Simülatör Tabanlı ATM Arayüz Yönetim ve Geliştirme Sistemi

Orhan ŞEN, Mehmet GÖKTÜRK

Gebze Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Bölümü, Kocaeli
osen@gyte.edu.tr, gokturk@bilmuh.gyte.edu.tr

Özet: Bankacılık alanında hızlı değişen pazar ihtiyaçlarına paralel olarak ATM cihazlarında da yenilikler yaşanmaktadır. Daha önceleri yalnızca para çekme ve bakiye sorma gibi oldukça sabit görevler yapan bu cihazların bilet satmak gibi yeni görevler aldığı, bu nedenle sürekli ve hızlı bir biçimde arayüz yönetimine ihtiyaç duydukları bilinmektedir. Bu çalışmada, ATM arayüz yönetimini bilişim firmasından alıp, bankacılık operasyon birimi personeli tarafından kullanılabilir bir yapıya kavuşturan simülatör tabanlı bir ATM arayüz yönetim ve geliştirme sistemi geliştirilmiştir. Arayüz tasarımı, durum kontrolü, senaryo analizi ve test aşamalarındaki süreçleri kısaltarak, bu aşamalardaki tüm birimlerin tek bir platform üzerinde koordine edilmesi hedeflenmiştir. Halihazırda arayüzler bankaların operasyon birimleri tarafından tasarlanmakta, ATM geliştiricileri tarafından ise geliştirilip cihazlar üzerinde test edilmektedir. Gerek arayüzün gerçek ATM cihazları üzerinde test etme zorluğu, gerekse iki farklı birim tarafından arayüzün kontrol edilerek geliştirilecek olması bu süreci zorlaştırmakta ve özellikle uzatmaktadır. Bu zor işlem sonunda da ATM kullanıcılarına karmaşık bir arayüz veya yeni isteklerin uygulanabilirliğinin az olduğu bir ATM yazılımı çıkmaktadır. Yapılan çalışmada birimlerin tek bir simülatör ortamı üzerinden arayüz tasarımlarını geliştirip yönetebilecekleri NCR NDC standartlarına uygun bir ara sistem gerçekleştirilmiştir. Yapılan ön değerlendirmelerde operasyon biriminden olumlu geri dönüş alınmıştır.

Anahtar Sözcükler: ATM Simülatörü, Arayüz Yönetim ve Geliştirme Sistemi, NDC.

Simulator-Based ATM Interface Management and Development System

Abstract: In parallel to the changes in the banking sector, ATM machines also face with various changes and advancements. These machines previously only used for cash withdrawal and balance inquiries. Yet, new tasks such as ticket sellling are emerging in a very fast fashion, requiring fast and intensive interface design, update and management on these. In this study, a simulator based ATM interface design and management system has been developed to take the development task from IT department to operational department by introducing an easier interface. Interface design, state control, scenario analysis and test steps are combined under a unified platform. In the traditional case, ATM interfaces are designed by the operational department and changes are implemented through information technology department by ATM software developers. The difficulties in both testing and in the flow if information between diferent departments, make the process difficult and time consuming. Thus, less usable and less updated interfaces are presented to end users. In this study, a single simulator platform can be used by all banking departments under conformance to NCR NDC standard. Preliminary evaluations by the operational department members suggest that the new system will have highly positive results when put in operation.

Keywords: ATM Simulator, Interface Management and Development System, ATM, NDC.

1. Giriş

Gelişen dünyada servis sektöründe artan maliyetler, bu alanda faaliyet gösteren firmaların maliyet düşürme ve ürün kalitesini yükseltme yönünde çözümlere gitmeleri sonucunu doğurmaktadır. Kendi kendine hizmet alınabilen (Self Service) terminal uygulamaları, hizmet sektöründe gerek maliyet düşürülmesini, gerekse servislerin farklı yerlerde farklı zamanlarda sağlanabilmesini sağlamıştır. Bu tür terminaller içinde en yaygın kullanım alanı bulan sistemlerin başında ATM cihazları (Automated Teller Machine) gelmektedir.

ATM sistemleri, bilişim sistemleri alanında yaygın kullanıma ilk geçenler arasındadır. İlk başlarda yalnızca çek yatırma amacıyla (bir nevi posta kutusu) kullanılan ATM sistemleri sıra ile bakiye öğrenme, para çekme, havale ve diğer işlemlerin yapılması amacıyla kullanılmaya başlamıştır. Teknolojinin gelişmesine paralel olarak ATM cihazlarında ekran ve etkileşim teknolojilerinin geliştiği, güvenlik alanında ise akıllı kart ya da temassız kart gibi yeniliklerin kullanılmaya başlandığı görülmektedir [1].

ATM sistemlerinde verilen servislerin sayı ve özelliklerinin sürekli gelişmesine karşın, günümüzde yaygın olarak kullanılan ATM cihazlarının arayüzlerinin göreceli olarak sabit kaldığı gözlenmektedir. Başlangıçta metin tabanlı olan arayüz gereksinimi, zaman içinde grafik nesneler ve animasyon içerecek biçimde geliştirilme ihtiyacı duymuştur. Öte yandan ATM sistemlerinin yüksek maliyeti ve kullanıcı profilinin çok değişken olması, ATM sistemlerinde hızlı ve köklü değişimlerin uygulanmasını güçleştirmektedir.

ATM üreticisi firmalar (IBM, NCR, Diebold vb.) genellikle kendi üretmiş oldukları ATM sistemlerini belirli bir iletişim ve yönetim standardı içinde tasarlamaktadırlar. Bu standart için NCR firmasının NDC protokolü (NCR Direct Connect Protocol) örnek gösterilebilir. Bu protokol dâhilinde, ATM sistemlerinin merkez haberleşmeleri, durum değişkenleri ve ekran yönetimleri bir standart dâhilinde yapılmaktadır [2].

ATM servislerindeki sayı ve karmaşıklık artışı ile mevcut sistemlerin geleneksel yollarla yönetiminin mümkün olmadığı, yoğun performans problemine ve destek hizmeti gereksinimine ihtiyaç duyduğu görülmektedir. Kullanıcı menüsüne eklenmesi gereken yeni bir maddenin büyük problemlere yol açabildiği, günlük ya da saatlik olarak sistem menülerinin ya da animasyonların değiştirilmesinin mevcut sistemlerle mümkün olmadığı görülmektedir.

Hâlihazırda bazı firmaların kullanmakta oldukları ATM yönetim yazılımlarının yetersiz olduğu bilgisi bu firmalar tarafından dile getirilmektedir. Problemlerin hizmet alımı ile çözülmesi çok yüksek maliyetler getirmekte, bir menü elemanının değiştirilmesi için yüksek meblağların ödenmesi beklenmektedir.

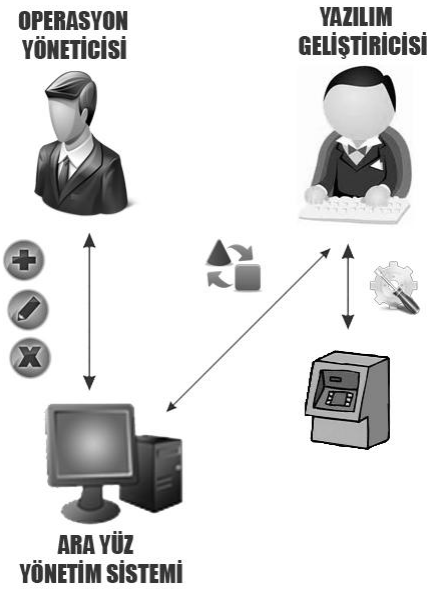
Bu problemler göz önüne alınarak yapılan çalışmada, ATM yazılımı geliştiricilerinin; arayüz tasarımı, durum kontrolü, senaryo analizi ve test aşamalarındaki süreçlerini kısaltarak, bu aşamalardaki tüm birimlerin NDC standartlarına uygun tek bir ortam üzerinden koordine olabilmesi amaçlanmıştır.

Makalenin ikinci bölümünde, yapılan bu çalışmanın uygulaması olan Simülatör tabanlı ATM arayüz yönetim ve geliştirme sisteminin detayları hakkında bilgi verilmiş, son bölümde ise sonuçlar ve çalışmaya eklenebilecek diğer çalışmalar hakkında öneriler sunulmuştur.

2. Simülatör Tabanlı ATM Arayüz Yönetim ve Geliştirme Sistemi

Halihazırda bulunan mevcut ATM cihazlarının çoğu, NDC standartlarına uygun olarak, bankaların ilgili birimlerindeki geliştiriciler tarafından veya dışarından satın alınan hizmet yazılımları tarafından kontrol edilmektedir. Bu yazılımların geliştirilmesi, yeni taleplerin eklenmesi, arayüzün değiştirilmesi, iyileştirmeler vb. gibi çalışmalar yazılım geliştiriciler tarafından yapılırken; bu taleplerde bulunan, arayüzü tasarlayan ve yazılımları test eden kişiler banka operasyon birimleri çalışanları olmaktadır. Aradaki bu fark; ATM arayüzünün yönetilmesini ve geliştirilmesini zorlaştıran temel faktörlerden biridir.

Bu sebeple bu çalışmada geliştirilen sistem ile farklı birimlerin aynı platform üzerinde çalışabilmesi amaçlanmıştır.



Şekil 1 : Sistem Modelleri

Şekil 1' de görüldüğü gibi sistem modelleri ATM arayüz yönetim sistemi üzerinden aynı ortamda bir birleri ile koordineli bir şekilde çalışabilmektedir. Burada operasyon yöneticisi, kendisine sunulan "Arayüz Tasarımı" ile ATM arayüz ekranlarını kolayca tasarlayabilir, arayüze yeni ekranlar ekleyebilir veya mevcut ekranlar üzerindeki bir değişikliği "sürükle bırak" yöntemi ile gerçekleştirebilmektedir.

2.1 Arayüz Tasarımı

Operasyon yöneticisinin üzerinde ekran tasarımı yapabildiği sistemdeki bu ekran, içinde her bir özelliğin nesne dayalı mimari ile tanımlandığı ve "sürükle bırak" yöntemi ile üzerinde kolayca işlem yapılabildiği bir ekrandır.

ATM arayüz yönetim sistemi NDC standartlarına göre 32x16'lık bir metin ekran matrisi ile 640x480 piksel ATM ekran boyutunu model olarak almaktadır. Buna göre yine her biri NDC standartlarına uygun olarak tanımlanmış nesnelerden olan yazı, resim, parametre ve veri girişi nesneleri ekranlar üzerinde yapılabilecek tüm işlemleri kapsamaktadır.

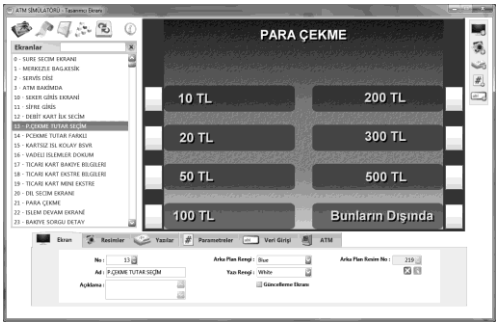
Operasyon yöneticisi ATM yazılımı karmaşasından uzak, fakat ATM arayüzünde yer alan tüm ekranları gerçek boyutları ile sistemde görebilmektedir. Buradan arayüze yeni bir ekran ekleyebildiği gibi, ekran üzerinde her hangi bir değişikliğin anında ekrana yansıtılması sayesinde de değişikliği kolayca gerçekleştirebilmektedir.

Mevcut sistemlerdeki bir ekranın tasarımının çeşitli NDC komutların birleşimi halinde oluşturulduğu ve bu komutların görsel olarak testinin yalnızca ATM üzerinde yapıldığı düşünüldüğünde, bu sistemin arayüz tasarımı üzerinde yarattığı kolaylık rahatça

görülebilmektedir. Operasyon yöneticisinin bir değişiklik isteğini yazılım geliştiricisine ileterek ondan bunu gerçekleştirmesini uzun uğraşlar sonunda beklemek yerine onu da bu sürece katarak istenileni görerek tasarla şeklinde bir sistem yaratılmıştır.

ATM arayüzündeki bir değişikliğin gerçekleştirilmesi sürecini ele alındığında, arayüz yönetim sistemini kullanan operasyon yöneticisi dışındaki diğer kullanıcı olan yazılım geliştiricisine ise, operasyon yöneticisinin tasarladığı ekranlar üzerinden istenilen fonksiyonları ATM yazılımlarına ekleme işlemi kalmaktadır. Buna göre belirlenen ekranlar üzerinde oluşacak durumlar, durum tablolarındaki ilgili yerlere yansıtılarak değişikliğin gerçeklemesi sağlanabilir.

ATM ekranlarının, ATM cihazlarında çalıştırılabilmesi için, NDC standartlarına göre anlamlandırıldığı bir takım komutlar dizisine ihtiyacı vardır. Bunlar NDC standartlarını kullanan tüm ATM cihazları için aynıdır. Buna göre ATM ekranların tamamı, geliştirdiğimiz sistemin bir fonksiyonu sayesinde NDC formatına otomatik olarak dönüştürülebilmektedir. Bu sayede yazılım geliştiricisinin önce ekranları tasarlayıp sonra da ATM test cihazında ekranları tek tek test etmek zorunluluğu ortadan kaldırılmıştır.



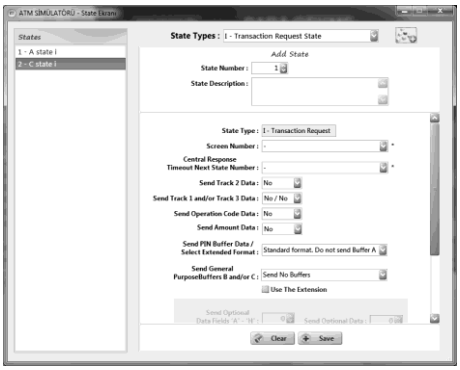
Şekil 2 : Arayüz Tasarımı Ekran Görüntüsü

2.2 State Tasarımı

Sistemin yazılım geliştiricisi kullanıcısına hitap eden bir diğer ekranı ise "State Tasarımı" ekranıdır. Yazılım geliştiricisinin, operasyon yöneticisinin ATM arayüzüne eklediği ya da değiştirdiği ekranları sistemin "Arayüz Tasarımı" ekranı üzerinden görebildiğinden, NDC formatındaki komutları dizisi şeklinde alabildiğinden bahsetmiştik. Bu oluşan ekranların ATM senaryoları içinde yer alması için ise, NDC standartlarında tanımlı her bir durumu temsil eden "State" kontrolleri şekline dönüştürülmesi gerekmektedir. Bir birini takip eden birçok durumun olması, birinin geri dönüşü diğerinin başlangıcı olabildiği gibi, birden fazla durum türünün standartlarda yer alması yine yazılım geliştiricisinin bu ekranları çalışır hale getirmesini zorlaştıran diğer bir etmendir.

Sistemin kolaylaştırmayı amaçladığı diğer bir problem olan durum kontrolleri tasarımı yine ATM arayüz yönetim sisteminde yer almaktadır. Buna göre sistemde yer alan "State Tasarımı" ekranı sayesinde yazılım geliştiricisi NDC standartlarında yer alan A'dan Z'ye tüm state'leri tek bir ekran üzerinden tasarlayabilmekte ve o karmaşık yapıdan uzaklaşabilmektedir. Her bir state'in anlamı ekran üzerinde tanımlı olup, durum geçişlerinin her biri bir bütün halinde görülebilmektedir. Yazılım geliştiricisi, operasyon yöneticisinin ürettiği bir ekranı "State Tasarımı" ekranı üzerinden istediği bir state'e bağlayabilmekte ve bu değişikliği anında görebilmektedir.

Burada iki birimin aynı ortam üzerinde çalışması ile onları durum senaryoları karmaşasından kurtararak, ATM arayüzünde yapacakları bir değişikliğin kolayca arayüze yansıtılmasını sağlatmak amaçlanmıştır.



Şekil 3 : State Tasarımı Ekran Görüntüsü

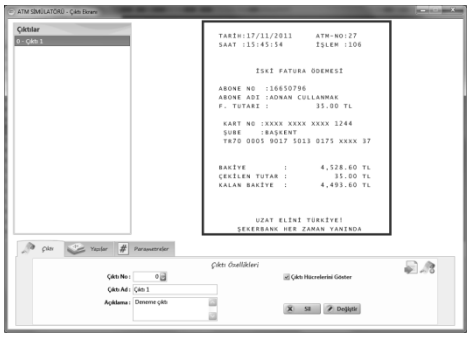
2.3 Çıktı Tasarımı

NDC standartlarına göre ATM'lerde kullanılan iki tip çıktı bulunmaktadır. Bunlar ATM kullanıcısına yaptığı işlem sonunda verilen "makbuz" ve ATM'in yaptığı işlemlerin kendi içinde logunu tuttuğu "journal" çıktılarıdır.

ATM arayüz yönetim sistemi NDC standartlarına göre 40x24'lük bir çıktı matrisini, çıktı tasarımı için model olarak almaktadır. Buna göre yine her biri NDC standartlarına uygun olarak tanımlanmış nesnelerden olan yazı ve parametre nesneleri çıktılar üzerinde yazdırılabilecek tüm işlemleri kapsamaktadır.

Mevcut sistemlerdeki bir çıktının tasarımının çeşitli NDC komutların birleşimi halinde oluşturulduğu ve bu komutların görsel olarak testinin yalnızca ATM üzerinde yapıldığı düşünüldüğünde, bu sistemin çıktı tasarımı için yarattığı kolaylık rahatça görülebilmektedir. Yazılım geliştiricisi; çıktı tasarımı ekranı ile ATM'den çıkarılacak makbuz veya journal çıktılarını, yine arayüz tasarımı ekranında olduğu gibi nesneye dayalı mimari ve "sürükle bırak" yöntemi ile kolayca tasarlayabilir. Tasarlanan çıktılarda yapılan değişikler anında ekrana yansıtıldığı gibi, tasarlanan çıktılarında kullanılmak üzere sisteme kayıt edilmesi sağlanmıştır.

NDC standartlarına göre çıktılarda yine ekranlarda olduğu gibi belli bir komut dizisinden oluşmak zorundadır. Çıktı tasarım ekranı, tasarlanan bu çıktıları NDC formatına da dönüştürerek, yazılım geliştiricisinin ATM yazılımında direk olarak bu komutları kullanması kolaylığını sağlar. Ayrıca oluşturulacak çıktıların kâğıt üzerinde nasıl duracağı ekran üzerinde görülebildiğinden, tasarlanan her çıktının tek tek ATM üzerinde test edilmesine gerek kalmayacak ve bu sayede uzun test sürecinden yazılımcı kurtarılmış olacaktır.



Şekil 4 : Çıktı Tasarımı Ekran Görüntüsü

2.3 Transaction Testi

ATM'ler bankalar dışı işlem cihazları olduğundan yapacakları işlemlerin bir kısmını banka sunucuları ile çeşitli mesaj akışları üzerinden yapmaktadırlar. Sunucu ile yapılan bu mesaj trafiğine "Transaction" işlemleri adı verilir. Bu mesaj akışı ATM'in "Transaction Request" mesajını sunucuya göndermesi ile başlarken, sunucunun "Transaction Reply" mesajını ATM'e göndermesi ve ardından ATM'nin "Solicited Status" mesajını sunucuya göndermesi ile tamamlanır. ATM üzerinde tanımlı state'lerin değişken değerleri sunucudan gelecek bu reply bilgileri ile

doldurularak arayüze yansıtılır. Bu yüzden operasyon yöneticisinin tasarladığı ATM arayüzlerinin, değişken değerleri ile de birlikte ekranda nasıl görüleceğinin test edilmesi gerekmektedir. Bu test işlemi yine test ATM'leri üzerinden yazılım geliştiriciler tarafından yapılır.

ATM'lerin arayüzlerinde yapılacak bir değişikliğin arayüzü tasarlayıp görebilecekleri ATM dışında bir ekran olmamasının yanında, değişken ekranlarında test edileceği bir ortamın olmaması gelişimi zorlaştırmaktadır. Bu zorluğu ortadan kaldırmak için ATM arayüz yönetim sisteminde "Transaction Test" ekranı bulunmaktadır. Yazılım geliştiricisi, operasyon yöneticisinin veya kendisinin tasarladığı değişken bir ekranı, NDC standartlarındaki Transaction mesaj akışı şeklinde bu ekrandan test edebilir. Buna göre sistem, transaction request mesajını sunucuya TCP protokolü ile ağ üzerinden gönderirken, yine aynı yolla sunucudan transaction reply mesajını alarak oluşan arayüzü ekranda gösterir. Bu işlem sayesinde mesaj akışının takibi ve testi ATM'e gitmeden yönetim sistemi üzerinden yapılabilir.

Çıktı oluştururken çıktıda bulunacak değişkenler yine Transaction Test ekranı ile doldurulabilir, oluşacak çıktıların görüntüsü yine ATM'e gitmeden test edilebilir.

ATM üzerinde bir ekran veya çıktının test edilebilmesi için, her seferinde ATM test cihazına gidilerek, test edilecek tüm senaryonun baştan çalıştırılıp denenmek zorunda olması çok uzun ve zor bir süreç olarak karşımıza çıkmaktadır. Bu nedenle arayüzün simülasyon ortamında test edilmesi bu süreci kısaltırken, ATM arayüzlerini yönetip geliştirmeyi kolaylaştırmaktadır.

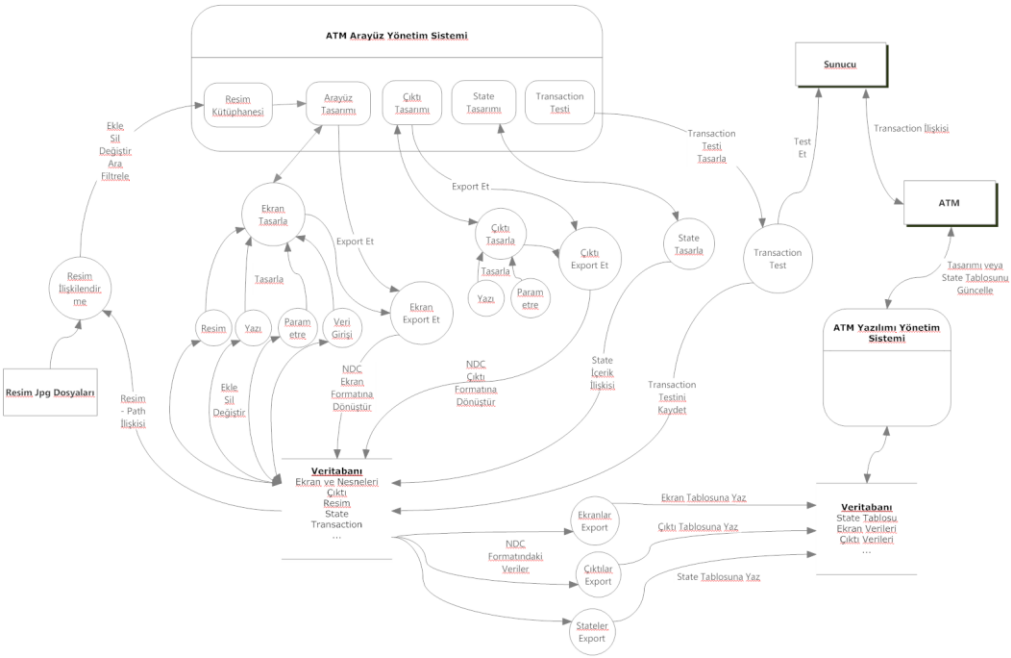
Şekil 5 : Transaction Testi Ekran Görüntüsü

3. Yöntem ve Teknikler

Simülatör tabanlı ATM arayüz yönetim ve geliştirme sistemi için tercih edilen programlama dili .NET framework üzerinde C# dili olup, tüm kayıtların tutulduğu veri tabanı MsSQL veri tabanıdır. C#'ın nesneye dayalı yazılım mimarisine uygun iyi bir dil olması ve MsSQL'in bu dile entegrasyonunun bire bir olması sebebiyle böyle bir seçim yapılmıştır.

Operasyon yöneticisinin ve yazılım geliştiricisinin aynı ortam üzerinden çalışmasının sağlanması için, ekranlar ayrı birer form olarak modellenmiş ve bir birini takip eden veriler Entity Framework nesneleri ile kontrol edilerek bir veri tabanı üzerinde işlemlerin yapılabilmesi sağlanmıştır.

Modellemenin modül bazında tasarlanması sebebiyle de sistem, geliştirilmeye ve yeni modüllerin eklenmesine uyumludur. ATM arayüz yönetim ve geliştirme sistemi ekranları, ATM prototipinin bire bir taklit ekranlarına sahip olup gelecek çalışmalar ile tamamıyla bir ATM Simülatör'ü olunması üzerine ilerlemektedir.



Şekil 6 : Data Flow Diagram

ATM arayüz yönetim ve geliştirme sistemine data flow diagram üzerinden de bakılacak olunursa; sistemin verilerinin bulunduğu bir veri tabanına, ATM yazılım geliştiricisinin kullandığı yazılımın işlemlerini yaptığı veri tabanına ve ağ üzerinden de sunucuya direk bağlantı olduğu görülmektedir. Buna göre operasyon yöneticisi ATM arayüz yönetim sisteminde tasarladığı ekran ve nesneleri sistemin veri tabanına kayıt etmiş olur. Resimler fiziksel olarak bir dosyada bulunurken bu ekrandaki resim kütüphanesi ekranı ile resimler üzerinde işlem yapabildiği ve kullanacağı resimleri dosyaya erişmeden düzenleyebildiği görülür. Buradaki filtreleme ve arama fonksiyonları ise çok sayıda resim olan sistemde işlem kolaylığı sağlar. Yazılım geliştiricisi operasyon yöneticisinin tasarladığı bu ara yüzleri "export" fonksiyonu ile NDC ekran formatına dönüştürürken, aynı zamanda ATM yazılımı yönetim sisteminin kullandığı veri tabanına da ekranları aktarabildiği görülür.

ATM ekranlarını tasarlarken yapılan işlemlerin bir benzeri de çıktı tasarımında yapılmaktadır. Çıktılar, çıktı tasarımı ekranı üzerinden tasarlanarak veri tabanına kaydedilirler. Bu işlem sonrasında yazılım geliştiricisi yine "export" fonksiyonu ile verileri, NDC çıktı formatına dönüştürerek, ATM yazılımı yönetim sisteminin kullandığı veri tabanına çıktıları aktarmış olacaktır.

ATM üzerindeki statelerin tasarımının ve kontrolünün zor olmasından bahsetmiştik. Bunu kolaylaştırmak için kullanılan state tasarımı ekranı, tasarlanan stateleri tip ve içeriklerine göre state-içerik ilişkisi şeklinde veri tabanına kaydetmektedir. Fakat burada da "export" fonksiyonu ile stateler NDC state formatına dönüştürülerek ATM yazılımı yönetim sisteminin kullandığı veri tabanındaki state tablosuna kolayca aktarabildiği görülür.

Ekran, çıktı ve state tasarımlarının NDC formatına dönüşümünün sistemde kolayca

yapılabilmesi, ATM yazılımı geliştiricilerinin tekrardan bu verileri yaratma zorunluluğundan kurtararak, onlara görerek tasarlama kolaylığını getirdiği görülmektedir.

ATM yazılımı geliştiricileri, transaction mesaj akışını ATM yazılımı yönetim sistemi üzerinden tasarlayarak, test ATM'lerine giderek sunucu ile test etmektedirler. Görüldüğü üzere sistemde bu işlem transaction testi ekranı üzerinden sunucu ile iletişime geçilerek gerçekleştirilmektedir. Bu sayede her bir transaction testi için ATM'e gidilme zorunluluğunun ortadan kaldırıldığı görülmektedir. Testler aynı zamanda sistemin veri tabanına kayıt edilerek tekrar farklı durumlar içinde kullanılabilmektedir.

4. Sonuç ve Öneriler

Günümüzde artan işlem taleplerine karşın servis hizmetlerindeki maliyet artışı göz önünde bulunduğunda bu alanda faaliyet gösteren firmalar için çeşitli iş kolaylaştırıcı ve süreç kısaltıcı çözümlerin kullanımı kaçınılmaz olmuştur. ATM cihazlarının arayüzlerinin geliştirilmesindeki problemler, çalışanlar arası koordinasyon güçlüğü ve test etmenin zorluğu düşünüldüğünde arayüz tasarımının yapılacağı bir simülatör zorunlu hale gelmektedir.

Geliştirilen sistem ile ATM arayüz tasarımlarının doğrudan bire bir ATM ekranları üzerinden sürükle bırak tekniği ile yapılabiliyor olması, state geçişlerinin tasarımının ve kontrollerinin, transaction mesaj akışı testinin tek bir ekrandan yapılabiliyor olması, bu aşamalarda çalışanlar için kullanışlı birer çözüm ortaya çıkarmıştır.

Mevcut ATM arayüz sistemlerinin gelişimindeki operasyon yöneticileri ve yazılım geliştiricileri arasındaki koordinasyon güçlüğü onları aynı ortam üzerinden çalıştırarak giderilmiştir. Ayrıca ATM arayüzlerinin test edilmesi için test ATM'lere gidilerek, her seferinde bir ekranın senaryosunun tamamının gözden geçirilmek zorunda olunmasından kurtarılmış, ATM arayüzünün testi için büyük bir test kolaylığı sağlanmıştır. Operasyon bölümü ile yapılan mülakatlarda geliştirilen sistemin oldukça yararlı olduğu görülmüş, en kısa zamanda çalışmanın ürün haline getirilerek uygulamaya alınması benimsenmiştir.

Simülatör tabanlı ATM arayüz yönetim ve geliştirme sistemi, senaryoların durumlar (state) ile entegrasyonunun yapılması, senaryo modellemenin gerçekleştirilebilmesi ve ekranların bir ATM gibi davranması sağlatılarak tamamen bir ATM Simülatörü haline doğru geliştirilebilir. Bu yöndeki çalışmalarımız ilerleyen zamanlarda devam edecektir.

Geliştirilmiş olan bu sistem; ATM yazılımlarını otomatik güncelleme, yazılımların dağıtımı ve "Kiosk" cihazları arayüzleri içinde uygulanabilir bir sistem olarak düzenlenebilecektir.

5. Kaynaklar

[1] Oberan, S.L, Non Cash Media Card and the Process For Dispensing from Automated Teller, US Patent App, 20080116259 A1, 2008.

[2] NCR PersonaS ATM System Description, NCR Corp**,** 1998.

[3] Rosado da Cruz, A.M, Automatic Generation of User Interfaces from Domain and Use Case Models, 208-212, 2007, IEEE QUATIC 2007.

[4] Zimmermann, C., M., Effects of dialogue design on automatic teller machine (ATM) usability: Transaction times and card loss, Behaviour & Information Technology V11, 441-449, 2000.