The Design of an XML-Based Experiment and Model Description Language

Andrew W. Hallagan

Department of Computer Science
Bucknell University

November 5, 2010

Big Idea

XML languages developed for

- **Experiment Description**: allows researchers to describe an experiment space in a way that is intuitive, elegant and can easily be transformed into a series of individual design points.
- **Model Description**: allows researchers to describe the structure of the components they are using without working directly with a C++ or Python script.

Roadmap

Experiment Description

- What it does.
- A conceptual example (i.e., no code).
- An implemented example.
- Results.

Model Description

- What it does.
- A conceptual example (i.e., no code).
- An implemented example.

Running Example: Csma Bridge

The OnOffHelper component in C++:

```
OnOffHelper onoff ( /* ... */ );
onoff.SetAttribute ("OnTime",
      RandomVariableValue (ConstantVariable (1)));
onoff.SetAttribute ("OffTime",
      RandomVariableValue (ConstantVariable (0)));
onoff.SetATtribute ("DataRate",
      DataRateValue (DataRate(50000)));
onoff.SetATtribute ("PacketSize",
      UintegerValue (uint32 (512)));
```

What the Experiment Description Language Does

- Lists each experimental factor in turn and describes some list of parameter values which the factor will take on.
- Provides constructs for building different types of lists.
- Describes a factorial experiment design.
- Prunes factorial experiment design.

Conceptual Example

Suppose the following experiment space:

$$
\begin{aligned}
onTime &\in [10, 16, 27, 31] \\
offTime &\in [10 \times i : 0 \leq i \leq 3] \\
packetSize &\in ListA \\
dataRate &\in ListA \\
ListA &= [1, 2, 3]
\end{aligned}
$$

With the restriction on each design point:
$offTime > onTime$ and $packetSize = dataRate$.

Implemented Example

The OnOffHelper experiment space in XML:

```
<factor>offTime</factor>
<test>EQUALS</test>
<lconst>10</lconst>
<op>MULT</op>
<rvar>i</rvar>
<where>
    <range>
        <var>i</var>
        <lo>0</lo>
        <hi>3</hi>
        <delta>1</delta>
    </range>
</where>
```

Results: Pruned Experiment Space

```
  6 {'packetSize': 1, 'offTime': 20, 'onTime': 10.0, 'dataRate': 1}
  9 {'packetSize': 1, 'offTime': 30, 'onTime': 10.0, 'dataRate': 1}
 16 {'packetSize': 2, 'offTime': 10, 'onTime': 10.0, 'dataRate': 2}
 19 {'packetSize': 2, 'offTime': 20, 'onTime': 10.0, 'dataRate': 2}
 22 {'packetSize': 2, 'offTime': 30, 'onTime': 10.0, 'dataRate': 2}
 32 {'packetSize': 3, 'offTime': 20, 'onTime': 10.0, 'dataRate': 3}
 35 {'packetSize': 3, 'offTime': 30, 'onTime': 10.0, 'dataRate': 3}
 42 {'packetSize': 1, 'offTime': 20, 'onTime': 16.0, 'dataRate': 1}
 45 {'packetSize': 1, 'offTime': 30, 'onTime': 16.0, 'dataRate': 1}
 55 {'packetSize': 2, 'offTime': 20, 'onTime': 16.0, 'dataRate': 2}
 58 {'packetSize': 2, 'offTime': 30, 'onTime': 16.0, 'dataRate': 2}
 68 {'packetSize': 3, 'offTime': 20, 'onTime': 16.0, 'dataRate': 3}
 71 {'packetSize': 3, 'offTime': 30, 'onTime': 16.0, 'dataRate': 3}
 81 {'packetSize': 1, 'offTime': 30, 'onTime': 27.0, 'dataRate': 1}
 94 {'packetSize': 2, 'offTime': 30, 'onTime': 27.0, 'dataRate': 2}
107 {'packetSize': 3, 'offTime': 30, 'onTime': 27.0, 'dataRate': 3}
```

What the Model Description Language Does

- Initially, model descriptions will have very little flexibility in order to serve more like templates.
- Eventually, provides an overarching *model* built as a composition of sub-components.
- Separate from simulation script.

Model Description XML

```
<component>
    <name>OnOffHelper</name>
    <id>onoff</id>
    <attrlist>
        <attr>onTime</attr>
        <attr>offTime</attr>
        <attr>dataRate</attr>
        <attr>packetSize</attr>
    </attrlist>
</component>
```

Results: A Single Model Instance

```
<OnOffHelper id="onoff">
    <attr name="onTime" value="10.0"/>
    <attr name="offTime" value="0"/>
    <attr name="dataRate" value="1"/>
    <attr name="packetSize" value="1"/>
</OnOffHelper>
```

From here we can transform into executable C++ or Python!

Questions, comments?