



Full validation of Atom feeds containing extensions using NVDL

MURATA Makoto (FAMILY Given)
International University of Japan



Atom Extensions

- Atom Threading Extensions (RFC 4685)
- Feed Paging and Archiving (RFC 5005)
- Atom License Extension (RFC 4946)
- Gdata, Google Calendar
- Open Search
- Atom Bidi Extension
- Simple Sharing Extensions
- Simple List Extension
- GeoRSS
- Dublin Core
- Yahoo Media
- ...

Atom feed containing extensions

- Foreign elements and attributes occur in Atom feeds.
 - Note: Other specifications such as ODF 1.1 and ISO/IEC 29500 also allow foreign elements and attributes.
- Example: XML documents provided by Google Calendar

Co-existence of multiple extensions

Present:

- OpenSearch + Gdata + Google Calendar
- A few of the available extensions

Future

- Many more at the same time?
 - Threading Extensions, Feed Paging and Archiving, License Extension, and Bidi?

Validation of Atom extensions(1)

- The schema (namely `atom.rnc`) in the Atom Syndication Format RFC simply skips extension elements and attributes
- Some extension RFCs provide schema fragments but they are *NOT* invoked by `atom.rnc`.

Validation of Atom extensions(2)

- Proprietary extensions (with the exception of Google Data API) do not provide any validation.
- Note: Still, this is better than ODF 1.1 and ISO/IEC 29500, where the standard schemas do not even allow foreign extensions and attributes!

Validation in Google Data (Version 2)

- ☞ Google modified many definitions in `atom.rnc` so that their extensions are validated.
- ☞ Advantages
 - Tight restrictions
 - Full validation of atom feeds as well as Google extensions
- ☞ Disadvantages
 - How can we add more extensions?
 - What happens when `atom.rnc` is extended?

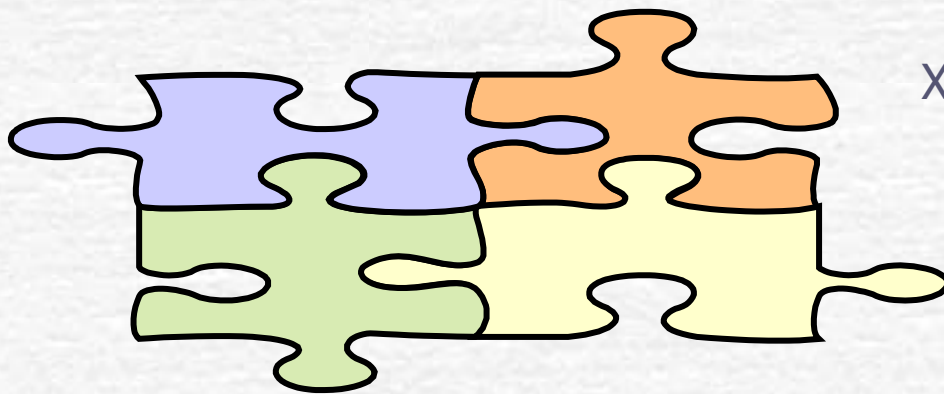
ISO/IEC 19757-4: Namespace-based Validation Dispatching Language -- NVDDL

XForms

XHTML

MathML

SVG



NVDL Key Idea #1

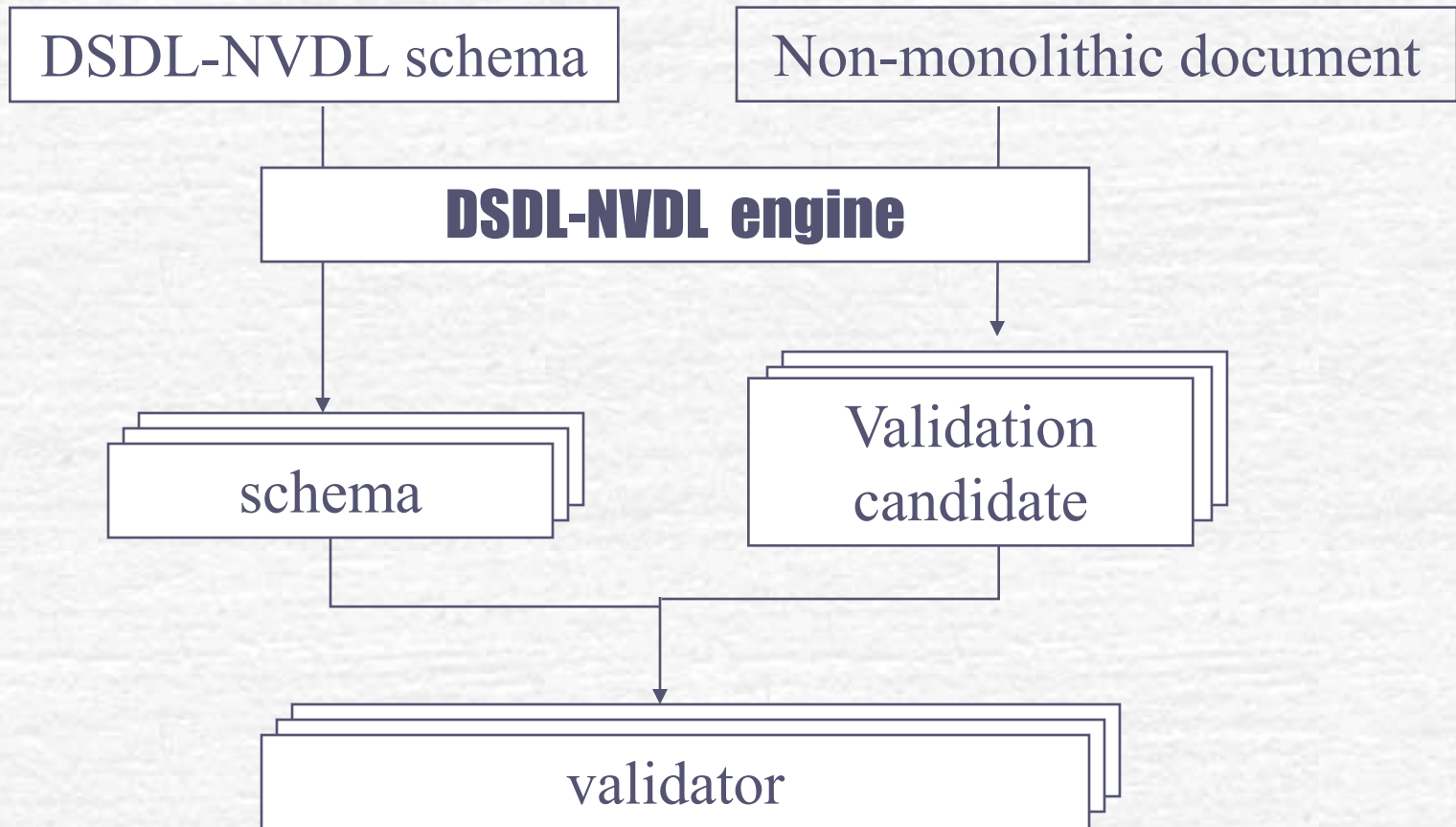
- Schema authoring by combining subschemas
 - Each subschema is concerned with one (or a few) namespaces.
 - Different subschemas may be written in different schema languages.
- Syntax
 - (namespace, schema) pairs

NVDL Key Idea #2

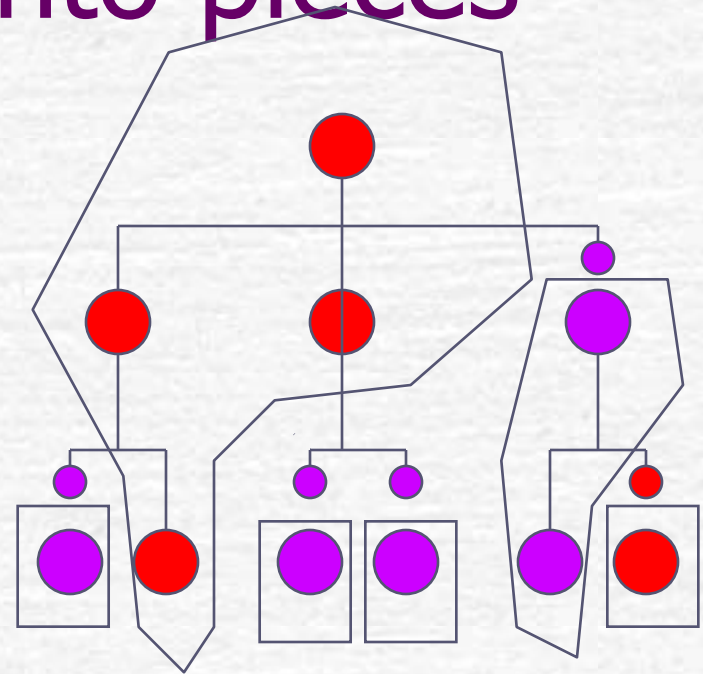
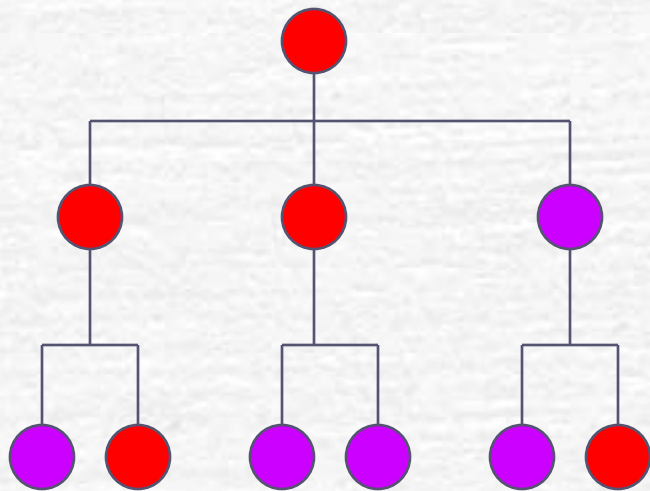
☞ Divide-and-validate

- Divide a non-monolithic document into validation candidates.
- Different validation candidates are dispatched to different validators.

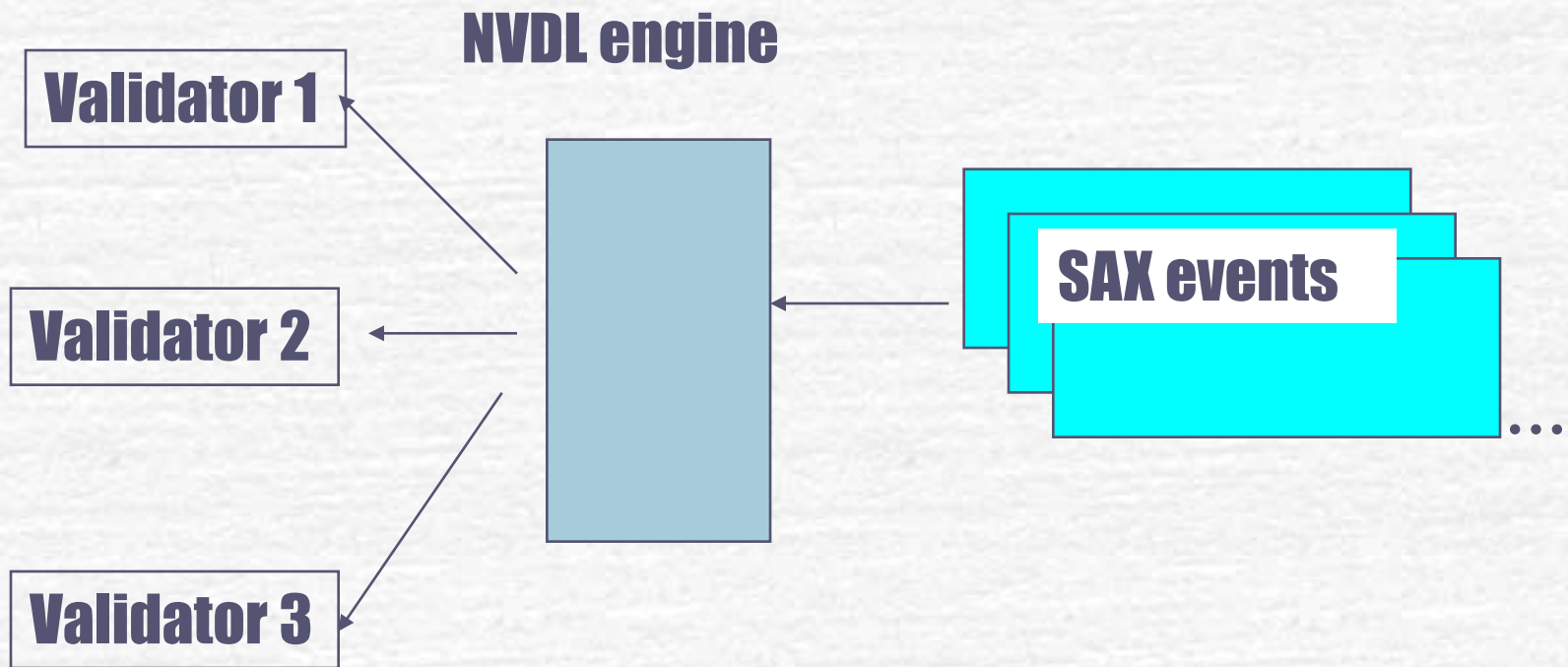
Processing model



Dividing non-monolithic documents into pieces



Implementation overview



Who uses NVDL?

- OOXML MCE (ISO/IEC 29500-3)
- ODF 1.2?
- W3C Internationalization Tag Set
<http://www.w3.org/TR/its/>
- W3C SVG Tiny 1.2
<http://www.w3.org/TR/SVGMobile12/>
- Open Publication Structure (OPS) of the International Digital Publishing Forum (Open eBook)

NVDL Implementations

- ▣ Jing
- ▣ Oxygen
- ▣ enovdl - Mono
- ▣ JNVDL
- ▣ Validator.nu
- ▣ SnRNV – Eclipse

Atom extensions: NVDL way (1)

- ❑ No changes to `atom.rnc`.
- ❑ Atom feeds without extensions are validated against `atom.rnc`
- ❑ Extensions are validated against different schemas.

Atom extensions: NVDL way (2)

Advantages

- More extensions can be added easily.
- No problems even when `atom.rnc` evolves.
- Full validation of atom feeds as well as Google extensions

Disadvantages

- Loose restrictions about mutual interactions
- What happens when `atom.rnc` is extended?

Summary

- Increasing use of atom extensions is a good thing, but validation is needed.
- Validation by RELAX NG is a short-term solution for validating a few extensions.
- Dispatching by NVDL is a long-term solution for validating many extensions at the same time although validation becomes somewhat loose.