Lojistik Merkez Konteyner Terminal Yönetimi Yazılımı için Alana Özgü Dil Geliştirimi ve Kullanımı

Tuğkan Tuğlular[1] Necmi Şentuna[2] Ali Koray Tuğ[3]

Gökhan Dağhan[4] Ceyhun Güven[5]

[1]İzmir Yüksek Teknoloji Enstitüsü, Urla, İzmir
[2,3,4,5]BİMAR Bilgi İşlem Hizmetleri A.Ş., Urla, İzmir

[1]tugkantuglular@iyte.edu.tr [2]necmi.sentuna@bimar.com.tr
[3]koray.tug@bimar.com.tr [4]gokhan.daghan@bimar.com.tr
[5]ceyhun.guven@bimar.com.tr

Özet. Lojistik merkezler için değişik özelliklere sahip konteyner terminal yönetim yazılımları geliştirme sürecini verimli hale getirmek için alana özgü bir dil geliştirilmiş ve kullanılmıştır. Bu çalışmada, bir görsel modelleme aracı kullanılarak konteyner terminal yönetimine özgü bilgi birikimi görsel bir alan modeli şeklinde ifade edilmiş ve bu alan modelinden alana özgü görsel bir dil elde edilmiştir. Ortaya çıkan konteyner terminal yönetimi görsel dili ile hazırlanan gösterimler, şablon tabanlı üretim yönteminden yararlanarak test durumlarına ve işletilebilir kodlara otomatik olarak dönüştürülmüştür. Alana özgü bir dil aracılığı ile temeli aynı ama özellikleri değişiklik gösteren yazılımları geliştirirken elde edilen faydalar; senaryoların doğrulanması için kullanılacak test durumlarının otomatik üretilmesi, nesne modeli ile buna ait kodun otomatik üretilmesi, nesne modeline uygun veritabanı tablolarını oluşturacak kodun otomatik üretilmesi, nesne modeline uygun web sayfası gibi sunum katmanı kodunun otomatik üretilmesi, olarak sıralanabilir.

Anahtar Kelimeler: model güdümlü yazılım geliştirme, alana özgü modelleme, alana özgü dil.

1 Giriş

Model-güdümlü yazılım geliştirme yaklaşımı ile alana özgü dil (AÖD) geliştirimi ve kullanımı 1990'ların sonundan bu yana süregelmektedir. AÖD geliştirim ortamlarının oluşması ile birlikte alana özgü görsel dillerin geliştirimi kolaylaşmış ve alana özgü dillerin önemli olan özelliklerinden biri olan ifade etme yetkinliği artmıştır [1]. Bu sayede modellerin alan uzmanları tarafından kolay anlaşılması sağlanmış hatta bazı araçlar vasıtası ile alan uzmanları bu modelleri geliştirebilir hale gelmiştir.

Bir AÖD kavramlar ile uygulama arasındaki açıklığı daraltarak yazılım geliştirim süreçlerinde kullanım kolaylığını, güvenilirliği ve verimliliği arttırmayı hedefler [2]. AÖD ile gelen otomatik kod üretimi sayesinde yazılımcının işi kolaylaşır, yazılım

geliştirme süreci hızlanır ve ortaya çıkan yazılımın kalitesi artar. Ayrıca, yazılım bakım ve özelleşmiş tekrar üretim süreçleri de kolaylaşır ve daha etkin yürütülür. Bu çalışmalara; ev otomasyonu için alana özgü dil [3], süreç yönetimi için alana özgü dil [4] ve rapor yazımı için alana özgü dil [5] örnek verilebilir. Anılan faydalarına karşın bir AÖD'yi geliştirmek hem kurumsal vizyon ve destek hem de ekip uzmanlığı ve adanmışlığı ile mümkün olmaktadır. AÖD geliştirim sürecini destekleyen araçlar bulunmakla beraber AÖD geliştirimi konusunda ciddi boyutta deneyim bilgisi eksikliği çekilmektedir. Bu alanda yegane bulunan yayın 2004 yılında Luoma et al. [6] tarafından hazırlanmıştır. Bu bildiri ile de, lojistik merkez konteyner terminal yönetimi (LMKTY) yazılımı için alana özgü görsel bir dil geliştirilmesi ve kullanımı sürecinde elde edilen deneyimler ortaya konmuştur.

2 Yöntem

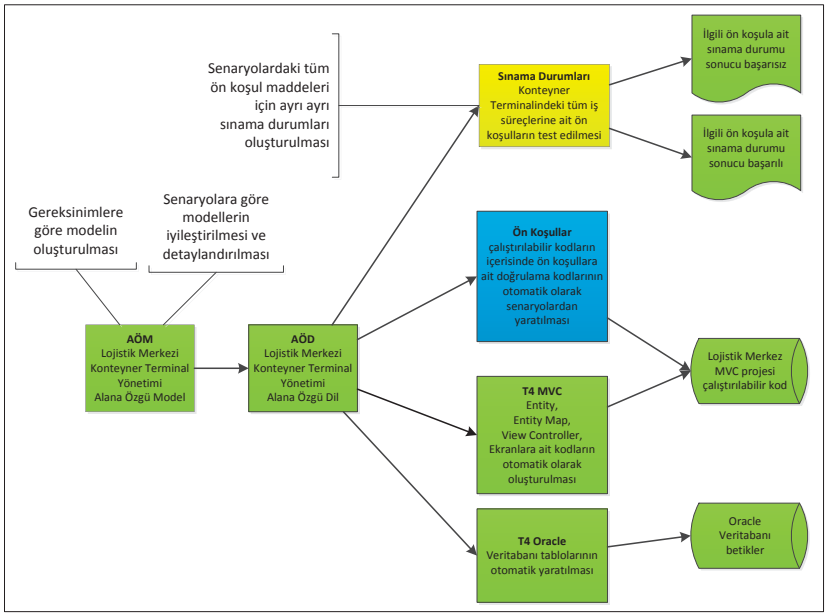
Bu çalışmada alan uzmanları ile birlikte aşağıda açıklanan adımlar izlenmiş ve sonucunda lojistik merkez konteyner terminal yönetimi yazılımı için alana özgü görsel bir dil geliştirilmiştir:

- Alan analizi ile alana özgü kavramlar ve aralarındaki ilişkilerin tanımlanması,
- Senaryo analizi ile kavramlar ve aralarındaki ilişkilerin gözden geçirilmesi, ayrıca kavramlara ilişkin özellikler ve senaryolara ilişkin koşulların belirlenmesi,
- Elde edilen bilgilerin bir görsel modelleme aracı kullanılarak bir alan modeli olarak ifade edilmesi,
- Alan modelinin bir dil tasarım aracı kullanılarak AÖD dönüştürülmesi,
- AÖD kullanılarak oluşturulacak sınama durumlarını ve kod parçalarını üretecek şablonların hazırlanması.

Lojistik merkez konteyner terminal yönetimi alana özgü dili ile otomatik sınama durumu ile otomatik kod üretim süreci Şekil 1.'de gösterilmiştir. Şekilde görüldüğü üzere, AÖD ile ifade edilen alana ait kavramlar ile bunların ilişkilerinin ana veri olarak nesne modeli kodunun, buna bağlı olarak veritabanı tablolarını oluşturacak betiklerin ve yine nesne modeline bağlı olarak sunum katmanı kodlarının ve senaryolar için sınama durumlarının otomatik olarak oluşturulması amaçlanmış ve sağlanmıştır.

Bu amaç doğrultusunda her alan modeli değişikliği ve alana özgü görsel dil geliştirilen her senaryo ile uyumlu çalışabilecek bir otomatik kod geliştirme yapısı oluşturulabilmesi için yansıtma (reflection) teknolojisinden faydalanılmıştır. Bu sayede alan modeli veya alana özgü görsel dil ile oluşturulan nesneler ve onların özelliklerinin (properties, attributes) otomatik olarak algılanıp kod üretiminde ve sınama durumu üretiminde gereken şekilde kullanılabilmesi sağlanmıştır. Yansıtma sırasında yeniden kullanılabilirlik konusu üzerinde hassasiyet ile durulmuştur. Bu sayede daha esnek bir yapıya sahip bir alana özgü dil ile ona ait üreteçler geliştirilmiş ve kullanılmıştır. Bu esnekliğe ek olarak, alan modeli temelinde kullanılması gereken sabit değer ve tanımlamalar–örn., otomatik oluşturulacak proje patikaları gibi tanımlamalar–kaynak (resource) dosyalarına eklenerek üreteçlerin gerekli bilgiyi

çalışma zamanı (runtime) esnasında bulması ve okuması sağlanmıştır. Böylece otomatik üretim sürecine yazılımcı müdahalesine gerek kalmamıştır.



Şekil 1. LMKTY Alana Özgü Model (AÖM) ve Alana Özgü Dil (AÖD) ile otomatik sınama durumu ile otomatik kod üretim süreci.

3 Kullanılan Araçlar

Lojistik merkez konteyner terminal yönetimi alana özgü dili geliştirilmesi ve kullanımı sürecinde yararlanılan araçlar aşağıda listelenmiş ve kısaca açıklanmıştır:

- Microsoft Visual Studio 2013 : kurumsal yazılım geliştirme ve süreç yönetim aracı.
- Microsoft Domain-Specific Language SDK 2013 : alana özgü modelleme ve alana özgü dil tasarlama aracı.
- Text Template (T4) : otomatik sınama durumu ve otomatik kod üretimi için gereken şablonların geliştirim aracı.
- Microsoft Entity Framework 6.0 : model-görünüm-denetleyici (model-view-controller) alt yapısında kullanılan veri erişim katmanı çerçevesi.
- Devart : Oracle veritabanı için varlık çerçevesi.
- Microsoft Visual Studio 2013 MVC4 : kurumsal mimaride kullanılan tepkisel (responsive) tasarım özellikli kullanıcı arayüzü geliştirme ortamı.

Bu araçlar alana özgü görsel dil ile yukarıda belirtilen kodların ve sınama durumlarının otomatik olarak üretilmesinde kullanılmıştır. Ancak bu araçların birlikte

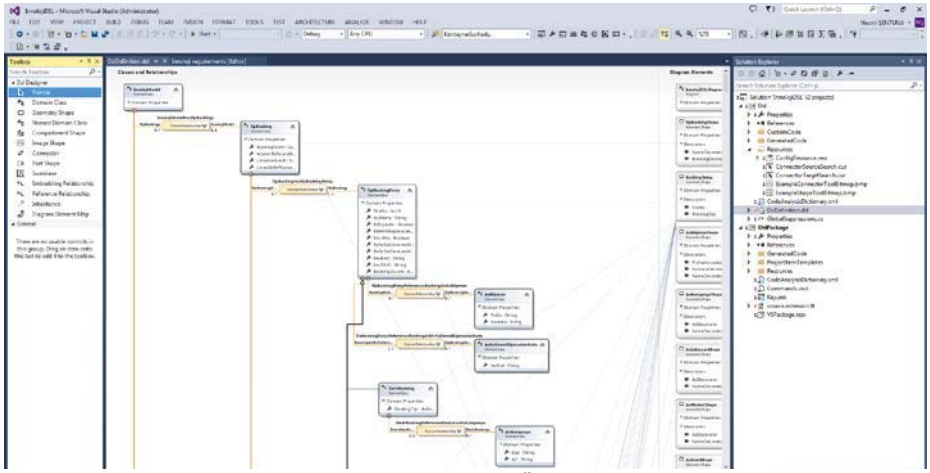
çalıştırılıp sonuç alınabilmesi için aşağıda listelenen iki tane yardım sınıfı firma bünyesinde geliştirilmiştir:

- ObjectHelpers : kod üretimi için ortaklaşa ve tekrar kullanılabilir metotlar sınıfı.
- ModelHelpers : sınama durumları için ortaklaşa ve tekrar kullanılabilir metotlar sınıfı.

Ayrıca, otomatik kod üretimi hedefli "entity", "entitymap" ve "viewmodel" kodlarını oluşturmak için 3 üç adet T4 şablonu yazılmıştır. Bunlara ek olarak, her sınama durumu için de birer T4 şablonu oluşturulmuştur. İzleyen bölümde bu araçların kullanımlarını da kapsayacak şekilde alana özgü görsel bir dil ile gerçekleştirilen otomatik üretime ait örnek bir çalışma açıklanmıştır.

4 Örnek Durum Çalışması

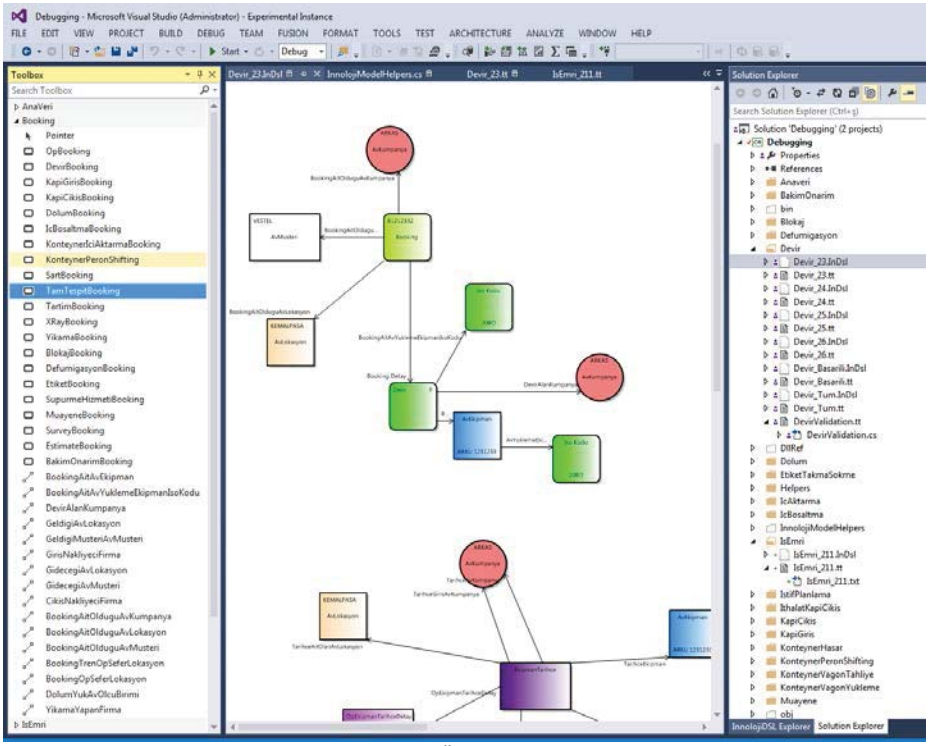
Alana özgü dil geliştirilirken Microsoft Domain-Specific Language SDK 2013 (MS DSL) aracının model ekranında Şekil 2'de görüldüğü gibi alana özgü kavramlar ve bunlar arasındaki ilişkiler yerleştirilmiştir. Diğer bir deyişle alana özgü model oluşturulmuştur. Tamamı şekilde gözükmemekle beraber 45 tane alan kavramı ve bunlar arasındaki 107 ilişki–bunların 7 tanesi kalıtım (inheritance) ve 36 tanesi de gömülü (embeded) ilişki olup– model ekranında bulunmaktadır. Bu ekranda ayrıca her kavram ile ilgili özellikler ve bu özelliklerin tipleri–ek olarak, zorunlu olup/olmadığı, saha uzunluğu v.b.–belirtilmiştir. Bu sayede nesne modeline ait kodun, veritabanı tablo betikleri ve sunum katmanı kodlarının otomatik olarak oluşturulması mümkün hale gelmiştir. Ayrıca, senaryo bazlı sınama durumlarının otomatik oluşturulabilmesi için, belirli bir senaryo içinde yer alan kavramların içinde bulunabilecekleri durum (state) bilgisinin de girilmesini zorunlu kılacak kod parçacıkları kavramlara eklenmiştir.



Şekil 2. MS DSL üzerinde LMKTY Alana Özgü Modeli Kısmi Görünümü.

Alana özgü modelden alana özgü dile dönüştürmenin yapılabilmesi için MS DSL aracında kavramların görsel varlıklar ile eşlendirilmesi gerekmektedir. Ancak MS DSL Designer aracının sunduğu görsel varlıklar sadece dikdörtgen ve daire türevleridir. Çalışmamız sonucunda kalıtım ve gömülü ilişkiler sebebi ile farklı 30 görsel imgeye ihtiyaç olduğu ortaya çıkmıştır. Farklı renkler ile görsel varlıklar, dolayısı ile onların temsil ettiği kavramlar ayrıştırılmıştır. Bir örneği Şekil 3'de verilmiştir.

MS DSL Designer aracı dönüştürme sonucunda standart olarak hiçbir özelliği göstermediği için görsel dili oluşturan varlıklar üzerinde görünmesi, hatta girilmesi istenen tüm özellikler alan modeli ekranında tanımlanmaktadır. Dönüşüm sonucunda ortaya çıkan görsel dil kullanıcıyı ilgili bilgileri doldurmaya zorlamaktadır. Bu sayede oluşturulan bir senaryo hem alan uzmanları ile incelenebilmekte hem de senaryo bazlı sınama durumlarını oluşturulabilmektedir.



Şekil 3. MS DSL üzerinde LMKTY Alana Özgü Dil ile Devir Senaryosu Kısmi Görünümü.

T4 şablonlarını oluşturmadan önce ortaya çıkması hedeflen kod parçaları MVC ve kurum standartları göz önüne alınarak oluşturuldu. Oluşturulan bu kod parçalarının hangi kısımlarının alana özgü dil ile oluşturulan görsel programdan geleceği belirlendi. Böylece T4 şablonlarını oluşturmanın mümkünlüğü gözlendiği gibi şablonları jenerik yapmanın da yolu bulunmuş oldu. T4 şablonları ile alana ilişkin nesne modeli kodu, veritabanı tablo oluşturan, okuyan/yazan ve güncelleyen kodlar ile sunum katmanı kodları ve senaryo sınama durumları otomatik olarak oluşturulabilir hale geldi. Kapı giriş senaryosu için kullanılan şablon örnek olarak Şekil 4'de verilmiştir.

Şekil 4. MS DSL üzerinde LMKTY için Hazırlanan Örnek bir T4 Şablonu.

Hem alana özgü sınamanın, hem de veri erişim ile sunum katmanı kod parçalarının AÖD'den otomatik olarak üretilmesi, yansıtma teknolojisinin uygulanması ile mümkün olmuştur. Oluşturulan AÖD kullanımı ile herhangi bir projenin yaratılmasında hem sunum katmanı hem de veri erişim katmanı (veritabanı ile birlikte) otomatik olarak hazır hale geldiği için projenin geliştirme zamanı kısalmıştır.

5 Sonuç

Bu bildiride, lojistik merkez konteyner terminal yönetimi yazılımı için alana özgü görsel bir dil geliştirilmesi ve kullanımı süreci açıklanmış ve elde edilen deneyimler paylaşılmıştır. Oluşturulan yöntemin işletilmesi sonucunda; müşteri senaryolarının hızlıca oluşturulduğu, bu senaryolara ilişkin sınama durumlarının alan uzmanları tarafından kolayca oluşturulduğu ve yazılımcılara sadece senaryo akış ile ilgili kısımın kodlanmasının kaldığı, ayrıca yazılımcıların otomatik üretilmiş kodlara müdahale etmesine gerek kalmadığı gözlemlenmiştir.

Kaynakça

1. Mernik, M., Heering, J. & Sloane, A. M. When and How to Develop Domain-specific Languages. ACM Comput. Surv. 37, 316–344 (2005).
2. Van Deursen, A., Klint, P. & Visser, J. Domain-Specific Languages: An Annotated Bibliography. Sigplan Not. 35, 26–36 (2000).
3. Jimenez, M., Rosique, F., Sanchez, P., Alvarez, B. & Iborra, A. Habitation: A Domain-Specific Language for Home Automation. Software, IEEE 26, 30–38 (2009).
4. Barzdins, J. et al. Domain specific languages for business process management: a case study. in Proc. DSM 2009 Work. OOPSLA 34–40 (2009).
5. Dantra, R., Grundy, J. & Hosking, J. A domain-specific visual language for report writing using Microsoft DSL tools. Vis. Lang. Human-Centric Comput. 2009. VL/HCC 2009. IEEE Symp. 15–22 (2009).
6. Luoma, J., Kelly, S. & Tolvanen, J.-P. Defining domain-specific modeling languages: Collected experiences. in 4th Work. Domain-Specific Model. (2004).