Global Information Management

Data Validation as a Service

Presented By: *Aleksandar Balaban*
m-click.aero

Date: *August 26, 2015*

Outline

- Introduction
- Business Rules & Validation Preprocessing
- Validation Service Interface & Implementation
- Future Work
- Summary

INTRODUCTION

Validation Basics

- Validation is used to check the structural and logical consistency of aeronautical data, compliance with international standards and fulfillment of best practices

- Subject of validation is AIXM 5.1 data and extensions (Event/D-NOTAM and other EUR and FAA extensions such as eASM, FNSE, MXIA):
  - http://www.aixm.aero/gallery/content/public/schema/5.1/extensions/index.html

- Validation against:
  - Structural rules (XML schema)
  - Predefined set of business rules like those specified here: https://ext.eurocontrol.int/aixmwiki_public/bin/view/Main/AIXM_Business_Rules

AIXM 5.1 Validation

AIXM 5.1 CM

AIXM 5.1 XSD

- Does the data make sense?
- Is it compliant with international standards?
- Does it respect recommended practices?

Semantic check

Syntactic check

Check the compliance of an XML dataset with the XSD grammar

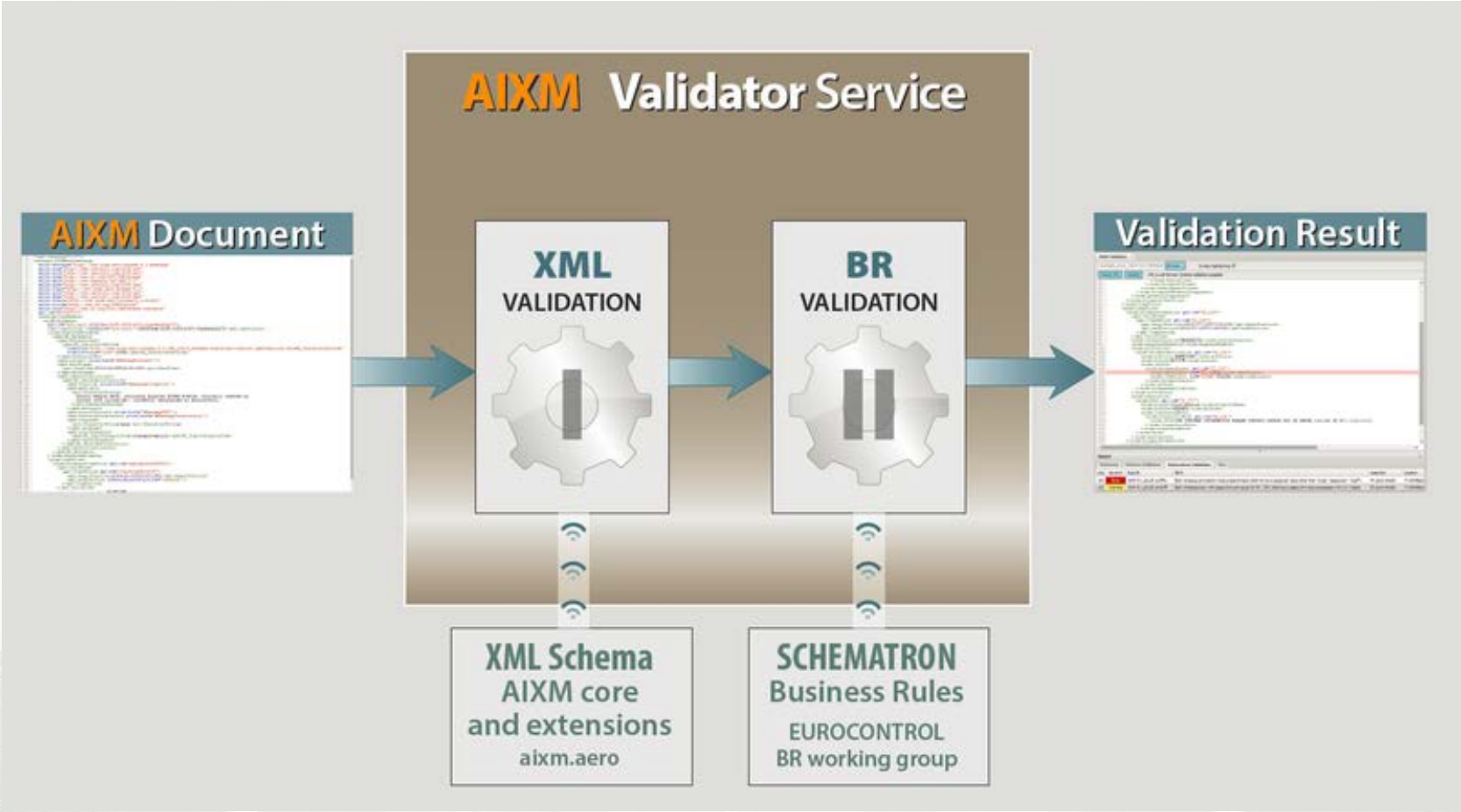
AIXM 5.1 XML

Features,

Events

Business Rules

Validation as Service, Workflow



Validation Basics 2

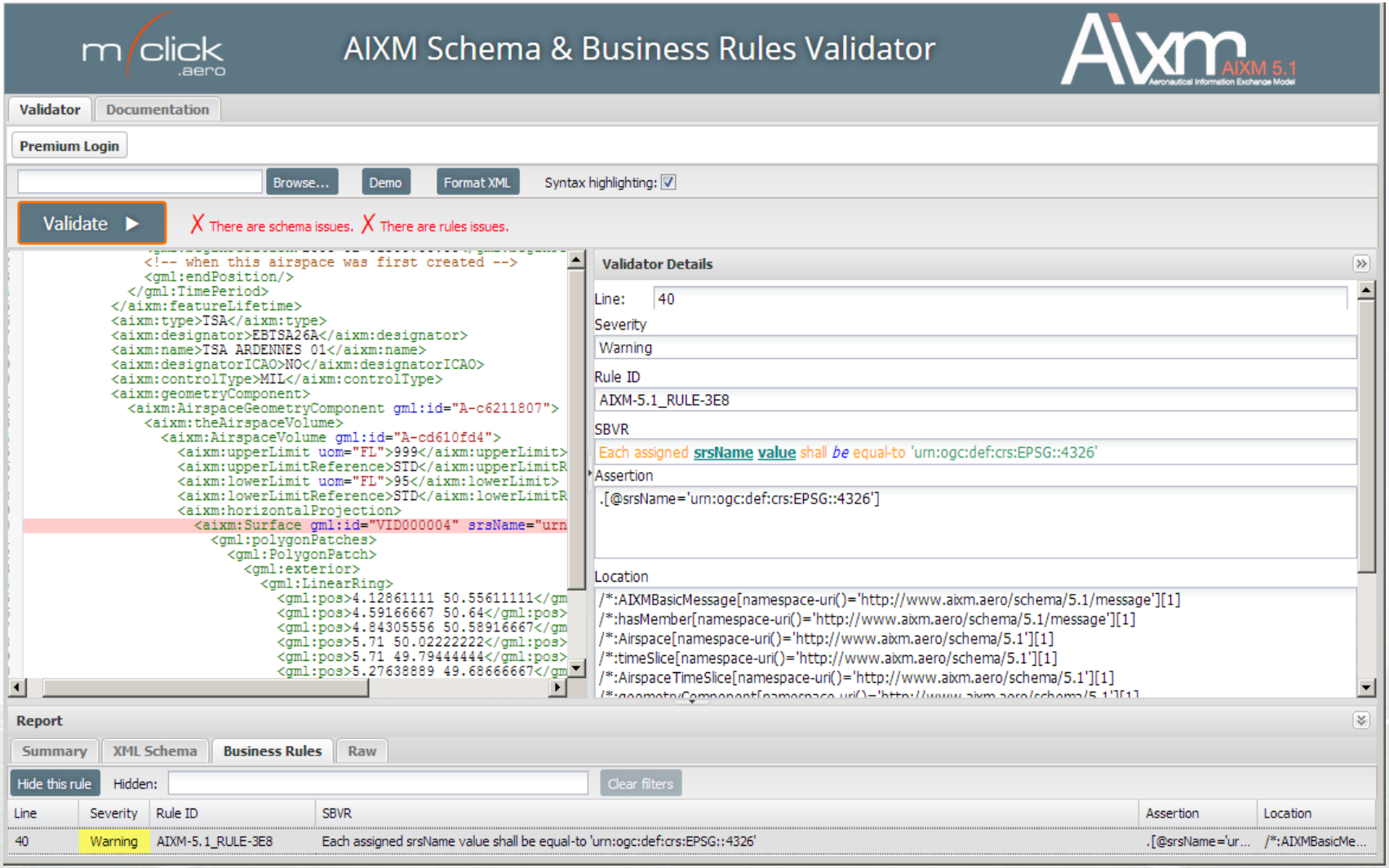
- Data validation is important both when data is issued and when data is used
- For static aeronautical data the data validation traditionally takes place at recipient side
- Data validation on own data repository requires local code
- Alternatively, the validation as a service based on concepts of cloud computing and SOA
- Validation service – a SWIM infrastructure service?

Validation as a Service, Advantages

- Standard SOA pro arguments
- Encapsulation and reuse by both data providers and data users (service reusability)
- Role as reference (authoritative) validator service
- Lower operational, maintenance and development costs

See m-click.aero implementation for an example of remote validation (web) service: https://swim.m-click.aero/validator/

Validation Service Example



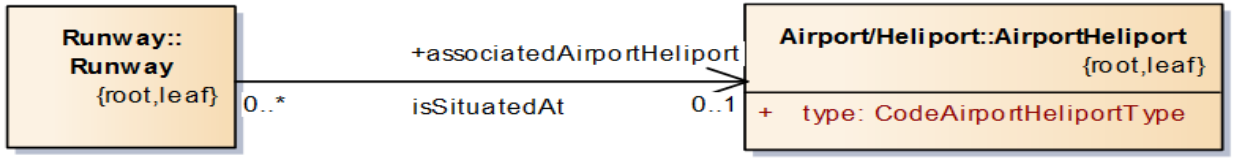
BUSINESS RULES & VALIDATION PREPROCESSING

Business Rules

- Document *AIXM 5.1 - Business Rules (data verification)* Rules are use case specific → only common rules should be defined in the schema itself
- Natural language (almost)
- Taxonomy like SVBR
- *Using SBVR and Schematron*: defines SBVR profile for writing AIXM business rules: https://ext.eurocontrol.int/aixmwiki_public/bin/view/Main/AIXM_Business_Rules

SBVR & AIXM Business Rules

- Semantics of Business Vocabulary and Business Rules (SBVR):
  - OMG Standard (http://www.omg.org/spec/SBVR/1.2/)
  - Formal and detailed natural language (like) declarative description of business rules
- Example:

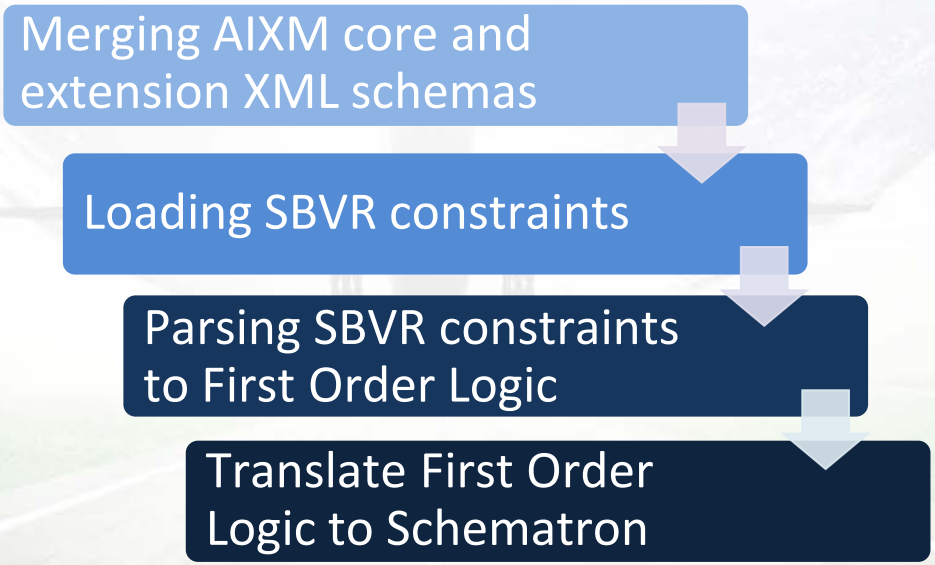


AIXM Business rule: *It is prohibited that a Runway with type equal-to 'RWY' isSituatedAt AirportHeliport with type equal-to 'HP'*

- Business rule specification contains many hundreds rules: https://ext.eurocontrol.int/aixmwiki_public/bin/download/Main/AIXM_Business_Rules/AIXM-5.1-BusinessRules-v.0.4.xlsx

Rules Transformation Process

- How to automatically enforce the validation rules (native English, SVBR)?
- Use Schematron for XSLT for service's rule execution engine.
- Generate Schematron for XSLT compatible, equivalent set of rules.



An Example

- **Business rule AIXM-5.1_RULE-EB5A0**:
  - Each **RunwayDirection.timeSlice** that belongsTo **Event** with scenario equal-to '**RWY.CLS**' and with version equal-to '2.0' shall have exactly one assigned availability value and shall have **availability.ManoeuvringAreaAvailability.operationalStatus** equal-to '**CLOSED**‘.
- **XPath within Schematron assertion** (context is aixm:RunwayDirection):
  - every $x1 in current()/aixm:timeSlice satisfies ((not(for $c3 in count(for $x2 in $x1/*/aixm:extension/*/event:theEvent return if (((for $c1 in count(for $x3 in //*[concat('#',@gml:id)=$x2/@xlink:href]/event:timeSlice/*/event:scenario return if ($x3 = 'RWY.CLS') then 1 else ()) return ($c1 >= 1)) and (for $c2 in count(for $x4 in //*[concat('#',@gml:id)=$x2/@xlink:href]/event:timeSlice/*/event:version return if ($x4 = '2.0') then 1 else ()) return ($c2 >= 1)))) then 1 else ()) return ($c3 >= 1)) or ((for $c4 in count(for $x5 in $x1/*/aixm:availability return if (not($x5[@xsi:nil='true'])) then 1 else ()) return ($c4 = 1)) and (for $c5 in count(for $x6 in $x1/*/aixm:availability/*/aixm:operationalStatus return if ($x6 = 'CLOSED') then 1 else ()) return ($c5 >= 1)))))

SBVR-to-Schematron Derivation

- Automated derivation of Schematron code from AIXM business rules:
  - **Saves time and resources** of experts
  - **Prevents errors** that can result from manual creation of Schematron code.
  - Works for rules written against conceptual model of **AIXM** and also of **AIXM extensions**
  - Translation rate: currently > 60% - can be increased (through future work)
  - Implemented in OGC Testbed 11:
    - As extension to ShapeChange (http://shapechange.net)
    - Further details available in *OGC Testbed 11 Aviation - Guidance on Using SBVR Engineering Report* (OGC document number 15-024, to be published on http://www.opengeospatial.org/standards/per)

VALIDATION SERVICE INTERFACE & IMPLEMENTATION

Introduction

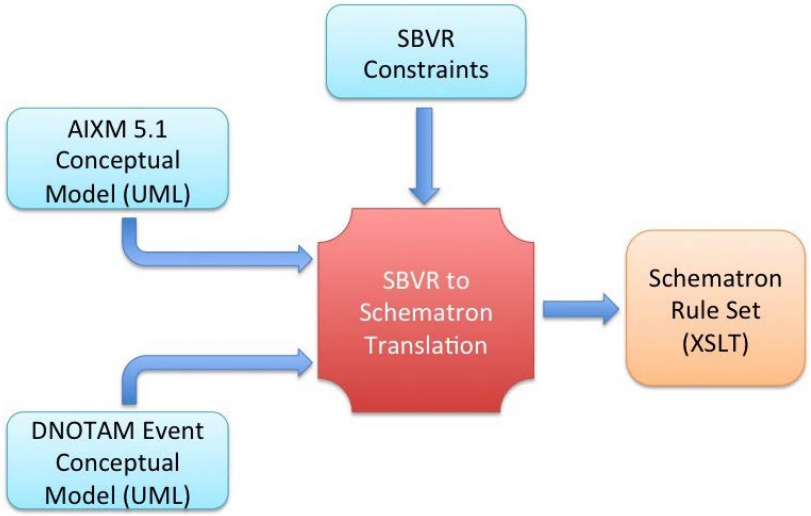
- Validation service:
  - Check AIXM 5.1 documents against set of predefined validation rules
  - Check D-NOTAMs against set of predefined validation rules
- May occur at data source, on client side or anywhere in the processing chain
- Digital NOTAM extension characteristics:
  - Extension to AIXM 5.1 schema
  - New feature type: Event
  - Additional reference “theEvent” for time slices (usually TEMPDELTA) of all feature types
- Further details available in *OGC Testbed 11 Aviation – Digital NOTAM Validation and Enrichment Service* (OGC document number 15-027, to be published on http://www.opengeospatial.org/standards/per)

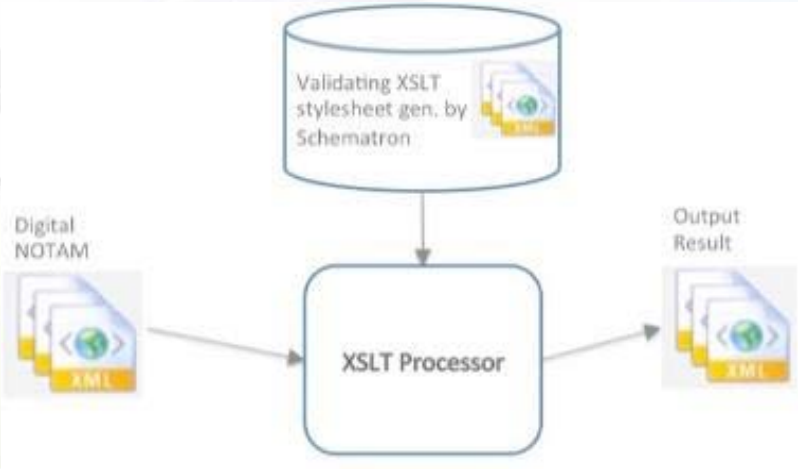
Service Interface

- OGC WPS 1.0 (Web Processing Service) compatible service endpoint (http://www.opengeospatial.org/standards/wps)
- Input
  - Feature collection (AIXM 5.1 entities or D-NOTAM messages)
  - In case of Digital NOTAM, Event feature and associated features if avaliable (RunwayDirection, etc.)
- Output
  - Validation report embedded in AIXM 5.1 feature's time slice metadata elements
  - Validation report encoded using ISO 19115/19139 derived Geographic MetaData extensible markup language (GMD).

Service Implementation

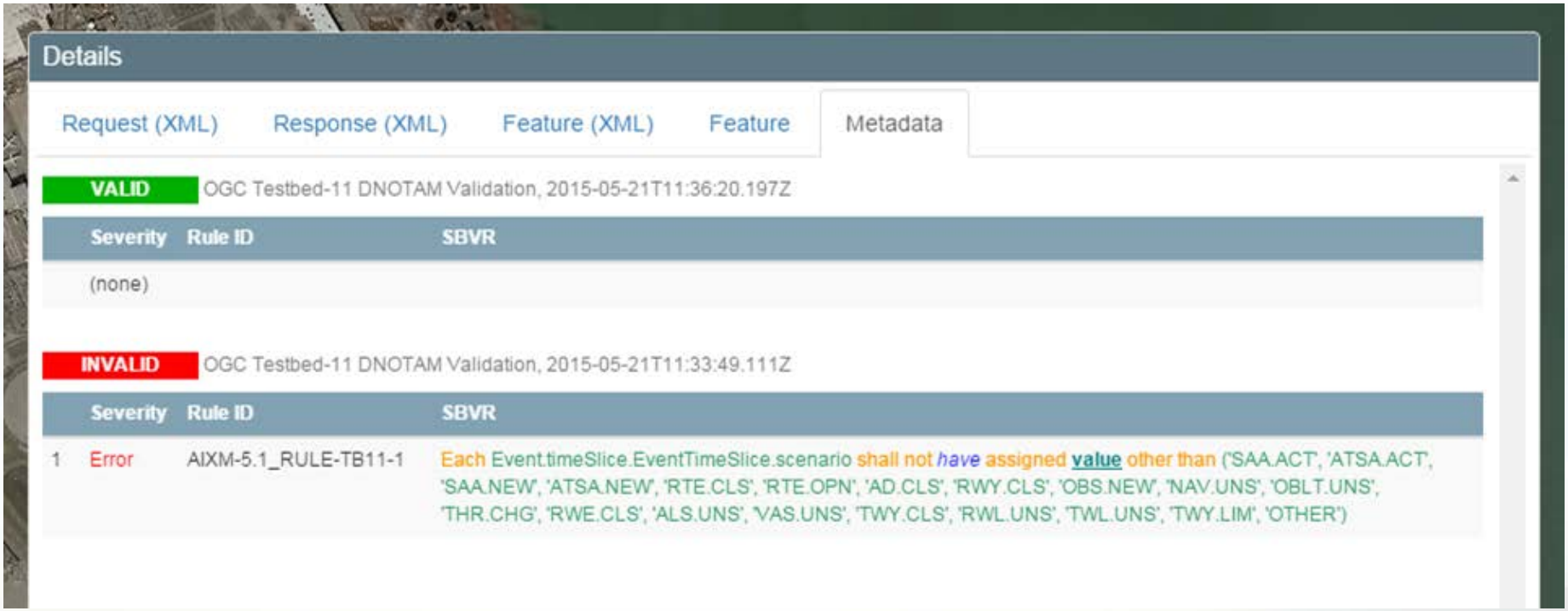
- Rules Repository
  - Business rules expressed in SBVR (formal, human-readable language)
  - Schematron rules are auto-generated from SBVR constraints
  - Rule repository as a XSLT document Schematron for XSLT rules
- Validation Engine
  - Schematron rules compiled to XSLT Using XPath 2.0 expressions
  - User defined XPath functions for reference resolving and geometries possible
  - WPS 1.0 service, requires well-formed XML input





Validation Report

Example, Web Client



Future Work

- Testing
- Complete AIXM 5.1 schema merging
- Continue working on SBVR rules
- WPS 2.0?
- Validation Service Interface improvements

Summary

- Validation as a Service work was performed within the OGC Testbed-11.
- The concept was developed in accordance with SOA, which promotes reusability.
  - Automated derivation of Schematron for XSLT code from AIXM 5.1 and D-NOTAM validation business rules (60%)
- Validation as a Service via standard OGC services such as the OGC WPS 1.0 (Web Processing Service) compatible service endpoint http://www.opengeospatial.org/standards/wps)
  - Validation report encoded using ISO 19115/19139 derived Geographic MetaData extensible markup language (GMD).

Thank you for your attention.

Any questions?

Aleksandar Balaban

Am Borsigturm 40, 13507
Berlin, Germany
aleksandar.balaban@m-click.aero