IODEF json-xml converter

Mio Suzuki <mio@nict.go.jp>

IETF101 London
Converter Overview

• Aiming to convert from JSON IODEF document to XML IODEF document, and vice versa
• Running code for latest JSON IODEF draft (draft-ietf-mile-jsoniodef-03)
• I’m currently implementing the converter including IETF Hackathon days in London
Considered Approaches

1. Customizing OSS converter
   • There exist many OSS converters and conversion libraries
     • json2xml, xml2json, xmltojson, x2j, j2x, etc.
   • Suitable for simple conversion
   • Difficult to implement custom mapping rules between JSON and XML

2. Using XSLT 3.0 fn:json-to-xml and fn:xml-to-json
   • Described in https://www.w3.org/TR/xslt-30/#json
   • Easy for simple conversion
   • Enables custom mapping rules using XPath function but complex
   • I’m not familiar with XSLT...

3. Making it own
   • Implementing my own translator
Converter Design

XML IODEF Schema -> Code Generator (generate DS.py) -> Generated Python Code -> Code for Custom Mapping Rules -> Converted XML IODEF Doc

Original XML IODEF Doc -> Generated Python Code

Original JSON IODEF Doc -> Converter -> Converted JSON IODEF Doc

Original JSON IODEF Doc -> JSON IODEF Schema

Python Object Generation Library (python-jsonschema-objects) -> Python Code

Code for Custom Mapping Rules
Problems

• Object generator (python-jsonschema-objects) makes errors caused by self reference recursions
  • On “Contact”, ”EventData”, and “IndicatorExpression” in IODEF JSON schema
  • Originated from IODEF XML Schema
  • Problem of implementation of python-jsonschema-objects?
  • It could cause errors on other generators/validators

• Related JSON schema snippet is described below

"Contact": {
  (snip)
  "Contact": {
    "type": "array", "items": {"$ref": "#/definitions/Contact"}},
  (snip)
"EventData": {
  (snip)
  "EventData": {
    "type": "array", "items": {"$ref": "#/definitions/EventData"}},