

Metadata Management

Final Project

Metamodel Repository Application

Umut BENZER

09.01.2012



İçindekiler

İçindekiler	2
Genel Tanıtım.....	3
Uygulamanın Kapsamı	3
Uygulamanın take-home çıktısı olan veritabanı modeli ile ilişkisi.....	3
Teknik Bilgiler	3
Uygulamanın Önemli Özellikleri	4
Kısaltmalar	4
Uygulama ile ilgili internet kaynakları	5

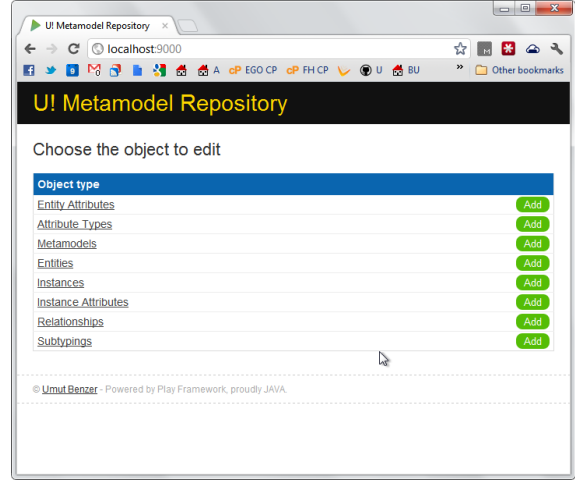
Genel Tanıtım

Uygulamanın Kapsamı

Geliştirilen metadata repository application, metamodelleri, bu modellere ilişkin entityler, attributelar, alt ve üst tip ilişkileri, entity arası ilişkiler ve entitylere ait instance bilgileri ile birlikte saklayabilecek kapasitededir.

Uygulamanın take-home çıktısı olan veritabanı modeli ile ilişkisi

Gerçekleştirilen uygulama veri kaynağı olarak take-home için oluşturulmuş veri modelinin ufak iyileştirmeler içermiş bir halini kullanmaktadır. Bu iyileştirmeler ileriki kısımlarda dokümanite edilmiştir.



Take-home'da yer alan metamodellerin içeriklerine bakıldığında Appendix B'de yer alan The Data Management Metamodel'in soyutlama seviyesinde diğerlerinden biraz daha üstte olduğu görülmüştü. Bu yüzden diğer metamodellerin hepsi The Data Management Metamodel için birer "veri" veya "instance" olarak düşünülmüştü. Hazırlanmış model uygulamada kullanılırken de bu fikir olduğu gibi korunmuştur. Uygulama şu an kitaptaki Data Management Metamodel ile çalışmakta ve diğer üç modeli saklamaktadır. Uygulama ve altında yer alan veritabanı modeli sınırsız sayıda metamodeli hem entity bazında (M1) hem de instance bazında (M0) tutabilecek seviyededir.

Teknik Bilgiler

Modellerin saklanması için tasarlanan veritabanı modeli MySQL DBMS'i kullanılarak yaratılmıştır. Programlama, JAVA platformunda çalışan Play Framework ile bir web uygulaması olarak gerçekleştirilmiştir. Veri erişimi için Hibernate kullanılmıştır. Hibernate, "relational database to object model mapping" için kullanılan bir araçtır.

Önceden hazırlanmış veritabanı modeli, Hibernate ile çalışması düzenlenmiştir. Yapılan değişikliklerin hiçbirisi take-home'da anlatılan veri modeli yapısını bozacak büyüklükte değildir. Bu değişikliklere aşağıdakiler örnek verilebilir:

- Veritabanı modelinin içerisine Foreign Key Constraint'ler eklenmiştir. Eskiden model bu kısıtlamalar üzerine inşa edilmiş olsa da bu kısıtlamalar explicit olarak belirtilmemiştir.
- Birden fazla kolonun birlikte Primary Key olduğu tablolarda, bu tarz tabloların Hibernate'te yönetilmesinin imkansız yakın olması nedeniyle auto-increment yeni ve tek bir primary key oluşturulmuştur. Eskiden primary key olarak kullanılan sahalarda eşsizlik özelliğinin devam edebilmesi için bu sahalara unique constraint eklenmiştir.
- Java ile MySQL'in birbirine daha kolay entegre olması için Bazı kolonların veri tipi veya uzunluklarında değişiklikler yapılmıştır.

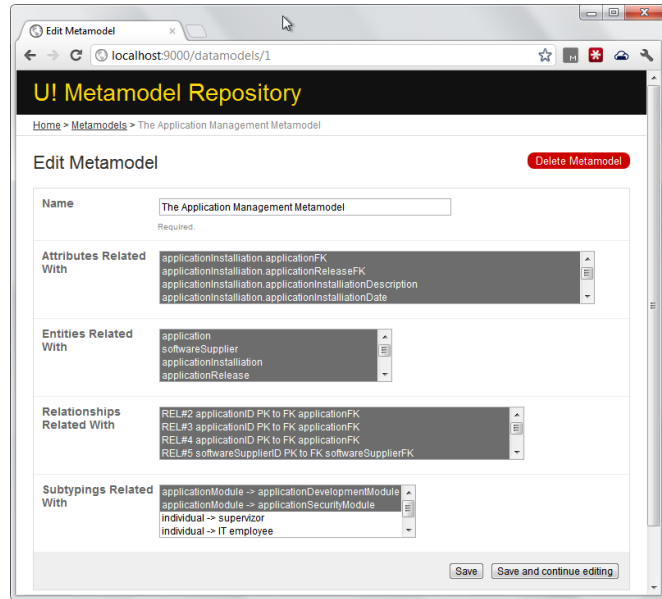
Uygulama yazılırken hazır kütüphanelerden sıklıkla faydalanılmıştır. Örneğin, bir metadata repository applicationun en temel özelliği ekleme/silme/düzenleme/listeleme, yani CRUD'dur. Bu sebeple, Play Framework içerisinde bir modül olarak hazır gelen CRUD kütüphanesinden yoğun olarak faydalanılmıştır. Benzer şekilde Play'in validation özellikleri de sıklıkla kullanılmıştır.

Projenin geliştirilmesi esnasında GIT sürüm kontrol sistemi kullanılarak, projenin ilerleme aşamaları kayıt altına alınmıştır.

Uygulamanın Önemli Özellikleri

Uygulama basit bir "metamodel tutan veritabanı" değildir. Metamodeler birbirleri ile ilişkili bir şekilde tutulur ve ortak parçalar içerebilirler. Örneğin, hangi entitylerin hangi metamodeler ile ilişkili olduğunu seçilebilmektedir. Bir entitynin hangi attributelarının hangi metamodelle ilgili olduğu, hangi ilişkilerin bir metamodel kapsamında değerlendirileceği, hangi alt/üst tip ilişkisinin ve hangi attribute tiplerinin hangi metamodellerin kapsamında olacağını ayrı ayrı belirlemek mümkündür.

Bunun ne işe yarayacağı aşağıdaki örnekle anlaşılabilir:



Bir kurumun metamodel yönetiminde "İK Metamodeli" "Personel" nesnesinin "Maaş" attributeunu içerirken, aynı kurumun "Proje Ekipleri Metamodeli" "Personel" nesnesinin "Çalıştığı Toplam Proje Sayısı" attributeunu içerir. Böylelikle verinin nerede ve nasıl tutulduğuna bağımlı olmaksızın erişilebilen bilgiler belirlenmiş olur.

Kısıtlamalar

Uygulamada modelden kaynaklanmayan, uygulamanın geliştirilmesi ise kolaylıkla çözülebilecek bazı kısıtlar mevcuttur:

- Bazı varlıklarda silme işlemi için bu varlıkla ilişkili varlıkların daha önceden silinmesi gerekmektedir. Örneğin, bir Entity'i kaldırmak için, bu Entity'nin tüm instance ve attribute bilgilerinin manuel olarak silinmesi gerekmektedir. Yazılım bunu otomatik yapmamaktadır.
- Yazılımın arayüzü karışıktır.
- İki entity ilişkisinde, primary olarak atanan entitydeki identifier attribute (key attribute) sayısı birden fazla ise, tüm identifierların ilişkilerinin tanımlanmasının kontrolü yapılmamakta, birisinin yapılması yeterli kabul edilmektedir. Yazılım ilişkinin geçerliliği için bunun kontrolünü yapmamaktadır.
- Yazılım alt ve üst tip bilgilerini tutmasına karşın, alt tip olarak tanımlanmış bir entitynin aynı zamanda üst tipin özelliklerini (attribute'larını) içerdiğini hesaba katmamaktadır.

Uygulama ile ilgili internet kaynakları

Uygulamanın kaynak kodlarının ve veritabanının içerisinde örnek bilgiler de taşıyan son hali aşağıdaki adresten erişilebilmektedir:

<http://meta.ubenzer.com/latest.zip>

Uygulamanın çalıştırılması için Play Framework ile ilgili teknik bilgiye, bazı programların kurulu olmasına ve çeşitli konfigürasyon değişikliklerine ihtiyaç duyulması mümkündür. Bu yüzden, kolay kontrol amacıyla, uygulamanın bir kopyası internet ortamında deploy edilmiştir. Metamodel uygulamasına aşağıdaki adresten erişilebilir:

<http://meta.ubenzer.com/>

Bu adres herkese açık olup, başkaları örnek veriler üzerinde değişiklik yapacağından gösterilen tüm içerik final projesi için hazırlanan içerik ile aynı olmayabilir.