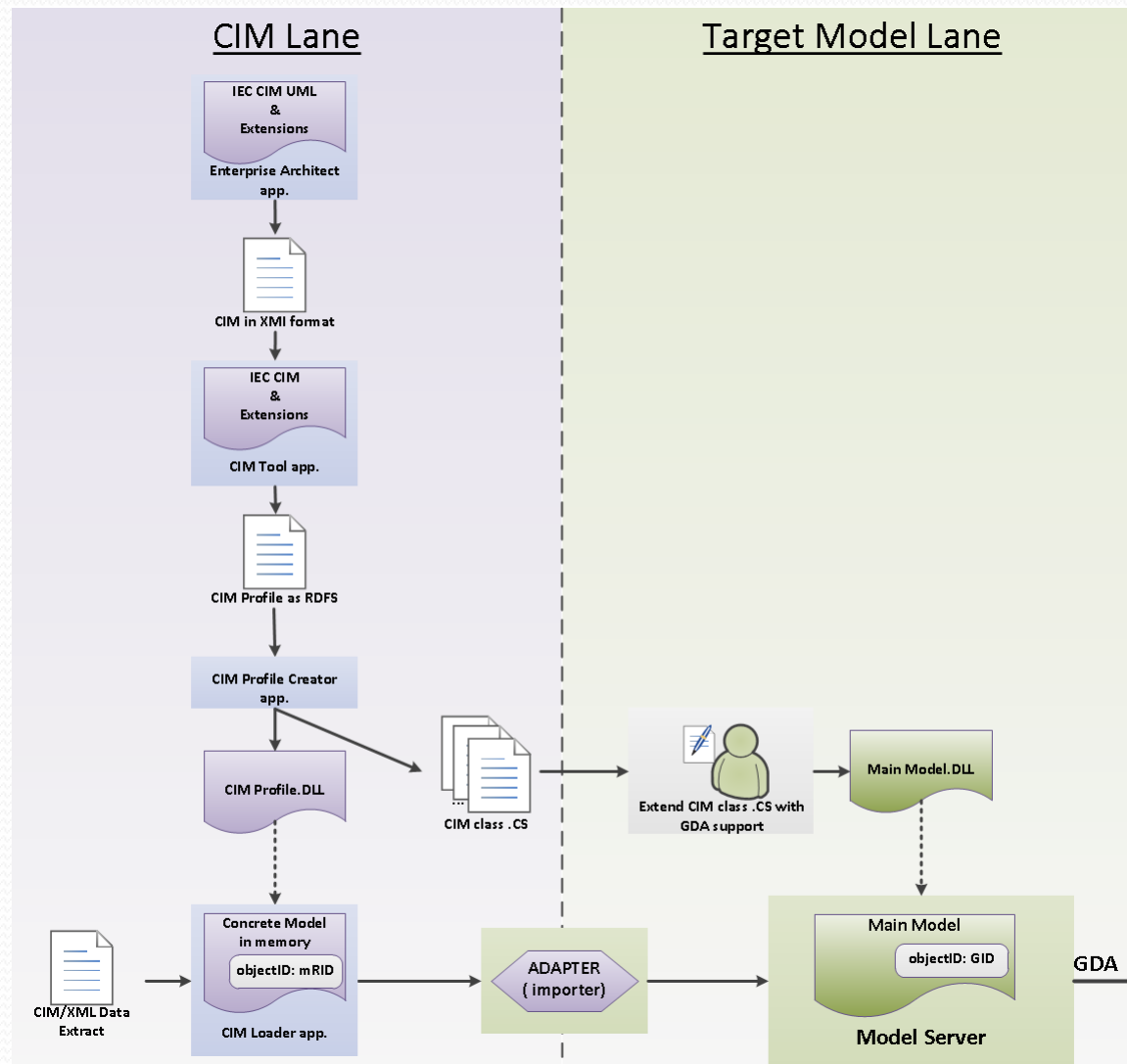


Standardi i modeliranje elektroenergetskih sistema

VEŽBA 4:

Parser CIM profila datog u RDFS formatu

Tok podataka pri inicijalizaciji modela elektroenergetske mreže



Pregled RDFS

formata zapisa CIM profila

The screenshot displays the Eclipse IDE interface for CIMTool, specifically showing the RDFS profile for a PowerTransformer. The title bar indicates the project is 'CIMTool - SIMESLabs/Profiles/PowerTransformer.legacy-rdfs - Eclipse Platform'.

Project Explorer: Shows the project structure with folders 'SIMESLabs', 'Incremental', 'Instances', 'Profiles', and 'Schema'. Under 'Profiles', files like 'LabsProfile.html', 'LabsProfile.legacy-rdfs', 'LabsProfile.owl', 'PowerTransformer.html', 'PowerTransformer.legacy-rdfs', and 'PowerTransformer.owl' are listed. Under 'Schema', 'simes-labs-schema.xmi' is shown.

PowerTransformer.owl: The main editor displays the RDFS profile content, which is an XML document with the following structure:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cims="http://iec.ch/TC57/1999/rdf-schema-extensions-19990926#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:msg="http://langdale.com.au/2005/Message#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:uml="http://langdale.com.au/2005/UML#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:j.0="http://iec.ch/TC57/2010/CIM-schema-cim15#"
  xml:base="http://iec.ch/TC57/2010/CIM-schema-cim15#" >
  <rdf:Description rdf:about="#TransformerWinding.windingType">
    <cims:stereotype rdf:resource="http://langdale.com.au/2005/UML#attribute"/>
    <rdfs:comment>The type of winding.</rdfs:comment>
    <rdfs:label>windingType</rdfs:label>
    <rdfs:range rdf:resource="#WindingType"/>
    <rdfs:domain rdf:resource="#TransformerWinding"/>
    <cims:multiplicity rdf:resource="http://iec.ch/TC57/1999/rdf-schema-extensions-19990926#" />
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdf:Description>
  <rdf:Description rdf:about="#Package_Wires">
    <cims:belongsToCategory rdf:resource="#Package_IEC61970"/>
    <rdfs:comment>An extension to the Core and Topology package that models information on t
    <rdfs:label>Wires</rdfs:label>
    <rdf:type rdf:resource="http://iec.ch/TC57/1999/rdf-schema-extensions-19990926#ClassCate
```

Project Model: Shows the package structure, including 'Package: IEC61970_IEC61968_combine', 'Class: Package', 'Class: Stereotype', and various datatypes like 'EnergyAsMWh', 'EnumeratedType', 'FlowgateAfcUseCode', 'FlowgateIdcType', 'FreqBiasFactor', 'PenaltyFactor', 'PotheadType', and 'PowerROCPeMin'.

Documentation: The bottom panel shows the 'PowerTransformerProfile' documentation.

Parsiranje XML-a

- DOM (Document Object Model) parser
 - Učitava se ceo XML dokument od jednom
 - .NET - XmlDocument
 - Može nastati problem ukoliko su u pitanju veliki XML dokumenti
- SAX (Simple API for XML) parser
 - Stream parser - Čita se element po element
 - Event driven – Kada se pročita predodređeni element XML-a, pozivaju se odgovarajuće metode za obradu
 - .NET - XmlReader
 - Obrada velikih XML dokumenata ne predstavlja problem

SAX parser 1/3

- Obrada XML dokumenta
 - Čita se element po element XML dokumenta
 - Od interesa su najčešće sledeći elementi: početak tag-a, atributi definisani u tagu, vrednost(tekst) navedena u tag-u, kraj tag-a
 - U zavisnosti od tipa elementa pozivaju se određene metode za obradu.
- Primer 1: `<item type="typeValue" > text</item>`
 - `<item>` - početak tag-a
 - `type` – atribut definisan za tag `item`
 - `"typeValue"` – vrednost atributa `type`
 - `characters` – vrednost (tekst) navedena u tag-u
 - `</item>` - kraj tag-a

SAX parser 2/3

- Primer 2: `<item type="typeValue" />`
 - Empty tag
 - Sadrži samo atribut, nema vrednosti (tekst)
- Metode za obradu XML dokumenta
 - StartDocument(...) - Poziva se pre početka parsiranja. U ovoj metodi se uglavnom rade inicijalizacije potrebnih promenljivih.
 - StartElement(...) – Poziva se kada se pročita početak tag-a.
 - EndElement(...) – Poziva se kada se pročita kraj tag-a.
 - Characters(...) – Poziva se kada se pročita vrednost (tekst) navedena unutar tag-a
 - EndDocument() - Poziva se na kraju parsiranja. Uglavnom se oslobađaju zauzeti resursi.

SAX parser 3/3

- Primer jednostavnog SAX parsera: pročitati zaposlene opisane u XML dokumentu i dodati ih u kolekciju svih zaposlenih

```
<?xml version="1.0"?>
  <Personnel>
    <Employee   typeOfEmployment="permanent">
      <Name>Marko</Name>
    </Employee>

    <Employee   typeOfEmployment="contract">
      <Name>Nikola</Name>
    </Employee>
  </Personnel>
```

- Algoritam
 - Čitati element po element
 - Za svaki otvoreni tag *Employee* programski kreirati objekat tipa *Employee*
 - Pročitati sve attribute dodeljene tagu *Employee* i programski ih dodeliti odgovarajućim property-ima objekta *Employee*
 - Pročitati sve podtagove taga *Employee* i njihove vrednosti (tekst) programski dodeliti odgovarajućim property-ima objekta *Employee*
 - Za svaki zatvoren tag *Employee* programski dodati *Employee* objekat u kolekciju zaposlenih
 - Implementacija u projektu *SaxParserEmployee*

RDFS SAX parser 1/2

- Koristimo ga radi programske manipulacije definisanog CIM profila
 - Rezultat parsiranja predstavlja objekat tipa *Profile* koji u sebi sadrži sve informacije potrebne za opis CIM profila definisanog RDFS-om
- Implementacija
 - `StartDocument()` – Inicijalizacija svih potrebnih promenljivih za kreiranje
 - `StartElement()`
 - Pre poziva metode prikupljaju se svi atributi odgovarajućeg resursa
 - Poziva se metoda i prikupljaju podaci potrebni za kreiranje objekata koji opisuju resurse
 - Ukoliko je u pitanju prazan element `<itam/>` ne očekujemo da ćemo pročitati zatvarajući tag za ovaj element, pa se nakon `StartElement()` metode automatski poziva `EndElement()` metoda.

RDFS SAX parser 2/3

- Implementacija...
 - EndElement() – Na osnovu prkupljenih informacija u metodi *StartElement()* kreiraju se objekti koji opisuju RDFS resurse i dodeljuju profilu.
 - Obrada je podeljena na nekoliko celina:
 - Prvi deo obrađuje resurse definisane RDFS-om. Određen je delom koda ograničenog uslovom: `if (qName.Equals(rdfProfileElement) || ...`
 - Ova celina je dalje izdeljena na obradu resursa različitog tipa:
 - Paket: `if (string.Compare(ExtractSimpleNameFromResourceURI(type), "ClassCategory") == 0)`
 - Klasa: `else if (string.Compare(ExtractSimpleNameFromResourceURI(type), "Class") == 0)`
 - Atribut: `else if (string.Compare(ExtractSimpleNameFromResourceURI(type), "Property") == 0)`
 - Ostatak metode obrađuje podatke vezane za
 - Labele: `else if (qName.Equals(rdfsLabel))`
 - Komentar: `else if (qName.Equals(rdfsComment))`
 - itd.

RDFS SAX parser 3/3

- Implementacija...
 - Characters() – Prikuplja definisane vrednosti (tekst)
 - EndDocument()-Oslobađa resurse i vrši dodatno procesiranje objekta tipa *Profile* ukoliko je potrebno

Parser CIM profila datog u RDFS formatu

- Opis pripremljenog rešenja
 - učitati RDFS dokument (koji definiše CIM profil) u odgovarajući objektni model u memoriji;
 - odštampati sadržaj objektnog modela u tekstualni izveštaj (paketi, klase u paketu, atributi u klasi).
- Zadaci
 - U jedinoj formi dodati TextBox u kome će se ispisati
 - Broj klasa definisanih u RDFSu
 - Ukupan broj atributa definisanih u RDFSu
 - Broj komentara definisanih u RDFSu
 - Demonstrirati rad sa conditional breakpoint-ima
 - Prikazati mogućnosti optimizacije datog koda