

WEB TABANLI SİSTEMLERİN PSM4MSF ÜSTMODELİ KULLANILARAK MODEL GÜDÜMLÜ GELİŞTİRİLMESİ

Geylani Kardaş^{1*}, Ferhat Erata²

¹Ege Üniversitesi Uluslararası Bilgisayar Enstitüsü, 35100, Bornova, İzmir

²UNIT Bilişim Teknolojileri Araştırma Geliştirme ve Otomasyon Sistemleri Danışmanlık San.Tic.Ltd.Şti., İzmir Teknoloji Geliştirme Bölgesi No:5 A2 Binası, 35430, Urla, İzmir

Özet

Web tabanlı yazılım sistemlerini model güdümlü geliştirmek için çeşitli metodolojiler bulunmaktadır. Bu metodolojilerin kapsamında, sistem modelleri sadece analiz ve tasarım amaçlı kullanılmamakta; web uygulamalarının otomatik üretilmesi için de modellerden yararlanılmaktadır. Ancak kabul görmüş ve üreticisi tarafından gelişimi sürekli desteklenen belirli bir uygulama geliştirme çatısı ya da platform teknolojisi olmaksızın otomatik yazılım üretimi, kurumun iş ihtiyaçlarının karşılanması ve geliştirilen uygulamaların değişen ihtiyaçlara hızlı bir şekilde uyum sağlayabilmesi açısından hem üretken hem de etkin olamamaktadır. Bu noktadan hareketle, bu makalede Microsoft Windows platformuna özgü bir teknoloji olan Sharepoint'in nesne modelini genişleten ve soyutlama seviyesini yükselten PSM4MSF ("Platform Specific Metamodel for Microsoft Sharepoint Foundation") adı verilen bir web mühendisliği üstmodeli önerilmekte ve bu üstmodelle dayalı bir model güdümlü web tabanlı yazılım geliştirme yöntemi tanıtılmaktadır. Microsoft SharePoint uygulama çatısı üzerinde geliştirilen ve Model Güdümlü Mimari'ye dayanan bir yöntem ile, önerilen üstmodelin kullanıcı modellerinden web tabanlı kurumsal çözümlerin otomatik olarak üretilmesi sağlanmaktadır. PSM4MSF'e dayalı yazılım çatısı bir çok kurumsal sistemin geliştirilmesinde başarıyla kullanılmıştır. Bu makalede de ülkemizin önde gelen petrokimya şirketlerinden birinde aktif olarak kullanılan ve şirketin temel performans göstergelerinin belirlenmesinde planlama ve üretim kesimlerinden gerekli bilginin toplanması ve işlenmesini yöneten bir web sistemi çözümü gerçekleştirimi anlatılarak önerdiğimiz yazılım geliştirme yönteminin kullanımı örneklenmiştir.

Anahtar Kelimeler: Üstmodel, Model Güdümlü Geliştirme, Web Mühendisliği, Microsoft Sharepoint Foundation

MODEL DRIVEN DEVELOPMENT OF WEB BASED SYSTEMS BY USING PSM4MSF METAMODEL

Abstract

There exist various methodologies for model driven development of web based software systems. In addition to analysis and design, system models are also used for automatic generation of web applications within these methodologies. However, without considering a widely-accepted application framework or platform whose development is constantly supported by its producer, the automatic code generation can not be productive and efficient when we consider meeting enterprise needs and quick adaptation to the changing requirements. Hence, in this paper, a metamodel for web engineering, called PSM4MSF ("Platform Specific Metamodel for Microsoft Sharepoint Foundation"), is proposed and a model driven software development methodology based on this metamodel is introduced. PSM4MSF both extends and increases the abstraction layer of Microsoft Sharepoint object model. Utilization of the metamodel inside a model driven architecture provides automatic generation of web based enterprise solutions from user defined models. The methodology based on PSM4MSF has been successfully used in the development of various enterprise systems. The usage is exemplified in the paper by discussing the implementation of a web system which is currently used in one of the prominent petrochemical companies in Turkey in order to process data gathered from company's planning and production departments for the evaluation of company's key performance indicators.

Keywords: Metamodel, Model Driven Development, Web Engineering, Microsoft Sharepoint Foundation

* E-posta: geylani.kardas@ege.edu.tr

1. Giriş

Web tabanlı yazılım sistemlerini model güdümlü geliştirmek için çeşitli web mühendisliği metodolojileri (örneğin [1], [2] ve [3]) bulunmaktadır. Bu metodolojilerin kapsamında, sistem modelleri sadece analiz ve tasarım amaçlı kullanılmamakta; web uygulamalarının otomatik üretilmesi için de modellerden yararlanılmaktadır. Ancak kabul görmüş ve üreticisi tarafından gelişimi sürekli desteklenen belirli bir uygulama geliştirme çatısı ya da platform teknolojisi olmaksızın otomatik yazılım üretimi, kurumun iş ihtiyaçlarının karşılanması ve geliştirilen uygulamaların değişen ihtiyaçlara hızlı bir şekilde uyum sağlayabilmesi açısından hem üretken hem de etkin olamamaktadır. Diğer yandan platform bağımlı yazılım geliştirme teknolojileri için genel amaçlı üstmodellere dayalı model eşleme ve dönüşümlerin gerçekleştirilebilmesi ve otomatik uygulama üretimi pratikte çok zordur. Bunun belirgin sebebi olarak platform bağımlı teknolojilere ait üstmodellerin uygulamadaki ihtiyaçlardan, gelişen teknolojilerden gelmesi ve doğası gereği hedef bir platform teknolojisi içermesi; platform bağımsız üstmodellerin ise daha çok akademik olarak çalışılması ve tüm platformlar için genel bir tanımlama çabasında olmasıdır. Bu makalede, bu iki yaklaşım köprülenerek, Microsoft Windows platformuna özgü bir teknoloji olan Sharepoint Temeli'nin ("Sharepoint Foundation")¹ [4] nesne modelini genişleten ve soyutlama seviyesini yükselten, web tabanlı kurumsal çözümler geliştirmeye yönelik bir yazılım ürün hattı oluşturabilecek PSM4MSF ("Platform Specific Metamodel for Microsoft Sharepoint Foundation") adı verilen bir web mühendisliği üstmodeli önerilmekte ve bu üstmodele dayalı bir model güdümlü web tabanlı yazılım geliştirme yöntemi tanıtılmaktadır.

Önerilen üstmodel, Microsoft Sharepoint Temeli nesne modelinin içerik yönelimli ayrık kavramlarını bakış açılarına bölüp bu kavramların arasındaki ilişkileri yeniden tanımlayarak soyutlama seviyesini yükseltmektedir. Web mühendisliği alanında araştırılmış uygun bir platform bağımsız üstmodeli SharePoint platformuna özgü teknoloji spesifik ilgileri kapsayacak şekilde genişletmek ya da Microsoft'un önerdiği MSF nesne modelinin (MSFNM) soyutlama seviyesini yükseltip web mühendisliği alanında kabul görmüş ortak görünümlere parçalayıp yeniden tümleştirmek gibi iki alternatifin birleşimi üzerinden bir çözümleme ile tamamen yazılım geliştirme ihtiyaçlarına dayalı yeni bir üstmodel önerilmiştir. "Object Management Group"'un (OMG) önerdiği Model Güdümlü Mimari'ye (MGM) [5] dayanan bir yazılım geliştirme yöntemi ile de PSM4MSF modellerinden web tabanlı kurumsal çözümlerin otomatik olarak üretilmesi sağlanmaktadır.

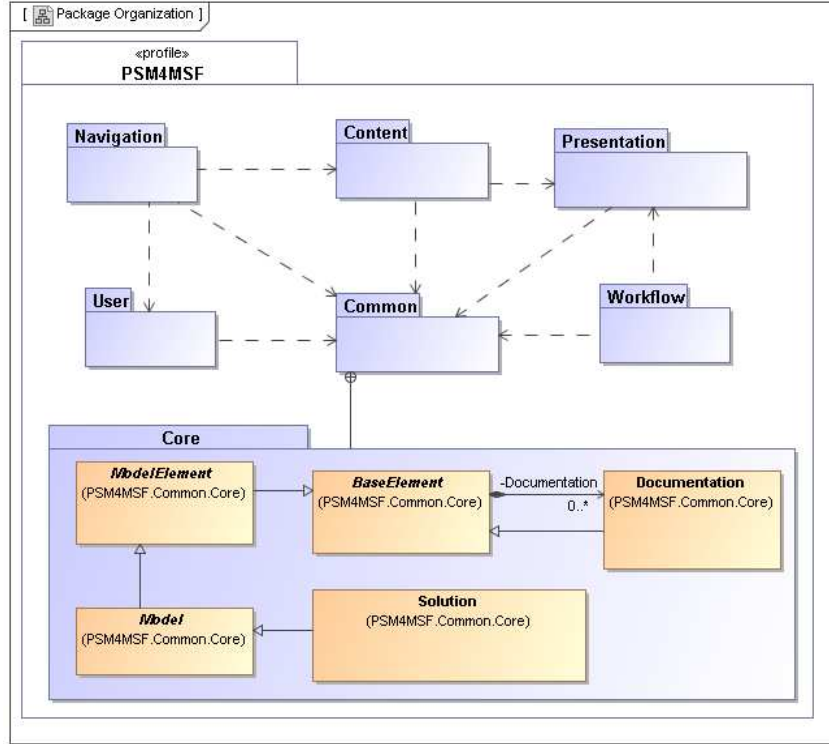
Makalenin geriye kalan kısmı şu şekilde düzenlenmiştir: 2. bölümde PSM4MSF üstmodeli tanıtılmaktadır. PSM4MSF'e dayanan model güdümlü yazılım geliştirme yöntemi örnek bir durum çalışması üzerinden 3. bölümde anlatılmaktadır. 4. bölümde PSM4MSF'e dayalı olarak önerdiğimiz geliştirme yönteminin değerlendirilmesi yer almaktadır. 5. bölümde literatürde yer alan ilgili diğer çalışmalar aktarılmıştır. 6. bölümde ise sonuç ve ileriye yönelik çalışmalar bulunmaktadır.

2. PSM4MSF Üstmodeli

Hem kullanıcı hem de teknolojik ihtiyaçların sürekli gelişim içerisinde olması, Web sistemlerinin tasarımının ve gerçekleştiriminin de devamlı bir gelişim içerisinde olmasını güdümler. Geliştirme sürecinin herhangi bir aşamasındaki modellerin kolaylıkla bu gibi değişikliklere uyum sağlaması için gereken esnekliği yakalamada PSM4MSF, geliştirilmesinin ilk fazından itibaren katı bir şekilde ilgilerin ayrımı ("separation of concerns") prensibine dayanır. Öncül versiyonu [6]'da tanıtılan PSM4MSF hem "Meta-Object Facility" (MOF) [7] tabanlı bir üstmodel hem de bir "Unified Modeling Language" (UML) [8] Profili olarak geliştirilmiştir. PSM4MSF'in tüm yapısı bir web uygulamasının içerik, görünüm, gezinim ve sunum gibi değişik ilgilere ("concerns") göre paketlere bölünerek ve her pakete bir model türü atanarak, bir web sisteminin farklı bakış açıları ("viewpoints") ile geliştirilebilmesi sağlanmaktadır. [6]'da web sistemi geliştirimi için sadece içerik modeline yoğunlaşmışken bu çalışmada PSM4MSF'in sunduğu tüm bakışaçıları ele alınmakta ve model güdümlü bir yazılım geliştirme süreci içerisinde kullanımları örneklenmektedir. Öte yandan [6]'da Windows Sharepoint servislerinin geliştirilmesi için UML genişletme ve profillemeye temelli bir üstmodel üretim mekanizması sonucunda PSM4MSF'in öncülü olan PSM4WSS'in nasıl hazırlandığı tartışılmaktadır. Örneğin yine içerik modeli için stereotip oluşturma detaylı bir şekilde [6]'da yer almaktadır. PSM4MSF'in geliştirilmesinde de benzer bir yol izlenmiştir ancak bu makalede PSM4MSF'in nasıl geliştirildiği yerine PSM4MSF'in web sistem geliştiricilere sunduğu ilgi modelleri ve bunların web tabanlı kurumsal çözümlerin geliştirilmesinde nasıl kullanılacağı anlatılmaktadır.

¹ Makalede orijinali İngilizce olan ve çeviride anlam kaybı yaşanabileceği düşünülen önemli isimlerin, tanımlamaların ve terimlerin önce Türkçe karşılıkları daha sonra ("") içerisinde İngilizceleri verilmektedir.

Şekil 1'de PSM4MSF paketleri birbirlerine girişim ve bağlılık yönleri ile beraber gösterilmektedir. Çekirdek ("Core") Paket, diğer tüm paketlerde ortak kullanılan elemanları içermektedir. Temel Eleman ("BaseElement") PSM4MSF elemanlarının neredeyse tamamına yakınının soyut süper sınıfıdır. Temel Eleman'dan kalıtlanan bütün PSM4MSF elemanları Dokümantasyon ("Documentation") elemanı ile bir veya birden fazla metinsel tanım barındırabilirler. Eğer bir elemanı ayırt etmek için Temel Eleman'dan gelen ID yeterli değil ise modellenen ortamdaki kavramları yansıtmaları açısından Model Elemanı ("ModelElement") üstvarlığı kullanılmaktadır.



Şekil 1. PSM4MSF paket düzeni

Soyut bir sınıf olan *Model*, Model Elemanı öğelerinden oluşur. Ayrıca *Model*'in kendisi de Model Elemanı'nı genişletmektedir. PSM4MSF 5 farklı model önerir: İçerik Modeli, Gezinim Modeli, Kullanıcı Modeli, Sunum Modeli ve İş Akışı Modeli. Her bir somut model bulunduğu paketdeki sınıf elemanlarını kendi içerisinde kompozisyon ilişkisi ile barındırır. Farklı bir görünümdeki somut bir elamana ise ilişkilendirme bağlantısı ("association") ile referans gösterir. Çözüm ("*Solution*") elemanı beş farklı modeli sarmalayan ve aynı zamanda kendisi de bir model olan bir sınıftır. Farklı olarak diğer somut 5 modelin her birini kompozisyon ilişkisi ile içerir. Tasarımcı bu çözüm ile üretilecek web uygulamasına bir proje ismini ve kodunu bu eleman sayesinde verebilir. Çözüm elemanı, Temel Eleman'ın niteleyici ve model ilişkilerini Model Elemanı üzerinden kalıtlamış olur.

PSM4MSF üstmodelinin Ortak ("*Common*") Paketi, kavramsal olarak belli bir görünüm içerisine ait olarak değerlendirilemeyen ve genellikle her görünümde kullanılabilen kavramları içerir (Şekil 1'de ortada). PSM4MSF Kullanıcı ("*User*") Paketi, kullanıcıları, kullanıcı gruplarını, rolleri, rollerin izin düzeylerini ve kullanıcı grupları ile roller arasındaki ilişkileri tanımlar. Kullanıcı grupları ve roller arasındaki ilişki sayesinde içeriklere (örneğin bir liste, web veya formlara) erişim kontrolleri ("access control") bu kullanıcı grupları üzerinden tanımlanarak çeşitli yetkilendirmeler gerçekleştirilmiş olur.

PSM4MSF İçerik ("*Content*") Paketi bir web uygulamasının tüm statik yapısını tanımladığı alana özgü kavramlar ile birlikte modellemeyi sağlamaktadır [6]. Bu paketin tanımlamalarına göre bir *Web Uygulaması* bir ya da daha fazla *Site Topluluğu*'ndan oluşur. Her bir site topluluğunun üst seviyede bir tane *Kök Web Sitesi* vardır. Bir kök web sitesi, içerisinde hiyerarşik bir biçimde, isteğe bağlı olarak yeni web siteleri ile çoğaltılabilir. Bir web sitesi birden fazla liste içerebilir ve her bir listenin bir ya da daha fazla *Görünümleri* ("*View*") olabilir.

PSM4MSF *Gezirim ("Navigation")* Paketi farklı yetkilerdeki kullanıcıların geliştirilecek çözüm ("solution") içerisindeki içeriklere ("contents") erişim şeklinin analiz edilmesini sağlamaktadır. Bu paketin içerisindeki *Gezirim Modeli ("NavigationModel")* aynı zamanda MSF platformu üzerinde çeşitli bileşenlerin geliştirilmesine olanak sağlar. Paketin üstmodelinin omurgasını, soyut üstvarlıklar olan *Gezinti Düğümü ("NavigationNode")* ve *Bağlantı ("Link")* ile bu sınıfların arasındaki ilişkiler oluşturur. Bu paketin sunduğu bakış açısına göre web, sayfa, görünüm ("view") ve formlar aslında kullanıcıların hedeflenen sistem üzerinde okuma ve yazma gibi çeşitli eylemleri gerçekleştirmek için erişmek istedikleri birer içeriktir. İşte her bir erişime açık içerik birimi ayrıca birer *Gezirim Düğümü* elamanı ile ilişkilendirilir.

PSM4MSF *Sunum ("Presentation")* Paketi formların ve sayfaların yerleşimi ve web uygulamasının bir takım davranışlarını modelleyen üstvarlıkları ve bunların ilişkilerini içerir.

Son olarak PSM4MSF *İş Akışı ("Workflow")* Paketi iş süreçlerini, PSM4MSF'in web uygulaması modeli ile tümleştiren model elemanlarını içerir. Bir iş akışı üst sınıfının 4 farklı türde gezirim formuna referansı vardır: Düzenleme Formu, Durum Formu, Örneklem ("Instantiation") Formu ve Görev Formu. Bu referanslar sayesinde PSM4MSF örnek modellerini okuyan kişi bir iş akışına gezirim üzerinden nasıl erişilebileceğini ve gezirimin daha sonra nasıl devam edeceğini anlamaktadır. PSM4MSF'in burada yer kısıtlarından ötürü sadece tanıtımı yapılan tüm paketleri ve üstvarlıklarına ait detaylı bilgiye [9]'dan erişilebilir.

3. PSM4MSF Üstmodeline Dayalı Model Güdümlü Yazılım Geliştirme Yöntemi

Bu bölümde, içerik etkin karmaşık kurumsal web uygulamalarının hazırlanması için geliştirdiğimiz ve PSM4MSF'e dayanan model güdümlü yazılım geliştirme yöntemi gerçek bir durum çalışması üzerinden anlatılmaktadır.

PSM4MSF'e dayalı model güdümlü yazılım geliştirme yöntemi, 2009-2010 yılları arasında çeşitli sektörlerde bir çok kurumsal sistemin geliştirilmesinde kullanılmıştır ve günümüzde de kullanılmaya devam etmektedir. Bu bölümde anlatılan iş süreci senaryosu da uygulamaya alınan gerçek bir senaryodur ve geliştiriminin PSM4MSF'e dayalı süreç ve araçlar ile yerine getirildiği yazılım sistemi ülkemizin önde gelen petrokimya şirketlerinden birinde aktif olarak kullanılmaktadır.

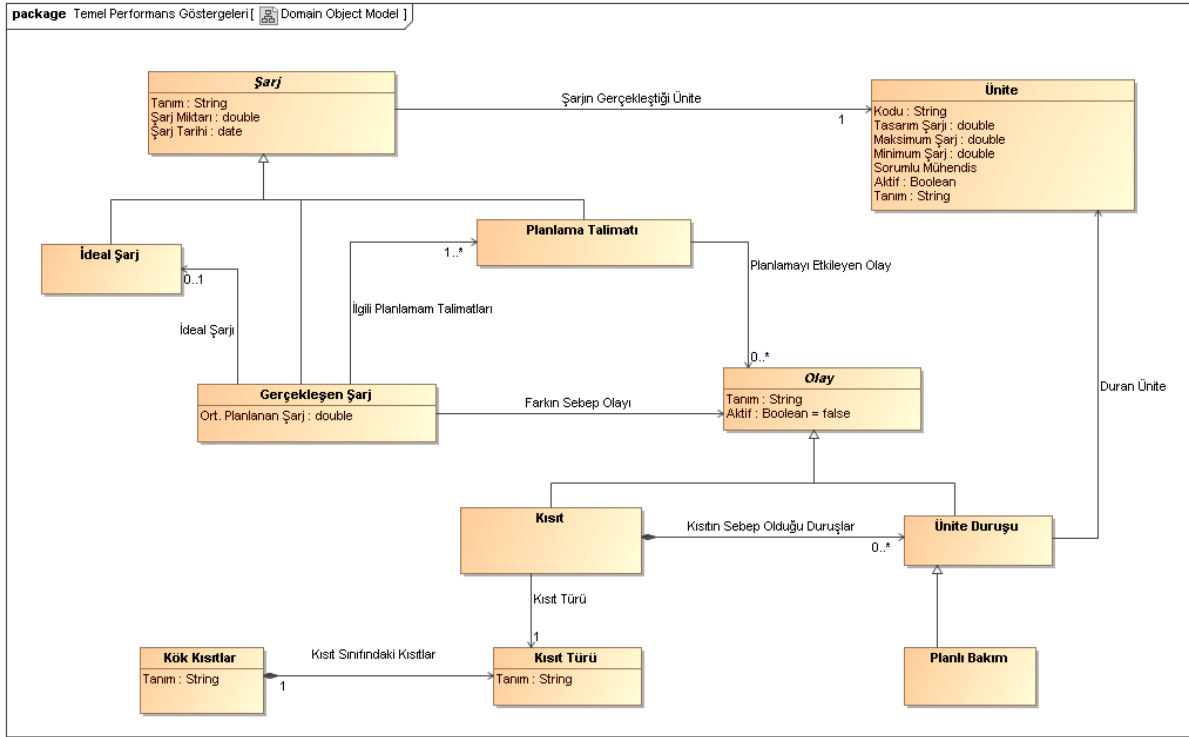
Durum çalışmamız, sözkonusu şirketin üretim kaybı, kapasite kullanımı, üretim birimlerinin (üniteler) işletme emirlerini yerine getirebilme oranı gibi temel performans göstergelerinin (TPG) belirlenmesinde planlama ve üretim kesimlerinden gerekli bilginin toplanması ve işlenmesini yöneten bir web sistemi çözümü gerçekleştirimidir. Planlanan (bir ünitenin ikinci bir talimat gelene kadar çalışması gereken miktar), ideal (ünitenin çalışabilir kapasitesi) ve gerçekleşen şarjların (ünitenin m³/gün birimindeki gerçek üretimi) takibi yanında, gerçekleşen şarjların planlanandan sapmasının kurum içerisinde gerçekleşen ve kök-sebeç çözümlemesi ("root-cause analysis") yapılarak sınıflandırılmış olaylar ile ilişkilendirilmesi sistemin işlevsel ihtiyaçlarıdır.

Üretim planlama müdürlüğü, ana üretim planına uygun olarak mevcut piyasa koşulları ve saha durumuna göre işletme departmanına günlük ünite çalışma talimatlarını gönderir. Ünite operatörü ise eğer planlanan şarjdan bir sapma olmuş ise bunun gerekçesini bir tür olay ("event") ile ilişkilendirerek gösterir. Olay soyut bir kavramdır; ancak sahada önceden başlama tarihi belirlenen ve tahmini bir bitiş tarihi olan planlı bakım ve önceden kestirilemeyen bir başlangıç tarihi olup giderilme tarihine kadar varlığını sürdüren kısıt ise olayın somut türleridir. Sapmaya, ilgili ünite ile aynı alt üretim süreci içerisinde bulunan başka bir ünitenin duruşu kısıt (neden) olabilirken; aynı ünitenin henüz başlatılmamış önceki bir duruşu da kısıt olabilir. Öte yandan operatörün gösterdiği kısıt başka ünitelerdeki duruşu da etkileyebilir. Bu durumda etkilenen ünitenin operatörü yeni bir kısıt listelemeden ilgili kısıtı göstererek sorumlu olduğu üniteye durumu belirtir. Kısıtlar, temel performans göstergelerinin hesaplamasında dikkate alınan "bakım", "işletme", "harici" gibi belli kök sebeplerden; yangın, enstrüman arızası gibi çeşitli alt kısıt türlerinden oluşmaktadır. Operatör ünite defterinde aşağıdakine benzer bir günlük tutar.

"17/04/2010 16:45 tarihinde gerçekleşen soğutma suyu kulesi yangını sebebiyle U-7 ünitesi 17/04/2010 21:00 tarihinde duruşa geçerek, 17/03/2010 08:00 tarihli ilgili planlama talimatındaki 600 m³/gün'lük planlanan şarjın 150 m³/gün altında kalmıştır. Olay takvime işlenmiş ve harici kök sınıfa ait soğutma suyu kesintisi alt kısıt türü altında sınıflandırılmıştır."

Yukarıda geçen tüm alan kavramları, sistem analistinin alan uzmanı paydaşlar ile yaptığı detaylı bir ihtiyaç mühendisliğinin ardından alan parçalaması ("domain decomposition") yöntemi ile şarjlar –ideal şarj, planlama

talimatı ve gerçekleşen şarj-, olaylar –kısıt, ünite duruşu ve planlı bakım-, kök kısıtlar –kök kısıt ve kısıt sınıfı- ve üniteler –ünite- olarak farklı seviyelerde sınıflandırılmıştır. Problem alanımızın kapsamındaki anahtar varlıkları, onların özelliklerini ve ilişkilerini yapısal olarak gösteren bir sözlük, hiçbir gerçekleştirim detayı içermeyecek bir şekilde alan nesne modeli “Domain Object Model” (DOM) [10] olarak Şekil 2’deki gibi belirlenmiştir. Bilindiği üzere bu kavramsal modelde varlıklar nesnelere karşılık gelerek aslında nesne yönelimli analizin temelini oluşturmaktadırlar [8].



Şekil 2. Temel Performans Göstergeleri Alan Nesne Modeli

İhtiyaç mühendisliğinin çıktıları arasında bir diğer ürün ise fonksiyonel ihtiyaçların sözlü olarak anlatıldığı, aynı zamanda alan nesne modelini de oluşturmada kullanılan yapısal kullanım senaryoları (“Use Case Scenarios”) dokümanlarıdır. “Operatör’ün Şarj Düşmesi Görevi” isimli bir kullanım senaryosu örneği Şekil 3’te gösterilmiştir.

Operatörün Günlük Şarj Düşmesi Görevi (Hedef Paydaş: Ünite Operatörleri, Ünite Mühendisleri)

Gerçekleşen şarj ile planlanan şarj arasında $\pm n\%$ ’den $-n$:TPG Yöneticisi tarafından ayarlanabilmektedir- fazla fark var ise ünitenin sorumlu operatörü görevlendirilir. Herhangi bir duruş o gün için söz konusu ise karşılaştırma yapılmaz. Sistemde operatörler için kontrol/takip edilmesi gerekli görevlendirmelere dair e-posta ile bir uyarı mekanizması mevcuttur. Sistem iki türlü görevlendirme yapar:

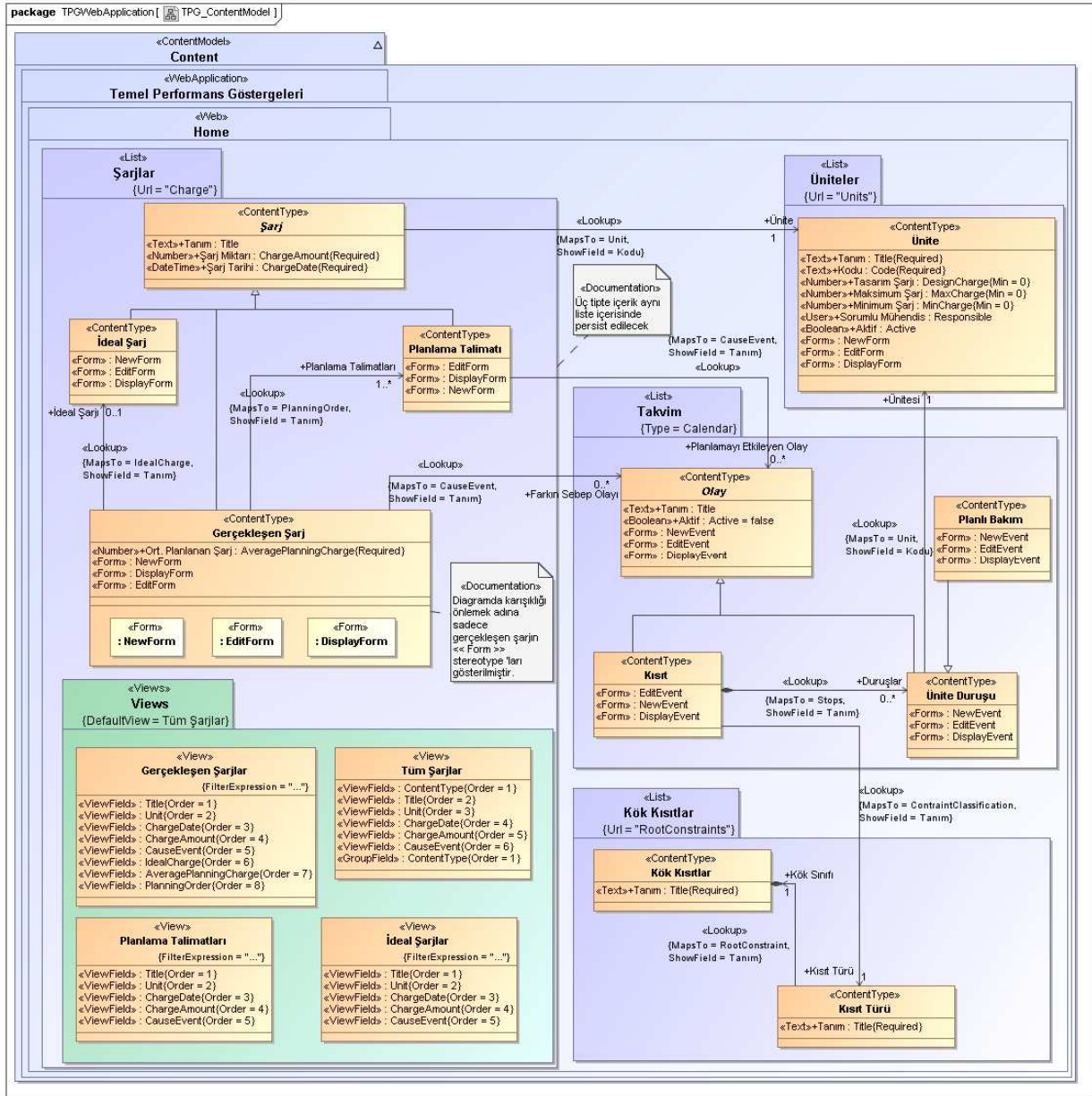
- Eğer “Şarj Düşümü” ise operatör, görev formu üzerinden bu şarj düşümünün sebep olayı olarak bir veya daha fazla kısıt olay ile ilişkilendirebilir. İlgili bir kısıt yok ise olay takvimine yeni bir kısıt ekleyebilir. Bu yeni eklediği olayların kök-sebebini sınıflandırması istenir.
- Eğer “Ünite Duruşu” ise operatör, görev formu üzerinden bu üniteyi bir Duruş ile ilişkilendirebilir. İlgili bir Ünite Duruşu yok ise olay takvimine yeni bir Duruş ekleyebilir.

Şekil 3. Operatör’ün Şarj Düşmesi Görevi Kullanım Senaryosu

Yazılım mimarı ve mühendisleri, belirtilen alan-nesne modeli ve kullanım senaryolarını, paydaşların bu model ve dokümanlarda belirttikleri fonksiyonel ihtiyaçlarını, teknik detaylarını da kapsayacak şekilde karşılayan ve çözümün nasıl gerçekleştirileceğini modelleyen nesne yönelimli tasarıma geçişte kullanırlar [8]. Önerdiğimiz yazılım geliştirme yöntemi işte bu noktada devreye girerek; “sistem analisti/alan uzmanı” ile “yazılım mimarı/yazılım mühendisi” arasındaki, analizden tasarıma geçişteki boşluğu köprüler ve bu iki kesimdeki modelleri sadece iletişimde kullandıkları ve çözümü doğruladıkları dokümandan bir adım öteye taşıyarak çalışan bir sistemin üretilmesinde kullanılmasını amaçlar.

Nesne yönelimli bir yaklaşıma sahip olan bu alan-nesne modeli daha sonra içerik yönelimli bir şekilde PSM4MSF üstmodelinin içerik paketi elemanları ile zenginleştirilerek sistem mimarı tarafından tekrar düzenlenir. Her ne kadar, üstmodelin kullandığı metafor içerik olsa da hala soyutlama seviyesi oldukça yüksektir ve içerik paketi varlıklarının tanıyan bir sistem analistinin PSM4MSF'in UML profilini, UML'in genişletme mekanizması olan profillemeyi destekleyen bir UML editörü yardımıyla kolaylıkla tasarlayabileceği düzeydedir.

Senaryomuzda, DOM'daki üst kavramlar MSF nesne modelindeki listelere, alt kavramlar ise içerik tiplerine eşleştirilmiştir. Mümkünse bir liste altına yerleşen içerik türlerindeki ortak özellikler, alana özgü soyut birer süper sınıf (şarj, olay) içerisindeki alt-alanlar olarak toparlanmıştır. İçerik tipleri arasındaki ilişkilendirme ise MSF nesne modelindeki arama alanının ("lookup-field") soyutlaması ile tamamlanır. Şekil 4'te söz konusu örnek senaryonun içerik yapısının PSM4MSF üstmodeline uygun UML modeli resmedilmiştir.



Şekil 4. PSM4MSF içerik paketi UML profili uygulanmış bir kullanıcı örnek modeli

Bu içerik modeline dayalı olarak, Şekil 3'teki kullanım senaryosu ışığında bir kullanıcı görev formu sunum modeli kapsamında ihtiyaçlara uygun olarak Şekil 5'teki gibi türetilir. Bu form gerçekleşen şarj ile planlanan şarj arasında

bir fark olduğunda operatörün önüne açılmakta ve operatörden bu duruma sebep olan yavaşlama ya da ünite duruşu için bir olay girmesi beklenmektedir.

The screenshot displays a software form titled "Şarj Düşmesi Operatör Görev Formu (Temel Performans Göstergeleri)". The form is structured as follows:

- TopToolBar**: A horizontal bar at the top.
- Information**: A box with an information icon (i) on the right.
- Horizontal Layout Container**: A dashed box containing two columns of sections:
 - Left Column (Şarj Başlık Bilgisi)**: Contains three input fields labeled "Başlık", "Ünite", and "Tarih".
 - Right Column (Şarj Değerleri)**: Contains two input fields labeled "Planlanan Şarj" and "Gerçekleşen Şarj".
- Vertical Layout Container**: A dashed box containing:
 - Edit Section (Kök Sebep Girişi)**: A box with two input fields labeled "Yavaşlama Sebebi" and "Duruş Sebebi".
 - Collection Section (İlgili Planlama Talimatları)**: A table-like structure with three columns: "Başlık", "Tarih", and "Şarj", each with a checkbox.
- BottomToolBar**: A horizontal bar at the bottom with two buttons labeled "Kaydet" and "İptal".

Şekil 5. Operatörün Şarj Düşmesi Görev Formu sunum modeli

Bir kullanıcı formunun ilgili içerik ile bağlantısı dikkat çekicidir. Bu kullanıcı formu amacına uygun olarak "Gerçekleşen Şarj" içerik tipi ile ilgili alanları sunacağından ve kullanıcıdan gerçekleşen şarjın alan modelinde belirlenmiş ilişkiler doğrultusunda bilgi girmesi beklendiğinden, gerçekleşen şarj içerik ("context") tipi bağlamında açılmalıdır. Yatay Yerleşim Taşıyıcısı'nın ("Horizontal Layout Container") şarj başlık bilgisi ve şarj değerleri olarak iki bölge içermesi ve her birinde "gerçekleşen şarj" içerik tipine ait ilgili alanların kullanıcıya sunulması arzu edilmektedir. Dikey Yerleşim Taşıyıcısı'nda ("Vertical Layout Container") ise bir adet bilgi girme bölgesi ve ortalama planlanan şarjı bulmada kullanılan ünite planlama talimatlarına dair başlık, tarih ve şarj kolonları olan bir koleksiyon bulunmaktadır. Bilgi girme bölgesinde ise operatörün yavaşlama ve duruş sebebi girmesi için seçimli bir liste yer alır. İçerik modelinde gerçekleşen şarj ile planlama talimatları arasındaki 1'e N ilişkisi (Şekil 4'te Şarjlar listesinin ortasında ilgili üstvarlıklar arasında gösterilmiştir) form modeldeki bir "Collection Section" örneği (Şekil 5'te alta yer alan İlgili Planlama Talimatları) tarafından planlama talimatları bilgisinin kullanıcıya liste şeklinde gösterilmesi amacıyla kullanılmaktadır. Böylelikle bu işlevsellik bir tek bile kod yazılmadan model güdümlü olarak gerçekleştirilmiş olur. Bu formu oluşturmada kullanılan üstmodel, önerdiğimiz bileşenler ve sistemin otomatik ürettiği Şekil 6'daki "Operatör'ün Şarj Düşmesi Görev Formu" gibi ekranlar tamamen öneri kapsamında gerçekleştirilen sistemin birer çıktılarıdır. MSF üstmodelinde bulunmayan ve Sharepoint platformu tarafından doğrudan desteklenmeyen bu yazılım bileşenlerinin üretilmesine ek olarak önerdiğimiz yazılım sistemi, MSF

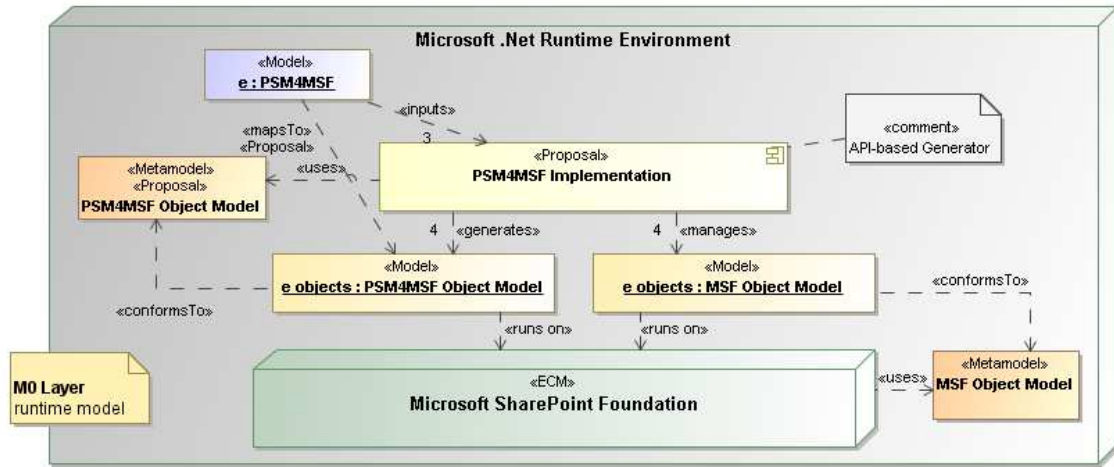
platformuna özgü bileşenlerin bu modellere göre organize edilmesini, tümleştirilmesini ve özelleştirilerek kullanılmasını sağlar.

Şekil 6. Operatörün Şarj Düşmesi Görev Formu ekran görüntüsü

Yer kısıtları nedeniyle bu makalede diğer modelleri oluşturan tüm kavramların tanıtılması ve bu kavramlara dayalı olarak PSM4MSF görünümünün modellenmesi detaylandırılmamıştır. İçerik modelinin “eXtensible Markup Language (XML) Metadata Interchange” (XMI) çıktısı, PSM4MSF modellerini işleyebilen bir uygulama sunucusuna diğer görünümün modelleri ile birlikte verildiğinde, modelden koda dönüşümler sonrasında tam otomatik olarak özel çözümün içeriği üretilir ve kullanıma hazır hale gelir. Şekil 7’de PSM4MSF modellerinden MSFNM’in türetilmesi ve web uygulamasının üretilmesi bir UML Bağımlılık Diyagramı içerisinde mimari açıdan gösterilmiştir.

PSM4MSF’e dayalı yazılım çatısı MSF platformunu hem genişletir hem de platformun uygulama sunucusunda çalışan örneğini kullanır. PSM4MSF modellerinden MSFNM için bir web uygulaması kurmayı (“deployment”) gerektiren modellerin çıkarılması ve gerekli dönüşümlerin yapılması PSM4MSF çatısı tarafından gerçekleştirilir. Bu modeller hem metin bazlı hem de .NET nesneleri olarak oluşturulabilir. Sonrasında üretilen modelleri girdi olarak kullanan MSF alt yapının kurulmasını gerçekleştirir. Son olarak PSM4MSF’e dayalı çatı PSM4MSF’in modellerine uygun olarak web uygulamasının statik yapısının üretilmesini tamamlar. Davranışsal yapısı ise Sharepoint’in alt yapısından gelen olay yakalayıcıları (“Event Receiver”) ve/veya Sharepoint’in iş akış modelleri ile 2 türlü gerçekleştirilir. Olay yakalayıcıları, bir içerik biriminde (liste, liste ögesi gibi) önceden tanımlanmış bazı olayları (öge oluşturma, öge silme gibi) senkron veya asenkron olarak yakalayan ve uygulamaya özel kullanıcı kod bloklarını çalıştıran olay-bazlı bir programlama modelidir. İş akışları ise model güdümlü olarak bir iş süreci (“business process”) ile tanımlanan akışın üzerindeki bir takım aktivitelerin sistem kullanıcıları ya da web servisleri tarafından tamamlanmasını koordine edebilen bir yapıdır. İş akışları ayrıca model güdümlü olarak “Business Process Execution Language”ın (BPEL) [11] platforma özgü örnekleri olan Sharepoint durum makinesi iş akışı (“Sharepoint State Machine Workflow”) ya da Sharepoint sırasal iş akışı (“Sharepoint Sequential Workflow”) şeklinde

gerçekleştirilebilir. PSM4MSF'in iş akış modeli ("workflow model") ile Sharepoint iş akışlarının entegrasyonu diğer görünümlemler ile ("NavigationModel", "ContentModel", vb.) sağlanır.



Şekil 7. PSM4MSF yazılım çerçevesi bünyesinde örnek modellerin çalışması

4. Değerlendirme

PSM4MSF üstmodeli sayesinde web mühendisliği çalışma alanına özgü kavramların Sharepoint platformu üzerinde kullanılabilirliği sağlanmıştır. Böylece Sharepoint platformuna özgü içerik yönetimine dair kavramların haricinde bir web uygulamasının geliştirilmesi sırasında ihtiyaç duyulan ilgilerin ve bu ilgilerin kapsadığı varlıkların yönetimine ve modellenmesine geçilmiştir. Modelleri belirtmede platforma özgü teknolojiye belli bir ölçüde soyutlanmış, belli ilgilere platform bağımsız seviyedeki UWE [2] benzeri üstmodel varlıklarına yaklaşılmıştır. Böylelikle bir yazılım mühendisi ya da mimarı, problem alanı ve fonksiyonel ihtiyaçlar üzerine daha fazla yoğunlaşabilmektedir.

PSM4MSF modellerinin bir çok senaryoda alan uzmanları ile iletişimde ve hedef sistemi kavramada anlaşılabilirliği arttırdığı gözlemlenmiştir. Bir alan uzmanı ya da sistem analisti problem alanına dair kavramları hızlı bir şekilde modelleyebilmekle sistemin içeriğini ve formlarını otomatik olarak üretebilmekte ya da var olan çalışma modellerini değiştirmek suretiyle bu içerik ve formları üzerinde değişiklikler yapabilmektedir.

Tasarım modellerinin analiz modelleri ile aynı olması, yani işletebilen modeller ("executable models") sayesinde çalışma zamanında sistem üzerinde genişletmeler ya da düzenlemelerin yapılabilmesi olanaklı kılınmıştır. Bunun bir adım ötesinde, alan uzmanlarına da sistem modellerini düzenleyebilme yetkisi verilerek bakım maliyetleri azaltılmış ve sonuç olarak ihtiyaçların değişmesi durumunda uygulamanın değişime hızlı uyum sağlayabilmesi sağlanmıştır. Böylelikle model güdümlü yaklaşım ile gerçekleştirdiğimiz PSM4MSF üstmodeline dayalı yazılım çatısı kullanılarak geleneksel kod merkezli programlama eforuna kıyasla önemli ölçüde üretkenlik artmış, geliştirme çevrimleri düşürülmüştür. Örneğin kullanılan kaynaklar (personel, donanım, vb.) sabit kalmak koşulu ile 3. bölümde anlatılan TPG sisteminin geliştirilmesinde beklenen 6 aylık geliştirme süresi 2 aya inmiştir. Sistem maliyetinde de adam/ay miktarındaki düşüşten dolayı yaklaşık %50 bir azalma elde edilmiştir.

Model güdümlü geliştirmenin anahtar önermelerinin hızlı gerçekleştiriminin sağlaması ve hedef sistemin otomatik üretilmesi olduğu [5] bilinmektedir. Ancak, her ne kadar tüm yazılımların, kullanıcı modellerinden otomatik bir şekilde üretilmesi hedeflense de, günümüzdeki diğer mevcut model güdümlü yaklaşımlarda olduğu gibi PSM4MSF'e dayalı geliştirmede iş akışı görünümü dikkate alındığında sistemin ihtiyaç duyduğu yazılımın %100'ü otomatik olarak üretilmemektedir. Tüm çözümlerin mutlaka genelleştirilen üstmodelin sınırları dışında probleme özgü özel ihtiyaçları da içermesi sebebiyle %100 kod üretimi gerçekleştirilememiştir. İş akışı görünümünde karşılaşılan sıkıntının, bu alanda olgunlaşan ve bir OMG standardı olan iş süreçleri modelleme notasyonunun (BPMN 2.0)² işletebilir olma uyumluluğu çalışmalarının tamamlanmasından sonra BPMN ve PSM4MSF arasındaki model dönüşümleri hazırlanarak giderilebileceğine inanılmaktadır.

² <http://www.omg.org/spec/BPMN/2.0> (son erişim: Temmuz 2011)

5. İlgili Çalışmalar

Literatürde MSF'in kendi nesne modeli ve onun XML şeması dışında web mühendisliği kavramlarını ve bakış açılarını içeren platforma özgü bir üstmodel bulunamamıştır. İlgili çalışmalar kapsamında daha çok platform bağımsız üstmodel önerilerinin olduğu söylenebilir.

Web mühendisliği alanında çalışan araştırmacıların geliştirdikleri alana özgü dilleri ve web modellerini tanımlamak için UML genişletme ve üstmodelleme yöntemlerinden faydalandıkları görülmektedir (örneğin [2], [12] ve [13]). W2000 [13] ile karmaşık Web uygulamalarının modellenmesi için bir gösterim önerilmektedir. Tüm kavramların temeli farklı gösterim elemanlarını karakterize eden ve bunların arasındaki ilişkileri tanımlayan bir üstmodelle dayanmaktadır. Conallen [12] standart UML diyagramları ile bağlantılı olarak, sunucu sayfaları, kullanıcı sayfaları, formlar, iskeletler ("frames") gibi tipik web ürünlerini ("artifacts") içeren Web uygulama genişletmelerini ("Web Application Extensions") (WAE) önermektedir. Ancak WAE'nin gezinim ve sunum bakış açılarının yüksek seviyeli soyutlamasının yeterli olmamasından hareketle Go'omez ve Cachero [1] OO-H adlı yöntemleri ile web arayüz modeli sağlamak için UML diyagramlarını genişleten yeni bir bakış açısı önermiştir.

WebML [3] ve UWE [2] önerilerinde, web uygulamalarının tasarımı ve geliştirilmesinde içerik, gezinim ve sunum gibi ayrı modeller bazında farklı ilgilere işaret edilmektedir. Ceri ve arkadaşları [3] WebML ("Web Modeling Language") yaklaşımında, kendilerine ait alana özgü bir modelleme ortamı ve kod üretme dili tanımlamışlardır. UWE yaklaşımı [2], UML uyumluluğu ve ticari platformların üstmodelleri içerisinde ortak kavramları bünyesinde bulundurması sebebiyle diğer web mühendisliği metotları içerisinde kendini ayırt etmektedir. Bu platform bağımsız üstmodeller arasında genel bir değerlendirme yapacak olursak UWE içlerinde MOF-uyumlu bir üstmodelle ve UML profiline sahip olması, diğer üstmodellere kıyasla web mühendisliği alanında farklı çalışmalarda sunulmuş üstvarlıkların genelini içermesi ve sunduğu bakış açıları ve model elemanları ile öne çıkmaktadır.

Çizelge 1'de yukarıda sözü edilen çalışmalarda sunulan üstmodeller ve bizim önerdiğimiz PSM4MSF'in karşılaştırmalı bir değerlendirmesi bulunmaktadır. Değerlendirme sırasında göz önüne aldığımız kriterler soyutlama seviyesi, desteklenen bakış açıları, kullanılan modelleme gösterimi ("notation"), kod üretimi ve araç desteğidir. Web sistemlerinin geliştirim süreci ve bunların dayandığı üstmodeller farklı soyutlama seviyelerine sahip olabilirler. Eğer bir üstmodel alt yapıdaki mimariye ait bileşenleri varlıklar olarak buldurmuyorsa bu üstmodelin soyutlama seviyesinin yüksek olduğu söylenebilir. Sistemin üzerinde çalıştırılacağı mimari ya da platforma ait bileşenlerin de üstmodelde yer alması durumunda üstmodelin soyutlama seviyesi düşük olacaktır. Bu açıdan PSM4MSF doğrudan Sharepoint platformuna ait varlıkları içerdiğinden soyutlama seviyesi "platforma özgü" olarak Çizelge 1'de yer almaktadır. UWE ve WebML üstmodelleri, PSM4MSF'e göre daha soyuttur, herhangi bir özel web mimarisine dayanmaz ve platform bağımsızdırlar.

Çizelge 1. PSM4MSF, UWE [2] ve WebML [3]'ün karşılaştırılması

	Soyutlama Seviyesi	Bakışaçıları	Modelleme Gösterimi	Araç Desteği	Kod Üretimi
PSM4MSF	Platforma özgü	İçerik, Sunum, Gezinim, Kullanıcı, İş Akışı,	UML gösterimi, MOF üst-üstmodeli	UML uyumlu araçlar	Çalışan uygulama ve çalışma zamanı bileşenleri
UWE [2]	Platform bağımsız	İçerik, Sunum, Gezinim, Kullanıcı, İş Akışı,	UML gösterimi, MOF üst-üstmodeli	UML uyumlu araçlar	Şablon veya iskelet kod üretimi
WebML [3]	Platform bağımsız	Yapısal, Üstmetin, Sunum, Kişiselleştirme	Özel gösterim	Özel tasarım aracı gerekli	Çalışan uygulama

İlgilerin ayrımı prensibine dayalı olarak bir web sistemi tasarlanırken farklı bakış açıları ile sistemin daha kolay geliştirilebilmesi sağlanabilir. Bu açıdan önerilen üstmodeller de çeşitli bakışaçıları sunmaktadır. PSM4MSF, UWE ve WebML üstmodellerini göz önüne aldığımızda PSM4MSF'in UWE ile isim ve ilgi açısından birebir örtüştüğü, WebML ile özellikle "içerik" bakışaçısı yönünden ayrıştığı söylenebilir. Örneğin bir web sisteminin içeriğinin

modellenmesini göz önüne aldığımızda WebML "yapısal" bakış açısında varlık-ilişki (E/R) diyagramlarını kullanırken PSM4MSF ve UWE "içerik" bakış açısında nesne modellerini kullanmaktadır.

Öte yandan modellemenin UML uyumlu ve MOF tabanlı olması bir kazanımdır. UML standardı da aynı zamanda MOF tabanlıdır ve geliştiriciler arasında yaygındır. UML Profillemeye mekanizması sayesinde herhangi bir UML uyumlu araç ile modelleme yapılabilir. UML profili, UWE ve PSM4MSF üstmodeli için alana özgü özel bir modelleme aracı geliştirmeksizin, profillemeye mekanizmasını destekleyen araçları kullanarak kurumsal bir web uygulamasının modellenmesini mümkün kılar. MOF tabanlı olsun ya da olmasın, UML'i desteklemeyen yaklaşımlarda özel bir araç desteği gerekmektedir. Eğer UML uyumlu bir araç yok ise bu web mühendisliği üstmodelini kullanacak geliştiricilerin ilgili yeni aracı tedarik etmeleri ve modelleme notasyonunu öğrenmelerini gerektirir. Bu özel ("proprietary") araçlar genellikle müşterilerin diğer araçlar ile uyumlu çalışmamalarına ve izole edilmiş geliştirme ortamlarının oluşmasına neden olur -ki bu tolere edilmesi zor ve maliyet artırıcı bir etkidir. Bu noktada WebML kendine has bir gösterime sahip olması ve özel tasarım araçlarına ihtiyaç duyması bakımından PSM4MSF ve UWE'ye göre dezavantajlıdır.

Modellerden kod üretimi kriteri dikkate alındığında PSM4MSF'e dayalı geliştirme süreci sonucunda çalışan bir uygulamanın ve bunun için gerekli çalışma zamanı bileşenlerinin elde edilebildiği söylenebilir. WebML'e dayalı yöntemde de çalışan uygulamaların doğrudan modelleme sonucu elde edilmesi mümkündür. Bunun için WebML modelleri önce platforma özgü ara modellere dönüştürülmekte daha sonra da bu modellerden yazılım kodlarının üretilmesi sağlanmaktadır. Ancak Çizelge 1'de listelendiği gibi UWE kod üretiminde PSM4MSF ve WebML kadar yetenekli değildir ve kod üretimi şablon veya iskelet seviyesindedir.

6. Sonuç ve İleriye Yönelik Çalışmalar

Bu makalede PSM4MSF isimli platforma özgü bir web mühendisliği üstmodeli ve buna dayanan model güdümlü bir yazılım süreci anlatılmıştır. Microsoft SharePoint uygulama çatısı üzerinde geliştirilen ve MGM'ye dayanan bir yöntem ile, önerilen üstmodelin kullanıcı modellerinden web tabanlı kurumsal çözümlerin otomatik olarak üretilmesi sağlanmaktadır.

PSM4MSF'e dayalı yazılım çatısı, bir çok kurumsal sistemin geliştirilmesinde başarıyla kullanılmıştır. Farklı projeler arasında ya da aynı proje içerisinde tekrarlanan süreçlerin model güdümlü olarak otomasyonu, otomatik üretilen prototipler ile projenin erken safhalarında kullanıcılardan hızlı geri dönüş sağlanması, geliştirilen bileşenlerin etkin tekrar kullanılabilirliği, ihtiyaçların değişmesi durumunda uygulamanın değişime hızlı uyum sağlayabilmesi gibi kazanımları ile PSM4MSF'e dayalı çatı, yazılım geliştirme ve bakım maliyetlerini önemli ölçüde düşürmüştür, iş süreçlerine ve iş mantığına uygun yazılım geliştirilmesini olanaklı kılmıştır. Makalenin 4. bölümünde de belirtildiği gibi PSM4MSF'e dayalı geliştirme sürecinin kod merkezli web sistemi geliştirmeye göre sistem geliştirme maliyetlerini yarı yarıya düşürdüğü ve ihtiyaç duyulan sistem geliştirme süresini zaman zaman 2/3'e ulaşan oranlarda azalttığı gözlemlenmiştir.

İleriye yönelik planlanan çalışma PSM4MSF ile UWE [2] üstmodeli arasında model dönüşümlerini sağlamaktır. Böylece içerik etkin yazılım sistemlerinin eksiksiz bir model güdümlü yazılım geliştirme yöntemine uygun olarak önce platform bağımsız seviyede tasarlanması ve otomatik model dönüşümleri ve kod üretimleri sonrasında MSF çerçevesi üzerinde gerçekleştirimi mümkün olacaktır. PSM4MSF ile UWE üstvarlıkları arasındaki eşlemeler şu an için tamamlanmıştır. Eşlemelere dayalı dönüşüm kurallarının yazılması ise hedeflenen ilk çalışmadır.

Kaynaklar

- [1] Go'mez, J. ve Cachero, C., "OO-H Method: Extending UML to Model Web Interfaces", *Information Modeling for Internet Applications*, van Bommel, P. (Ed.), Idea Group Publishing, Hershey, PA, ABD, 144-173 (2003).
- [2] Koch, N., Knapp, A., Zhang, G. ve Baumeister, H., "Uml-Based Web Engineering: An Approach Based on Standards", *Web Engineering: Modelling and Implementing Web Applications, Human-Computer Interaction Series, Part II*, Rossi, G., Pastor, O., Schwabe, D. ve Olsina, L. (Ed.), Springer, Londra, İngiltere, 157-192 (2008).
- [3] Ceri, S., Fraternali, P. ve Bongio, A., "Web Modeling Language (WebML): a modeling language for designing Web sites", *Computer Networks*, 33 (1-6): 137-157 (2000).
- [4] Pjalorsi, P., *Microsoft SharePoint 2010 Developer Reference*, Microsoft Press, ABD (2011).
- [5] Frankel, D. S., *Model Driven Architecture: Applying MDA to Enterprise Computing*, Wiley Publishing Inc., ABD (2003).

- [6] Erata, F. ve Kardaş, G., “Web Tabanlı Sistemlerin Geliştirilmesine Yönelik Platforma Özgü Bir Üstmodel”, *Üçüncü Ulusal Yazılım Mimarisi Konferansı (UYMK 2010)*, Ankara, Türkiye, 45-54 (2010).
- [7] Vandome, F. P., McBrewster, A. F. ve Miller, J., *Meta-Object Facility*, Alphascript Publishing, ABD (2010).
- [8] Larman, C., *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Prentice Hall, ABD (2004).
- [9] Erata, F., *Web Tabanlı Sistemlerin Geliştirilmesine Yönelik Platforma Özgü Bir Üstmodel*, Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü Uluslararası Bilgisayar Anabilim Dalı (2010).
- [10] Kelly, S. ve Tolvanen, J-P., *Domain-Specific Modeling: Enabling Full Code Generation*, John Wiley & Sons, Inc., New Jersey, ABD (2008).
- [11] Sarang, P., Juric, M. ve Mathew, B., *Business Process Execution Language for Web Services*, Packt Publishing, Birmingham, İngiltere (2006).
- [12] Conallen, J., *Building Web Applications with UML*, Addison-Wesley, ABD (2002).
- [13] Baresi, L., Colazzo, S., Mainetti, L. ve Morasca, S., “W2000: A Modelling Notation for Complex Web Applications”, *Web Engineering*, Mendes, E. ve Mosley, N. (Eds.), Springer, Almanya, 335-364 (2006).