

Bug ID S-17: Duplicate Checking Code Documentation

Table of contents

1 Problem.....	2
2 Solution.....	2
3 Modified file list.....	2
4 Code Explanation.....	2
5 JUnit test.....	3

1. Problem

When writing an XML model file for simBio, somebody may write two or more elements with the same value for the 'name' attribute at the same element nesting level, causing mistaken duplication of Nodes when simBio is run.

2. Solution

When an XML file is being parsed, check if a new Node already exists before adding it to simBio's data structure. If it already exists, display an error.

3. Modified file list

- org.simBio.serialize.xml.Parser.java

4. Code Explanation

The new code is an exception which is thrown when a duplicate Node is detected. Because the exception is thrown as soon as the duplicate element is found, only the first duplicate in the XML file will be detected.

In XMLSerializer's read method, which creates a model instance from an XML file, a SAXParser object is created and takes the XML source and a Parser object as parameters. The program checks for duplicates in the startElement method of org.simBio.serialize.xml.Parser. The startElement method is called when the SAXParser reaches the starting tag of an element in the XML file. An Initializer object is created using the parent Node (component) and the attributes of the XML element.

```
Initializer initializer =
    Initializer.getInstance( (Composite)component, attributes );
```

The parent Node (composite) calls getNode, taking the name of the Initializer object as a parameter, and returns a Component object if it finds a Node with the same name.

```
Composite composite = (Composite) component;
//try to find the Node which has the same name.
component = (Component)composite.getNode(initializer.getName());
```

If the getNode call returns a null, then the XML element is not a duplicate, and an instance of the Component is created using the initializer.

```
if (component == null) {
    try {
        //construct new instance
        component = serializer.getInstanceOf(qName, initializer);
    } catch (InstantiationException e1) {
        throw new SAXException(e1);
    }
}
```

If the `getNode` call returns a `Component` object, then a `Node` with the same name already exists, which means that the XML file has a duplication error, and a `SAXException` is thrown.

```
else{  
    //If the Node is a duplicate, throw an exception  
    throw new SAXException("Duplicate Node Name Error :" +  
        component.getName());  
}
```

The error message displays the full path from the root to the `Node` name. The `SAXException` is caught in the `read` method of `XMLSerializer` and then replaced with an `InstantiationException`.

This is the error message which is produced when the file `model2.xml`, which duplicates the ATP synthesis element, is run:

```
java.langInstantiationException: Cause:  
Duplicate Node Name Error :simulation/model/cell/ATP synthesis
```

5. JUnit test

The test class

- `org.simBio.serialize.xml.XMLSerializerTest`

The test method

Method	File Used	Test
<code>testReadDuplicateNode</code>	<code>duplicateNode.xml</code>	This tests if an error is thrown when an XML file with duplicates is parsed.
<code>testRead</code>	<code>xml/matsuoka_et_al_2003/mode</code>	This tests if an XML file without duplicates can be parsed without throwing an error.

The class `org.simBio.serialize.xml.XMLSerializerTest` runs JUnit tests on XML files that have duplicate elements as well as ones which don't have duplicates. `XMLSerializerTest` has two methods: `testReadDuplicateNode` and `testRead`. Each of these methods uses an XML file, and creates an `XMLSerializer` object for the file and calls that object's `read` method to parse the XML file. The `testReadDuplicateNode` method expects the xml file that it tests will have duplicates, and that an `InstantiationException` will be thrown. If an `InstantiationException` is not thrown then the test fails. The `testRead` method expects that the file it tests will not have duplicates, and so it fails if an `InstantiationException` is thrown.

The `duplicateNode.xml` file is the same as `model.xml`, except that in `duplicateNode.xml`, `simulation/model/cell/ATP synthesis` is duplicated.

How to run the test.

The code has been tested using JUnit 3. To run the tests, right-click on XMLSerializerTest.java and select [Run As]->[JUnit Test].