

X12 Talk handout

(to accompany PAUG presentation "OOP: I Did It Again")

Hierarchical data: Data which is logically structured as an inverted tree (like the filesystem). There's a root element (e.g., C:\, the "root directory"), and every element can have zero or more child elements of the same type (e.g., "C:\Program Files\Microsoft Office", a subdirectory). See reverse for an example structure used throughout the presentation.

XML: A text file with hierarchical structure. Promoted by the WWW Consortium (W3C).

Key XML concepts:

- Tag: primary unit of structure in XML. Example:
`<presenter>Sam Livingston-Gray</presenter>`
- Start tag: the first part of the overall XML tag. Highlighted here:
`<presenter>Sam Livingston-Gray</presenter>`
- End tag: the last part of the overall XML tag. Highlighted here:
`<presenter>Sam Livingston-Gray</presenter>`
- Empty tags can be designated this way: `<presenter />`
- Attributes: name/value pairs that provide additional data about a tag. Example:
`<presenter section="main">Sam Livingston-Gray</presenter>`
- SAX: Simple API for XML, a parser which makes a single pass through an XML textfile, firing events as it encounters language elements. Fast, but requires more code.
- DOM: Document Object Model, a parser which reads the entire document into code and makes it available as a hierarchical structure of objects.

X12: Another text file with hierarchical structure. This one was developed by an American National Standards Institute (ANSI) committee for electronic document interchange (EDI) of insurance data. If that doesn't scare you, it should.

Key X12 concepts:

- Segment: primary unit of structure in XML. Corresponds to a tag in XML. Segments are defined as any text from the current position in the file to the next segment delimiter. (Delimiters are not nailed down in X12, but "@" is not uncommon.)
- Base Element: What I call the first element (index 0).
- Element: Values that provide additional data about a segment. Corresponds to an attribute in XML. Elements are delimited by the element delimiter, commonly "*".
- End tags: No such thing. This makes parsing X12 rather complicated...

X12 Parser object

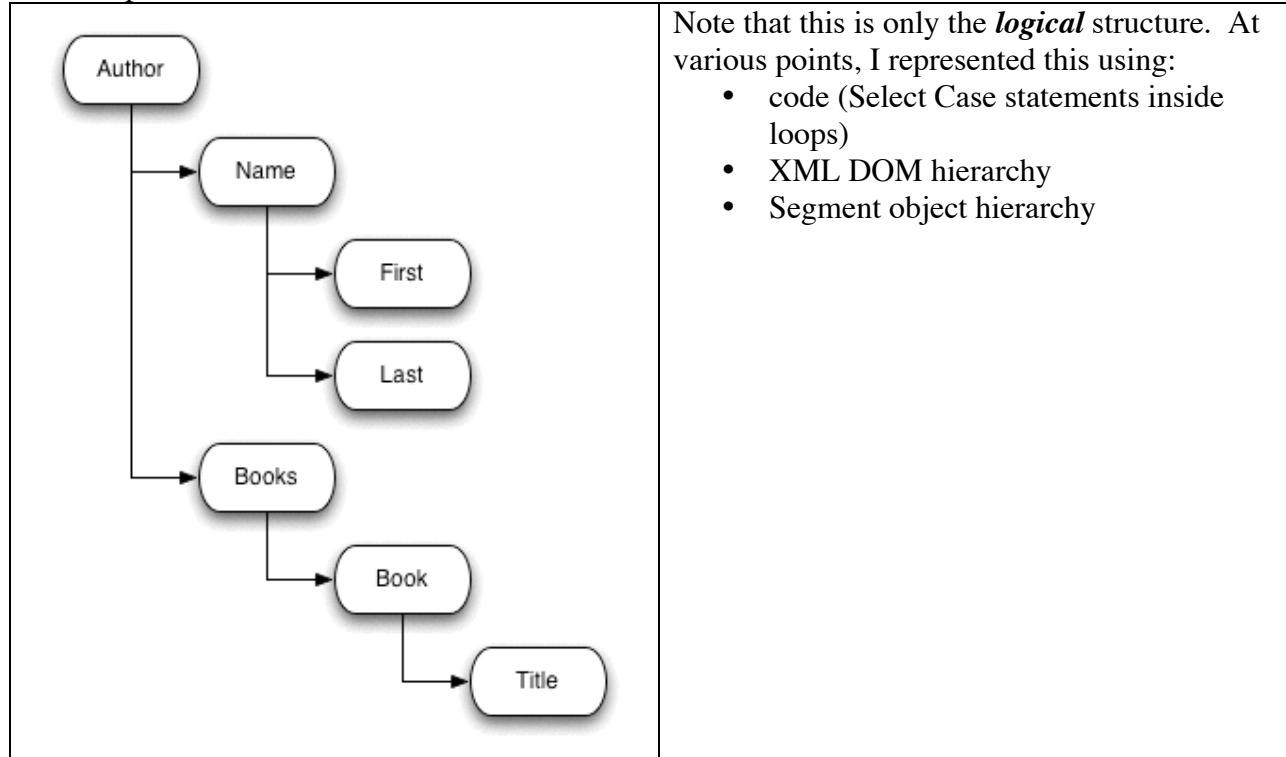
Methods:

- OpenInputFile – tells the parser where to find its input
- ReadNextSegment – tells the parser to read through the input file looking for the next segment delimiter, then chop up the segment to be exposed by the Element property.

Property:

- Element(index) – Returns the section of the current segment before the *index*th element delimiter, where *index* is zero-based. Example: if segment is "AUTHOR*42*Sam*Livingston-Gray@", Segment(0) = "AUTHOR" and Segment(2) = "Sam".

The Map



The Segment class

Key properties:

- BaseElement: Tells us what we expect in the zeroth element. (This is our identifier.)
- ChildSegments: Collection of other Segment objects “below” this one in the hierarchy.
- ParentSegment: Refers to the Segment that has this one in its ChildSegments collection.

Turning the XML DOM hierarchy into a Segment hierarchy:

This was fun. Basically, I wrote a single recursive function. The function knows how to turn a single node in the DOM hierarchy into a single Segment object, and it also knows how to call itself on all children of that DOM node. So if you pass it the root of the DOM hierarchy, you get a complete representation of the same map returned as a root Segment object. (Confused? Email me.)

Thanks!

If you're looking for a consultant, please consider me.

Sam Livingston-Gray
slg@timestream.net
<http://timestream.net/resume/>