Lab 2.3

Auto-generating Java classes using JAXB

Outline

- Perform a blast search using your favorite protein, and collect the results in XML format. Read the outline for the DOCTYPE fix.
- Use the XML Schema provided to auto-generate the Java classes needed by the JAXB runtime environment.
- Create a class which:
  - Initializes the JAXB runtime environment with your content tree.
  - Unmarshalls your blast results from file to the content tree.
  - Validates the content tree (unmarshall-time).
  - Outputs a summary of the results to console.

Getting the source code

- Use CVS!
- Module name is “blastoutput”

Making eclipse happy (optional – you can use command line)

- You may have conflicting versions of xerces.
- This can be overriden by overriding the ANT classpath
- Under the Window->preferences menu, expand the Ant item, and choose Runtime.
- Add the following External JARs to the classpath:
  - /opt/diro/jwsdp-1.3/jaxp/lib/endorsed/xercesImpl.jar
  - /opt/diro/jwsdp-1.3/jaxp/lib/endorsed/dom.jar
- Create a classpath variable called JWSDP_HOME and point it to /opt/diro/jwsdp-1.3

Where did this XML Schema come from?

- NCBI makes the Document Type Definition (DTD) for BlastOutput available through their website
  - http://www.ncbi.nih.gov/dtd
- Converted to XML Schema using Datatool program (part of the NCBI C++ toolkit)
  - Note: Autogeneration
- NOTE: BioJava Blast Parser will be introduced in lecture 3.1

Reality Check

- XML Schema specification is not fully mature.
- The decimal datatype cannot handle scientific notation (Example: 1.6E-311).
- A tool developer must always take note of the special cases.
- XML Schema was modified to handle this type by hand

Solution

- A SimpleType was created.

```
<xs:simpleType name="sn-double">
    <xs:restriction base="xs:double">
        <xs:pattern value="-?\d\.\d+((E|e)(\+|\-)\d{2,3})?"/>
        <xs:pattern value="\d+"/>
        <xs:pattern value="-?\d+\.\d+"/>
    </xs:restriction>
</xs:simpleType>
```

Step1: Using ANT

- Use the build.xml ANT script provided
- Targets are:
  - generate
  - compile
  - run
  - docs

- ant –Djwsdp.home=“” <target>

Step2: Generating The Source

- ant generate
  - This target will run the JAXB Binding Compiler and generate your content tree Java source files.
- ant compile
  - This target will compile all sources in the src subdirectory.
- ant run
  - This target will run the ca.bioinformatics.blastoutput.main.Main class.
- ant docs
  - This target generates the javadoc for your content tree

Step3: Creating the Main class

- Modify the ca.bioinformatics.dtt.main.Main class.
- This will be your startup class which will initialize the JAXB environment, unmarshall, validate, and output a summary.

Step 4: Generate output

- Your application should generate the following information about your blast result.
- Accession of hit
  - E-value and score of HSPs

```
Results of BLAST search

Accession:NP_055202
        evalue=1.07425E-77      score=744.0
Accession:CAA11218
        evalue=1.76723E-58      score=579.0
Accession:NP_110480
        evalue=7.72818E-34      score=366.0
Accession:NP_034819
        evalue=3.83768E-31      score=343.0
Accession:AAH52340
        evalue=1.23418E-30      score=338.0
Accession:BAC40554
        evalue=1.75215E-30      score=337.0
Accession:NP_004723
        evalue=9.85871  score=72.0
[marc@blueprint-14 bin]$
```