

YAZILIM MÜHENDİSLİĞİ
KAVRAMLARI
3. ÖDEV (OKUMA ÖDEVİ)

İSİM: SERDAR
SOYİSİM: GÜL
NUMARA: 20421689

NESNEYE YÖNELİK ANALİZDEN TASARIMA GEÇİŞ

Nesneye yönelik geliřtirmenin yararlarından biri de iřlem yaklařımında geliřtiricilerin nesneleri modelleme aracı olarak kullanabilmesidir. Analiz ve tasarım kısmında nesneleri ve aralarındaki iliřkileri birebir olarak aynı řekilde kullanabilmekteyiz. Buna gre nesneye ynelik analizden nesneye ynelik tasarıma geiřin gayet dzgn ve olay olduėu iddia edilebilir. Ama greceėiz ki bu iř sanıldıėı kadar kolay bir iř deėildir.

Analizden tasarıma geiřin zorluėu analizde ve tasarımda kullanılan nesnelerin znde farklı řeyleri temsil etmesinden kaynaklanır. Sonu olarak analiz modeli tasarım modelini kolaylařtırmaz. Bu yzden geliřtiriciler 2 adet zor tasarım grevini uyulařan bir řekilde gerekleřtirmek zorundadır. Bunlar , yazılım iin bir mimari belirtmek ve yazılım tarafından kullanılacak esas bir modeli oluřtırmaktır.

Bu görüş için gerçek zamanlı bir projenin (otomatik banka veznesi makinası ki bu da günlük hayatta sık kullanıldığından) analiz ve tasarım konularını çalıştım.

NESNEYE YÖNELİK ARKAPLAN

Nesneye yönelik yaklaşımdaki temel düşünce , iletişim içindeki nesnelerin bir sisteme oturtulmasıdır.Bu gelişim sürecinde onların belirtilmesinde kullanılabilir.Ama bir tehlikeyi de gözler önüne serer , aynı işlemdeki farklı durumdaki nesnelerin farklı türde olma ihtimalinden kaynaklanır.

NESNE TANIMLARI

Genelde uzmanlar nesnenin problem yerindeki nesneler olduğu konusunda hemfikirdirler.Daha özel olarak nesne

- problemin esasındaki bir şeyin soyutlaması
- ele alınan problemde keskin sınırlar ve anlamlara belirtilen soyutlama,kavram,şey
- gerçek dünyadaki şeylerin bir kümesinin soyutlaması olarak tanımlanır.

Ama ters bir durum olarak tasarım düzeyindeki nesneler çözüm evrenindeki nesnelerdir.Nesneye yönelik tasarımın amacı , geliştirme başlamadan sistemin nasıl geliştirileceği konusunda üretilen çözümdür.

Tasarımdaki nesneye verilen bazı özel tanımlarda bu nesnenin de problem düzeyinde bir soyutlama olduğu , bu sebeple geliştiricilerin analizdeki nesneye ufak eklentilerle tasarım nesnesi oluşturabilecekleri

öngörülüyor.Bir diğer tanıma göre tasarım nesnesi kodun soyutlamasıdır.Bu durumda tasarım nesnesinin problem alanındaki bir şey mi yoksa çözüm dünyasının bir elemanı olup olmadığı hala bir soru işareti olarak kafamızda duruyor.

Nesneye yönelik programlamada nesne bir kısım bellek alanı ve onunla ilgili yaptırımlar olarak belirtiliyor.Pascal'daki kayıt ve C'deki yapı kavramı burada bellek alanını temsil ediyor.

Bütün bu tanımlar ışığında nesne güzelce tanımlanmış bir sınır , durumu sarmalayan bir kimlik ve davranıştan ibarettir.

NESNEYE YÖNELİK MODEL TANIMLARI

Analiz modeli problemin esasına ait bir modeldir.Ve analiz modeli mühendislerin esas anlaması için lüzumlu olan gereksinimlere sahiptir.

Çoğu analiz modeli açık olmadığından , hem gerçek dünyada olabileceklerin bir soyutlaması hem de ek olarak olmayabileceklerin de soyutlamasını içerir.

Analiz modeli problemi çözecek donanım veya yazılım tanımlarını belirtecek bir gereksinim modeli olarak belirtilmelidir.Bu bakış açısına göre analiz modeli bir problem belirtimidir.

Ama tasarım modeli geliştiriciye sistemin doğasını anlaması için yardımcı olmalıdır.Tasarım modeli , kavramsallık,iletişim,geliştirme konularında tasarıma bir ortalama verir , nesneler ve ilişkileri tanımlar , ve sistemin harici davranışlarındaki etkileşimi oluşturur.

GEÇİŞLERDEKİ MEVCUT GÖRÜŞLER

Burada řu soru kafa karıřtırıyor.Analiz modeli tasarım modelini oluřturmak iin nasıl kullanılabilir?oėu insan analiz modelinin tasarım modelinin bir parası olması hususunda tartıřıyor ama mimari aıda nedense bu pek de tartıřılmıyor.Bu konuda bir ka grř var.

KOLAY GEİř GRř

Bu grře gre tasarım modeli analiz modelini kapsar.Tasarım modeli 4 elemandan oluřur.İlki analiz elemanlarını direk konduėu problem esası , ikincisi veri ynetimi elemanı ki burda bileřenleri eklemek gerekir , bu elemanı detekleyebilmek iin nesnenin nasıl saklanacaėına dair bilginin problem esası kısmında tanımlanması gerekir.

OMT gibi diėer byk metodlarda anlařılır ki tasarım modeli analiz modelinin bir ka zellik ve fonksiyon eklenmiř halidir.

Bu grřn esası analiz modelindeki elemanlara bir ka zellik ve fonksiyon ekleyerek olduėu gibi tasarım modelinde kullanma eėilimine dayanır.Ama bu yaklařımda baı problemler ortaya ıkar.

-zellikler ve fonksiyonlar eklendiėinden bir gerek dnya nesnesinden ıkıp , bir yazılım nesnesine dnřr.

-Analiz yazılımının iinde gerek dnyayı esas almalı , gereksinimleri yazılımının iinde ele almamalı.

2. durumu zme ulařtırsak bile karřımıza řunlar ıkıyor.

-Nesneye ynelik metodlar , modellenen řeyi tam olarak belirtemiyor.

-Analiz modeli tasarım modelini iermesi gereken řeyi ieriyor ve yazılımının bir parası haline geliyor.Ki tasarımda analize eklenenler tam olarak alıřmayabilir.

-Eėer analiz yazılımının bir n tasarımıysa ki ayrıca bir gereksinim modeli , bu bir problem tanımımı?

-Eğer analiz bir problem tanımıysa nasıl bu tanıma birşeyler ekleyerek bir çözüm tanımı oluşturabiliriz?

TEZAT OLUŞTURAN GÖRÜŞLER

Problem tabanlı görüş ve hedef sistem görüş arasındaki farkı işleyen bir makaleye göre , hedef sistem görüşü ele alınırsa analiz modeli söz verdiği problem tanımı ve gereksinimler konusunda başarısız olmaktadır.

Başka bir makaleye göre de analiz modeli sistemin harici davranışlarını ve gereksinimlerini belirleyemiyor, ve ayrıca gereksinimdeki değişiklikler amaç , tasarım , önem konularında büyük değişikliklere yol açacağından analizden tasarıma geçişin zorluğu vurgulanıyor.

YENİ BİR GÖRÜŞ

Burada geçişteki zorluğun anlaşılması için analiz ve tasarım nesnelerine odaklanmak gerekiyor.Burada 2 örnek üzerinden anlatılacak olaylar.

MODELLER VE NESNELERİ ARASINDA İLİŞKİLER

Burada ATM ile ilgili bir örnek var.

Öncelikle analizde olması gereken en az şey ile birlikte olabileceklerin kapalı tanımı veriliyor.

Sonra tasarım kısmında bu olabilecekler ayrıntılı bir şekilde tanımlanıyor.

En sonunda da yazılım kısmında tasarım kısmında bulunan elemanların gerçek dünyadaki çalışma prensiplerine göre yazılmış halleri bulunuyor.

Burada her ilerleme aşamasında müşterinin değişimi ve daha çok isteğe cevap verebilmesi , gerçek hayattakine çok yakın bir çözüm olması anlaşılıyor.Analiz kısmındaki müşteri nesnesi de bir müşteridir.Ama günlük hayatta yapılabilecek pek çok şeyi yapamayabilir.Bu sebeple tasarım ve yazılım

geliştirme kısımlarında da ayrıntılı bir şekilde nesneler yapılır.

Analiz ile tasarımın nesneleri görüldüğü gibi birbirinden oldukça farklıdır.

Buradan anlaşılıyor ki analizdeki nesne , gerçek hayattaki müşterinin soyutlaması , tasarımdaki nesne programdaki nesnenin soyutlaması , programdaki nesne tasarımdaki nesnenin belirtilmeleri yapılmış tamamlanmış halidir. Analizden tasarıma kolayca ulaşılabilceği iddiaları da burada suya düşüyor diyebiliriz.

GÖREV TASARIMINDAKİ ZORLUKLAR

Burada 2 zorluk dikkat çekiyor. Birincisi ATM örneğindeki silik olarak gösterilen analiz modelinde gösterilen nesnelerin bilgisayar destekli bir çözümde nesneye yönelik tanım tarafından kabul edilmesi , ikincisi de daha önce belirtildiği gibi tasarımcıların tasarımda hangi nesnenin ne bilgisini nasıl saklayacağı hususunda karar vermesi durumudur.

MİMARİ

Bu durumda tasarımın nasıl kolay hale geleceği sorusu kafaları kurcalıyor. Bunun için yazılımı oluşturmada önce modellemek gerekiyor ve bu da bayağı maliyetli bir iş çünkü henüz olmayan bir şeyin soyutlamasını yapmak ayrıca bunu karmaşık yapmak gerekir.

Eğer bu tip bir mimari kullanılırsa tasarım sanıldığından kolay olur ve analiz modeli de sadece problem esası elemanının içine konulabilir.

Bu durum kullanışlı değil gibi görünmesine karşın pek çok örüntünün kullanımına zemin hazırlar. Bi problemin çözümü için gerekli mimariyi bulmak oldukça zordur. Problem Analiz modelinde belirtildiğinden ve analizde aşırıya kaçılmayacağından bu böyledir.

PROGRAM İÇİNDEKİ ESASIN MODELİ

Daha önce de belirtildiği gibi analiz modeli tasarım modelinin basit bir hali değildir. Eğer gösterimdeki farkları ihmal edersek, problemin esasının analiz modelinde belirtilenden çok farklı olduğunu görebiliriz. Sistem analiz modelinde olmayan pek çok şeyi içerebilir. Ama unutmamak da gerekir ki analiz modeli tasarım kararı alımında önemli bir rol oynar.

Bir önemli olay da sistemin saklayacağı bilginin sonradan telafi edilebilmesidir. Kullanıcının görebileceğine bilgi modeli, bilgi saklamak için bir veritabanı tasarlarken kullanılabilecek da kısıtlı bilgi modeli (veri modeli) diyebiliriz. İkisi farklıdır. Aynı zamanda ikisi de analiz modelinden etkilenir. Bilgi modeli gerçek dünya nesnesinin bilgilerin saklamak zorunda değildir.

Buna göre bir problem analiz modeli tanımına bakılarak çözülebilir. Yalnız verilecek görevler bağımlı ve kısa süreli olmalıdır.

TAVSİYELER

Bu durumda bazı tavsiyelerde bulunmak gayet yararlı olacak ama illa ki bu tavsiyelere uyun diye de bir zorlama getirilmiyor. Bazıları direk olarak olaya konu ile ilgili olmasa da hepsinin bir sebebi var , ama illa uygulanacak diye de bir zorlama yok.

NESNEYE YÖNELİK ANALİZ TAVSİYELERİ

- Hazırlayıcı gereksinimleri istemek
- Esasla birlikte alışılmış bir hale gelmek
- Yeni sistemi yerleştirdikten sonra esas gözlerinin önüne getirip düşünmek
- Esasa göre durgun veya devingen sorunlara yoğunlaşmak
- Nesneye yönelik modellemeyi geliştiren gereksinimlerle birlikte doğal dilde tanımlamak

-Programın içindeki nesneler hakkında hemen özelleştirilmiş kararlar almamak

Bunlar uygulandığında gerçek dünya problem ve gereksinimler elde edilir ama herhangi bir yazılım tasarımı yapılmamış olur.

NESNEYE YÖNELİK TASARIM TAVSİYELERİ

-Nesneye yönelik analizdeki önerilen sistemin kara kutusu üzerinde düşünmek , ona yoğunlaşmak
-Önerilen sistemin mimari görüntüsünü geliştirmek

-Analiz modelinde belirtilen her nesne sınıfı için şu soruyu cevaplamak."Acaba önerilen program gerçek dünyadaki nesnelere ilişkin bilgiye ihtiyaç duyacak mı???"Eğer ilk soruya evet cevabını verirsiniz şu iki soruyu da sorun

"Analiz modelindeki bütün niteliklere ihtiyaç var mı??"

"Ek niteliklere ihtiyaç var mı??"

-Analiz modelindeki her ilişki için şu soruyu cevapla

"Analiz modelinin içindeki nesne sınıfları , tasarım modeli içindeki nesne sınıfları ile ilişki içerisinde mi?"Bana göre bunun bazen olmaması gerekir.

-Sistemin başka (ek) ilişkilere ihtiyacı olup olmadığını araştır.

-Analiz modelinin devingen kesimini,mimari görüntüyü,tasarım modelinin yapım aşamasındaki halini kılavuzlar eşliğinde kullanarak mimariyi oluştur.

-Eğer mimariye gerekiyorsa tasarım modelindeki k nesne sınıflarını ekle

-Gerekli fonksiyonelliği sağlaması açısından nesneye yönelik tasarım elemanlarına sorumluluk ataması yap.

Bütün bu tavsiyeleri birebir uygulamasak da , bir kısmını uygulasak bile nesneye yönelik analizden

nesneye yönelik tasarıma geiş hakkında düşüncelerin deėişeceėine , daha sağlam tasarımlar , analizler ve yazılımlar yapılabileceėne inanıyorum.

Umarım bu makale sizin nesneye yönelik programlama hakkındaki görüşlerinie ek olarak bir kaç şey daha sunabilmiştir...