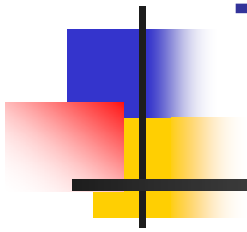
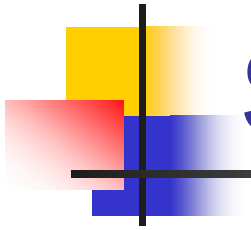


COMP4317: XML & Database Tutorial 2: SAX Parsing

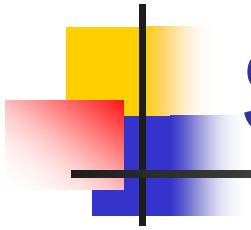
Week 3

Thang Bui @ CSE.UNSW



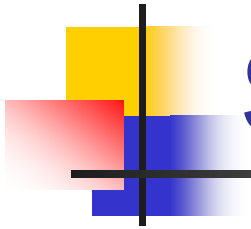
SAX – Simple API for XML

- is NOT a W3C standard.
- SAX parser sends events on-the-fly
  - startDocument event
  - endDocument event
  - startElement event
  - endElement event
  - ...



SAX example

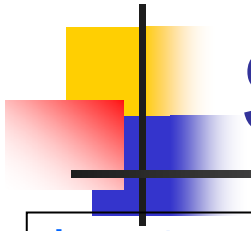
```
<book isbn="1-2345-6789-0">
  <title>
    TCP/IP Illustrated
  </title>
  <author>
    Stevens W.
  </author>
  <publisher>
    Addison-Wesley
  </publisher>
</book>
```



SAX example

	Events

<book isbn="1-2345-6789-0">
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <publisher>
  Addison-Wesley
  </publisher>
</book>
	1. startDocument
	2. startElement: “book”, attrs={...}
	3. startElement: “title”
	4. characters: ...
	5. endElement: “title”
	6. startElement: “author”
	7. characters: ...
	8. endElement: “author”
	9. startElement: “publisher”
	10. characters: ...
	11. endElement: “publisher”
	12. endElement: “book”
	13. endDocument



SAX Parser

Java code

```
import org.xml.sax.XMLReader;
import org.xml.sax.helpers.XMLReaderFactory;
...
 public static void main() {
    SAXTester tester = new SAXTester();
    XMLReader myReader;
    String parserName = "org.apache.xerces.parsers.SAXParser";
    try {
      myReader = XMLReaderFactory.createXMLReader(parserName);
    } catch (...) { return;  }
    myReader.setContentHandler(tester);
    myReader.setErrorHandler(tester);
    try {
      myReader.parse("input.xml");
    } catch (...) {  return; }
  }
```

SAX Parser

Java code

```
import org.xml.sax.XMLReader;
import org.xml.sax.helpers.XMLReaderFactory;
...
 public static void main() {
    SAXTester tester = new SAXTester();
    XMLReader myReader;
    String parserName = "org.apache.xerces.parsers.SAXParser";
    try {
      myReader = XMLReaderFactory.createXMLReader(parserName);
    } catch (...) { return;  }
    myReader.setContentHandler(tester);
    myReader.setErrorHandler(tester);
    try {
      myReader.parse("input.xml");
    } catch (...) {  return; }
  }
```

Create a parser

SAX Parser

Java code

```
import org.xml.sax.XMLReader;
import org.xml.sax.helpers.XMLReaderFactory;
...
 public static void main() {
    SAXTester tester = new SAXTester();
    XMLReader myReader;
    String parserName = "org.apache.xerces.parsers.SAXParser";
    try {
      myReader = XMLReaderFactory.createXMLReader(parserName);
    } catch (...) { return;  }
    myReader.setContentHandler(tester);
    myReader.setErrorHandler(tester);
    try {
      myReader.parse("input.xml");
    } catch (...) {  return; }
  }
```

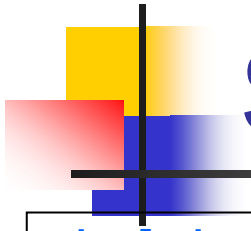
Register a content handler: handle all events

SAX Parser

Java code

```
import org.xml.sax.XMLReader;
import org.xml.sax.helpers.XMLReaderFactory;
...
 public static void main() {
    SAXTester tester = new SAXTester();
    XMLReader myReader;
    String parserName = "org.apache.xerces.parsers.SAXParser";
    try {
      myReader = XMLReaderFactory.createXMLReader(parserName);
    } catch (...) { return;  }
    myReader.setContentHandler(tester);
    myReader.setErrorHandler(tester);
    try {
      myReader.parse("input.xml");
    } catch (...) {  return; }
  }
```

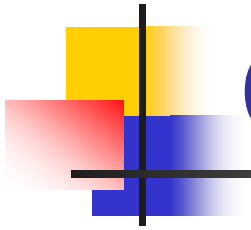
Parse the file and send events to the registered content handler



SAX Parser

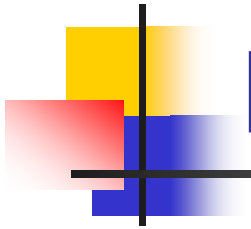
C++ code

```
#include <xercesc/util/PlatformUtils.hpp>
#include <xercesc/sax2/XMLReaderFactory.hpp>
#include <xercesc/sax2/SAX2XMLReader.hpp>
XERCES_CPP_NAMESPACE_USE
...
int main() {
  XMLPlatformUtils::Initialize();
  SAX2XMLReader * parser = XMLReaderFactory::createXMLReader();
  SAX2Tester * tester = new SAX2Tester();
  parser->setContentHandler(tester);
  parser->setErrorHandler(tester);
  try {
    parser->parse("input.xml");
  } catch (...) { return 1; }
  XMLPlatformUtils::Terminate();
}
```



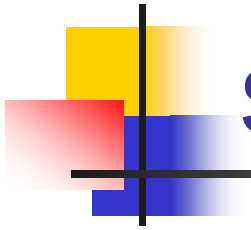
ContentHandler class

- to handle all events from the parser
- (ErrorHandler: to handle all warning, error, or fatal error)
- class DefaultHandler
  - Interfaces => do nothing
  - Need to override its methods



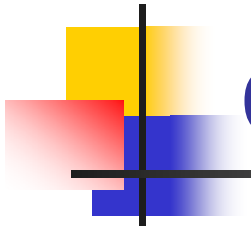
Document events

- startDocument()
- endDocument()



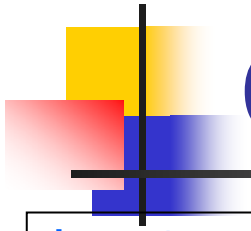
startElement events

- startElement (String uri, String localName, String qName, Attributes attrs)
  - qName: name of the element
  - attrs: List of all attributes of that element
    - getLength()
    - getQName(index): name of an attribute
    - getValue(index): value of an attribute



endElement events

- endElement(String uri, String localName, String qName)



ContentHandler example

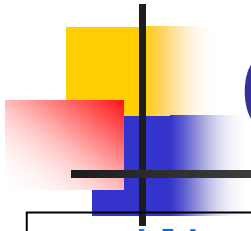
Java code

```
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;
import org.xml.sax.Attributes;
import org.xml.sax.helpers.DefaultHandler;

class SAXTester extends DefaultHandler {
  public SAXTester() {}

  public void startDocument() throws SAXException {
    System.out.println("<?xml version=\"1.0\"?>");
  }

  public void endDocument() throws SAXException {
  }
```

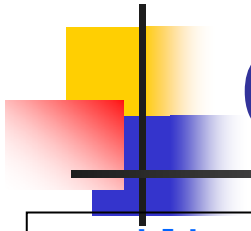


ContentHandler example

Java code

```
public void startElement
            (String uri, String localName, String qName, Attributes attrs)
                throws SAXException {
  System.out.print("<" + qName);
  for (int i = 0; i < attrs.getLength(); i++)
    System.out.print(" " + attrs.getQName(i)
                        + "=\"" + attrs.getValue(i) + "\"");
  System.out.println(">");
}

public void endElement(String uri, String localName, String qName)
              throws SAXException {
  System.out.println("</" + qName + ">");
}
```



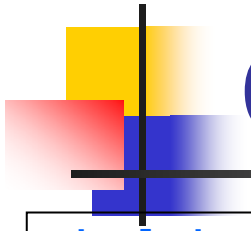
ContentHandler example

Java code

```
  public void warning(SAXParseException ex) throws SAXException {
    System.err.println("Warning: " + ex.getMessage());
  }

  public void error(SAXParseException ex) throws SAXException {
    System.err.println("Error: " + ex.getMessage());
  }

  public void fatalError(SAXParseException ex) throws SAXException {
    System.err.println("Fatal Error: " + ex.getMessage());
    throw ex;
  }
} //class SAXTester
```

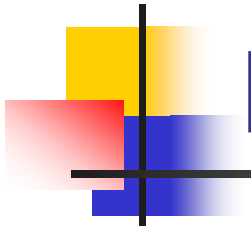


ContentHandler example

C++ code

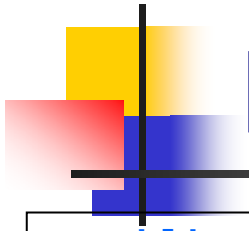
```
#include <xercesc/sax2/Attributes.hpp>
#include <xercesc/sax2/DefaultHandler.hpp>
XERCES_CPP_NAMESPACE_USE

class SAX2Tester : public DefaultHandler {
public:
  SAX2Tester();
  ~SAX2Tester();
  void startElement(const XMLCh* const uri, const XMLCh* const localname, const
XMLCh* const qname, const Attributes& attrs);
 void endElement(const XMLCh* const uri, const XMLCh* const localname, const
XMLCh* const qname);
  void warning(const SAXParseException& exc);
  void error(const SAXParseException& exc);
  void fatalError(const SAXParseException& exc);
}
```



Building a tree

- Using a stack
- startX event:
  - peek the stack for the parent
  - push to the stack
- endX event:
  - pop from the stack
- Refer to Marc.H.Scholl's slides pages 90-94

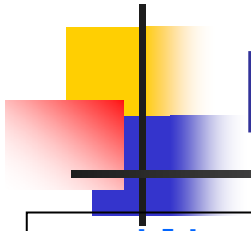


Building a tree

Java code

```
public void startDocument() throws SAXException {
  stack = new Stack();
}
public void startElement
            (String uri, String localName, String qName, Attributes attrs)
                throws SAXException {
  String parent = null;
  if (!stack.empty()) parent = stack.peek();
  stack.push(qName);
  System.out.println(qName + " is a child of " + parent);
}

public void endElement(String uri, String localName, String qName)
              throws SAXException {
  stack.pop();
}
```



Building a tree

C++ code

```
  public void startDocument() {
    stack = new stack<XMLCh *>();
  }
  void startElement(const XMLCh* const uri, const XMLCh* const localname, const
XMLCh* const qname, const Attributes& attrs) {
    XMLCh * parent = NULL;
    if (!stack.empty()) parent = stack.top();
    stack.push(qName);
    cout << myXMLStr(qName) << " is a child of " << myXMLStr(parent) << endl;
  }


 void endElement(const XMLCh* const uri, const XMLCh* const localname, const
XMLCh* const qname) {
    stack.pop();
  }
```

Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument	stack = {}
		tree = null

Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
		stack = {book}
		tree = (book)

Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
		stack = {title, book}
		tree = (book)
		  |
		  (title)

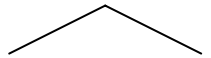
Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
		stack = {book}
		tree = (book)
		  |
		  (title)

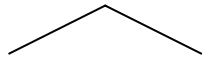
Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
	5. startElement: “author”
		stack = {author,book}
		tree = (book)
		 
		(title) (author)

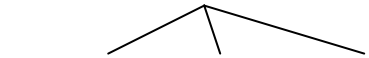
Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
	5. startElement: “author”
	
	6. endElement: “author”
		stack = {book}
		tree = (book)
		 
		(title) (author)

Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
	5. startElement: “author”
	
	6. endElement: “author”
	7. startElement: “pub”
		stack = {pub, book}
		tree = (book)
		 
		(title)(author)(pub)

Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
	5. startElement: “author”
	
	6. endElement: “author”
	7. startElement: “pub”
	
	8. endElement: “pub”
		stack = {book}
		tree = (book)
		 
		(title)(author)(pub)

Building a tree - Example

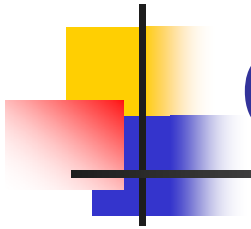
	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
	5. startElement: “author”
	
	6. endElement: “author”
	7. startElement: “pub”
	
	8. endElement: “pub”
	9. endElement: “book”
		stack = {}
		tree = (book)
		 
		(title)(author)(pub)

Building a tree - Example

	Events	Stack & Tree

<book>
  <title>
  TCP/IP Illustrated
  </title>
  <author>
  Stevens W.
  </author>
  <pub>
  Addison-Wesley
  </pub>
</book>
	1. startDocument
	2. startElement: “book”
	3. startElement: “title”
	
	4. endElement: “title”
	5. startElement: “author”
	
	6. endElement: “author”
	7. startElement: “pub”
	
	8. endElement: “pub”
	9. endElement: “book”
	10. endDocument
		stack = {}
		tree = (book)
		 
		(title)(author)(pub)



Q&A

- class Tree {
- String label;
- Tree left,right;

```
add_child(Tree a) {
    if (left==null) left = a;
    else { Tree child = left;
            while (child.right != null)
              child = child.right;
            child.right = a; }
}
```

DAG example