



MODEL DRIVEN SOFTWARE DEVELOPMENT

LECTURE : 9

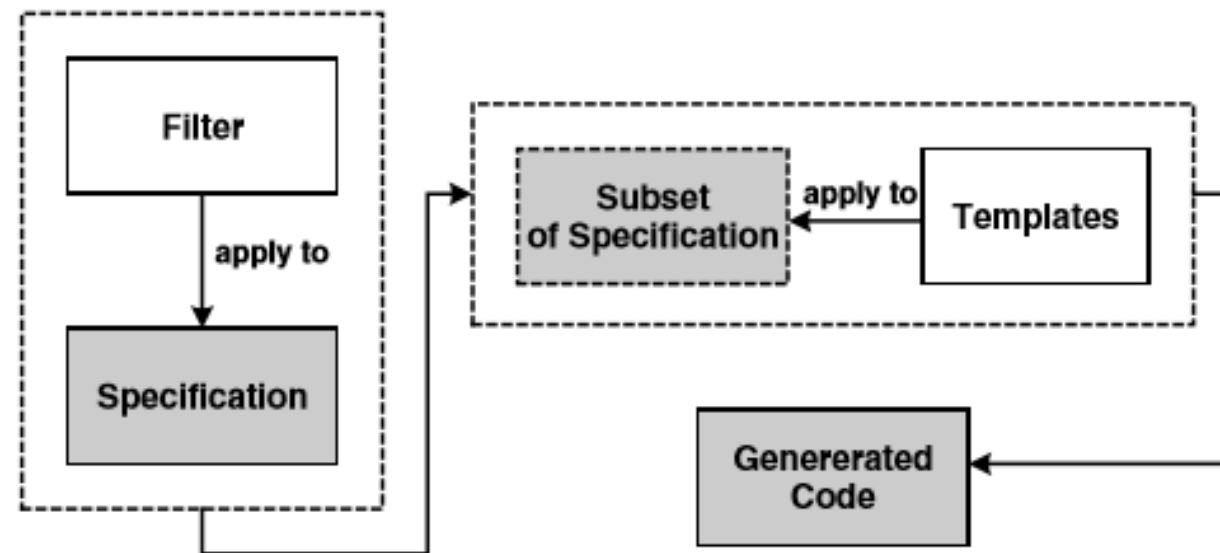


CODE GENERATION TECHNIQUE

- Code generation is the technique of using or writing programs that write source code.
- **Model-to-Text**, whereas text may be distinguished in
 - Program code
 - Documentation
 - Test cases
 - Model serialization (XMI)

TEMPLATES AND FILTERING

- This generation technique describes the simplest case of code generation
- we use templates to iterate over the relevant parts of a textually-represented model, for example using XSLT via XML.



TEMPLATES AND FILTERING

- The code to be generated is found in the templates.
- Variables in the templates can be bound to values from the model.
- Below, we will present a simple example in which a Java Bean Person is generated from an XML specification (for simplification purposes, we do not use XML as the model representation, instead we use a custom schema). This is the specification

```
<class name="Person" package="com.mycompany">  
  <attribute name="name" type="String"/>  
  <attribute name="age" type="int"/>  
</class>
```

Code Generated →

```
package com.mycompany;  
public class Person {  
    private String name;  
    private int age;  
    public String get_name() {return name;}  
    public void set_name( String name) {this.name = name;}  
    public int get_age() {return age;}  
    public void set_age( int age ) {this.age = age;}  
}
```

```
<xsl:template match="/class">
  package <xsl:value-of select="@package"/>;
  public class <xsl:value-of select="@name"/> {
    <xsl:apply-templates select="attribute"/>
  }
</xsl:template>

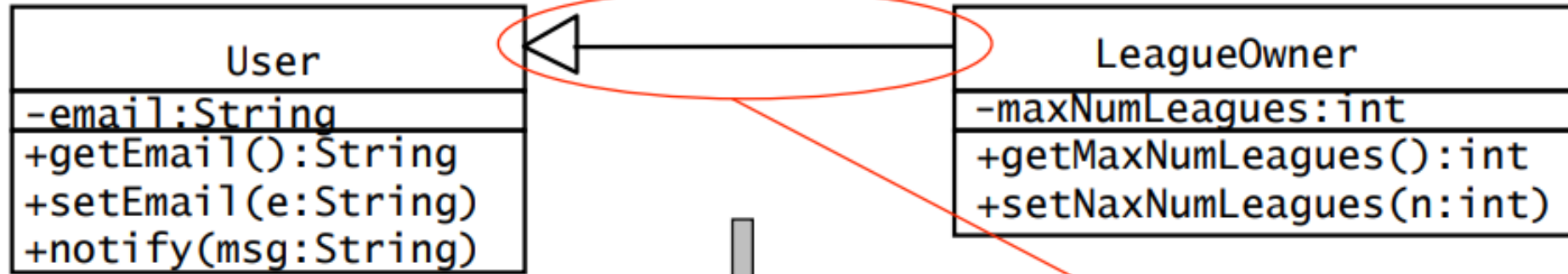
<xsl:template match="attribute">

  private <xsl:value-of select="@type"/>
    <xsl:value-of select="@name"/>;

  public <xsl:value-of select="@type"/>
    get_<xsl:value-of select="@name"/>() {
      return <xsl:value-of select="@name"/>;
    }

  public void set_<xsl:value-of select="@name"/> (
    <xsl:value-of select="@type"/>
    <xsl:value-of select="@name"/>) {
    this.<xsl:value-of select="@name"/> =
      <xsl:value-of select="@name"/>;
  }
</xsl:template>
```

Object design model before transformation:



Source code after transformation:

```
public class User {
    private String email;
    public String getEmail() {
        return email;
    }
    public void setEmail(String value){
        email = value;
    }
    public void notify(String msg) {
        // ....
    }
}
```

```
public class LeagueOwner extends User {
    private int maxNumLeagues;
    public int getMaxNumLeagues() {
        return maxNumLeagues;
    }
    public void setMaxNumLeagues
        (int value) {
        maxNumLeagues = value;
    }
}
```