parçacigi tarafindan görülebilmesine ragmen hiç bir zaman kullanilmayan nesnelerdir. Bu örneklerde oldugu gibi gerçekte ortak olarak kullanilmayan ancak kendini yaratan is parçaciklarindan kaçan nesnelerin ayiklanmasi gerekmektedir. Aksi takdirde, mevcut kaçis analizi araçlarinda (Bogda 2001; Indus) oldugu gibi, pek çok nesne ortak kullanim içerisinde belirlenecektir. Bu projede gerçeklestirilen yazilimda bu sorunlari ele alinarak genel kaçis analizi teknikleri nden farkli bir yol izlenmis ve dogrulama çerçeve içerisine yerlestirilmistir.

Projenin katkida bulundugu diger sorun ise is parçaciklarinin yalitimidir. Ölçeklenebilirlilik için her is parçaciginin yalitilmasi büyük önem tasimaktadir. Bu izolasyon parçaciklarin ortamini olusturma problemi içine girer. Var olan çerçeve, diger tekniklerin aksine, senkronizasyon bilesenlerinin ortamini olusturma sorunu kaldirsa da is parçaciklarinin otomatik izolasyonu yönünden gelismelere açiktir. Bu konuda özellikle otomatik veri bagimsizligi analizi gereklidir. Veri bagimsizligi analizi Wolper tarafindan modellenmistir (Wolper, 1986) ancak bu analiz amacimizin isteklerini karsilamamaktadir. Projede gerçeklestirilen yeni bir analiz ile senkronizasyonu etkilemeyecek parametreler ve/veya parametre degerler bulmakta ve bu eksikligi doldurmaktadir. Ortamin modellenmesinde bu parametrelere sabit degerler verilerek durum uzayi indirgenmesi önemlidir. Bu analiz için gerekli algoritma gelistirilmis ve otomatik ortam olusturma araçlarina entegre edilmistir.

Proje süresince bir adet makale uluslararasi hakemli dergide yayimlanmis; uluslararasi hakemli konferanslarda bir adet kisa bildiri ve ulusal bir adet tam bildiri sunulmus ve yayimlanmis; bir adet bildiri uluslararasi hakemli konferansta sunulmak ve yayimlanmak üzere kabul edilmistir. Ayrica bu proje sonucunda bir tezli yüksek lisans tezi ve bes tezsiz yüksek lisans programi bitirme projesi olusturulmustur ve desteklenmistir. Bitirme projelerinde olusan teknik raporlar dergilere gönderilmek üzere hazirlanmaktadir. Halen yürütülmekte olan bir doktora tezi bu proje ile kismen desteklenmistir ve çalismanin ilk bulgulari bir makale olarak dergiye gönderilmek üzeredir.

2. Genel Bilgiler

Yazilim güvenilirligi teyit etmek üzere gelistirilen model denetimi kullanan otomatik dogrulama teknikleri günümüzde oldukça önemli ilerlemeler kaydetmistir. Model denetlemenin büyük ölçekli yazilim sistemlerinde etkili olabilmeleri, yazilimin yapisinin ve modelinin çikarilmasi ile orantilidir. Bu sebeple yazilimin yapisi hakkinda bilgi toplama isleminin, geriye dönük mühendislik (reverse engineering) teknikleri yerine, tasarim asamasinda yapilmasi daha verimli olacaktir. Tasarim kaliplari (design patterns) (Gamma vd., 1994) bu amaç için kullanilabilir. Bu kaliplarla tasarimi ve gerçeklestirimi yapilmis yazilim bilesenleri, hem yürütüm sirasinda görev alir hem de bilesenin modelini belirtir. Bu yöntem model çikarimi sorununa çözüm getirebilecektir. Ayrica tasarim kaliplarinin bilesenlerin davranissal arayüzlerini belirtmesi gerekmektedir. Bu arayüzler bilesenlerin kontratlaridir. Bu yöntem, kontrat tabanli tasarim (design by contract) (Meyer, 1992) kavraminin bir uygulamasi olarak düsünülebilir. Davranissal arayüzler sayesinde sistemin geri kalani modellenebilir. Bu sayede modüler denetleme için büyük olanak saglanmaktadir. Model denetleme tekniklerinin ölçeklendirilmesi için modülerlilik gereklidir.

Dogrulamaya yönelik tasarim (DYT) kavrami yazilim mühendisligi alaninda yeni olusmaktadir. Holzmann ve arkadaslari (Holzmann vd., 2008) model denetleme alaninin geleceginin programlari dogrulamaya uygun sekilde yapilandirmanin oldugunu belirtmektedirler. Bu yeni açilim DYT kavraminin gelecek arastirmadaki önemini vurgulamaktadir. Bu alandaki çalismalarin bazilari sunlardir. Mehlitz ve Phenix (Mehlitz ve Phenix, 2003) tasarim kaliplarinin kullanimiyla dogrulamanin verimliligini gelistirilmesini öne sürmüstür. Tasarim kaliplari büyük programlarda bagimsiz olarak dogrulanabilir bilesenlere ayirmada kullanilmistir. Bu çalisma henüz olgunlasmadigindan, kaliplar için kullanim kurallari olmasini öne sürmesine ragmen, bu kurallarin tanimlama sekli kesin olarak belirtilememistir. Bir diger çalisma, Sharygina ve ekibi (Sharygina vd., 2001)tarafindan DYT kavrami ile UML modellerinin dogrulanmasi üzerinedir. Adindan da anlasilabilecegi gibi, bu çalismada programin kodu göz önüne alinmamistir. Yazilimin yani sira, DYT kavrami donanim sistemleri ve devre tasariminda da ilgi görmüstür (Sforza vd. 2001; Krasniewski, 1990).

2.1. Yaklasima/Ilgiye Dayali Programlama için Dogrulama Teknikleri

Programlama paradigmalarina yeni eklenen *ilgiye dayali programlama* (aspect-oriented programming) (Kiczales, 1997), yazilimlara yeni bir tür modülerlik eklemistir. Bu yeni paradigma ile programcilar, *ilgi* (aspect) olarak adlandirilan, nesneleri çapraz kesen olgulari (crosscutting concerns) yazilimin diger tüm yapisal olgularindan ayrik olarak ele almaktadirlar. Örnegin, nesneye dayali programlarda kayit tutma yazilim içinde bir çok nesnenin içine yayilirken, ilgiye dayali programlamada kayit tutma bir ilgi olarak diger nesnelerden ayri gerçeklestirilir. Bir baska ilgi örnegi ise kosutzamanli programlarda senkronizasyon kurallaridir.

Bu yeni paradigmanin popülerlilik kazanmasi nedeniyle otomatik dogrulama alaninda ilgiye dayali programlarin güvenilirligini (reliability) teyit edecek yeni teknik ve araçlara ihtiyaç dogmustur. Ilgilerin getirdigi ifade gücü, potansiyel hata sayisini arttirmistir. Bu yönde çesitli çalismalar yürütülmektedir. Krishnamurthi ve ekibi, ilgileri önceden belirtilmis olan kesim noktalarina göre dogrulayabilen modüler bir yöntem gelistirmistir (Krishnamurthi vd., 2004). Ancak burada hatasiz gerçeklestirimi güç olan kosutzaman ilgileri açik birakilmistir. Denaro ve Monga ilgileri ortamlarindan izole bir biçimde denetlemektedir (Denaro ve Monga, 2001). Ancak dogrulama için model çikarimi otomatik olmayip, model ve programin arasinda uyumsuzluga meyillidir. Ayrica sadece ilgilerin degil, yeni hatalara sebep olabilecek olan, bu ilgilerin dokunmalari (weaving) ve ana programa etkisi de incelenmelidir. Ilgilerin hiç birisinde hata olmadigi halde, dokunmalari sirasinda etkilesimler nedeniyle yeni hatalar ortaya çikabilmektedir (Ubayashi ve Tamai, 2002). Bu sebeple dokunmus kodun otomatik dogrulanmasi için Ubayashi ve Tamai, dogruluk kriterlerini birer ilgi olarak kaynak koda eklemis ve Java PathFinder (Visser vd., 2003) ile otomatik incelemistir. Ancak bu çalismanin ölçeklenebilirligi süphe götürmektedir. Durum uzayi böyle bir model denetleyicisinin kolayca kapasitesini asacak büyüklüktedir. Weston ve ekibi kontrol akis ve veri analizine dayali bir modüler dogrulama yöntemi sunmustur (Weston vd., 2005). Bu yöntem ile ancak sabit kesim noktalarina göre analiz yapilabilmektedir. Dokumanin nasil olacaginin önceden bilinmesi gerekmektedir.

Çalismamizin bu bölümünde ilgilerin model denetleme teknikleri ile dogrulanmasi için yöntemler ve bir kuramsal model olusturulmustur. Ölçeklenebilirligi saglamak açisindan arayüz tabanli bir varsayim-garanti çerçevesi gelistirilmistir. Ilgilerin kuramsal olarak dogrulanabilecek sekilde adaptasyonunu ve tekrar kullanimini saglayan yöntemler arastirilmis ve bu konuda bilgi birikimi elde edilmistir. Çalisma sirasinda yasanan zorluk ise gerçek hayatta kullanilan yeterli büyüklükte bir ilgiye dayali program bulmak olmustur. Bu sorun ilgilerin nispeten yeni olmasindan kaynaklanmaktadir. Bu konuda kisa dönemli olarak dört yüksek lisans ögrencisinin bitirme projeleri tamamlanmis (Uzunoglu, 2008) (Karaca,2008) (Özgür, E, 2008) (Özgür, A, 2008); bir doktora ögrencisi ile yapilan çalisma uluslararasi bir konferansta kisa bildiri olarak (Ispir ve BetinCan, ASE 2008) ve bir ulussal konferansta bildiri olarak (Ispir ve BetinCan, UYMK 2008) sunulmus ve yayimlanmistir. Doktora ögrencisi su anda konu üzerine çalismalarina devam etmekte ve bir dergi yayini hazirlama asamasindadir.

2.2. Kosutzamanli Programlama için Dogrulamaya Yönelik Tasarim

Kosutzamanli programlarin yürütümü sirasinda ortaya çikan hatalarin çogunlugu senkronizasyon birimlerinin (Java'da bulunan Hoare monitörleri gibi) yanlis kullanimindan kaynaklanmaktadir. Sinama teknikleri gibi geleneksel geçerlilik yöntemleri bu tarz programlarda yetersiz kalmaktadir. Bu yetersizlik, durum uzayinin eszamanli çalisan islemlerin sayisi ile üstel olarak artmasindan ileri gelmektedir. Buna karsin, yakin zamanda kosutzamanli programlar için model denetlemeye dayali

otomatik yazilim dogrulama teknikleri önemli asamalar kaydetmistir (Corbett vd., 2000; Chaki vd., 2003; Dwyer vd. 2001; Alur vd., 1998, Magee ve Kramer, 1999). Bu çalismalarda ölçekliligi artirabilmek için çesitli modüler denetleme teknikleri ve kismi-sira indirgeme teknikleri kullanilmistir. Bu teknikler durum uzayini indirgese de ölçeklenebilirlik sorunu asilamamistir.

Kosutzamanli programlama için dogrulamaya yönelik tasarim alaninda yakin zamanda çalismalar yapilmistir (Betin-Can ve Bultan 2004; Betin-Can vd., 2005). Bu çalismalarda ortak kullanilan nesneler ve bunlari korumakla görevli senkronizasyon kisimlari yazilim içerisindeki kullanici bilesenlerden ayristirilmistir. Senkronizasyon bilesenleri eylemlerle (action), bu bilesenlerin kullanim kurallari ise arayüzlerle tanimlanmasi için tasarim kaliplari gelistirilmistir. Böylelikle senkronizasyon hatalari ve kullanim hatalari ayri ayri modüler olarak otomatik denetlenebilmektedir Bu modülerlilikle, durum uzayi çok büyük olan bir program yerine, durum uzayi küçük bir çok bilesen denetlenmis olmaktadir.

Ancak bu çalismalarda açik kalan önemli bir kaç konu bulunmaktadir. Bunlardan en önemlisi, is parçaciklarinin (thread) ortak eristigi nesnelerin otomatik bulunmasidir. Su anki çerçevede ortak nesnelerin belirlenmesi tamamen programciya birakilmis bir durumdadir. Programci ortak nesneleri ve bunlara uygun senkronizasyonu tasarim asamasinda belirleyip, gerçeklestirimi kosutzaman denetim kalibi (*concurrency controller pattern)* ile yapmaktadir. Oysa yazilimda programcinin tasarim asamasinda göremedigi gizli ortak nesneler olabilmektedir ve var olan çerçevenin geçerliligi tüm ortak nesnelerin biliniyor olmasina dayanmaktadir. Proje kapsaminda bu sorun üzerine gidilmis ve verilen bir kosutzamanli programin kaynak kodu üzerinden ortak kullanimda olan ve erisimi sorun yaratacak olan nesneleri otomatik olarak raporlayan bir araç gelistirilmistir. Bu araç yazilimi bir yüksek lisans tezini olusturmakta ve bir uluslararasi konferansta sunulmak üzere kabul edilmistir (Tolubaeva ve BetinCan, 2008).

Sözü geçen DYT çörçevesinin diger açik kalan konusu ise is parçaciklarinin yalitimi bölümündedir. Yalitim sirasinda is parçaciklari arasi bilgi tasiyan parametrelerin alabilecegi degerlerin kisitlanmasi/soyutlanmasi çerçeve içerisinde ad-hoc olarak yapilmistir. Projede gerçeklestirilen yeni bir analiz ile senkronizasyonu etkilemeyecek parametreler ve/veya parametre degerler bulmakta ve bu eksikligi doldurmaktadir. Ortamin modellenmesinde bu parametrelere sabit degerler verilerek durum uzayi indirgenmesi önemlidir. Bu analiz için gerekli algoritma gelistirilmis ve otomatik ortam olusturma araçlarina entegre edilmistir. Bir uluslararasi dergi makalesi ile bulgular yayimlanmistir (BetinCan vd., 2007).

3. Gereç ve Yöntem

Projede gerçeklestirilenler asagida kisaca özetlenmistir

- Is parçaciklarinin yalitimi için bir veri bagimsizligi analizi olusturulmustur. Bu analiz duragan zaman analizi (static analysis) olup baglam (context) duyarli degildir. Analiz sirasinda program bagimlilik çizgesinin tümünün olusturulmasi yerine kontrol ve veri akislari aninda (on the fly) hesaplanmistir. Analiz gerçeklestirilecek is parçacigi yalitimi yaziliminda sürücü ve taklit nesneleri olusturulmasina entegre edilmistir. Olusan sistem bir hava trafigi kontrolü yazilimi üzerinde sinanmis ve bulgular Automated Software Engineering Journal derigisinde yayimlanmistir. Analiz hakkinda detayli bilgi bu bildirinin 5.3 bölümünde yer almaktadir.

- Dagitimli sistemler için bir DYT uygulamasi gelistirilmis, özellikle bozukluk isleme (failure handling) için bir tasarim önerilmistir (Simsek, 2007). Bu çalisma bir Özlem Simsek'in tezsiz yüksek lisans bitirme projesini olusturmustur.

- Kosutzamanli Java programlari için ortak kullanim nesnelerinin otomatik belirlenmesi için kaynak kod üzerinde analiz yapan bir araç gelistirilmistir. Aracimiz diger yöntemlerin aksine her is parçaciginin eriminden (scope) disari çikan nesneyi raporlamamaktadir. Bunlari sadece potansiyel ortak nesne olarak görmektedir ve eger bir potansiyel ortak nesnenin durumu herhangi bir is parçacigi tarafindan degistirilirse aracimiz bu nesneyi ortak kullanimda olarak göstermektedir. Örnegin sadece sorgu metotlari çagrilan bir nesne aracimiz tarafindan ortak kullanimda olarak gösterilmez. Bu açidan diger kaçis analizlerinden farkli olup daha çok programciya yardimci olmaya yöneliktir. Aracimiz sayesinde programci senkronizasyonu düsünmeden yazilimini yapabilir, daha sonra aracimizin buldugu ortak nesneleri uygun bir senkronizasyon ile koruyabilir. Bu sekilde zor olan senkronizasyon gerçeklestirimi en sona birakilir ve is mantigi üzerine yogunlasilir. Ayrica daha önce adi geçen DYT yöntemi bu son asamada oldukça verimli çalisacak ve güvenilir bir yazilim ortaya çikacaktir. Aracimizin bir önemli özelligi ise uzek metot çagirimi (RMI) ve kullanici arayüzleri (GUI) için de ortak nesneleri bulabilmesidir. Analiz baglam duyarli olup tepeden asagi ve alttan yukari iki yöntemi birlestirmektedir. Sinif yapisindan dolayi gizli kalan alan nesnelerinin kaçis bilgisini de bulabilmektedir. Bu analiz ve araç olarak gerçeklestirimi Munara Tolubaeva'nin yüksek lisans tezini olusturmaktadir. Detayli bilgiler uluslararasi bir konferansta sunulmak ve yayimlanmak üzere kabul edilen bildiride bulunmaktadir.

- Ilgilerin dogrulanmasi için bir varsayim-garanti metodolojisi ve ilgilerin birimsel dogrulanmasi üzerine alan arastirmasi çalismalari Mustafa Ispir adli doktora ögrencisi ile yapilmis ve bulgular ulusal ve uluslararasi birer bildiri ile yayimlanmistir. Bu çalisma Ispir'in doktora tezinin temellerini olusturmaktadir.

Yapilan alan arastirmasi ile edinilen bilgiler çalismanin devaminda kullanilmaktadir. Sunulan dogrulama yöntemi iki asamadan olusmaktadir. Kullanicinin kuramsal modellerden ve modelleme-geçerleme detaylarindan soyutlanmasi için teknigimiz ilgi tanimlarini, ifade sekli karmasik olmayan arayüzler ile birlikte verilmesini gerektirmektedir. Bir ilgi arayüzü durum makineleri ve ön/art kosullardan olusmaktadir. Bu arayüz sayesinde ilginin ortami en genis anlamda modellenmis olarak bir sonlu durum model denetleyicisi olan SPIN aracina verilerek ilginin davranisi sorgulanmaktadir. Bu yöntemin matematiksel modelleri hazirlanmistir. Ikinci asamada ise verilen herhangi bir programin ilgiye uyumlulugu sorgulanmaktadir. Bu asamada ilgi arayüzleri kullanilarak otomatik denetleyici ilgiler olusturularak ve direk program üzerinden uyumluluk kontrolü önerilmistir. Bu teknigin dogruluk (soundness) ispati üzerinde çalismalar devam etmekte ve bilgiler bir dergiye gönderilmek üzere düzenlenmektedir.

- Ilgi davranislarinin otomatik dogrulamasi için PROMELA modeline çevrimi otomasyonu gelistirilmistir. Bu araç popüler bir model denetleyicisi olan SPIN (Holzmann, 1997) kullanarak ilgilerin zamansal özelliklerini dogrulamaktadir. Gelistirilen yazilim, SPIN aracinin girdisi olacak bir PROMELA model çikarimi otomatik olarak yapmaktadir. Ilgiler için yeni bir model çikarimi algoritmasi olusturulmus ve Emre Uzunoglu tarafindan hayata geçirilmistir. Bu çalisma Uzunoglu'nun tezsiz yüksek lisans projesini olusturmustur. Ilgilerin aktive olmalari *channel* yapisi ile modellenmis ve her bir öneri için bir islem birimi olan *proctype* tanimlanmistir. Grubumuz bu çalismadaki araç ve teknik rapor, projedeki doktora ögrencisinin çerçevesi ile birlestirilerek bir dergi yayini hazirligi içerisindedir.

- Ilgilerin dogrulanmasina yönelik bir tasarim sablonu olusturmustur (Karaca, 2008). Bu sablon programcilarin bir ilgi gerçeklestirirken otomatik model çikarimini kolaylastirici bir program yapisi olusturmalarini saglamaktadir. Bu sekilde ilgi modülü denetleyici için gerekli bilgiler bu sablonun kullanimindan otomatik olarak elde edilmekte ve tasarim ile dogrulama arasinda bir köprü kurulmustur. Irfan Karaca tarafindan olusturulan sablon bu ögrencinin tezsiz yüksek lisans bitirme projesidir. Olusturulan sablon kullanicinin genisletecegi ana ilgi modülleri ve arayüz tanimlamak için gerekli sonlu durum makinasi yapisindan olusmaktadir. Bu sablon yukarida adi geçen Uzunoglu'nun aracina model bilgisi saglamak için kullanilabilmektedir.

- Ilgilerin yeniden kullaniminda davranis özelliklerinin korunmasi üzerine bir çalisma yapilmistir (Özgür,E 2008; Özgür,A 2008). Yapilan çalismada verilen bir ilgi kodundan ön/art kosullar ve degismezler otomatik olarak elde edilmekte ve ilginin tekrar kullanilacagi programi denemek üzere denetleyici ilgiler üretilmektedir. Bir programcinin daha önce çalismakta olan bir yazilimdaki bir ilgiyi tekrar kullanmak istediginde o ilginin yeni örülecegi sistem üzerine olan kosullari çikarip kontrol edebilecegi bir teknik sunuyoruz. Daha önceki yazilim bir prototip olabilir ve bu teknik ile programci test ettigi prototip ortamindaki ilgiyi asil yazilimina tasiyabilir. Gerçeklestirilen yöntem bu tasima sirasinda yardimci olacaktir. Yöntem, iki yüksek lisans ögrencisi tarafindan olusturulmus

ve yari otomatik bir araç gelistirilmistir. Bu denetlemeyi geçen yazilimlarda ilginin davranis özelliklerinin korunacagi öngörülmüstür. Bu çalismanin biçimsel ispati daha sonraki çalismalara birakilmistir. Yukarida açiklanan arayüz belirtimi ve ilginin varsayimlarini kullanicidan almak yerine, bu çalismada ana program üzerinden arayüz kontratlari çikarilmaktadir. Çalisma sonucu iki teknik rapor olarak sunulmustur. Proje sonucunda olussan teknik raporlar bir dergiye gönderilmek üzere düzenlenmektedir.

4. Bulgular

Proje bir kaç adimdan olustugundan her biri için su bulgular elde edilmistir. Kosutzamanli programlar için veri bagimsizligi analizinin ve is parçacigi yalitim yapilan durum çalismasinda oldukça önemli bir etkide bulunmustur. Durum uzayinin küçültülmesinde etkin dogrulama sonuçlari alinmasinda büyük katkilari görülmüstür. Ortak nesnelerin bulunmasi programci için bir kolaylik saglamakta ve programcinin gözden kaçirabilecegi senkronizasyon gereken alanlari bulmasinda yardimci olmaktadir.

Ilgiye dayali programlar üzerine yapilan çalismalarda bir otomatik dogrulamaya ihtiyaç duyuldugunu ve dolayisiyla kullaniminda çekincelerin bulundugunu gözlemledik. Bu sebep ile durum çalismasi için gerçek hayatta kullanilan yeterli büyüklükte ve karmasiklikta bir ilgiye dayali program bulmak zorlugu yasanmistir. Ancak çalismalarimiz sonucu gelistirilen yöntemler bu güven eksikligini giderme açisindan etkili olacagini düsünmekteyiz. Dogrulama çalismalarinin ve araçlarinin kullanilabilmesi için programciyi biçimsel yöntemlerin detaylarindan soyutlayarak yazimi kolay arayüzler ve sablonlar sunmanin etkili olacagi ve bu alana yogunlasilmasi gerektigi sonucuna varmis bulunmaktayiz. Ilgilerin temel yazilim üzerindeki programcinin düsünemeyecegi etkileri bulmanin ve ya istenen etkiyi sagladigi göstermenin ilgiye dayali programlarin kullanim alanlarinin genisletecegini öngörmekteyiz.

5. Tartisma/Sonuç

Bu proje sirasinda Munara Tolubaeva adli yüksek lisans ögrencisi tam desteklenmistir. Proje sirasinda ögrenci bulma sorunlari yasansa da çesitli dönemlerde olmak üzere bes tezsiz yüksek lisans ögrencisi disaridan katkida bulunmuslardir. Son dönemde Mustafa Ispir adli doktora ögrencisi proje ekibine katilmis ve konferans sunumlari desteklenmistir. Kuramsal modellerin olusumunda Dicle Düvencioglu son dönem katkida bulunmus ve desteklenmistir. Bu aksakliklara ragmen projede asagida belirtilen yayimlar yapilmistir.

Bu proje sonucunda bir tezli yüksek lisans tezi ve bes tezsiz yüksek lisans programi bitirme projesi olusturulmustur ve desteklenmistir. Ayrica halen yürütülmekte olan bir doktora tezi bu proje ile kismen desteklenmistir.

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

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

4. Irfan Nuri Karaca, "A Design Template for Development of Reuasable Aspects ", Teknik rapor ODTÜ/II-TR-2008-6, Orta Dogu Teknik Üniversitesi, 2008.

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

6. Munara Tolubaeva, "DoSSO: Automatic Detector of Shared Objects in Multithreaded Java Programs" yazim asamasindadir.

Proje çerçevesinde, 1 adet makale uluslararasi hakemli dergide yayimlanmis, uluslararasi hakemli konferanslarda 1 adet kisa bildiri ve ulusal 1 adet tam bildiri sunulmus ve yayimlanmistir. Ayrica 1 adet bildiri uluslararasi hakemli konferansta sunulmak ve yayimlanmak üzere kabul edilmistir. 1 makale de dergilere gönderilmek üzere hazirlanmaktadir. Tüm bunlarla ilgili detayli bilgi asagida 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 Ispir 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 Ispir ve Aysu Betin Can “Ilgilerin Birimsel Dogrulanmasi Üzerine Alan Arastirmasi”, 2. Ulusal Yazilim Mühendisligi Konferansi Bildiri Kitapçigi, 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) kitapçiginda yayimlanmak üzere kabul edildi.

5. Mustafa Ispir ve Aysu Betin Can, “Modular Verification of AspectJ Programs”. Bir dergiye gönderilmek üzere hazirlanmaktadir.

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.

BETIN-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.

BETIN-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.

BETIN-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), Portlan, 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 (IWPSE),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.

ISPIR, 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.

ISPIR, M., Betin Can A., Ilgilerin Birimsel Dogrulanmasi Üzerine Alan Arastirmasi, 2. Ulusal Yazilim Mühendisligi Konferansi Bildiri Kitapçigi, Izmir, Türkiye (2008).

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

SFORZA, F., Battu, L., Brunelli, M., Castelnuovo, 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.

SIMSEK, Ö. , Failure Handling in Distributed Systems, (Tezsiz MS Bitirme projesi) Teknik rapor ODTÜ/II-TR-2007-13, Orta Dogu 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) kitapçiginda yayimlanmak 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.

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

TÜBITAK
PROJE ÖZET BILGI FORMU

Proje No: 106E032

Proje Basligi: Kosutzamanli ve Yaklasima Dayali Programlar için Otomatik Dogrulamaya Yönelik Tasarim

Proje Yürütücüsü ve Arastirmacilar: Dr. Aysu Betin Can

Arastirmacilar: Munara Tolubaeva, Mustafa Ispir

Projenin Yürütüldügü Kurulus ve Adresi:

Orta Dogu Teknik Üniversitesi, Inönü Bulvari, 06531, Ankara

Destekleyen Kurulus(larin) Adi ve Adresi:

Türkiye Bilimsel ve Teknolojik Arastirma Kurumu(TÜBITAK)

Tunus Caddesi No:80 06100 Kavaklidere / Ankara

Projenin Baslangiç ve Bitis Tarihleri: 01.07.2006 – 01.10.2008

Öz (en çok 70 kelime)

Bu projede yakin zamanda popülerligi artan yaklasima/ilgiye dayali programlar ile kosutzamanli programlar üzerinde çalisilmistir. Yaklasima dayali programlar için arayüz tabanli modüler otomatik dogrulama teknikleri gelistirilmis ve bir dogrulamaya yönelik tasarim çerçevesi olusturma yönüne gidilmistir. Buna paralel olarak, kosutzamanli programlarin dogrulamaya yönelik tasarim yöntemi içerisindeki otomasyonu için gerekli gizli kalan ortak nesnelerin otomatik bulan bir araç gelistirilmis, ayrica is parçaciklarinin yalitimi için otomatik veri bagimsizligi analizi ve otomatik yalitim metotlarina entegrasyonu gerçeklestirilmistir.

Anahtar Kelimeler:

Otomatik dogrulama, kosutzamanli programlama, yaklasima/ilgiye dayali programlama (AOP), arayüzler, model denetimi

Projeden Yapilan Yayinlar:

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)

Mustafa Ispir 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) kitapçiginda yayimlanmak üzere kabul edildi.

Mustafa Ispir ve Aysu Betin Can “Ilgilerin Birimsel Dogrulanmasi Üzerine Alan Arastirmasi”, 2. Ulusal Yazilim Mühendisligi Konferansi Bildiri Kitapçigi, 2008

Özlem Simsek, “Failure Handling in Distributed Systems”, Teknik rapor ODTÜ/II-TR-2007-13, Orta Dogu Teknik Üniversitesi, 2007.

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

Irfan Nuri Karaca, “A Design Template for Development of Reuasable Aspects “, Teknik rapor ODTÜ/II-TR-2008-6, Orta Dogu Teknik Üniversitesi, 2008.

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