

Oracle Forms Arayüzlerinin HTML5 Arayüzlerine Otomatik Dönüştürümü

Hasan Ali Özdaş^{1,2}, Mehmet Şevki Ütebay^{1,3}, Özhan Düz^{1,4}, Rıdvan Salih Kuzu^{1,5}

1 OBSS Bilişim Bilgisayar Hiz. Dan. San. Tic. Ltd. Şti., İstanbul

2 Dokuz Eylül Üniversitesi, Bilgisayar Mühendisliği

3 Bilkent Üniversitesi, Bilgisayar Mühendisliği

4 Doğu Akdeniz Üniversitesi, Bilgisayar Mühendisliği

5 Boğaziçi Üniversitesi, Sistem ve Kontrol Mühendisliği

ali.ozdas@obss.com.tr,

mehmet.utebay@obss.com.tr

ozhan.duz@obss.com.tr

ridvan.kuzu@obss.com.tr

Özet: Bankacılık, sigortacılık ve resmi kurumlarda kullanılan uygulama arayüzleri için Oracle Forms arayüzlerinin uzun yıllardır öncelikli tercihlerden biri olduğu bilinmektedir. Fakat Forms artık eski bir teknoloji olduğu için ve 2017 yılından sonra Oracle tarafından desteklenmeyeceği için, bu uygulamaların minimum emek ve maliyetle daha modern arayüz teknolojilerine dönüştürülmesine yönelik arayışlar söz konusudur. Forms uygulamalarında ekranlar "FMB" dosyaları şeklinde tutulmaktadır. Oracle tarafından sağlanan bir araç olan "FRMF2XML" ile bu dosyaların XML formatına dönüştürülmesi mümkündür. Bu şekilde XML'e dönüşen Forms arayüzlerinden, bir dönüştürücü vasıtası ile HTML5 uyumlu bir kullanıcı arayüzünün üretilmesi için izlenmesi gereken süreçler bu çalışmada detaylandırılmıştır.

Anahtar Sözcükler: Kullanıcı Arayüzü, Otomatik Dönüşüm, HTML5, Oracle Forms, Migration

Automatic Conversion of Oracle Forms to HTML5 Interface

Abstract: In banking, insurance and official institutions, Oracle Forms interfaces are known as one of the first preferences in terms of developing user interface for many years. Nevertheless, Forms is going to become an obsolete technology and will not be supported by Oracle after 2017. For this reason, the migration of these applications to a more modern interface technology with minimum cost and labor is under consideration. Forms interfaces take shape from "FMB" files. A Tool named as "FRMF2XML" provided by Oracle can be used to transform "FMB" files into XML formatted files. In this way, XML formatted Forms interface files can be migrated to an interface compatible with HTML 5 with the assistance of a converter. This paper explains steps which should be followed during migration process in detail.

Keywords: User Interface, Automatic Conversion, HTML5, Oracle Forms, Migration

1. Giriş

Yazılım uygulama dönüşümleri, bilişim sektöründe sıklıkla karşılaşılan süreçlerden biridir. Organizasyonlar geliştirmiş oldukları uygulamaları bir yazılım dilinden diğerine, bir platformdan başka bir platforma ya da bir işletim sisteminden bir başkasına farklı sebeplerle aktarma ihtiyacı duyabilmektedir. Güncelliğini yitiren teknolojiyi terk etmek, bütünleşmiş kurumsal bilgi sistemleri oluşturmak, geleneksel sunucu-istemi mimarisinden, çok katmanlı mimarilere geçiş yapmak gibi örnekler bu sebepler arasında sayılabilir [1][2][3]. Bu tür dönüşümlerin geçmişteki somut örneklerinden bir kaçını listelersek:

- Uygulamayı bir yazılım geliştirme dilinin yeni sürümüne yükseltmek (COBOL 68 –COBOL 85 dönüşümü)
- Bir yazılım geliştirme dilinden diğerine geçiş yapmak (C# - Java dönüşümü)
- Bir dosya saklama sistemini, hiyerarşik ya da ilişkisel bir veritabanına taşımak (VSAM – DB2 dönüşümü)
- Bir uygulama *framework* altyapısından diğerine geçiş yapmak (Oracle Forms - .NET dönüşümü)

Dönüşüm esnasındaki kaynak ve hedef platformlar arasındaki semantik farklılık, dönüşüm işlemlerinin gerçekleştirilebilirlik derecesini belirleyen en önemli etmendir. Eğer çok büyük semantik farklılık söz konusu ise, karmaşıklığı yüksek ve daha az otomatik ve düşük verimli diyebileceğimiz bir dönüşüm mümkün olacaktır.

Bu bildirinin amacı, Oracle Forms'dan HTML 5'e otomatik dönüşüme olanak vermeyi amaçlayan FORMS2HTML5 adlı ticari amaçlı uygulamanın geliştirilmesi safhasında karşılaşılan zorlukları, tasarım kararlarını ve çözüm stratejilerini ortaya koymaktır.

Böyle bir dönüşümün hayata geçirilme amacı, son 20 yıldır kurumsal uygulamaların hızlı ve

kolay adımlarla gerçekleştirilmesi için yaygınlıkla kullanılan Forms'un, Oracle tarafından desteğinin kesileceğinin açıklanmış olmasıdır[4]. Bu bakımdan, mevcut kurumsal uygulamaların, standart web teknolojiler ve servise dayalı mimarilerin (SOA) avantajlarını kullanabilir duruma gelmeleri için bu dönüşümü mümkün olduğunca erken yapmaları gerekmektedir.

Farklı kurumsal altyapılarda, farklı ihtiyaçlara cevap vermekte olan Oracle Forms uygulamalarının tek tek manuel dönüşümünü yapmak yerine, bu bildiride teknik detayları açıklanan FORMS2HTML5'in, zaman ve maliyet avantajı sağlayacak otomatik bir dönüşüm çözümü olması hedeflenmektedir.

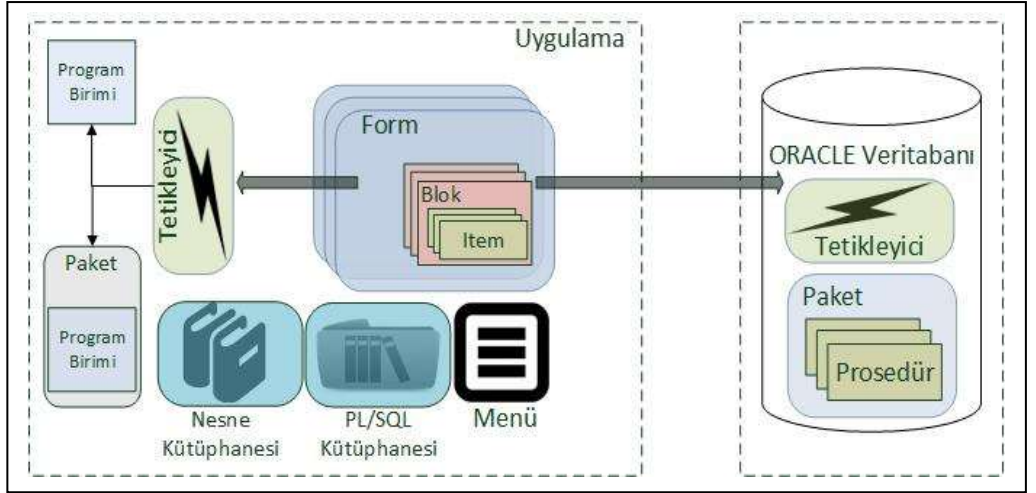
2. FORMS2HTML5 Genel Yapı

Oracle Forms bir 4GL hızlı veritabanı uygulamaları geliştirme altyapısı olduğu gibi, aynı zamanda bu veritabanı uygulamalarının koştuğu bir platformdur. Şekil 1'de bir Oracle Forms uygulamasının ana bileşenleri ve bunların birbiri ile ilişkisinin, uygulama geliştiriciler açısından nasıl görüldüğü özetlenmiştir [5].

Veritabanı, standart verilerin yanı sıra, tetikleyiciler (trigger), paketler (package) ve prosedürler gibi öğelere de sahip olmaktadır. Fakat bu öğeler, Oracle Forms uygulama dönüşümü kapsamı dışında tutulacaktır. Kullanıcı arayüzü açısından ise, *window*, *canvas*, *frame*, *block*, *item*, vb. gibi öğeler Oracle Forms uygulamalarında mevcut olduğu gibi, görsel olarak HTML 5'te de karşılığı bulunabilir durumdadır. Bu bakımdan dönüşüm operasyonunun öncelikli amacı, kullanıcı arayüzüne ait öğelerin Forms'dan HTML 5'e aktarılmasıdır.

2.1 Çözümleyici (Parser) Geliştirme Adımları

Parser geliştirme işleminde amaçlanan şey, Oracle Forms bünyesinde yer alan FRMF2XML aracıyla üretilmiş olan geçerli



Şekil 1: Uygulama Geliştirici Gözünden Oracle Forms Yapısı

XML dosyasının çözümlenip bir uygulama geliştirme arayüzü (API) olarak üretici (Generator) modülüne sunulmasıdır.

FRMF2XML aracıyla form modülleri, nesne kütüphaneleri ve menü modüllerinin bir XML dosyasına dönüşümü mümkün olmaktadır. Parser modülünde XML dosyasına dönüşümü yapılmış olan Forms modülleri çözümlenmektedir. Yapılan bu çözümleme sonucunda bir model ortaya çıkarılmakta ve sunulan API sayesinde bu modele erişim sağlanmaktadır.

Parser modülünde öncelikle girdi (input) olarak verilen XML dosyası çözümlenerek form modülde yer alan öğelerin model nesnelere oluşturulur. Bu model nesnelere oluşturulurken, öğelerin öncelikle standart olarak gelen özellikleri nesnelere atanır. Ardından XML dosyasının çözümlenmesiyle elde edilen öğelere ait özellik değerleri bu nesnelere atanır. Bu şekilde yol izlenmesinin sebebi XML dosyasında öğelerin alabileceği tüm özelliklere ait olan değerlerin yer almama ihtimalidir. Bu yüzden Oracle Forms ürünü incelenerek var olan tüm öğelere ait olan özellikler ve bunların başlangıç değerleri ortaya çıkarılmıştır. Bu özellikler çözümleyici modülünde JSON formatındaki dosyalar halinde yer almaktadır. Yukarıda da

belirtildiği gibi, öncelikle JSON dosyaları çözümlenir ve oluşturulan Forms öğe nesnelere bu dosyalar içerisindeki başlangıç bilgileri atanır. Ardından girdi XML dosyası çözümlenerek içerisinde yer alan Forms öğelerine ait olan özellik bilgileri başlangıç değerleriyle değiştirilir.

Oracle Forms'da bir öğe oluşturulurken var olan bir başka öğe referans gösterilerek oluşturulabilir [6]. Bu özelliğe dayanarak, Forms öğelerine ait olan özelliklerin model nesnelere atanmasının ardından eğer varsa bu öğelere ait olan referans öğe bilgileri de nesnelere atanmıştır.

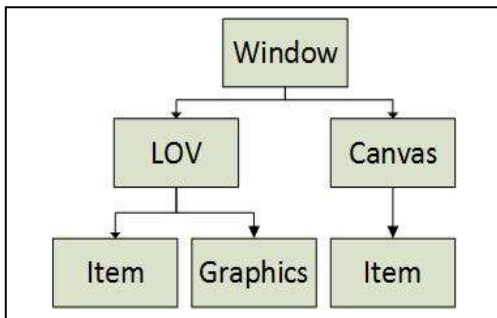
Referans değerlerinin atanmasının ardından Form modülünde yer alan *Item* öğelerinin "Number of Items Displayed" özelliği ve bu *Item* öğelerinin yer aldığı *Block* öğelerinin gösterilecek kayıt sayısı (number of records display) özelliklerine bakılarak yeni *Item* öğeleri oluşturulur ve modele eklenir.

Bir sonraki adımda Forms modülünde yer alan LOV(List of Items) öğeleri ele alınmıştır. LOV, içerisinde birden fazla sayıda kolondan oluşabilen bir tablonun sunulduğu bir *pop-up* penceresidir. Bu konuda LOV'ye ait olan SQL sorgusu (query) ve tabloda yer alan kolonlara ait olan bilgiler ve benzerleri,

XML dosyası çözümlenerek LOV nesnesi oluşturulur. Ardından LOV içerisinde yer alan alt öğelere ait nesnelere oluşturulur. Bu öğeler ile LOV öğesi arasında kullanıcı arayüzü (UI) açısından bir baba-oğul ilişkisi kurularak LOV nesnesi altından bu bilgi sunulur. LOV ile alakalı son olarak SQL sorgu (query) bilgisinde parametre olarak yer alabilecek olan Forms öğeleri ortaya çıkarılarak LOV nesnesine bu bilgi de eklenir.

LOV ile ilgili işlemlerin tamamlanmasının ardından dönüşüm sırasında ihtiyaç duyulan her bir Forms öğesine ait eşsiz bir kimlik bilgisi üretilir.

Forms modülünde yer alan öğelerin oluşturulmasının ardından bu öğeler arasında UI tarafından anlamlı olacak şekilde bir baba-oğul ilişkisi kurulur. Bu duruma şöyle bir örnek verebiliriz. Bir *Item* öğesi normalde bir *Block* öğesi altında yer almaktadır. Dolayısıyla bu iki öğe arasında mantıksal olarak bir baba-oğul ilişkisi bulunmaktadır. Ancak bu *Item* öğesi neticede bir *Canvas* öğesi üzerinde ekranda yer almaktadır. Bu durumda bu *Canvas* öğesi *Item* öğesini içerisinde barındırmaktadır. Yani bu iki öğe arasında da UI anlamında bakacak olursak bir baba-oğul ilişkisi kurulabilir. Bu ilişki çözümlayicinin oluşturduğu model altından üretici (generator) modülüne sunulur.



Şekil 2: UI Ağacı

Çözümlayici, aldığı XML girdi dosyasının sonucu olarak, üretici modülüne bir model

sunmaktadır. Bu model içerisinde, Şekil 2’te görüldüğü gibi, en üst öğe olarak (root) *Window* öğesi yer alır. LOV ve *Canvas* öğeleri, *Window* öğesinin çocukları olarak yer almaktadır. *Canvas* öğelerinin altından *TabPage*, *Item* ve *Graphics* öğelerine, LOV altından da ilgili *Item* öğelerine erişim sağlanmaktadır.

2.2 Üretici (Generator) Geliştirme Adımları

Üretici, çözümlayici tarafından oluşturulan nesne modelini girdi olarak kullanarak, çıktı olarak; HTML, CSS ve LOV’ler için XML ve Java kaynak kodu üretebilmektedir.

Üreticiye girdi olan nesnelere, çözümlayici tarafından atanan özellik ve değerlere sahiptir. Dolayısıyla, girdi olan bu nesnelere, Forms bileşenlerinin yapısına ve özelliklerine sahip durumdadır. Üretici ilk olarak girdi olan, Forms bileşeni yapısındaki nesnelere, ilgili HTML bileşeni ile eşleştirip, ihtiyaç duyulan yapıya geçirerek bir model kümesi oluşturmaktadır. Bu dönüşüm sırasında Forms bileşenlerinin özelliklerinden, HTML sayfada uygulanabilir olanlar süzülerek yeni nesnelere atanmaktadır. Dönüşüm sonucunda, Forms bileşenlerinin sahip olduğu hiyerarşi, yeni model kümesinde de aynen korunmaktadır.

Bu işlem sonucunda, her bir Forms bileşeni için, eşleniği olan HTML bileşeni yapısında bir nesne oluşmaktadır. HTML bileşeni yapısına uygun nesnelere her biri, Forms bileşeninden miras olarak alınan gerekli özellik ve değerlere ek olarak, HTML içeriğinin bulunduğu XML dosyalarına sahiptir.

HTML yapısına uygun olarak dönüştürülen nesnelere hiyerarşik olarak işlenmektedir. Bu aşamada her bir nesne, öz devinimli bir fonksiyon aracılığıyla, varsa önce ardıl nesnesi sonra kendisi işlenmek suretiyle oluşan çıktı, HTML ve stil detaylarını içeren CSS dosya içeriğine eklenmektedir. Tüm

nesneler bu işlemten geçtikten sonra HTML ve CSS dosyaları diske yazılarak üretici işlevini yerine getirerek çıktıyı üretmektedir.

İşlenen nesne eğer bir LOV nesnesi ise, yukarıdaki sürece ek olarak, XML dosyası olarak sorgu içeriği ve bu sorguyu çalıştırabilecek Java kaynak kodu dosyaları üretilmektedir. LOV nesnesi görsel olarak bir açılır pencereye eşleniktir. Bu doğrultudaki görsel üretimi esnasında, açılır pencereyi tetikleyen bileşenlerle LOV arasında ayrıca bağ kurulmaktadır. Devamında ise, ilk olarak LOV içeriğindeki SQL sorgusu için bir XML dosyası üretilir. Sonrasında, bu XML dosyasını girdi olarak alan ve içeriğindeki sorguyu JDBC API ile çalıştırarak sonucunu dönen bir fonksiyona sahip Java kaynak kodu üretilir.

Özetle üretici, çözümleyicinin sağlamış olduğu girdiyi kullanarak sırasıyla, HTML yapısına uygun nesnelere, bunların HTML ve CSS karşılıklarını, LOV'ler için ayrıca sorgu içeriği ve sonucu için XML ve Java kaynak kodu çıktılarını üretebilmektedir.

2.3 Sorgu Dönüşümleri

Sorgular Oracle Forms'da kayıt grubu (*Record Group*) ögesinin bir özelliği olarak yer almaktadır. Kayıt grubu ögeleri satır ve sütunlardan oluşan bir veri yapısıdır. Veritabanı tablolarına da benzetilmektedir. Ancak bundan farklı olarak kayıt grubu ögesi Forms modülüne bağlı olan bir nesnedir ve kayıtların kullanıcı tarafında saklanmasında başvurulmaktadır.

Kayıt grubu ögelerinde yer alan bu sorguların Oracle Forms dışarısında da kullanılabilmesi için çözümlenip dönüştürülmesine ihtiyaç vardır.

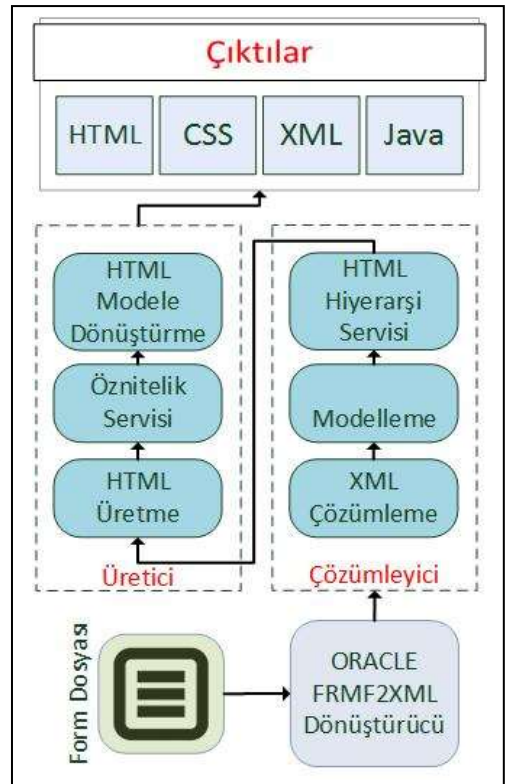
Sorguların dönüşümde dikkat edilen nokta içerisinde bind değişkeni (bind variable) içerip içermemesidir. Eğer sorgu içerisinde bu değişkenler kullanılmışsa, sorguların çözümlenmesi sırasında ortaya çıkarılarak

sorgunun kullanıldığı ögenin bir özelliği olarak değişkenin modeline eklenir.

Örnek vermek gerekirse, "SELECT * FROM table0 WHERE column0 = :block0.item0" şeklinde bir sorguyu ele alalım:

Sorgunun *Where* koşulunda "block0" isimli *Block* içerisinde bulunan "item0" isimli *Item* ögesi, bind değişkeni olarak kullanılmaktadır. Bu sayede uygulama koşturmadan önce değeri bilinmeyen ögeler, sorgu içerisinde kullanılabilir hale gelmektedir. Bind değişkenlerinin sorgu içinde kullanılmalarının bir diğer nedeni ise performans ve kaynak tüketimi konularında avantaj sağlamasıdır [7].

Yukarıdaki sorgunun bir LOV ögesi içerisinde kullanıldığını düşünürsek sorgunun çözümlenmesi sonucunda elde edilen bind değişken değerleri bu ögenin modeline eklenir. Böylece üretici (generator)



Sekil 3: FORMS2HTML5 Mimarisi

modülünde, bu öğeye erişilerek değeri sorguya eklenir ve sorgu koşulabilir duruma gelir.

3. Otomatik Dönüşüm Uygulaması

FORMS2HTML5 Genel Yapı başlığında detaylı bir şekilde süreç adımları açıklanan uygulamanın modüllerini kısaca özetleyecek olursak, Şekil 3’te de görüldüğü gibi Oracle Forms dosyaları FRMF2XML tarafından girdi olarak alınarak bir XML dosyası oluşturulmaktadır. Sonrasında bu dosya, bizim geliştirmiş olduğumuz çözümlenici tarafından bir nesne modeline dönüştürülmektedir. Üretici, bu nesne modelini işleyerek, HTML ve CSS çıktıları üretmekte, LOV’ler için de XML ve Java kaynak kodlarını üretmektedir. Dönüşüm sonucunda ise, sisteme girdi olarak verilmiş olan bir Form dosyasındaki bileşenlerin HTML 5 çıktılarındaki karşılıkları Tablo 1’de özetlenmiştir. Tabloda (*) işareti ile HTML karşılığı verilmiş olan dönüşümlerde, sözdizimsel (syntactic) taşımının yeterli olmadığı görülmüştür. Bu sebeple, Kendu UI ve Oracle Jet kütüphanelerinin kullanımına başvurularak HTML 5’te karşılığı olmayan

Forms bileşenlerinin semantik dönüşümü gerçekleştirilmiştir. Otomatik dönüşüm uygulamasına girdi olarak verilmiş olan örnek bir Forms arayüzü Şekil 4’te, dönüşüm (transformation) işlemi sonucunda elde edilen HTML ve CSS çıktılarının arayüzü ise Şekil 5’te görülmektedir.

Şekil 4: Oracle Forms Arayüzü

Şekil 5: Dönüştürülmüş HTML5 Arayüzü

Oracle Forms Bileşeni	HTML Karşılığı
Content Canvas	Div
Tab	Tab*
TextSegment	Label
Frame	Fieldset
Image	Image
Checkbox	Checkbox (input type="checkbox")
Display Item	Textfield (input type="text" readonly)
Hierarchical Tree	Tree*
List Item	Select
Push Button	Button (input type="button")
Radio Group	Radio Button (input type="radio")
LOV	Window*

Tablo 1: Dönüşüm Karşılıkları

FORMS2HTML5, HTML 5 dönüşümü kapsamında, birden fazla Javascript kütüphanesi seçeneği sunmaktadır. HTML 5 dönüşümü işlemi, halihazırda mevcut olan Kendu UI ve Oracle Jet javascript kütüphaneleri opsiyonel olarak kullanılabilir. Bu bakımdan FORMS2HTML5 modüler yapıyla, yeni Javascript kütüphanelerine

"plug & play" kolaylığıyla entegre olabilmektedir. Bu sayede, Kendo UI ve Oracle Jet'e ek olarak istenilen yeni Javascript kütüphanelerinin de dönüşüm işleminde kullanılabilmesini rahatlıkla sağlamaktadır.

4. Sonuç ve Öneriler

Bu bildiri Oracle Forms uygulamalarının HTML 5 web arayüzlerine dönüşümüne olanak veren FORMS2HTML5 adlı uygulamanın mimarisini ve dönüşüm süreç adımlarını ortaya koymaya çalışmıştır. Bu amaçla öncelikli olarak Oracle Forms platformu hakkında temel bilgiler paylaşılmış ve dönüşüm ihtiyacını ortaya çıkaran motivasyonlar açıklanmıştır.

Forms uygulamaları ile HTML 5 uygulamaları arasında arayüz bileşenleri açısından bazı benzerlikler söz konusu olsa da, aralarındaki semantik farklılığın çok fazla olması ve bazı fonksiyonlitenlerin Forms'da çok ilkel (primitive) kalması sebebi ile dönüşüm işlemlerinin otomatikleştirilmesi çok karmaşık ve yüksek çaba isteyen bir süreç olmuştur.

Benzer yazılım dönüşümü (transformation) ya da taşıması (migration) süreçlerinde, öğeler arasındaki sözdizimsel (syntactic) eşlemeye dayalı aktarım yapmanın otomatik dönüşümü sağlamak için yetersiz kalacağı anlaşılmış, kaynak ve hedef platformlar arasındaki semantik boşlukların giderilmesi için köprüleme (bridging) işlemlerinin de göz önünde bulundurulması gerektiği anlaşılmıştır.

5. Teşekkür

Bu çalışma TEYDEB 1501 Sanayi Ar-Ge Projeleri Destekleme Programı'nın 3140575 numaralı proje desteği kapsamında OBSS

Bilişim Bilgisayar Hiz. Dan. San. Tic. Ltd. Şti. bünyesinde gerçekleştirilmiştir.

6. Kaynaklar

[1] Microsoft: Solution Guide for Migrating Oracle on UNIX to SQL Server on Windows, Chapter 17 - Developing: Applications - Migrating Oracle Forms. **Microsoft TechNet** (2005)

[2] Mossienko, M., "Automated Cobol to Java recycling", **Proceedings of the 7th European Conference on Software Maintenance and Reengineering (CSMR)**, IEEE Computer Society, (2003) 40-50

[3] Sneed, H., "Risks Involved in Reengineering Projects", **Proceedings of the 6th Working Conference on Reverse Engineering (WCRE)**, IEEE Computer Society (1999) 204-211

[4] Oracle Business Intelligence Discoverer Statement of Direction, March 2014 <http://www.oracle.com/technetwork/developer-tools/discoverer/overview/discoverer-sod-jan2009-132849.pdf>

[5] Andrade, Luis, et al. "Forms2Net-migrating oracle forms to microsoft. NET." **Generative and Transformational Techniques in Software Engineering, Springer Berlin Heidelberg**, 2006. 261-277.

[6] Gamer, Pam., "Oracle Forms Developer 10g: build Internet applications; course manual. 2. Student guide", **Oracle University**, 2006.

[7] Agrawal, S., C. Barclay, D. Bronnikov, S. Castledine, T. Chang, B. Cheng, R. Dani et al. "Oracle Database PL/SQL Language Reference, 11g Release 1 (11.1) B28370-05."

