The State Space in XML Format

EECS 4315

www.cse.yorku.ca/course/4315/

Today's Plan

- Implementing a ***PublisherExtension***
- Parameterizing a Listener

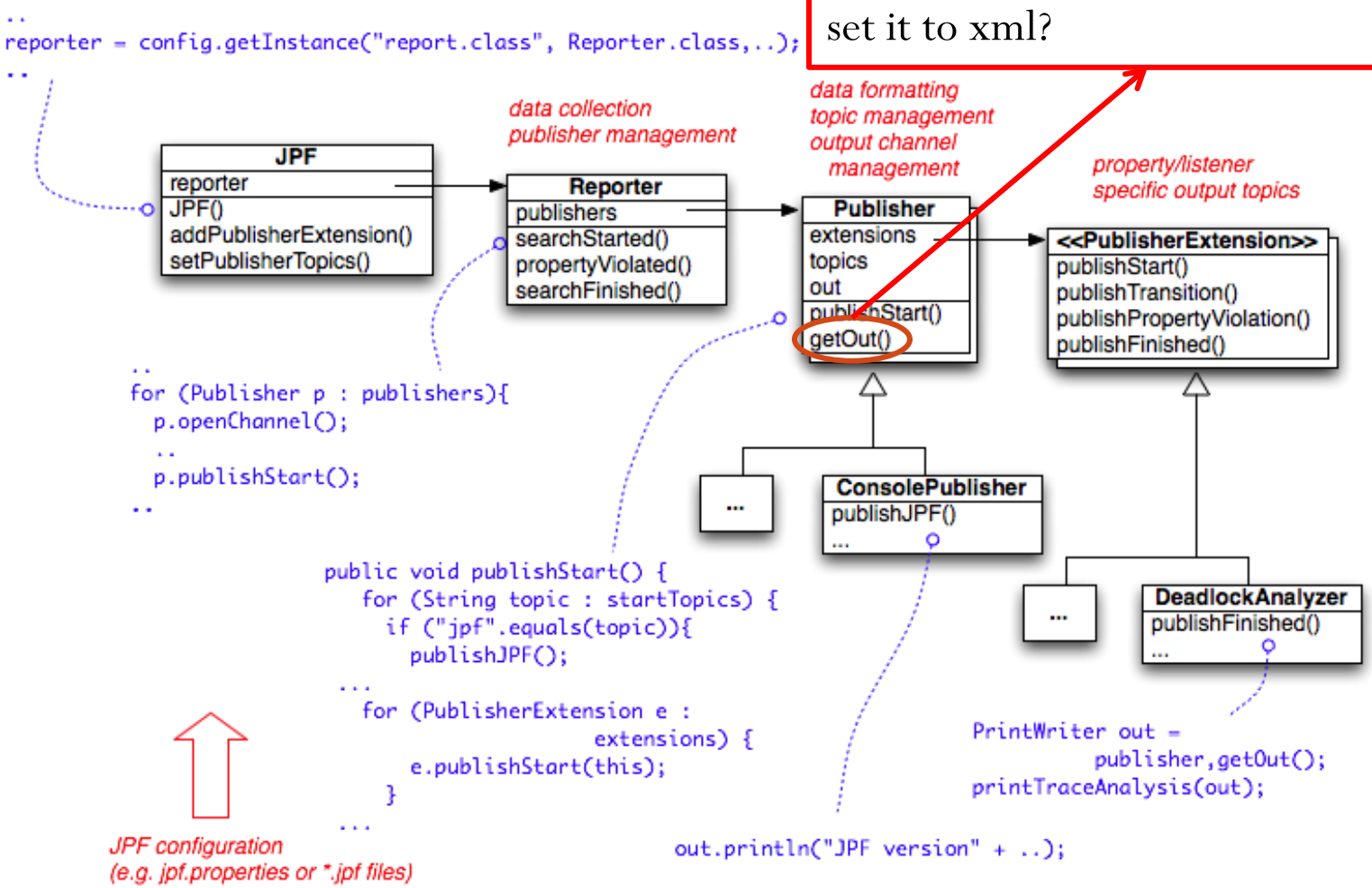
JPF Report System

Three major components:

- the ***Reporter***
- any number of format specific ***Publisher*** objects
- any number of tool-, property- and Publisher-specific ***PublisherExtension*** objects

JPF Report System

Return the ***PrintWriter*** object which is used by JPF to print data. Its default value is console. How to set it to xml?



Picture from:
http://babelfish.arc.nasa.gov/trac/jpf/wiki/devel/report

Configure the Properties

- Set the publisher to be console and xml where console is the default value

  report.publisher = console, xml

- Set the output file name

  report.xml.file = HelloWorld

The *PublisherExtension* Interface

```
public interface PublisherExtension {
        void publishStart(Publisher publisher);
        void publishTransition(Publisher publisher);
        void publishPropertyViolation(Publisher publisher);
        void publishConstraintHit(Publisher publisher);
        void publishFinished(Publisher publisher);
        void publishProbe(Publisher publisher);
}
```

The State Space in XML Format

<scxml>: The top-level wrapper element, which carries version information.

Example:

```
<state id="s">
  <transition event="e" cond="x==1" target="s1"/>
  <transition event="e" target="s2"/>
  <transition event="*" target="s3"/>
</state>
```

SCXML: https://www.w3.org/TR/scxml/#scxml

New *StateSpacePrinter*

```
public class StateSpacePrinter extends ListenerAdapter implements SearchListener, PublisherExtension{
        private int source;
        private int target;

        public StateSpacePrinter(Config config, JPF jpf) {
            source = -1;
            target = -1;
            jpf.addPublisherExtension(Publisher.class, this);
        }

        @Override
        public void publishTransition(Publisher publisher) {
            PrintWriter out = publisher.getOut();
            if (source != -1) {
                ...
            }
        }
}
```

Parameterizing a Listener

```
private String separator;
public StateSpacePrinter(Config config) {
    source = -1;
    target = -1;
    separator = config.getString("stateSpacePrinter.separator", "->");
}
```

We can set the separator in the application properties file:

```
stateSpacePrinter.separator =  -->
```

The default value is given in the constructor.