XML Processing

Stream Processing (SAX)

Lecture "XML in Communication Systems"
Chapter 7

Dr.-Ing. Jesper Zedlitz
Research Group for Communication Systems
Dept. of Computer Science
Christian-Albrechts-University in Kiel

Recommended Reading

- http://www.saxproject.org/

Parsers

- What is a parser?
  - A program that analyses the grammatical structure of an input, with respect to a given formal grammar
  - The parser determines how a sentence can be constructed from the grammar of the language by describing the atomic elements of the input and the relationship among them.

XML Parsing Standards

- Two parsing methods for accessing XML
  - DOM (Document Object Model)
    - convert XML into a tree of objects
    - "random access" method
    - can update XML document (insert/delete/update nodes)
  - SAX (Simple API for XML)
    - event-driven parsing
    - "serial access" method
    - read-only API

SAX vs. DOM Comparison

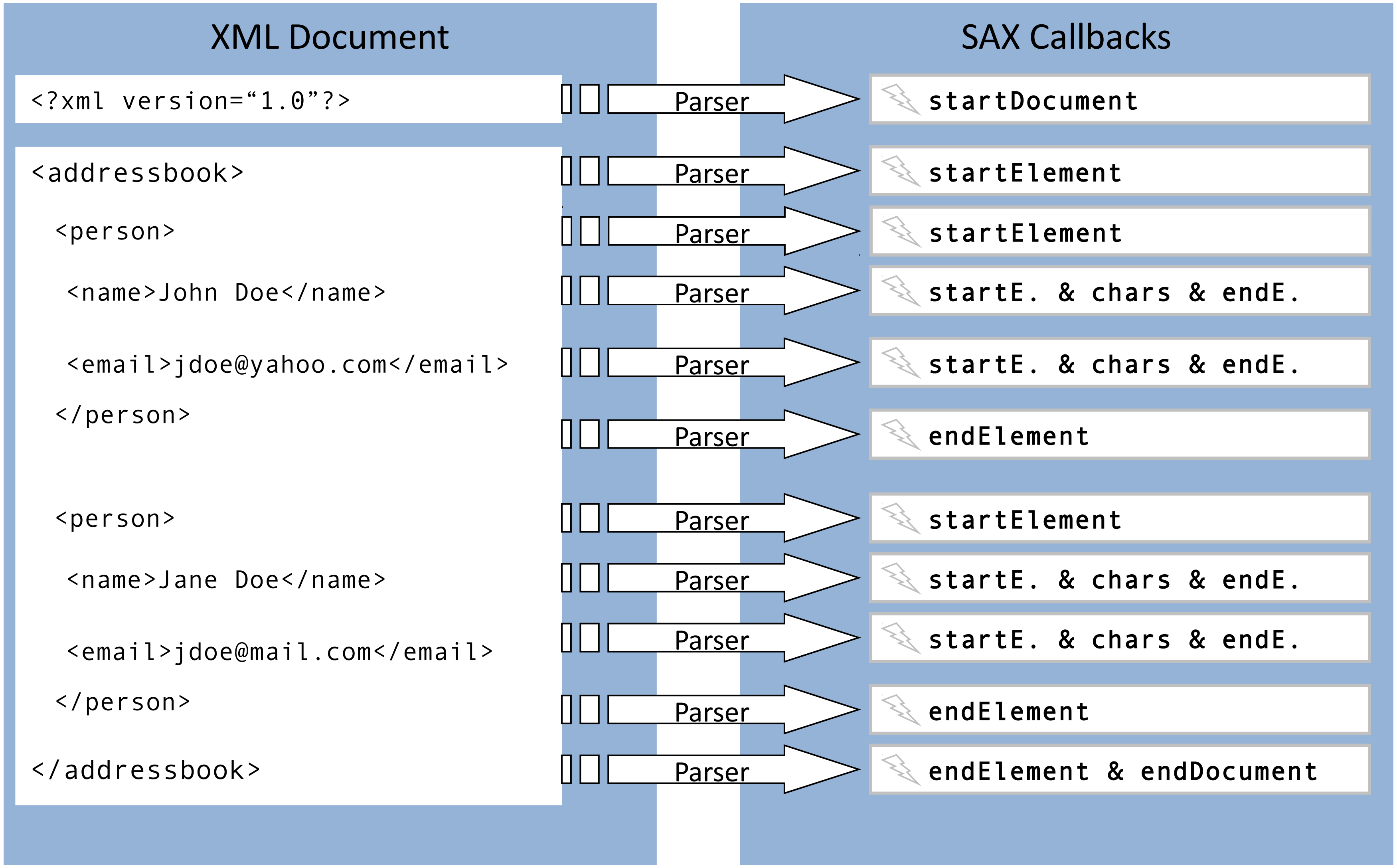
- SAX
  - Java-specific
  - interprets XML as a stream of events
  - You supply event-handling callbacks.
  - SAX parser invokes your event-handlers as it parses.
  - doesn't build data model in memory
  - serial access: very fast, lightweight
  - good choice when
    - no data model is needed, or
    - natural structure for data model is list, matrix, etc.
- DOM
  - W3C standard for representing structured documents
  - platform and language neutral

SAX Parser

- SAX = Simple API for XML
  - XML is read sequentially
  - When a parsing event happens, the parser invokes the corresponding method of the corresponding handler
  - The handlers are programmer's implementation of standard Java interfaces and classes
  - Similar to an I/O-Stream, goes in one direction

How Does SAX Work?

Informatik · CAU Kiel



```
<?xml version="1.0"?>
<addressbook>
  <person>
   <name>John Doe</name>
   <email>jdoe@yahoo.com</email>
  </person>

  <person>
   <name>Jane Doe</name>
   <email>jdoe@mail.com</email>
  </person>
</addressbook>
```

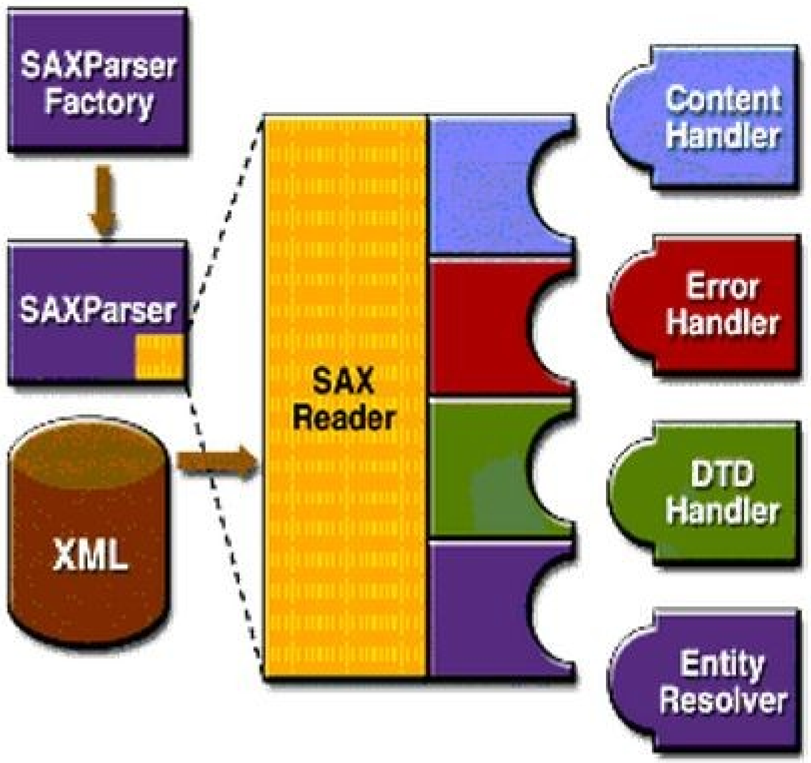
XML Document	SAX Callbacks
`<?xml version="1.0"?>`	startDocument
`<addressbook>`	startElement
`<person>`	startElement
`<name>John Doe</name>`	startE. & chars & endE.
`<email>jdoe@yahoo.com</email>`	startE. & chars & endE.
`</person>`	endElement
`<person>`	startElement
`<name>Jane Doe</name>`	startE. & chars & endE.
`<email>jdoe@mail.com</email>`	startE. & chars & endE.
`</person>`	endElement
`</addressbook>`	endElement & endDocument

How Does SAX Work?

```
<?xml version="1.0" encoding="UTF-8"?>

<dots>
  this is before the first dot
  and it continues on multiple lines
  <dot x="9" y="81" />
  <dot x="11" y="121" />
  <flip>
    flip is on
    <dot x="196" y="14" />
    <dot x="169" y="13" />
  </flip>
  flip is off
  <dot x="12" y="144" />
  <extra>stuff</extra>
  <!-- a final comment -->
</dots>
```

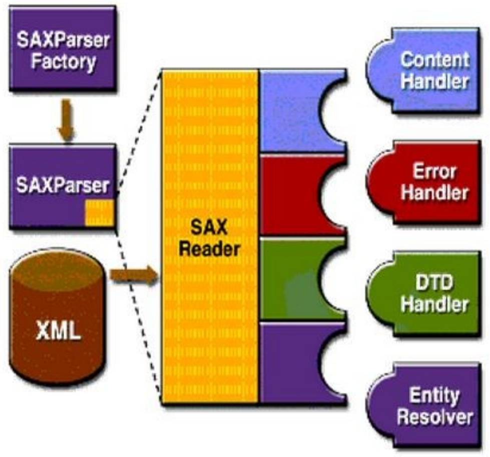
```
startDocument
startElement: dots (0 attributes)
characters:   this is before the first dot
  and it continues on multiple lines
startElement: dot (2 attributes)
endElement:   dot
startElement: dot (2 attributes)
endElement:   dot
startElement: flip (0 attributes)
characters:   flip is on
startElement: dot (2 attributes)
endElement:   dot
startElement: dot (2 attributes)
endElement:   dot
endElement:   flip
characters:   flip is off
startElement: dot (2 attributes)
endElement:   dot
startElement: extra (0 attributes)
characters:   stuff
endElement:   extra
endElement:   dots
endDocument

Finished parsing input.  Got the following dots:
[(9, 81), (11, 121), (14, 196), (13, 169), (12,
144)]
```

SAX Structure



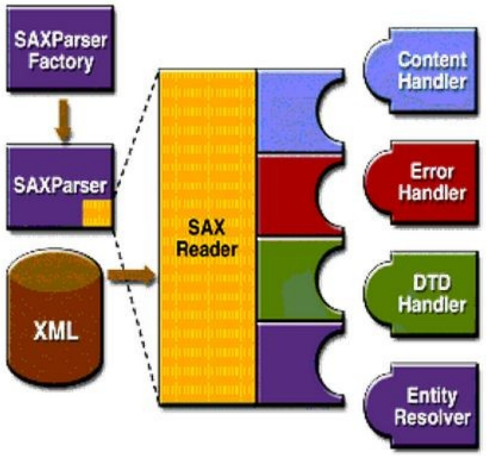
SAX Structure

- SAX structure
  - SAXParserFactory
    A SAXParserFactory object creates an instance of the parser determined by the system property, javax.xml.parsers.SAXParserFactory.
  - SAXParser
    The SAXParser interface defines several kinds of parse() methods. In general, you pass an XML data source and a DefaultHandler object to the parser, which processes the XML and invokes the appropriate methods in the handler object.



SAX Structure

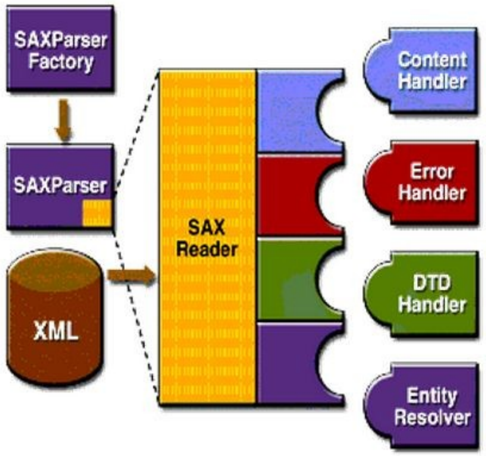
- SAX structure
  - SAXReader
    The SAXParser wraps a SAXReader. Typically, you don't care about that, but every once in a while you need to get hold of it using SAXParser's getXMLReader() so that you can configure it. It is the SAXReader that carries on the conversation with the SAX event handlers you define.



  - DefaultHandler
    Not shown in the diagram, a DefaultHandler implements the ContentHandler, ErrorHandler, DTDHandler, and EntityResolver interfaces (with null methods), so you can override only the ones you're interested in.

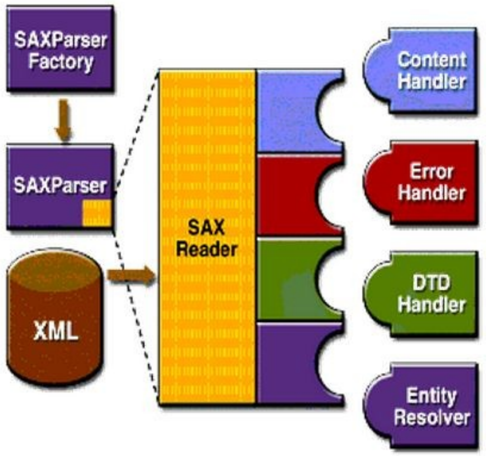
SAX Structure

- SAX structure
  - ContentHandler
    Methods such as startDocument, endDocument, startElement, and endElement are invoked when an XML tag is recognized. This interface also defines the methods characters and processingInstruction, which are invoked when the parser encounters the text in an XML element or an inline processing instruction, respectively.



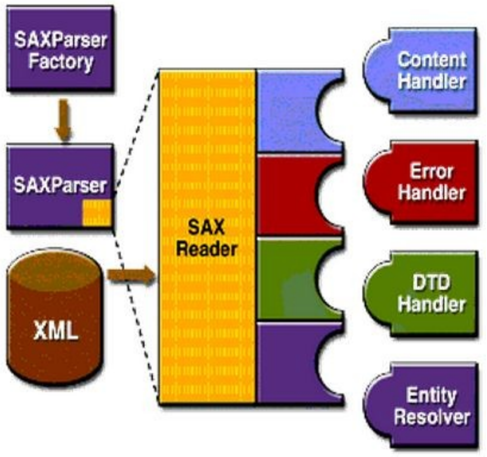
SAX Structure

- SAX structure
  - ErrorHandler
    Methods error, fatalError, and warning are invoked in response to various parsing errors. The default error handler throws an exception for fatal errors and ignores other errors (including validation errors). That's one reason you need to know something about the SAX parser, even if you are using the DOM. Sometimes, the application may be able to recover from a validation error. Other times, it may need to generate an exception. To ensure the correct handling, you'll need to supply your own error handler to the parser.



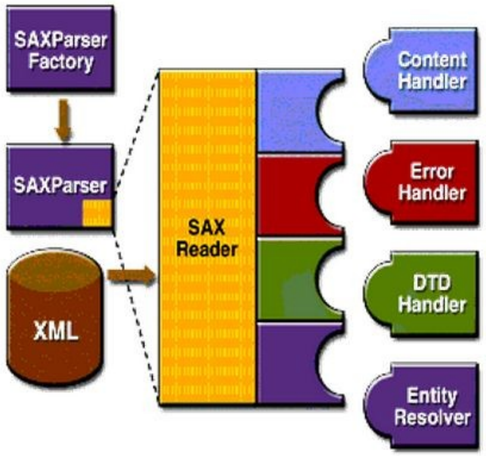
SAX Structure

- SAX structure
  - DTDHandler
    Defines methods you will generally never be called upon to use. Used when processing a DTD to recognize and act on declarations for an unparsed entity.



SAX Structure

- SAX structure
  - EntityResolver
    The resolveEntity method is invoked when the parser must identify data identified by a URI. In most cases, a URI is simply a URL, which specifies the location of a document, but in some cases the document may be identified by a URN—a public identifier, or name, that is unique in the web space. The public identifier may be specified in addition to the URL. The EntityResolver can then use the public identifier instead of the URL to find the document—for example, to access a local copy of the document if one exists.



SAX Programming

- In some cases, programming with SAX is difficult
  - How can we find, using a SAX parser, elements e1 with ancestor e2?
  - How can we find, using a SAX parser, elements e1 that have a descendant element e2?
  - How can we find the element e1 referenced by the IDREF attribute of e2?
  - SAX parsers do not provide access to elements other than the one currently visited in the serial (DFS) traversal of the document.
    - They do not read backwards
    - They do not enable access to elements by ID or name