

ORMAN AMENAJMAN MODELLERİNİN SİSTEM ANALİZİ VE NESNE TABANLI TASARIMI VE PROGRAMLANMASI

Özkan BİNGÖL¹, Emin Zeki BAŞKENT², Ali İhsan KADIOĞULLARI², Sedat KELEŞ³

¹Karadeniz Teknik Üniversitesi, Enformatik Bölümü, 61080 Trabzon, bingolo@ktu.edu.tr

²Karadeniz Teknik Üniversitesi, Orman Fakültesi, Orman Mühendisliği, Trabzon

³Çankırı Karatekin Üniversitesi, Orman Fakültesi, Orman Mühendisliği, Çankırı

ÖZET

Orman amenajman planlama sisteminin karmaşık olması, farklı planlama bileşenlerini içermesi, zamana ve mekana bağımlı olması, çok amaçlı olması ve sürekli yenileniyor (dinamik) olması plan Karar Destek Sistemlerinin (KDS) geleneksel yazılım mimarisi ile hazırlanmasını güçleştirmektedir. Ayrıca amenajman planlarının zengin içerikli olması da farklı planlama tekniklerini (simülasyon, optimizasyon, kombine optimizasyon vb) kullanmayı gerektirmektedir. Bu derece karmaşık ve dinamik bir yapı gösteren orman amenajman KDS'nin tasarımı ve kodlanması ancak sistem analizi prensiplerine ve nesne tabanlı programlama (NTP) tekniği ile mümkün görülmektedir. Aynı zamanda planlamalardaki güncel değişikliklerin ve yeniliklerinin KDS'ne anında yansıtılması için sistemin modüler yapıda hazırlanmasını gerektirmektedir.

Bu tebliğde, ülkemizde modern anlamda yeni geliştirilmiş ETÇAP yaklaşımı ile orman amenajmanı model yazılımının (ETÇAPSimülasyon ve ETÇAPOptimizasyon) genel mimarisi verilmiş ve nesne tabanlı programlama tekniğinin bütün özelliklerini yansıtan Birleşik Modelleme Dili (UML) diyagramları ile tasarımı ve kodlanması anlatılmıştır.

Anahtar Kelimeler: UML, ETÇAPSimülasyon, ETÇAPOptimizasyon, ETÇAP

SYSTEM ANALYSIS, OBJECT ORIENTED DESIGN AND PROGRAMMING OF FOREST MANAGEMENT MODELS

Developing Decision Support Systems (DSS) involving complex management planning systems with various components and multiple values to be sustained over time and space is quite hard to be programmed with conventional programming techniques. Due to the complex components and features of management planning, different quantitative decision techniques such as simulation, optimization and combinatorial optimization are required. The design and programming of a forest management model to accommodate such complexities can only be achieved with object oriented analysis and programming techniques. Furthermore, the model needs to be flexible and modular to incorporate the new developments into the planning system.

This paper is designed to demonstrate the system architecture of the new forest management planning concept called ETÇAP and explain the design and coding principles with Unified Modelling Language (UML) that represents all features of the object oriented programming languages.

Key Words: UML, ETÇAPSimülasyon, ETÇAPOptimizasyon, ETÇAP

GİRİŞ

Ormanlığın uluslar arası boyut kazanması, orman kaynaklarının giderek azalması, orman kaynaklarına olan talebin çeşitlenmesi ve artması, orman ekosistemlerinin sürdürülebilirlik bazında çok amaçlı planlanma zorunluluğu, ormanlardan en uygun yararlanma şeklinin belirlenmesine yönelik Karar Destek Sistemlerine (KDS) olan ihtiyacı ortaya koymuştur (Keleş,2009). Orman amenajman planlarının yapımına hizmet etmek

amacıyla bugün gelişmiş ülkelerde, ormancılık alanında konumsal bilgiyi kullanarak yöneylem araştırması teknikleriyle bütünleştiren KDS'leri geliştirilmekte ve orman işletmelerince kullanılmaktadır. KDS'ler, özellikle Amerika, Kanada ve Finlandiya gibi ormancılığı gelişmiş ülkelerde orman amenajman planlarının hazırlanmasında etkili bir şekilde, orman işletmelerinde kullanılmaktadır. Bu sistemler, ormanların sunmuş olduğu pek çok değer/fonksiyonları aynı anda dikkate almakta, farklı optimizasyon ve simülasyon tekniklerini kullanmaktadır. Geliştirilen bu KDS'lerinin büyük bir bölümü çağdaş bilgi sistemleri ve özellikle CBS tarafından desteklenmektedir. (Başkent ve Keleş, 2004; Başkent, 2005; Keleş, 2008).

Gelişmiş ülkelerde orman amenajmanı planlaması alanında yaşanan gelişmelerden hareket ederek, ülkemiz de gerek akademik çalışmalarla ve gerekse ormancılık teşkilatındaki çalışmalarla bilişim teknolojileri ve yöneylem araştırması tekniklerini etkin şekilde kullanmaya başlamıştır. Ülkemizde orman amenajman plan yapım sürecinde yöneylem araştırması tekniklerinin ilk kullanımı Soykan (1979) tarafından gerçekleştirilmiştir. Köse (1986) tarafından iki adet planlama modeli amaç programlama yöntemine göre geliştirilmiştir. Mısır (2001), nesne tabanlı bir programlama tekniği ile birlikte karar verme tekniklerinden amaç programlama tekniğini kullanarak çok amaçlı bir orman amenajmanı planlama modeli geliştirmiştir. Gümüşhane ili Karanlıkdere planlama biriminde Keles (2003) orman ekosistemlerinin su üretim fonksiyonunu, Karahalil (2003) toprak koruma fonksiyonlarını doğrusal programlama tekniği kullanarak amenajman planlarına yansıtılmışlardır. Yolaşmaz (2005), Artvin planlama birimi ormanlarını ETÇAP anlayışı kapsamında doğrusal programlama tekniği kullanarak uzun dönem amenajman planlama senaryoları geliştirmiştir. Keleş (2008) tarafından, ETÇAP planlama yaklaşımı temel esaslarını dikkate alan, ülkemiz teknik, ekosistemin tüm öğelerini bir bütün olarak kabul eden, Türkiye ormancılık politikası, mevzuat ve sosyo-kültürel koşullarına uygun Karar destek sistemi (KDS) tasarımı ve yazılımı geliştirilmiştir. Kadioğulları (2009) ise ETÇAP planlama yaklaşımı çerçevesinde Konumsal Karar Destek Sistemi (KKDS) tasarım ve yazılımını geliştirerek Uğurlu planlama biriminde test etmiştir.

Orman amenajman planlarının hazırlanmasına yönelik KDS'nin geliştirilmesinde farklı yazılım teknikleri kullanılmaktadır. KDS için geliştirilecek yazılım yalnızca kod geliştirme olarak düşünülemez. Sistem içerisinde nelerin yapılacağı, yazılımın nasıl kodlanacağı, sistemin hangi bileşenlerden oluşacağı, tasarım ve kodlamanın süresi ve maliyetinin ne kadar olacağı ve nihayetinde hangi sorunlarla karşılaşılıp çözümlerinin neler olacağı önceden düşünülmelidir. Sistem analizi prensiplerine göre bu sorunlar daha küçük problemlere parçalanarak her bir problemin ayrı ayrı çözümleri düşünülmelidir. Her problemin cevaplandırılmasıyla, yani sistemin detaylı olarak analiz edilmesiyle, bileşenler kesinleşecek, hata olasılıkları en aza indirgenecek ve yazılımın kodlanması kolaylaşacaktır.

Sistem analizi iyi yapılmış bir modelin bileşenleri daha net bir şekilde ortaya çıkarılmış olacaktır. Kodlama aşamasına ancak analiz aşamasından sonra geçilebilir. KDS yazılımları geniş kapsamlı oldukları için modüler bir yapı düşünülmelidir. Bu yapı sayesinde sistemdeki modüller birbirlerinden bağımsız bir şekilde sürekli güncellenebileceği gibi zamanla ortaya çıkacak yeni tekniklerin veya yeni modüllerin sisteme dahil edilmesi kolaylaşacaktır. Modüler yapının kodlanması sırasında bileşenlerin birbirleriyle haberleşmesi, bir modülü baz alarak yeni modüllerin türetilmesi, etkili bellek yönetimi, modüllere erişim kolaylığı gibi özellikleri ancak nesne tabanlı tasarım ile mümkündür.

Buradan hareketle, bu tebliğde orman amenajman planlama modellerinin geliştirilmesinde; nesne tabanlı tasarım ve programlamanın temel ilkeleri ortaya konulmuş,

nesnelere ve ilişkileri belirlenmiş, model bileşenleri geliştirilmiş UML diyagramları ile ilişkilendirilerek model tasarımı gerçekleştirilmiştir.

Nesne Tabanlı Programlama

Bilgisayar teknolojisinin gelişmesi ve kullanım alanlarının yaygınlaşması sonucunda yordamsal olarak geliştirilen yazılımlar giderek daha karmaşıklaşmaya başlamış, yazılan kodların değiştirilmesi, yeni özelliklerin eklenmesi ve takip edilmesi daha zor bir hale gelmiştir. Giderek büyüyen bu sorun 1960'lı yılların sonunda yazılım geliştirme süreçlerinde büyük değişikliklere sebep olmuş ve yordamsal programlamaya alternatif olarak NTP mantığı geliştirilmiştir.

NTP, nesne kavramı üzerine kurulmuş ayrıştırılabilir (modüler) bir yazılım tasarımı yaklaşımıdır (Booch, 1986). Bu yaklaşımda yazılımın temel bileşenleri birbirleriyle iletişim kurabilen nesnelere topluluğu olarak tasarlanırlar. Kod işletimi nesnelere içinde yapılır ve her nesne diğer nesnelere iletili gönderilerek haberleşirler (Kanoğlu, 2009).

Nesneler metotlar ve niteliklerden oluşurlar. Nitelikler, nesnelere sahip oldukları verilere, metotlar ise bunlar üzerinde yapılabilecek işlemlere karşılık gelir. Bir başka deyişle nesne, kendisini işleyecek kod kesimini kendisi ile birlikte tanımlayan ve taşıyan ve kendi tanımladığı biçimden daha farklı amaçlarla kullanılmayan veri türü olarak yorumlanabilir (Kanoğlu, 2009). KDS'lerinin geliştirilmesinde etkin olarak kullanılan nesne tabanlı tasarım ve programlamanın dört belirgin özelliği öne çıkmaktadır.

1. Modülerlik: Yazılımların kapsamı büyüdükçe gerçekleştirilmesi de aynı oranda zorlaşmaktadır. Bu tarz yazılımlar, tek bir parça halinde geliştirilmek yerine iş bölümü yapıp parçalanarak birbirinden bağımsız alt sistemler (modüller) şeklinde tasarlanabilirler. Büyük çaplı sistemlerin modüller halinde tasarlanması; test, bakım ve onarımlarının daha kolay yapılmasını sağlar.

Modüller birbirlerinin nasıl çalıştıklarıyla ilgilenmezler. Her modülün dışarıya açılan arayüzleri vardır ve bu arayüzler sayesinde başka modüllerle bilgi alışverişinde bulunabilirler. Dolayısıyla bir modülün arayüzlerini bilmek onu kullanmak için yeterlidir. Bu da geliştirilen yazılımların hem programcıdan bağımsız hale gelmesini sağlar hem de aynı anda birden fazla programcının aynı sistem üzerinde çalışmasında olanak tanır. Yani farklı programcılar farklı modüller hazırlayıp tek bir sistem altında arayüzler aracılığıyla birleştirebilirler.

2. Kapsülleme: Tanımlanan sınıflar bütün özellikleriyle dışarıya açık olmak zorunda değildir. Yani bir sınıfa ait bir nitelik veya metot yalnızca o sınıfın içerisinde kullanılabilir veya başka sınıfların bu özelliklere erişmesi istenmemeyebilir. Bir sınıf içindeki özellik ve metotlar erişim izinleri ile dışarıya açık veya kapalı hale getirilebilir. Bu işlem kapsülleme veya sarmalama olarak adlandırılmaktadır.

3. Kalıtım: NYP'de tasarım yapılıırken başlangıçta temel sınıfların tanımlanması büyük avantaj sağlar. Temel sınıflar genel özellikleri ve metotları içericecek şekilde hazırlanırsa, yeni oluşturulacak sınıflarda bu metotların ve özelliklerin yeniden tanımlanmasında gerek kalmaz. Bunun yerine yeni sınıf daha önce tanımlanmış bir sınıftan kalıtım özelliği kullanılarak türetilebilir ve sadece farklı olan özellikleri bu sınıfta tanımlamak yeterlidir.

4. Çok biçimlilik: Kalıtım özelliği kullanılarak oluşturulmuş bir nesne, türetilmiş olduğu temel nesnenin bütün özelliklerine sahiptir. Dolayısıyla temel nesnelere kullanıldığı bir yerde türetilmiş nesnelere rahatlıkla kullanılabilir. Bu aslında türetilmiş nesnelere bağımsız bir yapı ortaya çıkarır ki buna çok biçimlilik denmektedir.

NTP'nin yaygınlaşmasıyla birlikte programlama dili üreten büyük firmalar standart bir dil kullanmadıkları için 1990'lı yıllarda, sadece nesne tabanlı paradigmanın içinden yüzden fazla modelleme dili ortaya çıkmıştır (Güngören, 2005). Ancak, zamanla NTP'de çok büyük fikir adamları olan Booch, Rumbaugh ve Jacobson'un modelleme teknikleri ve dilleri yaygınlık kazanmış, ardından bu üçlü kendi modelleme dillerinin güçlü yanlarını bir araya getirerek tek bir dil oluşturmaya karar vermişlerdir. Birleşik Modelleme Dili – Unified Modelling Language (UML) 1997 yılında bu çabanın bir ürünü olarak ortaya çıkmıştır. UML'yi ortaya çıkaran bu üçlüye de Üç Amigo adı takılmıştır (Güngören, 2005).

UML: Birleşik Modelleme Dili

UML bir programlama ya da yazılım geliştirme dili olmaktan ziyade, iş sistemlerinin nasıl modellenebileceğini belirleyen ve açıklayan yöntemlerin bir araya toplanmış halidir (URL-1). Grafikselleştirilmiş ve ilişkisel bir modelleme dili olan UML, modelleme için değişik diyagramlar kullanır. UML standartlaşmış uluslararası bir dildir ve bu dili bilenler diyagramlardan aynı şeyleri anlar.

UML, çeşitli diyagramlardan oluşan üç ana bölümden oluşur. Yapısal diyagramlarda modellenen sistemde nelerin var olması gerektiği vurgulanır. Davranış diyagramlarında modellenen sistemde nelerin meydana gelmesi gerektiğini belirtir. Davranış diyagramlarının bir alt kümesi olan Etkileşim diyagramlarında ise modellenen sistemdeki elemanlar arasındaki veri ve komut akışı gösterilir (URL-2, URL-3). Yazılım geliştirilirken bu diyagramların hepsi kullanılmak zorunda değildir. Yazılım geliştiren kişi ihtiyaç duyduğu noktalarda bu diyagramlar içinden istediklerinden faydalanabilir.

Çizelge 1. UML Diyagramları

Yapısal Diyagramlar	Davranış Diyagramları	Etkileşim Diyagramları
Sınıf diyagramı	Kullanım Senaryosu diyagramı	Sıralama diyagramı
Nesne diyagramı	Durum diyagramı	İletişim diyagramı
Bileşen diyagramı	Faaliyet diyagramı	Etkileşime bakış diyagramı
Paket diyagramı		Zaman akış diyagramı
Dağılım diyagramı		
Birleşik yapı diyagramı		

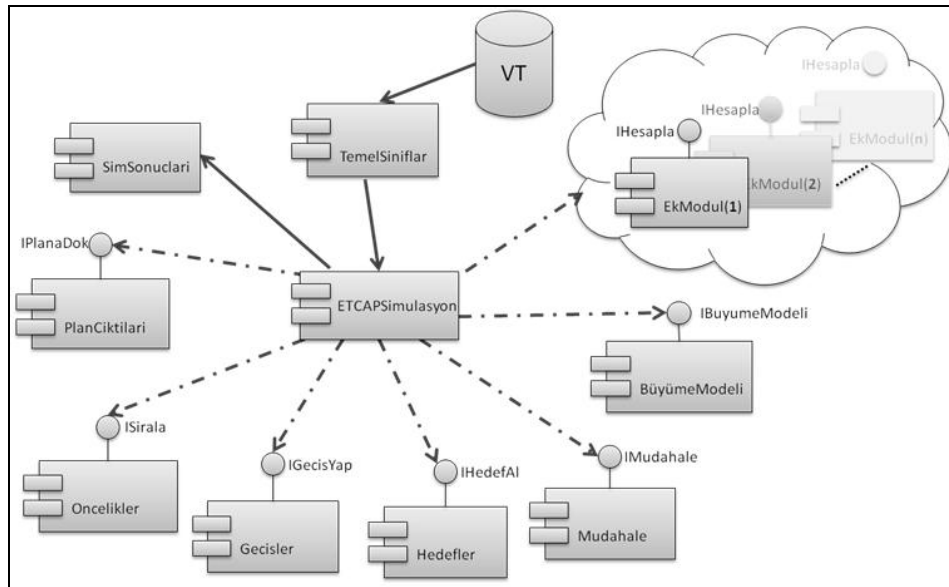
ORMAN EKOSİSTEMİNİN MODELLENMESİ

Temel olarak bir orman ekosisteminin birçok bileşeni mevcuttur. Bunlar, genelde a) sistem girdisi, b) müdahaleleri (eylemleri) ve c) çıktılar olarak üç grupta toplanır. Daha genel bir ifadeyle bu bileşenler; planlama birimleri, fonksiyonları, yetiştirme ortamı özellikleri-bonitetleri, işletme sınıfları, bölmeler, meşcere tipleri, meşcereler, bölmecikler ve her türlü konumsal parametreleri barındıran "aktüel kuruluş" ile aktüel kuruluş üzerinde gerçekleştirilebilecek silvikültürel "müdahaleler", müdahaleler ile oluşan meşcere geçişleri ve yine müdahaleler sonucunda oluşan çıktılar ya da "ürün ve hizmetler" (odun, su, karbon, rekreasyon, biyoçeşitlilik koruma değeri vb) olarak bilinir. Bir planlama modelinde bu üç bileşen etkin bir şekilde tasarlanır ve kodlanır. Ancak, ormancılıkta zaman ve mekansal düzenleme önemli olduğundan, orman parçalarının (bölmeciklerin) zamana bağlı müdahale bağımlı ya da bağımsız değişimlerinin de (büyüme ve artımlarının) modellenmesi gerekmektedir. Orman amenajman modellerinin lokomotifleri olan bu büyüme-artım modelleri de, aktüel optimal kuruluşları arasındaki ilişkilerden hareketle nesne

tabanlı olarak tasarlanmış ve kodlanmıştır. Son olarak, bölmeciklerin konumsal yapıları (alan, yer, şekil ve konumsal dağılım) da Arc/Info nesne sağlayıcısı kullanılarak modellenmiştir. Tüm bu bileşenler temelde simülasyon yaklaşımına göre birleştirilerek *ETÇAPSimülasyon* modeli ve optimal karar verme tekniklerine (doğrusal programlama) göre birleştirilerek de *ETÇAPOptimizasyon* modeli geliştirilmiştir.

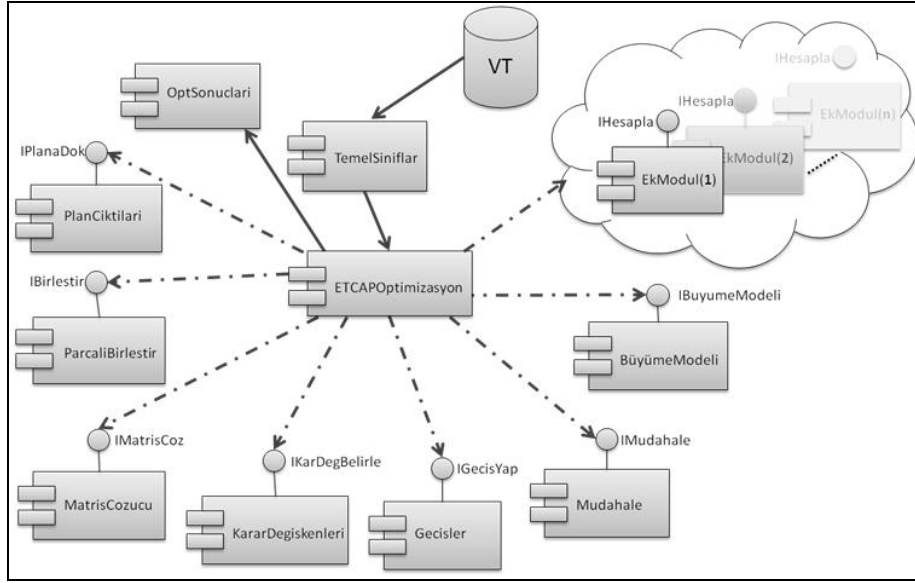
Modelin Bileşen Diyagramları İle Tasarımı

Keleş (2008), ETÇAPSimülasyon ve ETÇAPOptimizasyon modüllerinin tasarlanması için UML diyagramlarından, kullanım senaryosu ve sınıf diyagramlarını oluşturmuş, Bingöl (2009) ile Kadioğulları (2009) ise bileşen ve faaliyet diyagramlarını da kullanarak tasarımı genişletmişler, ayrıca sınıf diyagramlarını konumsal parametreleri de içerecek şekilde yeniden tasarlanmıştır. Şekil-1.'de *ETÇAPSimülasyonun* Şekil-2'de *ETÇAPOptimizasyonun* bileşen diyagramları ile alt-modüllerin birbirleriyle olan ilişkileri gösterilmiştir.



Şekil 1. ETÇAPSimülasyon modülünün bileşen diyagramı tasarımı (Bingöl, 2009)

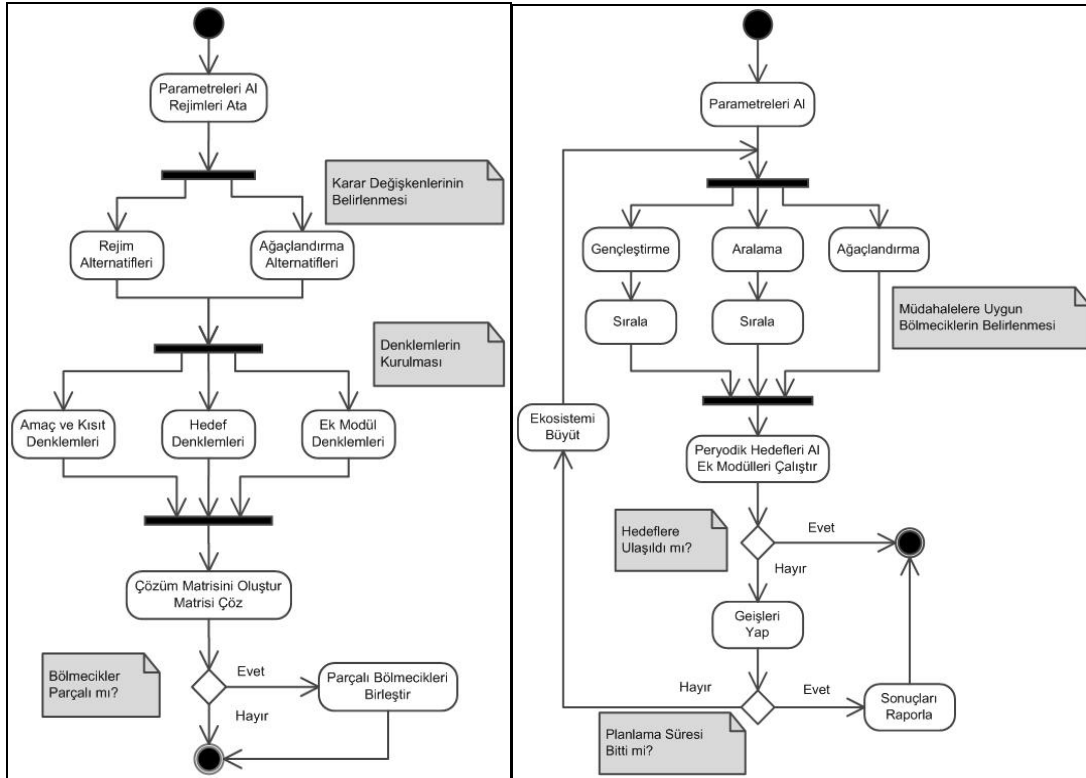
Şekil-1 ve Şekil-2'de görüldüğü gibi ETÇAPSimülasyon ve ETÇAPOptimizasyon modülleri de aslında birçok alt bileşenden oluşmaktadır. Aslında bu alt bileşenlerin birçoğu (Temel Nesnelere, BüyümeModeli, Hedefler, Gecişler vb) her iki diyagramda ortak kullanılmaktadır. Yani bir modül için tasarlanan bir bileşen uygun arayüzlerin kullanılması sayesinde farklı modüllerde rahatlıkla kullanılabilir. Bu da sistemin daha rahat tasarlanması ve değişikliklerin sisteme çok kolay bir şekilde monte edilmesini sağlamaktadır.



Şekil 2. ETÇAPSimülasyon modülünün bileşen diyagramı tasarımı (Bingöl, 2009)

Modelin Faaliyet Diyagramları İle Tasarımı

Şekil-3'de ETÇAPSimülasyon ve ETÇAPOptimizasyonun faaliyet diyagramları görülmektedir. Bu diyagramlar sayesinde modüllerin çalışma aşamaları izlenebilir ve hangi modüller hangi aşamada işleme alındığı görülebilir.



Şekil-3. ETÇAPSimülasyon (solda) ve ETÇAPOptimizasyon (sağda) modüllerinin faaliyet diyagramı tasarımları (Bingöl,2009).

Modelin Sınıf Diyagramları ile Tasarımı

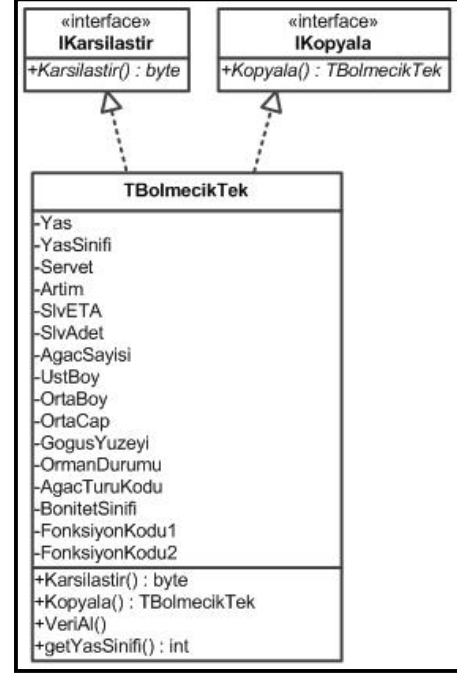
Temel sınıflar alt modülü

Sistemin tasarlanmasında aktüel verilerin tutulduğu bölmecik sınıfı en temel sınıf olarak ele alınmıştır. ETÇAPSimulasyon ve ETÇAPOptimizasyon modellerinde kullanılacak olan bölmecik nesnesinin sınıf yapısı zamana bağlı olarak değiştiği göz önüne alındığında Şekil-4'deki gibi tasarlanması gerekir.

Şekil-4'te de görüldüğü üzere TBölmecikTek sınıfı IKarsilastir ve IKopyala gibi iki arayüze sahiptir. IKarsilastir arayüzü bölmeciklere müdahaleler uygulanırken hangi öncelik dikkate alınacaksa bu kritere göre sıralama yaptırılmasını sağlar. IKopyala arayüzü ise bir planlama periyodu sonunda değişen aktüel bölmecik değerlerini hafıza saklamak için kullanılacaktır. Böylece aktüel değerler değişse bile, istenildiğinde kopyalanmış değerlere bakılarak geriye dönük izlemeler sağlanabilecektir.

Şekil-4'deki bölmecik sınıfı tasarımı kendi başına yetersizdir. Çünkü bölmeçiğin bir de değişmeyen değerleri vardır. Bu değerler, TBolmecikSabit isminde bir sınıf yapısı ile tasarlanmış ve TBolmecikTek ve TBolmecikSabit sınıfları arasında birebir ilişkisi kurulmuştur. TBolmecikSabit sınıfı aynı zamanda Konumsal parametreleri de barındırmaktadır. TBölmeçiğin sınıfı yalnızca bir bölmeçiği temsil etmektedir. Halbuki modüllerde bölmecikler bazı özelliklerine göre gruplandırılıp bir liste yapısı şeklinde kullanılması gerekecektir. Bu yüzden birden fazla bölmeçiği temsil eden TBolmecikListesi sınıf yapısı oluşturulmuştur. TBolmecikListesi sınıf yapısı kullanılarak bir bölmecik grubundan özel alt gruplar oluşturulabilmelidir (örneğin bir işletme sınıfındaki bölmecikler içerisinde Kızılçam ağaç türüne ait olan bölmecikler bir grupta toplanmak istenebilir). Bu işlemi gerçekleştirebilmek için de TOzetBolmecikGrubu isminde bir sınıf yapısı oluşturulmuştur.

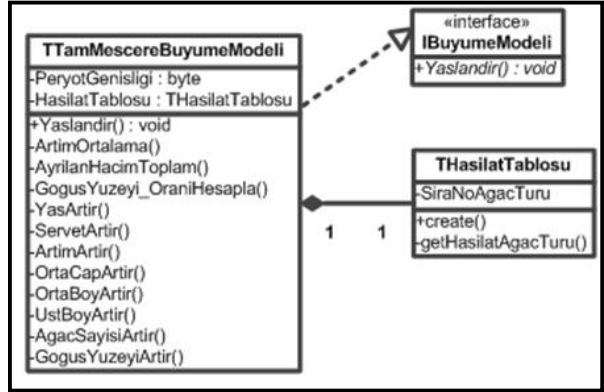
Konumsal simülasyon ve optimizasyonu gerçekleştirmek için, kullanılan komşuluk belirleme kriterleri, bir bölmeçiğin komşuluk kriterlerine uygun komşuları, bölmeçiğin içinde dahil olduğu açma alanları veya blok alanlarının hepsini barındıran temel sınıflar alt modülünün UML diyagramı Şekil-5'de gösterilmiştir.



Şekil-4. TBolmecikTek sınıfı

Büyüme modeli alt modülü

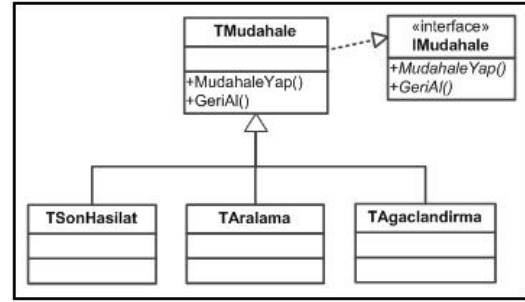
Hazırlanan sistemde büyüme modeli olarak hasıllata dayalı tam meşcere büyüme modeli kullanılmıştır. Fakat farklı büyüme modellerinin de bu sistem üzerinde kullanılabilmesi için tasarımda genel bir büyüme modeli yapısı sunulmuştur. Böylece daha iyi bir büyüme modeli geliştirilirse IBuyumeModeli arayüzünü desteklemek şartıyla bu sistem üzerinde kullanılabilir. Büyüme modelini temsil eden UML sınıf yapısı tasarımı şekil-7'de gösterilmiştir.



Şekil 7. Büyüme modeli sınıf diyagramı

Müdahaleler alt modülü

Sistem tasarlanırken silvikültürel müdahale alternatifleri olarak gençleştirme, aralama ve ağaçlandırma müdahaleleri göz önüne alınmış, bunların ayrıntılı yöntemleri irdelenmemiştir. Fakat sistem, farklı müdahalelerin eklenmesi veya var olan müdahalelerin değiştirilmesine açık bir yapıdadır. Şekil-8'de TMudahale ana sınıfının yapısı gösterilmiştir. Bu şekilde görüleceği üzere TMudahale sınıf yapısı IMudahale arayüzünü kullanabilecek şekilde tasarlanmıştır. Böylece eklenecek farklı müdahaleler için imkan sunulmuştur.

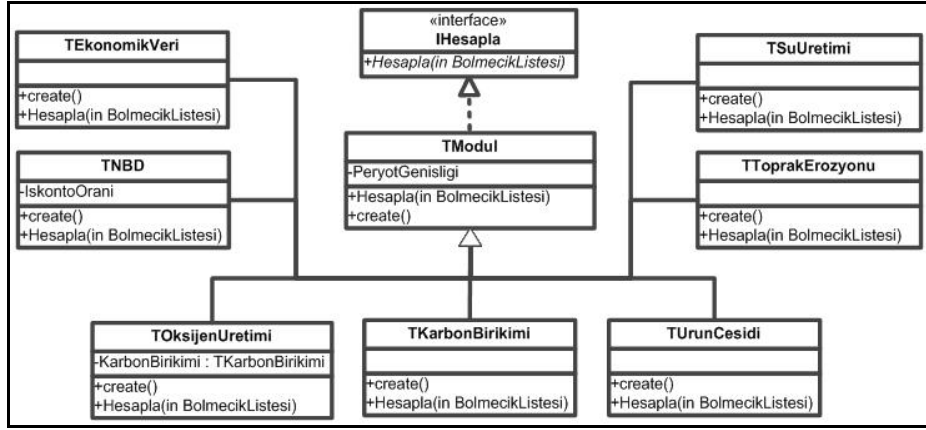


Şekil 8. Müdahaleler sınıf diyagramı

Ek modüller

Sistem tasarlanırken yalnızca büyüme modeline bağlı olarak ormanlık alanın ağaç yapısının gelişiminin izlenmesi göz önüne alınmamış, aynı zamanda var olan verilere bağlı olarak değişen bazı verilerin de elde edilebilmesi için ek modül desteği verilmiştir. Burada sisteme eklenecek herhangi bir modül IHesapla arayüzünü destekleyen TModul sınıf yapısında bir modül olmalıdır. TModul ve IHesapla yapısı ile ETCAPSimulasyon ve ETCAPOptimizasyon yazılımların eklenmiş farklı ek modüllerin yapısını gösteren UML Sınıf yapısı Şekil-9'da gösterilmiştir. Yazılıma şu ana kadar eklenmiş olan ek modüller şunlardır;

- Su üretimi
- Toprak erozyonu
- Ürün çeşitleri
- Karbon birikimi ve emisyonu
- Oksijen üretimi
- Ekonomik veri (Net bugünkü değer -NBD)



Şekil-9. EkModüller alt modülünün sınıf diyagramı tasarımı

SONUÇ VE DEĞERLENDİRME

Orman amenajmanı planlama sisteminde KDS'ne duyulan ihtiyaç her geçen gün artmaktadır. Bu çalışmada UML diyagramları ile yardımıyla ETÇAP yaklaşımına uygun geliştirilebilecek bir KDS'nin yapısı genel olarak ortaya konulmuştur. Sistem tasarlanırken modül yapının gereksinimine değinilmiş ve bunun avantajlarından bahsedilmiştir.

Sistemin ihtiyaçlarının belirlenmesi, modüllerin birbirleri ile olan ilişkilerinin netleştirilmesi ve kodlama aşamasındaki iş yükünün hafifletilmesi açısından bir tasarım modelleme tekniğine olan ihtiyaç olduğu görülmüştür. UML, KDS'nin tasarlanması için gerekli tüm aşamaları içerdiği için kullanılabilir bir tasarım modelleme tekniğidir. UML modelleme tekniğinin kullanılmasındaki bir gerekçe de, sistem ile ilgilenen fakat tasarım aşamasında yer almayan birisi de rahatlıkla bu diyagramlardan sistem yapısı hakkında detaylı bilgilere ulaşabilir. Üstelik bu sistem içerisinde gerekli arayüzleri desteklemek şartıyla bir yeni model eklenebilir veya modüller değiştirilebilir.

Bu eserde orman amenajman planlama model tasarımı ve kodlanmasının UML diyagramları ile gerçekleştirimi gösterilmiştir. Ayrıca, bu modellerin kullanımı ve güncellenmesini kolaylaştıracak arayüz tasarımları ve kodlanması da model bileşenleri ile birlikte değerlendirilmiştir. Sonuçta, giderek karmaşık bir yapı arz eden ormancılıkta planlamanın ancak nesne tabanlı ve modüler bir yapıdaki programlama ile gerçekleştirilebileceği vurgulanmıştır.

KAYNAKLAR

- Başkent, E.Z. ve Keleş, S., 2004. Ormancılıkta Model ve Modelleme Kavramlarının Kullanımı ve Genel Değerlendirmesi (1. Bölüm), Orman Mühendisliği Dergisi, Ocak-Şubat-Mart 2004, Yıl 41, Sayı 1-2-3, 19-24.
- Başkent, E. Z., 2005. Orman Amenajman Planlarının Ekosistem Tabanlı Ve Çok Amaçlı Planlanması (Etçap) ve Uygulanmasına Yönelik Eylemler, Türk Ormancılığında Uluslar arası Süreçte Acil Eyleme Dönüştürülmesi Gereken Konular-Mevzuat ve Yapılanmaya Yansımaları, Orman Mühendisleri Odası Sempozyumu, 22-24 Aralık, Antalya.

- Bingöl, Ö., 2009, Orman Ekosisteminin Modellenmesi Ve Model Üzerinde Sezgisel Yöntemler Kullanarak Silvikültürel Müdahale Alternatiflerinin Belirlenmesi, K.T.Ü. Fen Bilimleri Fakültesi, Trabzon.
- Booch, G., Object-Oriented Development, IEEE Transactions on Software Engineering., Vol. SE-12, No.2, pp. 211-221, Feb, 1986.
- Buzzard, G. ve Mudge, T., Object-based computing and The Ada programming language, Computer, vol. 18, no.4, Apr. 1985
- Güngören, B., UML ile Nesne Tabanlı Çözümleme ve Tasarım, Seçkin Yayıncılık, İstanbul, 2005.
- Kanoğlu, A., İnşaat Projeleri Yönetimi Bilgi Sistemleri Ders Notları, İstanbul Teknik Üniversitesi, Erişim: <http://atlas.cc.itu.edu.tr/~kanoglu/crs-iscpm.htm>, 27 Aralık 2009.
- Kadioğulları, A.İ., 2009, Orman Amenajman Planlarının Hazırlanmasında Konumsal Yapının Kombine Optimizasyon (Heuristic) Teknikleri İle Kontrolü; Konumsal Planlama, Doktora Tezi, K.T.Ü. Fen Bilimleri Enstitüsü, Trabzon.
- Karahalil U. 2003. Toprak Koruma ve Odun Üretimi Fonksiyonlarının Doğrusal Programlama ile Modellenmesi, Yüksek Lisans Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon.
- Keleş S. 2008. Orman Amenajman Planlarının Hazırlanmasına Yönelik Karar Destek Sisteminin Tasarımı ve Prototip Modelinin Geliştirilmesi, Doktora Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon.
- Keleş S. 2003. Ormanların Su ve Odun Üretimi Fonksiyonlarının Doğrusal Programlama Tekniği İle Optimizasyonu (Karanlıkdere Planlama Birimi Örneği), Yüksek Lisans Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon.
- Köse, S., 1986. Orman İşletmelerinin Planlanmasında Yöneylem Araştırması Yöntemlerinden Yararlanma Olanakları, Doktora Tezi, KTÜ Fen Bilimleri Enstitüsü, Trabzon, 123 sayfa.
- Mısır, M., 2001. Çok Amaçlı Orman Amenajman Planlarının Coğrafi Bilgi Sistemlerine Dayalı Olarak Amaç Programlama Yöntemiyle Düzenlenmesi (Ormanüstü Planlama Birimi Örneği), K.T.Ü. Fen Bilimleri Enstitüsü, Doktora Tezi, Trabzon.
- Pokkunuri, B. P., Object Oriented Programming, ACM SIGPLAN Notices, v.24 n.11, p.96-101, Nov. 1989.
- Soykan, B., 1979. Aynıyaşlı Ormanların Aktüel Kuruluşlarının Optimal Kuruluşa Yaklaştırılmasında Yöneylem Araştırması Metotlarından Yararlanma Olanaklarının Araştırılması, KTÜ Orman Fakültesi Yayın No: 106, Orm. Fak. Yayın No: 5, Trabzon, 252 sayfa.
- Yolasığmaz H. A., 2004. Orman Ekosistem Amenajmanı Kavramı ve Türkiye’de Uygulaması, Doktora Tezi, K.T.Ü., Fen Bilimleri Enstitüsü, Trabzon.
- URL-1, Erişim: <http://tr.wikipedia.org/wiki/UML>, 25 Aralık 2009.
- URL-2, Erişim: <http://e-bergi.com/2009/Ocak/UML>, 26 Aralık 2009.
- URL-3, Erişim: http://en.wikipedia.org/wiki/Unified_Modeling_Language, 26 Aralık 2009.

TEŞEKKÜR

Bu çalışmayı destekleyen TÜBİTAK’a teşekkür ederiz.