Attaching Semantics

A T2TRG proof of concept

Attaching Semantics

"Semantic Style" (looking for a better word)

Attach information via **selectors** into the instance

— Similar to adding style semantics to HTML via CSS

One "style" can apply to:

— single instance
— "class" of instances (making it "metadata")

Usage

— Supply "style" with a self-description

— More general use for "metadata extraction"

— Write generic styles covering a whole media type specification

— Specify a media type using a semantic style

Approaches

Transformation language (DSSSL/XSLT)

— What is the target language (generic data model)?

Augmentation language (CSS)

— In-model augmentation:
Stay in generic data model of input

— Extra-model augmentation:
Generate into expanded generic data model

Rules

selector → effect

```
.warning {color: red}
```

Selector selects zero or more structural elements into a "node set"

Effect is then applied to each selection

Selector Language

Generically (or specifically!) identify internal structure of the resource

— Selector for structured data (JSON, CBOR)
— Selectors for text strings, binary data

— Navigation, predicates
— WILL evolve into Turing-equivalence

Selectors for structured data

Selects zero or more structural elements into a "node set"

— HTML Elements in CSS
— XML Elements, Attributes or other parts of the ESIS in XSLT and XPath
— JSON:
RFC 6901 (JSON Pointer): instance-oriented
JSON Path: A bit better for class-oriented, but not much

Each node could have additional names (entry points) for substructures selected with it

Selectors for text strings

Regexps are widely used

"Capturing" as a way to expose substructure to the effect

What if languages are not regular
(e.g., parse expressions?)

Selectors for binary data

Needs some creative copying

Identify good sources of experience, e.g.:

— Erlang
— Struct Packing/Unpacking in Perl, Python, Ruby
— bindata and similar modules
— Haskell

Similar considerations as for text strings, just not as much experience

Selection components

Navigational:
E.g., XSLT/XPath axes (ancestor ancestor-or-self attribute child descendant descendant-or-self following following-sibling namespace parent preceding preceding-sibling self @ // .. .)

Type-based, Attribute-based:
E.g., XML GI, XML classes, ids, other attributes

Computational, inference-based

Computational, inference-based

— :odd, :even, … ➔ structural computation

— can the effect of other rules be used?

— Full "T-boxes" as in OWL

➔ Turing-equivalent

Consulting external resources (e.g., by dereferencing computed URIs)?

Effect

Simple augmentation: Add attribute to each selected node
(kept separate from input data model in CSS (properties))

Generate additional structure (even in augmentation), tree grafting:

- CSS: `::before`, `::after`…
- Creation of augmented structure can involve:
  - computation (e.g., counters)
  - picking from selected node (selectors again?)
  - picking from external resources
    (e.g., by dereferencing computed URIs)?