XIncluding plain-text fragments for symmetry and profit

Piotr Bański, University of Warsaw, bansp@o2.pl

Summary

XML for the annotation of images, audio or video is by necessity of the stand-off kind: the annotations are virtually anchored to the binary stream (typically by x,y coordinates or time offsets). Work on language resource annotation has extended this approach to plain text, where annotations are anchored by character offsets. This is referred to as **extreme stand-off markup**. No generic XML tools or standards support it, but... they could, fairly easily.

Rationale

ISO Language Annotation Framework (e.g. Ide & Romary, 2007) recommends that data be kept as read-only plain text, with one or more layers of XML annotation identifying individual spans of characters and adding metadata to them (identifying e.g. segments of various levels of granularity, parts of speech, syntactic or discourse functions, etc.)

However, the only way to handle this (as evidenced by the American National Corpus), is to use in-house tools working on correspondence semantics for hyperlinks (see Bański, 2010, for terminology and more remarks).

But character streams can do better! **Because text is what XML is built out of and what it contains**, instead of correspondence semantics, we can move a step higher and upgrade the representation to inclusion semantics, performed by generic XML tools. The XInclude Rec has recognised this only partially by allowing for the inclusion of entire text resources. Why not fragments as well? Maybe no one has envisioned any use for that up till now.

This poster muses on bringing extreme stand-off annotation techniques closer to those language-resource creators who recognize the virtues of having plain text as the base data but do not want to commit to tools available so far only from the ANC and creating ANC formats. Non-linguists would hopefully also benefit (panini for your thoughts!).

Suggestions

Half-way solution: use RFC 5147 syntax for @pointer.

Use-the-power,-Luke solution: treat the text resource as a giant text node (after escaping <'s, &'s, etc.); use the XPointer Framework, with syntax analogous to the TEI string-range() scheme (cf. a), or to something that ISO LAF would like to use (cf. b), or to the string-range() function of the W3C xpointer() scheme, cf. (c):

a. `(string)text-range(offset, length)`

b. `(string)text-span(startoffset, endoffset)`

c. `(string*)text-range([string-to-match],off,lgth)`

Let RFC 5147 become the definition of an analog of shorthand pointers that would complement the schemes above.

Either way: lift the XInclude Ban!

Issues that would have to be addressed

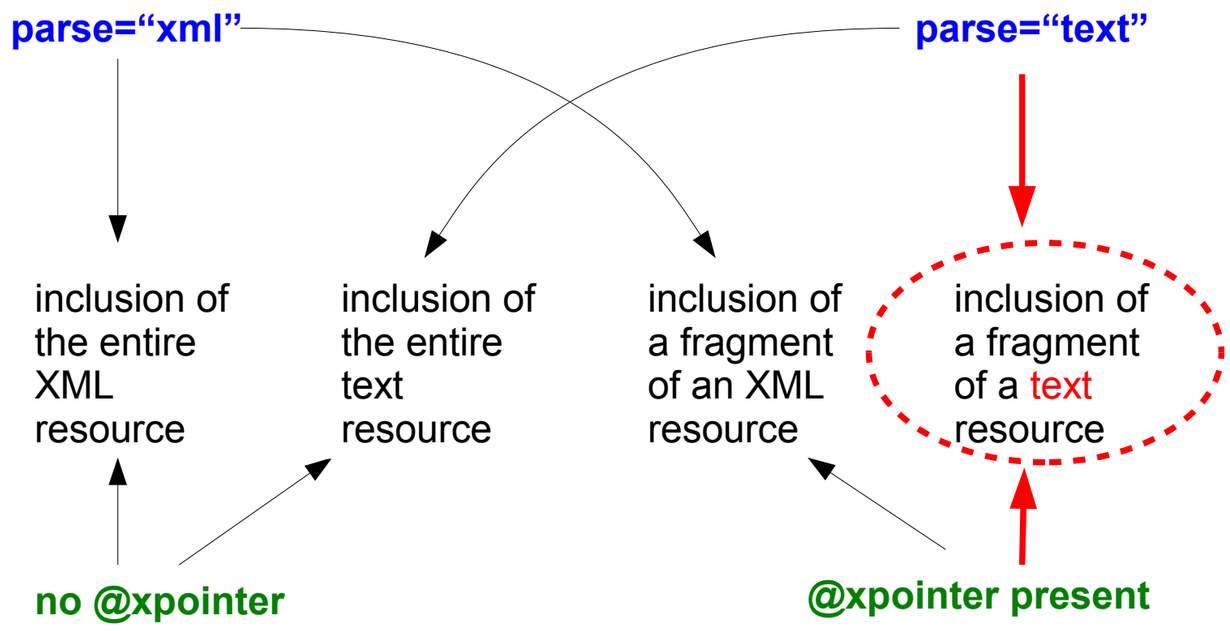
- treatment of the BOM: ignore if present;
- charset issues: XInclude has @encoding; (more generally, would an encoding()/charset() scheme be useful/necessary with the extension outlined here?)
- line breaks: RFC 2046 demands CRLF, but RFC 5147 states that all line breaking sequences must be treated as a single character.
- integrity checks: RFC 5147 proposes schemes for that purpose;
- content negotiation: Simon St. Laurent proposed an I-D for an XPointer scheme to handle that, for example:

```
#content-type(application/xhtml+xml)element(/1/7)
content-type(image/svg+xml)element(/1/4) content-
type(application/mathml+xml)element(/1/3)
```

well, why not, then, have e.g.:

```
content-type(text/plain)text-range(112,7)
```

<xi:include href="resource.xml" parse="text|xml"
xpointer="fragment_identifier" />



What makes it impossible right now?

1. XInclude Ban: XInclude Rec, 3.1: "The xpointer attribute must not be present when parse="text"."

2. XPointer Framework Rec, introduction: "The framework is intended to be used as a basis for fragment identifiers for any resource whose Internet media type is one of text/xml, application/xml, ([...xml..xml...]). Other XML-based media types are also encouraged to use this framework in defining their own fragment identifier languages."

What makes it worth trying?

- RFC 3986, sect. 3.5:
  "The semantics of a *fragid* are defined by the set of representations that might result from a retrieval action on the primary resource. The fragment's format and resolution is therefore dependent on the media type of a potentially retrieved representation (...) the *fragid* is not used in the scheme-specific processing of a URI; instead, the fragid is separated from the rest of the URI prior to a dereference, and thus the identifying information within the fragment itself is dereferenced solely by the user agent, regardless of the URI scheme."

→ So let XInclude delegate the task of erroring out to the agent.

- W3C WD on Media Fragments URI suggests the syntax for constructing *fragids* for audio, video, and images;
- RFC 5147, "URI Fragment Identifiers for the text/plain Media Type" – well, it looks like there is interest in fragmenting plain text, and part of the work is done. They define things such as `#char=100,` `#line=10,20;length=9876,UTF-8` – nice and simple.

References

- ANC (American National Corpus): http://www.americannationalcorpus.org/
- ISO LAF: http://www.tc37sc4.org/
- TEI Guidelines, ch. 16: http://www.tei-c.org/release/doc/tei-p5-doc/en/html/SA.html
- RFC 2046, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types" http://www.ietf.org/rfc/rfc2046.txt
- RFC 3986, "Uniform Resource Identifier (URI): Generic Syntax" http://www.ietf.org/rfc/rfc3986.txt
- RFC 5147, "URI Fragment Identifiers for the text/plain Media Type" http://tools.ietf.org/rfc/rfc5147.txt
- Media Fragments URI 1.0, W3C Working Draft 24 June 2010: http://www.w3.org/TR/media-frags/
- XInclude: http://www.w3.org/TR/xinclude/
- XPointer Framework: http://www.w3.org/TR/xptr-framework/

Bański, Piotr (2010). "Why TEI stand-off annotation doesn't quite work: and why you might want to use it nevertheless." Presented at Balisage-2010: http://www.balisage.net/Proceedings/vol5/html/Banski01/BalisageVol5-Banski01.html

Ide, Nancy & Laurent Romary (2007). Towards International Standards for Language Resources. In Dybkjaer, L., Hemsen, H., Minker, W. (Eds.), Evaluation of Text and Speech Systems, Springer, pages 263–284.

Laurent, Simon St. (2002). The XPointer content-type() Scheme (IETF Internet Draft, expired): http://simonstl.com/ietf/draft-stlaurent-content-type-frag-00.txt