



Network Computing

Lecture on XML

Instructor: Louise Moser

TA: Yung-Ting Chuang

Quarter: Spring 2012

Agenda

- Websites and google group websites.
- Review XML:
 - How we write toXML() in our first project
 - How we parse/read XML in our first project
 - How to merge XML with GUI
- Overview of Sockets

- Class website:

http://www.ece.ucsb.edu/courses/ECE155/155B_S12Moser/

- Google group website:

<http://groups.google.com/group/ece155bs12/>

- Email to google group:

Send email to: ece155bs12@googlegroups.com

Example XML file for project1

```
<?xml version='1.0' ?>
<Doctor>
  <DoctorName>Yung-Ting</DoctorName>
  <DoctorLastName>Chuang</DoctorLastName>
  <Patients>
    <Patient>
      <PatientSSN>123-45-2355</PatientSSN>
      <PatientFirstName>Po-Yo</PatientFirstName>
      <Notes></Notes>
    </Patient>
    <Patient>
      <PatientSSN>236-34-0292</PatientSSN>
      <PatientFirstName>Shu-Ha</PatientFirstName>
      <Notes>
        <Note>
          <Author>Yung-Ting Chuang</Author>
          <Date>2011/3/24</Date>
          <Content>Has foot injury</Content>
        </Note>
      </Notes>
    </Patient>
  </Patients>
</Doctor>
```

How to write XML in project1

- In project1, when the doctor need to retrieve his/her patient information, we need to use the `toXML()` method to write the information in XML format.
- This method will be similar to the `toString()` method, but will return the String in XML format.

Example for Doctor code:

Doctor class

```
public class Doctor
{
    public String name = "MyName";
    public String lastname = "MyLastName";

    Vector <Patient> patients; // doctor keep patient information in a vector

    public String toXML()
    {
        String xml = "<Doctor>";
        xml += "<DoctorName>" + name + "</DoctorName>";
        xml += "<DoctorLastName>" + lastname + "</DoctorLastName>";
        xml += "<Patients>";
        for (int i = 0; i < patients.size(); i++)
            xml += patients.elementAt(i).toXML(); // add xml code for each patient
        xml += "</Patients>";
        xml += "</Doctor>";

        return xml;
    }
}
```

Just a reminder...

- Remember,
<Patients></Patients> is not the same as <Patient></Patient>
 - So in Doctor.java, we have <Patients> </Patients> tag
 - But in Patient.java, we have <Patient> </Patient> tag
- Similarly, <Notes></Notes> is different than <Note></Note>
 - So in Patient.java, we have <Notes></Notes>
 - In Notes.java, we have <Note></Note>

How to read XML in project1

- We use import `javax.xml.parsers` and `org.xml.sax` packages to read XML data. Example:
`SAXParserFactory spf = SAXParserFactory.newInstance()`
`SAXParser sp = spf.newSAXParser()`
- We first create a parser object. We need to provide a parser for a `HandlerBase` object which has the following methods for us to overwrite:
 - **`void startElement(String name, AttributeList list)`**
 - **`void endElement(String name)`**
- In the ***startElement*** method, if the object name such as Patient, Note, or Doctor , we create an object for it and insert it into the vector.
- In the ***endElement*** method, we set the values of Objects. If it is the object name such as Patient, Note, Doctor we do not need to do anything about them.
- It is important that the tag names that you use when writing and when parsing are the same (and case sensitive). Otherwise, you will have trouble reading what you have written.

Example of XMLParser reading XML file we had in slide 5:

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.parsers.*;

public class XMLParser extends DefaultHandler
{
    public Doctor doctor;
    private Patient patient;
    private Note note;
    private StringBuffer accumulator = new StringBuffer();    // Buffer used during parsing

    public XMLParser()
    {
        doctor = new Doctor();
    }
    public void characters(char[] buffer, int start, int length)
    {
        accumulator.append(buffer, start, length);
    }

    //Each time a new tag is opened, this method is called
    public void startElement(String uri,String lname, String name, Attributes attributes)
    {

```

Example of XMLParser (continued):

```
// -----  
//          New Note starts  
if(name.equals ("Note"))  
{  
    note = new Note();  
    patient.addNote(note);  
    // we will set values of this later  
    // add to patient  
}  
// -----  
//          New Patient starts  
else if(name.equals ("Patient"))  
{  
    patient = new Patient();  
    doctor.addPatient(patient);  
    // we will set values of patient later  
    // add to doctor  
}  
// -----  
// Reset accumulator  
accumulator.setLength(o);  
}
```

Example of XMLParser (continued):

```
public void endElement(String uri,String lname,String name)
{
    String value = accumulator.toString();
    // -----
    // Doctor information
    if(name.equals ("DoctorName"))
        doctor.name= value;
    else if(name.equals ("DoctorLastName"))
        doctor.lastname = value;
    // -----
    // Patient information
        else if(name.equalsIgnoreCase("PatientSSN"))
            patient.ssn = value;
        else if(name.equalsIgnoreCase("PatientFirstName"))
            patient.name = value;
    // -----
    // Note information
        else if(name.equalsIgnoreCase("Author"))
            note.author = value;
        else if(name.equalsIgnoreCase("Date"))
            note.date = value;
    }
}
```

Combining GUI and XML...

- Below is the example layout of DoctorApplet.java:

```
public class DoctorApplet extends JApplet implements ActionListener{

    public void init() { makeGUI(); }

    public void destroy() { //toXmlFile(doctor,fileurl);}

    public Doctor readXmlFile(String url) throws Exception { //same code as in demoXML.java}

    public void makeGUI(){}

    class PatientTab extends JTabbedPane implements ActionListener{}

    public void actionPerformed(ActionEvent e){}
}
```