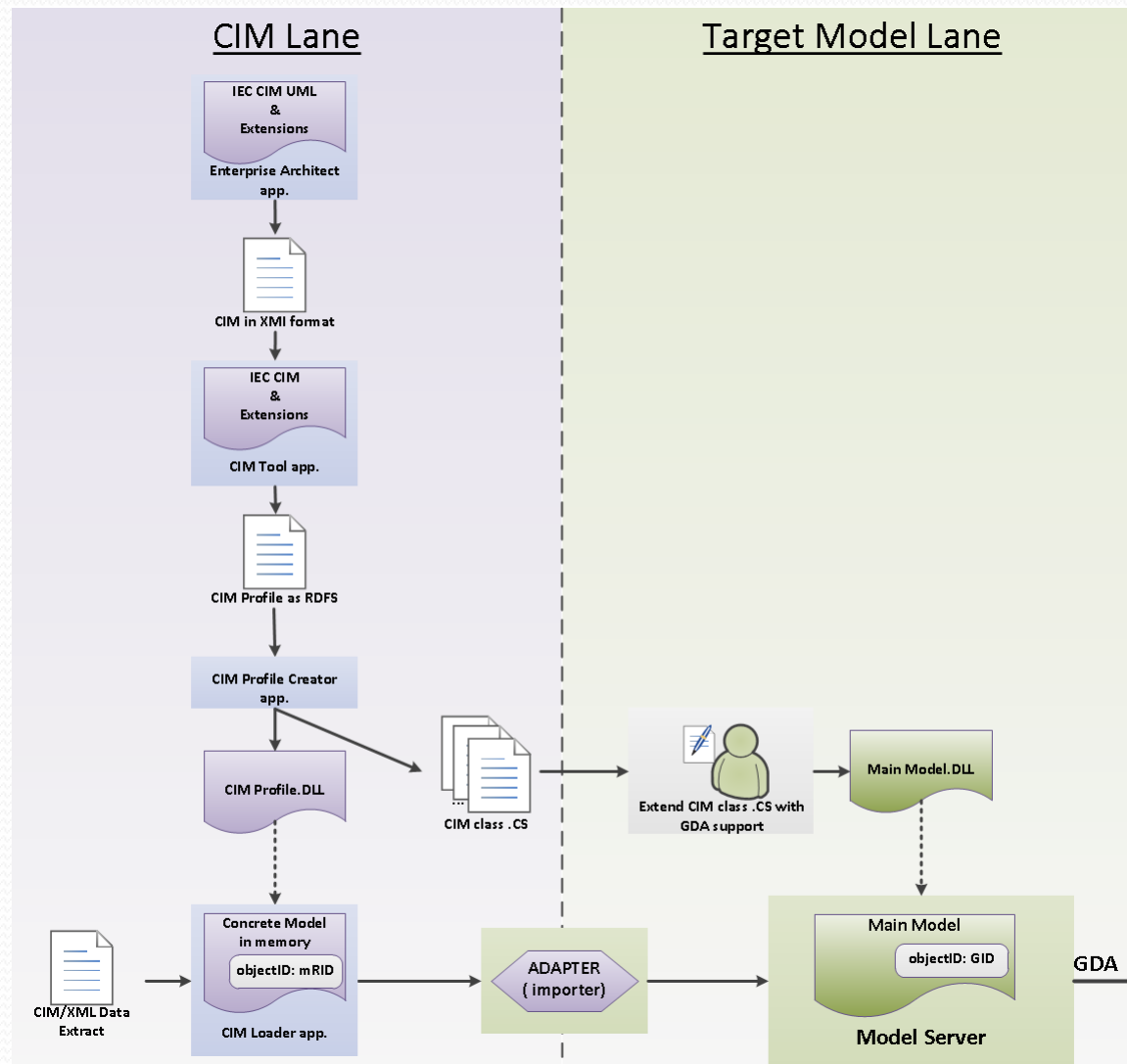


Standardi i modeliranje elektroenergetskih sistema

VEŽBA 10:

Generic Data Access podrška za čitanje
podataka sa Network Model Service-a

Tok podataka pri inicijalizaciji modela elektroenergetske mreže



Generic Data Access (GDA) 1/2

- Omogućava pristup podacima bez ikakvog znanja o logičkoj šemi koja se koristi za unutrašnje skladištenje - implementaciji. Dovoljno je poznavanje definisanog modela, a u ovom slučaju je to CIM.
- Prvobitno je standardizovan u okviru OMG-a kao Data Access Facility (DAF) koji je pružao osnovnu sposobnost postavljanja upita nad nekim modelom. Daljom nadogradnjom nastaje GDA koji obezbeđuje naprednije funkcionalnosti (npr. filtriranje), čitanje i upis podataka.
- Bilo koja klasa i njeni atributi iz internog modela prikazuju se u formi resursa pridržavajući se određenih pravila.

Generic Data Access (GDA) 2/2

- Standard definiše upite i odgovore u vidu resursa, svojstva i vrednosti.
- Resurs je svaki objekat koji poseduje jedinstveni identifikator
- Svojstvo je neki aspekt resursa koji se može opisati. Kada se pojavi u upitu predstavljen je preko *ModelCode*-a. Svaki svojstvo ima svoj domen – niz resursa na koji se može primeniti.
- Asocijacije između resursa se kreiraju preko svojstva koji su tipa *Reference*.

Metode za čitanje podataka

- Resource Query Service

- GetValues(...) – čitanje jednog resursa

`public ResourceDescription GetValues(long resourceId, List<ModelCode> propIds)`

- GetExtentValues(...) – čitanje niza resursa za isti tip entiteta

`public int GetExtentValues(ModelCode code, List<ModelCode> propIds)`

- GetRelatedValues(...) – čitanje referenciranog resursa

`public int GetRelatedValues(long source, List<ModelCode> propIds, Association association)`

- GetDescendentValues(...) – čitanje niza referenciranih resursa

`public int GetDescendentValues(List<long> sources, List<ModelCode> propIds,
List<Association> path, List<Association> tail)`

GetValues(...)

- Čitanje jednog resursa (entiteta)

```
public ResourceDescription GetValues(long resourceId, List<ModelCode> propIds)
```

- Metodi se prosleđuje globalni identifikator entiteta i lista *ModelCode*-ova koji predstavljaju identifikatore svojstva vezanih za entitet
- Rezultat je *ResourceDescription* koji sadrži prosleđeni identifikator i listu zahtevanih svojstva

```
long globalId = 0x0000000300000001;

List<ModelCode> propertyIDs = new List<ModelCode>();
propertyIDs.Add(ModelCode.IDOBJ_MRID);
propertyIDs.Add(ModelCode.IDOBJ_DESCRIPTION);
propertyIDs.Add(ModelCode.EQUIPMENT_ISPRIVATE);
propertyIDs.Add(ModelCode.POWERTR_FUNC);

rd = GdaQueryProxy.GetValues(globalId, propertyIDs);
```

GetExtentValues(...)

- Prva iz grupe metoda koje vraćaju niz resursa
- Čitanje niza resursa za isti tip entiteta

```
public int GetExtentValues(ModelCode code, List<ModelCode> propIds)
```

- Metodi se prosleđuje *ModelCode* koji predstavlja tip entiteta i lista *ModelCode*-ova koji predstavljaju identifikatore svojstava vezanih za entitet
- Rezultat je identifikator upita (identifikator iteratora) na osnovu koga klijent može da pročita rezultujuće *ResourceDescription-e*

```
ModelCode resourceType = ModelCode.POWERTR;
```

```
List<ModelCode> propertyIDs = new List<ModelCode>();  
propertyIDs.Add(ModelCode.IDOBJ_MRID);  
propertyIDs.Add(ModelCode.IDOBJ_DESCRIPTION);  
propertyIDs.Add(ModelCode.EQUIPMENT_ISPRIVATE);  
propertyIDs.Add(ModelCode.POWERTR_FUNC);
```

```
int iteratorId = GdaQueryProxy.GetExtentValues(resourceType, propertyIDs);
```


Iteriranje kroz rezultat upita 1/2

- Metode čiji je rezultat niz *ResourceDescription*-a, ne vraćaju niz već identifikator upita. Niz može biti veoma velik i nije pogodno da klijent sve podatke prihvati od jednom. Servis čuva rezultate upita, a klijent je u mogućnosti da iterira kroz rezultujući niz.
- Da bi ovo omogućio, pored metoda koje predstavljaju upit, neophodno je da servis obezbedi metode za iteriranje kroz rezultate upita:
 - `List<ResourceDescription> IteratorNext(int n, int id);` - pozivom ove metode klijent zahteva od servisa da mu prosledi narednih *n ResourceDescription*-a za upit čiji je identifikator *id*
 - `bool IteratorRewind(int id);` - zahtev servisu da se vrati na početak rezultujućeg niza
 - `int IteratorResourcesTotal(int id);` - koliko ukupno postoji *ResourceDescription*-a u rezultujućem nizu
 - `int IteratorResourcesLeft(int id);` - koliko je ostalo nepročitanih *ResourceDescription*-a
 - `bool IteratorClose(int id);` - oslobađanje resursa, poziva se kada se završi rad nad upitom čiji je identifikator *id*

Iteriranje kroz rezultat upita 2/2

```
ModelCode resourceType = ModelCode.POWERTR;

List<ModelCode> propertyIDs = new List<ModelCode>();
propertyIDs.Add(ModelCode.IDOBJ_MRID);
propertyIDs.Add(ModelCode.IDOBJ_DESCRIPTION);
propertyIDs.Add(ModelCode.EQUIPMENT_ISPRIVATE);
propertyIDs.Add(ModelCode.POWERTR_FUNC);

int iteratorId = GdaQueryProxy.GetExtentValues(resourceType, propertyIDs);

int resourcesLeft = GdaQueryProxy.IteratorResourcesLeft(iteratorId);

int numberOfResources = 500;
List<ResourceDescription> result = new List<ResourceDescription>();

while (resourcesLeft > 0)
{
    List<ResourceDescription> rds = GdaQueryProxy.IteratorNext(numberOfResources, iteratorId);
    result.AddRange(rds);
    resourcesLeft = GdaQueryProxy.IteratorResourcesLeft(iteratorId);
}

GdaQueryProxy.IteratorClose(iteratorId);
```

GetRelatedValues(...) 1/3

- Čitanje niza resursa vezanih za neki resurs (entitet)

```
public int GetRelatedValues(long source, List<ModelCode> propIds , Association association)
```

- Metodi se prosleđuje *source* koji predstavlja globalni identifikator entiteta za koji želimo da nađemo povezane resurse(entitete), lista *ModelCode*-ova koji predstavljaju identifikatore svojstva rezultujućih resursa i *association* koja predstavlja vezu između *source*-a i rezultujućih resursa
- Rezultat je identifikator upita (identifikator iteratora) na osnovu koga klijent može da pročita rezultujuće *ResourceDescription-e*

GetRelatedValues(...) 2/3

- Association – objekat kojim se definišu veze između entiteta u GDA upitima
- Sadrži tri atributa: *inverse*, *propertyId*, *type*
- *bool Inverse* – nije od interesa na ovom kursu
- *ModelCode propertyId* – identifikator svojstva tipa *reference* koji sadrži *source*. Vrednost tog svojstva sadrži globalne identifikatore resursa koji su povezani sa *source*-om preko ovog svojstva.
- *ModelCode type* – služi za filtriranje resursa koji se dobiju na osnovu *propertyId*-a. Ukoliko mu je vrednost nula ne radi se filtriranje, u suprotnom kao rezultat primene asocijacije vraćaju se samo resursi tog tipa.

```
[DataContract]
public class Association
{
    private bool inverse;
    private ModelCode propertyId;
    private ModelCode type;

    public Association()
    {
        this.inverse = false;
        this.propertyId = 0;
        this.type = 0;
    }
}
```

GetRelatedValues(...) 3/3

```
long source = 0x0000000200000001;

Association association = new Association();
association.PropertyId = ModelCode.LOCATION_PSR5;
association.Type = ModelCode.POWERTR;

List<ModelCode> propertyIDs = new List<ModelCode>();
propertyIDs.Add(ModelCode.IDOBJ_MRID);
propertyIDs.Add(ModelCode.IDOBJ_DESCRIPTION);
propertyIDs.Add(ModelCode.EQUIPMENT_ISPRIVATE);
propertyIDs.Add(ModelCode.POWERTR_FUNC);

int iteratorId = GdaQueryProxy.GetRelatedValues(source, propertyIDs, association);
int resourcesLeft = GdaQueryProxy.IteratorResourcesLeft(iteratorId);

int numberOfResources = 500;
List<ResourceDescription> result = new List<ResourceDescription>();

while (resourcesLeft > 0)
{
    List<ResourceDescription> rds = GdaQueryProxy.IteratorNext(numberOfResources, iteratorId);
    result.AddRange(rds);
    resourcesLeft = GdaQueryProxy.IteratorResourcesLeft(iteratorId);
}

GdaQueryProxy.IteratorClose(iteratorId);
```

GetDescendentValues(...) 1/2

- Čitanje niza resursa vezanih za niz prosleđenih resursa (entiteta)

```
public int GetDescendentValues(List<long> sources, List<ModelCode> propIds,  
                               List<Association> path, List<Association> tail)
```

- Metodi se prosleđuje lista *source*-eva koji predstavljaju globalne identifikator entiteta za koje želimo da nađemo povezane resurse(entitete), lista *ModelCode*-ova koji predstavljaju identifikatore svojstva rezultujućih resursa i lista asocijacija (*path*) koje predstavljaju veze između *source*-eva i rezultujućih resursa
- Rezultat je identifikator upita (identifikator iteratora) na osnovu koga klijent može da pročita rezultujuće *ResourceDescription*-e

GetDescendentValues(...) 2/2

```
List<ModelCode> properties = new List<ModelCode>();
properties.Add(ModelCode.WINDINGTEST_LOADLOSS);
properties.Add(ModelCode.WINDINGTEST_NOLOADLOSS);
properties.Add(ModelCode.IDOBJ_MRID);

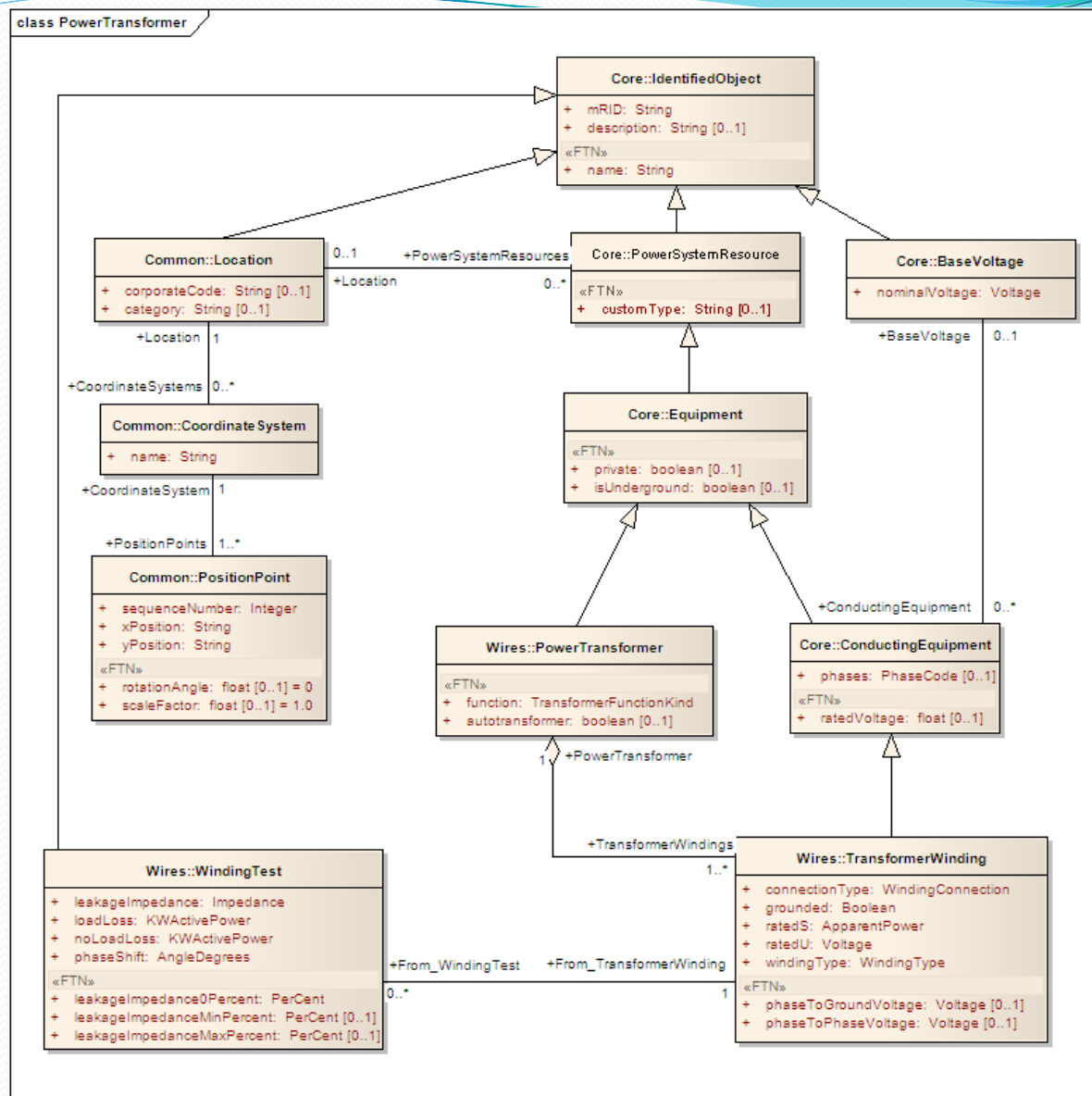
Association firstAssociation = new Association();
firstAssociation.PropertyId = ModelCode.BASEVOLTAGE_CONDEQS;
firstAssociation.Type = ModelCode.POWERTRWINDING;

Association secondAssociation = new Association();
secondAssociation.PropertyId = ModelCode.POWERTRWINDING_TESTS;
secondAssociation.Type = 0;

List<Association> path = new List<Association>() { firstAssociation, secondAssociation };
List<Association> tail = new List<Association>();

List<long> sources = new List<long>() { 0x00000000100000001, 0x00000000100000002, 0x00000000100000003 };

int iteratorId = GdaQueryProxy.GetDescendentValues(sources, properties, path, tail);
```



Zadaci

1. Pročitati sve *PowerTransformer*-e sa servisa. Za svaki od od njih tražiti sve *property*e koje *PowerTransformer* sadrži. Rezultate prikazati na proizvoljan način (štampati u XML, prikazati na formi...).
2. Za *PowerTransformer* koji ima najmanji globalni identifikator pročitati *TransformerWinding*-e koji su vezani za njega. Tražiti sve *property*e koje *TransformerWinding* sadrži. Rezultate prikazati na proizvoljan način (štampati u XML, prikazati na formi...).