

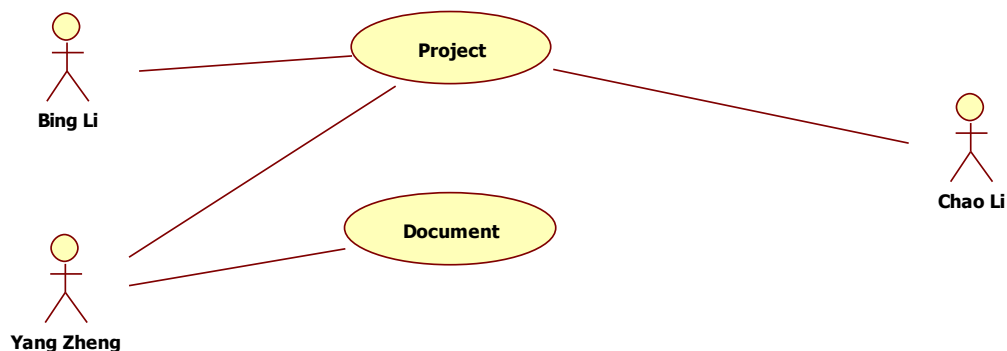
Parser for extracting use case information from XMI

----- Sprint2 Report by Chao Li, Yang Zheng, Bing Li

By using the DOM4J library, we successfully extracted the key information of the Use Case diagram from the XMI file. But the constraint of our achievement is that only supporting the XMI file that is generated by the StarUML tool.

As we mentioned above, the following information have been extracted from XMI file:
For example, there is a Use Case diagram that is created by the StarUML tool:

Use Case Diagram:



(1) The XMI file of the Use Case of the diagram:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<XMI xmi.version = "1.1" xmlns:UML="href://org.omg/UML/1.3" timestamp = "Mon Feb
03 15:33:31 2014">
  <XMI.header>
    <XMI.documentation>
      <XMI.owner></XMI.owner>
      <XMI.contact></XMI.contact>
      <XMI.exporter>StarUML.XMI-Addin</XMI.exporter>
      <XMI.exporterVersion>1.0</XMI.exporterVersion>
      <XMI.notice></XMI.notice>
    </XMI.documentation>
    <XMI.metamodel xmi.name = "UML" xmi.version = "1.3"/>
  </XMI.header>
  <XMI.content>
    <UML:Model xmi.id="UMLProject.1">
      <UML:Namespace.ownedElement>
        <UML:Model xmi.id="UMLModel.2" name="Use Case Model" visibility="public"
```

```

isSpecification="false" namespace="UMLProject.1" isRoot="false" isLeaf="false"
isAbstract="false">
    <UML:Namespace.ownedElement>
        <UML:Actor xmi.id="UMLActor.3" name="Bing Li" visibility="public"
isSpecification="false" namespace="UMLModel.2" isRoot="false" isLeaf="false"
isAbstract="false" participant="UMLAssociationEnd.12"/>
        <UML:UseCase xmi.id="UMLUseCase.4" name="Project" visibility="public"
isSpecification="false" namespace="UMLModel.2" isRoot="false" isLeaf="false"
isAbstract="false" participant="UMLAssociationEnd.13 UMLAssociationEnd.16
UMLAssociationEnd.19"/>
        <UML:UseCase xmi.id="UMLUseCase.5" name="Document" visibility="public"
isSpecification="false" namespace="UMLModel.2" isRoot="false" isLeaf="false"
isAbstract="false" participant="UMLAssociationEnd.10"/>
        <UML:Actor xmi.id="UMLActor.6" name="Chao" visibility="public"
isSpecification="false" namespace="UMLModel.2" isRoot="false" isLeaf="false"
isAbstract="false" participant="UMLAssociationEnd.15"/>
        <UML:Actor xmi.id="UMLActor.7" name="Yang" visibility="public"
isSpecification="false" namespace="UMLModel.2" isRoot="false" isLeaf="false"
isAbstract="false" participant="UMLAssociationEnd.9 UMLAssociationEnd.18"/>
        <UML:Association xmi.id="UMLAssociation.8" name="" visibility="public"
isSpecification="false" namespace="UMLModel.2">
            <UML:Association.connection>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.9" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.8" type="UMLActor.7"/>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.10" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.8" type="UMLUseCase.5"/>
            </UML:Association.connection>
        </UML:Association>
        <UML:Association xmi.id="UMLAssociation.11" name="" visibility="public"
isSpecification="false" namespace="UMLModel.2">
            <UML:Association.connection>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.12" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.11" type="UMLActor.3"/>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.13" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.11" type="UMLUseCase.4"/>
            </UML:Association.connection>

```

```

        </UML:Association>
        <UML:Association xmi.id="UMLAssociation.14" name="" visibility="public"
isSpecification="false" namespace="UMLModel.2">
            <UML:Association.connection>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.15" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.14" type="UMLActor.6"/>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.16" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.14" type="UMLUseCase.4"/>
            </UML:Association.connection>
        </UML:Association>
        <UML:Association xmi.id="UMLAssociation.17" name="" visibility="public"
isSpecification="false" namespace="UMLModel.2">
            <UML:Association.connection>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.18" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.17" type="UMLActor.7"/>
                <UML:AssociationEnd xmi.id="UMLAssociationEnd.19" name=""
visibility="public" isSpecification="false" isNavigable="true" ordering="unordered"
aggregation="none" targetScope="instance" changeability="changeable"
association="UMLAssociation.17" type="UMLUseCase.4"/>
            </UML:Association.connection>
        </UML:Association>
        <UML:Stereotype xmi.id="X.24" name="useCaseModel"
extendedElement="UMLModel.2"/>
    </UML:Namespace.ownedElement>
</UML:Model>
<UML:Model xmi.id="UMLModel.20" name="Analysis Model" visibility="public"
isSpecification="false" namespace="UMLProject.1" isRoot="false" isLeaf="false"
isAbstract="false">
    <UML:Namespace.ownedElement>
        <UML:Stereotype xmi.id="X.25" name="analysisModel"
extendedElement="UMLModel.20"/>
    </UML:Namespace.ownedElement>
</UML:Model>
<UML:Model xmi.id="UMLModel.21" name="Design Model" visibility="public"
isSpecification="false" namespace="UMLProject.1" isRoot="false" isLeaf="false"
isAbstract="false">
    <UML:Namespace.ownedElement>
        <UML:Stereotype xmi.id="X.26" name="designModel"

```

```

extendedElement="UMLModel.21"/>
    </UML:Namespace.ownedElement>
</UML:Model>
<UML:Model xmi.id="UMLModel.22" name="Implementation Model"
visibility="public" isSpecification="false" namespace="UMLProject.1" isRoot="false"
isLeaf="false" isAbstract="false">
    <UML:Namespace.ownedElement>
        <UML:Stereotype xmi.id="X.27" name="implementationModel"
extendedElement="UMLModel.22"/>
    </UML:Namespace.ownedElement>
</UML:Model>
<UML:Model xmi.id="UMLModel.23" name="Deployment Model"
visibility="public" isSpecification="false" namespace="UMLProject.1" isRoot="false"
isLeaf="false" isAbstract="false">
    <UML:Namespace.ownedElement>
        <UML:Stereotype xmi.id="X.28" name="deploymentModel"
extendedElement="UMLModel.23"/>
    </UML:Namespace.ownedElement>
</UML:Model>
</UML:Namespace.ownedElement>
</UML:Model>
</XML.content>
</XMI>

```

The parser to the XMI file:

```

public void XMLReader() throws DocumentException
{
    SAXReader reader = new SAXReader();
    Document doc = reader.read(new File("testxmi.xmi"));
    Element root = doc.getRootElement();
    parse(root);
}

public void parse(Element ele)
{
    parseAttribute(ele);
    List el = ele.elements();
    for(Object e:el)
    {
        Element element = (Element)e;
        if(!element.isTextOnly())
        {
            parse(element);
        }
    }
}

```

```

        else
        {
            parseAttribute(element);
            System.out.println(element.getQName().getName()+
                "--->" + element.getText());
        }
    }

}

public void parseAttribute(Element ele)
{
    String associationName="";
    ClassDiagramItems cdi = new ClassDiagramItems();

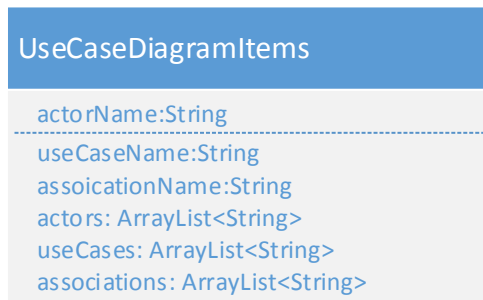
    List attList = ele.attributes();
    for(Object e:attList)
    {
        Attribute attr = (Attribute)e;
        String str = new String(attr.getValue());

        if(ele.getName().equals("Association")&&attr.getQName().getName().equals("xmi.id"))
        {
            associationName = attr.getValue();
            System.out.println(associationName);
        }

        if(ele.getName().equals("AssociationEnd")
            &&attr.getQName().getName().equals("association"))
        {
            if(attr.getValue().equals(associationName));
            {
                System.out.println(ele.attribute(10).getValue());
            }
        }
    }
}

```

The class diagram to store use case, actor and associations:



This class will be used by merging and evaluation functions.

In future, we will use strategy design pattern to support XMI file generated by other UML tools.