

Açık Kaynaklı Araçlarla Bütünleşik Bilgi Sistemi Yapmak: ODTÜ Örneği

Eren Şenelmiş¹

¹ OYAK Teknoloji Bilişim ve Kart Hizmetleri AŞ, Ankara
esenelmis@oytek.com.tr

Özet: Açık Kaynaklı Araçlarla Bütünleşik Bilgi Sistemi Yapmak: ODTÜ Örneği isimli makalede ODTÜ için geliştirilen Bütünleşik Bilgi Sistemi (BBS) projesinde açık kaynaklı araçların kullanımı ile ilgili deneyimler paylaşılacaktır. Açık kaynaklı araçların seçiminde izlenen yöntem açıklanacaktır. Hangi araçların ne amaçla kullanıldığı ve bu araçların ne şekilde entegre edilerek bütünleşik bir sistemin oluşturulduğu açıklanacaktır. Uygulamada karşılaşılan problemler ve bu problemlere karşı geliştirilen çözümler anlatılacaktır.

Anahtar Sözcükler: Açık Kaynak, Bütünleşik Bilgi Sistemi, Entegrasyon

1. Giriş

Herhangi bir tedarikçi olmadan gönüllülük esasına göre kullanıcılar tarafından başka kullanıcıların faydalanması için geliştirilen, dağıtılan ve desteklenen açık kaynaklı çok başarılı yazılımlar vardır [1]. Kullanıcıların, maddi bir gelir beklemeden açık kaynaklı bir projeye neden destek verdiği akademisyenler ve uygulayıcılar için hala bir araştırma konusudur. Ama kullanıcıların açık kaynaklı yazılım geliştirme çalışmaları içerisinde yer almasındaki temel motivasyon kaynakları şu şekilde sıralanabilir (1) kullanıcıların doğrudan o yazılıma ihtiyaç duymaları ya da kullandıkları yazılımda çeşitli iyileştirme gereksinimi duymaları (2) yapılan işi seviyor olmaları (3) açık kaynaklı bir projeye üst düzey katkı sağlamanın kazandırdığı tanınmışlık [2]. Andrea Bonaccorsi ve Cristina Rossi, “Why Open Source Software Can Succeed” isimli makalelerinde kullanıcıların açık kaynaklı araçların geliştirilme sürecinde yer almalarındaki motivasyon kaynaklarını ayrıntılı bir şekilde değerlendirmişlerdir [3].

Geliştirilen araçlar kullanıldığı zaman yapılan çalışmalar amacına ulaşmış olur. Bundan sonraki kısımda kurum ya da kuruluşların açık kaynaklı ürünleri kullanmak ya da kullanmamak konusunda verdikleri kararı etkileyen etmenlere değinilecektir.

İş hayatının yoğun temposu içerisinde, otomasyon sistemlerindeki aksaklıkların mümkün olan en alt seviyede olması ve oluşan problemlerin çok kısa sürede giderilmesi beklenmektedir. O nedenle araç seçimi çok önemlidir. Bu makalenin ikinci kısmında araç seçiminin nasıl olması gerektiği konusu ele alınmıştır.

Kullanılan aracın açık kaynaklı olması dış bir yapı ya da firma bağımlılığını büyük oranda ortadan kaldırmaktadır. Araçta ortaya çıkabilecek hatalara geliştirme ekibi doğrudan müdahale edip sistemin tekrar çalışır duruma gelmesini sağlayabilmektedir.

Başta kamu kuruluşları olmak üzere bir kısım kurum ya da kuruluşlar otomasyon sistemlerinde kullandığı araçlar için firma desteğine önem vermektedir. Olası bir hata durumunda firmanın en kısa zamanda çözüm getireceğine inanmaktadırlar. Basit yapılandırma problemlerinde ve kullanım hatalarında bu şekilde destek alınabilmektedir. Ama üründe ortaya çıkan bir

hata (bug) durumunda çözüm süreci çok uzun olabilmektedir. Artık birçok açık kaynaklı araç için de firma desteği verilmektedir.

Olgunluğunu ve güvenilirliğini kanıtlamış birçok açık kaynaklı araç mevcuttur. Bu tür araçlara en güzel örnek Apache Web Server’dir. Netcraft’ın Ağustos 2012 tarihli araştırmasına göre internet üzerinden erişilen yaklaşık 630 milyon web sitesinin %55’i açık kaynaklı Apache Web Server üzerinden hizmet vermektedir [4].

ODTÜ’de gerçekleştirilmekte olan Bütünleşik Bilgi Sistemi projesi açık kaynaklı araçlar üzerinde oluşturulmuştur. Genel kabul görmüş ve aktif bir topluluk tarafından geliştirilen araçların kullanılmasına önem verilmiştir. Aşağıdaki tabloda projede kullanılan açık kaynaklı araçların başlıcaları ve kullanım amaçları listelenmiştir.

Araç Adı	Kullanım Amacı
Activiti	Süreç Motoru, Süreç Tasarlama Aracı
Drools	Kural Motoru
CAS	Erişim Kontrolü
DSpace	Bilgi Varlıkları Kütüphanesi
Liferay	Portal
WSO2 ESB	Kurumsal Servis Yolu
WSO2 Governance Registry	Servis Kayıt Kütüğü
OpenLDAP	LDAP Sunucusu
Lucene/SOLR	Arama Motoru
PostgreSQL	Veritabanı Yönetim Sistemi
Debian	İşletim Sistemi

Tablo 1. BBS Projesinde Kullanılan Başlıca Açık Kaynaklı Araçlar

Proje kapsamında bu araçların son kullanıcı gereksinimlerini karşılayacak şekilde uyarlanması sağlanmıştır. Uyarılama aşamasında çeşitli yöntemler izlenmiştir. Aracın tartışıldığı forumlarda inceleme yapmak ya da sorunu tanımlayıp yardım istemek, aracın hata takip sisteminde hata kaydı oluşturmak ve

doğrudan aracın kodunda değişiklik yapmak bu yöntemlerden bazılarıdır.

Projenin amaçlarından biri de kullanılan araçların bir bütün oluşturacak şekilde entegrasyonunun sağlanmasıdır. Bu amacın gerçekleştirilmesi için araçlarda mevcut bulunan entegrasyon yeteneklerinden faydalanılmıştır. Aracın entegrasyon yeteneklerinin yetersiz kaldığı durumlarda aracın kodunda değişiklik yapılarak entegrasyon sağlanmıştır.

Aşağıdaki alt başlıklarda son kullanıcı gereksinimlerini karşılamak için açık kaynaklı araçlarda yapılan uyarlamalar ile ilgili tecrübeler paylaşılmaktadır.

2. Kullanılacak Aracı Belirlemek

CMMI for Development Version 1.3 modelinde yer alan Karar Analizi ve Çözümleme sürecinde tanımlanan yöntem izlenerek hangi aracın kullanılacağına karar verilmiştir [5]. Bu yöntemde öncelikle son kullanıcı gereksinimleri ve teknik gereksinimlerden yola çıkarak değerlendirme kriterleri belirlenmiştir. Açık kaynaklı araçları değerlendirirken gereksinimlerden kaynaklanan bu kriterler tek başına yeterli olmamaktadır. Açık kaynaklı araçların yapısı gereği daha başka kriterler de dikkate alınmalıdır. Aracın kaç kişi ya da kaç firma tarafından desteklendiği önemli bir etkidir. Nispeten küçük topluluklar tarafından geliştirilen araçlar sonlanma riskini taşımaktadırlar. O nedenle kalabalık topluluklar tarafından geliştirilen araçlar daha çok tercih edilmiştir. Aracı geliştiren topluluğun ne kadar aktif olduğu, kapatılmamış kaç tane işin kaldığı, işlerin ortalama kapatılma süresi gibi ölçüm değerleri dikkate alınmıştır. Aracın hangi sıklıkta sürüm çıkardığı da yine değerlendirme kriterleri arasında yer almaktadır. Daha ayrıntılı bir değerlendirme için David Cruz ve arkadaşlarının çalışmasından faydalanılabilir [5]. Belirlenen kriterlerin % olarak ne kadar ağırlıkta olduğu belirlenmiştir.

Bu gereksinimler için hangi alternatiflerin mevcut olduğu belirlenmiştir. Önemli bir kısım kullanım senaryoları belirlenmiştir. Bu kullanım senaryoları üzerinden gidilerek araçların başta belirlenen kriterlere göre puanlaması yapılmıştır. Değerlendirme sonucunda tercih edilen araç müşteri onayına sunulmuştur. Müşterinin onayladığı ürün kullanıma alınmıştır.

3. ODTÜ BBS Projesinde Kullanılan Araçlar

Tablo 1’de yer alan araçlar başta olmak üzere ODTÜ BBS projesinde yaklaşık 20 farklı açık kaynaklı araç kullanılmıştır. Bundan sonraki kısımda bu araçların kullanımı ve entegrasyonu sırasında yaşanan tecrübeler paylaşılacaktır.

3.1. Süreç Yönetimi

BBS, süreç tabanlı bir sistem olduğu için Süreç Yönetim altyapısı olarak kullanılacak aracın seçimi ve uyarlanması çok önemli bir konudur. Uzun süren inceleme ve değerlendirmelerden sonra Süreç Yönetimi altyapısı olarak Activiti kullanıma karar verilmiştir [7].

Activiti, Intalio, Bonita ve BizAgi isimli araçlar değerlendirilmiştir. Bu değerlendirme şu kriterlere göre yapılmıştır; BPMN 2.0 standardını destekleyip desteklemediği, ne sıklıkta sürüm yayınladığı, kişi ve firma sayısı olarak topluluğunun büyüklüğü, forumlarındaki aktivite, hataların ya da yeni isteklerin ortalama gerçekleştirilme süresi, diğer araçlarla entegrasyon yetenekleri vb. Değerlendirmeye tabi tutulan araçlar için tüm kriterlerin yorumlanması bu makalenin kapsamını aştığı için örnek olarak birkaç değerlendirme verilecektir. Değerlendirmeye alınan 4 araç da BPMN 2.0 standardını desteklemektedir. Intalio, komple bir sistem olarak tasarlanmıştır. Sadece kendisini imkân sağladığı noktalarda uyarlamaya izin vermektedir. BBS projesi gereksinimlerinden olan XACML desteği mevcut değildir ve sonradan eklenmesi de mümkün değildir. Bonita ve BizAgi ise bütünleşik bir bilgi sisteminde ihtiyaç duyulan esnekliğe, açıklığa ve fonksiyonallığa sahip değildir. Bahsedilen bu 3 araç da temelde tek bir firma tarafından gerçekleştirilmektedir. Topluluk destekleri ve forum aktiviteleri son derece yetersizdir. Activiti, yaklaşık 10 firma ve 30 kişi tarafından geliştirilmektedir. Geniş bir topluluk tarafından destekleniyor olması önemli bir avantajdır. Son derece aktif bir forumu vardır. Forum üzerinden sorulan bir soru çok kısa bir zamanda cevaplanmaktadır. 1 saat içerisinde cevaplanan soruların sayısı azımsanmayacak kadar çoktur. Bu karşılaştırmanın yapıldığı sırada her ay düzenli olarak sürüm yayınlaması yapılmaktaydı.

Activiti’nin geçmişi jBPM’e dayanmaktadır. jBPM, JBoss çatısı altında geliştirilmektedir. jBPM’in 4. sürümünün geliştirilme çalışmaları devam ederken geliştiren topluluğun bir kısmı ayrılıp Alfresco çatısı altında Activiti’yi geliştirmeye başlamışlardır. Bunun bir sonucu olarak Activiti’nin ilk sürümü 5 versiyon numarası ile yayınlanmıştır.

Activiti, süreç motoru (Activiti Engine) ve süreç tasarlama aracı (Activiti Designer) olarak kullanılmaktadır. BPMN 2.0 standardında yer alan olay (event), görev (task), geçit (gateway) vb model elemanlarından birçoğunu desteklemektedir.

BBS projesinde süreç modelleme 2 aşamada gerçekleştirilmektedir. İlk aşama analiz çalışmalarının sonunda analiz grubu tarafından gerçekleştirilmektedir. Bu aşamada BPMN modellerinde daha çok başlangıç/bitiş olayları, kullanıcı görevleri ve bir kısım geçitler tanımlanmaktadır. Bu yaklaşım ile sonraki aşamaya göre nispeten sade bir süreç modeli oluşturulmuş olmaktadır. Son kullanıcı ile bu model üzerinden kontrol yapılmakta, eksik ya da hatalı noktalar erken bir aşamada tespit edilmektedir.

Süreç modellemenin ikinci aşaması geliştirme çalışmaları sırasında geliştirme ekibi tarafından gerçekleştirilmektedir. Bu aşamada servis görevleri, iş kuralı görevleri, aktivite çağırma (Call Activity) vb türdeki tanımlamalar yapılmaktadır.

Activiti, esnek bir araç olması nedeniyle çok iyi bir şekilde uyarlanabilmekte ve sistemin geri kalan kısmı ile rahat bir şekilde bütünleşmesi sağlanmaktadır. Activiti’de servis görevleri web servisleri çağırabilmenin yanı sıra Spring bean’lerini de çağırabilmektedir. Bu özellik, Java’nın tüm gücünü sürecinizde kullanabilme özgürlüğünü kazandırmaktadır.

BPMN’in çıkış amaçlarından biri de son kullanıcı, analizci ve geliştiricinin aynı modeli kullanması ve otomasyonun yine aynı model üzerinden gerçekleşmesidir. Yukarıda anlatılan şekliyle BBS projesinde yürütülen çalışmalarda bu amaç gerçekleştirilmiştir.

Proje kapsamında yapılan çalışmalar sırasında Activiti’nin kodunda yapılan değişiklik çok azdır. Proje gereklerini karşılamadığı durumlarda eklenti mekanizması ile ek özellikler kazandırılmıştır. Örneğin Activiti süreç değişkenlerini serialize ederek saklamaktadır. Süreç değişkeni olarak kullanılan veri nesnelere ilişkin sürecin tablolarında tutulması ihtiyacı doyumlanmıştır. Bu gereksinimin sonucu olarak Activiti’ye eklenti yapılaraksüreç değişkeni olarak kullanılan veri nesnelere ilişkin süreç tablolarında tutulması sağlanmıştır.

Oluşan problemlerle ilgili genellikle forumlarda bilgi bulunmaktadır. Bilgi olmadığı durumlarda forumlarda soru sorulmuştur. Bu sorular çok kısa sürede cevaplanmıştır. Activi Designer’ın 5.9.2 sürümünde tespit edilen bir hata, hata takip sistemine girilmiştir. Birkaç gün sonra yayınlanan 5.9.3 sürümünde bu problemin giderildiği görülmüştür.

3.2. Kural Motoru

Kural motoru olarak Drools kullanılmaktadır [8]. Activiti, iş kuralı görevlerini (Business Rule Task) sadece Drools ile gerçekleştirmektedir. O nedenle kural motoru olarak kullanılacak araç için seçim yapılmasına gerek kalmamıştır. Aynı zamanda açık kaynaklı araçlardan beklenen kriterleri sağladığı için kullanımında bir problemle karşılaşmamıştır.

Drools, JBoss tarafından geliştirilmektedir. Geniş bir topluluğu, aktif bir forumu vardır ve çok sık aralıklarla sürüm yayınlamaktadır. Drools da, Tibco Business Events, CLIPS, Jess ve BizTalk Rules Engine gibi Rete algoritmasını gerçekleştirmiştir.

3.3. Erişim Kontrolü

Erişim kontrolü için CAS kullanılmaktadır [9]. CAS, OpenSSO ve JOSSO değerlendirilmiştir. Tüm açık kaynaklı araçlar için geçerli olan kriterlerin yanı sıra şu kriterlere göre değerlendirme yapılmıştır; Single Sign-On desteği, OpenLDAP desteği, Liferay

entegrasyonu, Java tabanlı araçlarla entegrasyonu ve PHP tabanlı araçlarla entegrasyonu.

Değerlendirilen tüm araçlar yeterli olgunluk seviyesindedir. Topluluklarının büyüklüğü ve aktivitesi farklılık gösterse de yeterli seviyededir. CAS, birçok farklı programlama dili ile geliştirilmiş sistemlerle entegre olabilmesi, açık kaynak kodlu birçok araç tarafından entegre edilmiş olması ve kolay uyarlanabilir esnek bir yapısı olması nedeniyle tercih edilmiştir. Craig Dickson, JavaOne 2007’de yaptığı Fast and Free SSO: A Survey of Open-Source Solutions to Single Sign-on isimli sunumunda bu araçları ayrıntılı bir şekilde değerlendirmiştir [10].

Proje kapsamında ilk olarak Liferay entegrasyonu sağlanmıştır. Geliştirme ortamında erişim kontrolü test ve canlı ortamlardan farklı olarak Spring Security ile gerçekleştirildiği için Spring Security’nin CAS entegrasyonu da sağlanmıştır. Bilgi Varlıkları Kütüphanesi olarak kullanılan DSpace’in CAS entegrasyonu olmamasına rağmen proje çalışmaları kapsamında bu entegrasyon da gerçekleştirilmiştir. ODTÜ’nün uzun süredir kullandığı PHP tabanlı bir sisteminin de erişim kontrolünü CAS üzerinden gerçekleştirmesi sağlanmıştır.

Proje gereksinimleri nedeniyle CAS’ın kodunda da çeşitli değişiklikler yapılmıştır. CAS’ın arayüzleri ODTÜ kurumsal kimliğini yansıtacak şekilde değiştirilmiştir. Görüntülü ve sesli captcha desteği de CAS’ın kodunda yapılan değişikliklerle sağlanmıştır.

3.4. Yetkilendirme Servisi

Yetkilendirme servisi olarak Axiomatics kullanılmaktadır [11]. Axiomatics, projede kullanılan ve açık kaynaklı olmayan tek araçtır.

Proje gereksinimlerinde yetkilendirme servisinin XACML standardını desteklemesi istenmektedir. XACML standardını sağlayan açık kaynaklı araçlardan SunXACML, Heras AF ve Enterprise XACML incelenmiştir. Bu araçlardan hiçbirisi açık kaynaklı araçlardan beklenen kriterleri sağlamamaktadır. Yeterli topluluk destekleri mevcut değil ve yeterli olgunluk seviyesine ulaşmamışlardır. Projedeki aktivite son derece düşüktür. Bu gibi sebepler nedeniyle yetkilendirme servisi olarak açık kaynaklı bir araç kullanılamamaktadır.

4. Sonuç ve Öneriler

Projelerin en önemli amacı müşteri gereksinimlerini karşılayan ürün oluşturmaktır. Bu amaca ulaşmak için çeşitli gereksinimleri karşılayacak birçok açık kaynaklı araç vardır.

Açık kaynaklı bir araç seçilirken fonksiyonel ve teknik gereksinimlerin yanı sıra topluluğun büyüklüğü, forum aktivitesi gibi kriterler de dikkate alınmalıdır.

Açık kaynaklı araçkullanan bir geliştirme ekibi karşılaştığı problemler karşısında araştırma yapma yöntemini bilmeli ve gerektiğinde bu aracın kodu üzerinde değişiklik yapabilecek yetkinliğe sahip

olmalıdır. Bu yetkinliğe sahip bir geliştirme ekibi çok kısa sürede problemleri giderebilmektedir. Kapalı kaynaklı araçlarda yaşanan bir problem karşısında kısa sürede destek alınabilmektedir ama üründe bir hata varsa bu hatanın giderilmesi uzun zaman alabilmektedir. Pek çok durumda problemlere alternatif çözüm oluşturma görevi yine geliştirme ekibine düşmektedir.

Bu çalışmada, yaygın olarak kullanılan açık kaynaklı araçlarla bir bütünleşik bilgi sisteminin nasıl oluşturulduğu örneklendirilmiş ve yaşanan deneyimler paylaşılmıştır.

Her projenin kendisine göre bir yapısı ve gereklilikleri vardır. Bir proje için oluşturulan çözüm başka bir proje için uygun olmayabilir. BBS projesinde açık kaynaklı araçlarla çözüm üretilmiş olması tüm projelerin açık kaynaklı araçlarla geliştirilebileceği anlamına gelmez. Yazar, projelerin araç seçimi aşamasında açık kaynaklı araçların da bir alternatif olarak değerlendirilmesini önermektedir.

5. Kaynaklar

[1] E. von Hippel, "Sticky information and the locus of problem solving: implications for innovation", *Management Science*, 40 (4) (1994)

[2] Karim R Lakhani, Eric von Hippel, "How open source software works: "free" user-to-user assistance", *Research Policy*, Volume 32, Issue 6, June 2003

[3] Andrea Bonaccorsi, Cristina Rossi, "Why Open Source software can succeed", *Research Policy*, Volume 32, Issue 7, July 2003

[4] Netcraft, August 2012 Web Server Survey, <http://news.netcraft.com/archives/2012/08/02/august-2012-web-server-survey.html>

[5] CMMI Product Team, CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033), Software Engineering Institute, Carnegie Mellon University, 2010, <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>

[6] David Cruz, Thomas Wieland ve Alexander Ziegler, "Evaluation Criteria for Free/Open Source Software Products Based on Project Analysis", *Software Process Improvement and Practice*, 2006; 11: 107–122

[7] Activiti, <http://www.activiti.org>

[8] Drools, <http://www.jboss.org/drools>

[9] CAS, <http://www.jasig.org/cas>

[10] Craig Dickson, "Fast and Free SSO: A Survey of Open-Source Solutions to Single Sign-on", *JavaOne*, 2007

[11] Axiomatics, <http://www.axiomatics.com>