Design Suggestions for XML++

XML Prague 14.02.2020

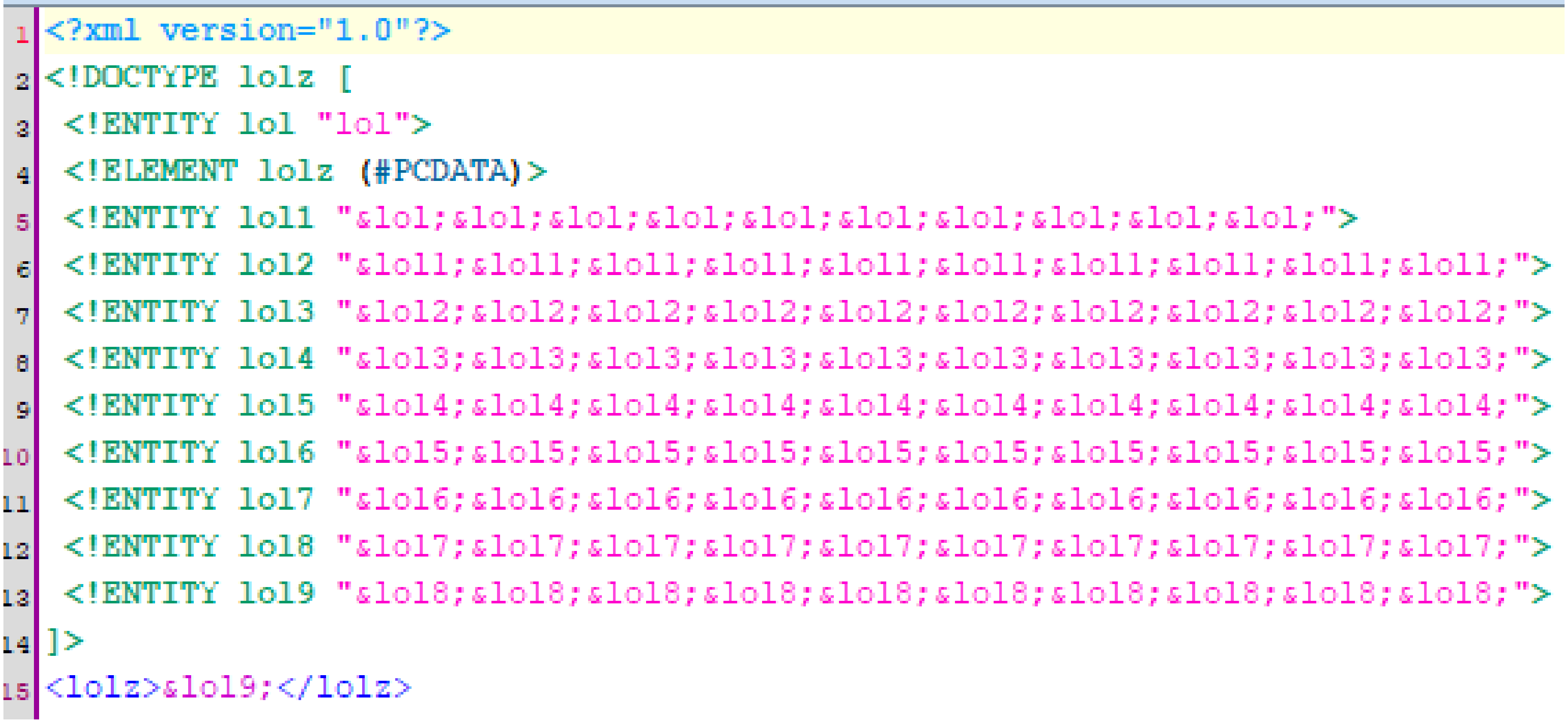
Ruthless Dictator Slogan

Make XML Great Again!

XML Design flaws

DoS Attack

Expand to >3 gigabytes memory



```xml
<?xml version="1.0"?>
<!DOCTYPE lolz [
 <!ENTITY lol "lol">
 <!ELEMENT lolz (#PCDATA)>
 <!ENTITY lol1 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
 <!ENTITY lol2 "&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;&lol1;">
 <!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
 <!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
 <!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
 <!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
 <!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
 <!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
 <!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

See Wiki: One Billion Laugh Attack / XML Bomb

Cut old Braids

German Expression: **Cut old braids** - Students during Wars of Liberation in 1813-1815

Simplify XML

- MicroXML (2012) solves XML Security & Complexity:
  - **MicroXML prohibits 12 features of XML 5th edition**
    - No custom entities
    - Only UTF-8
    - …
    - No draconian XML error handling (see XML 5 - 2014-2016)
  - See James Clark Blog Entry (Dec. 2010)
  - Better marketing: LeanXML, SecureXML,…, XML2.0
  - Desired: Transformations from old to new XML

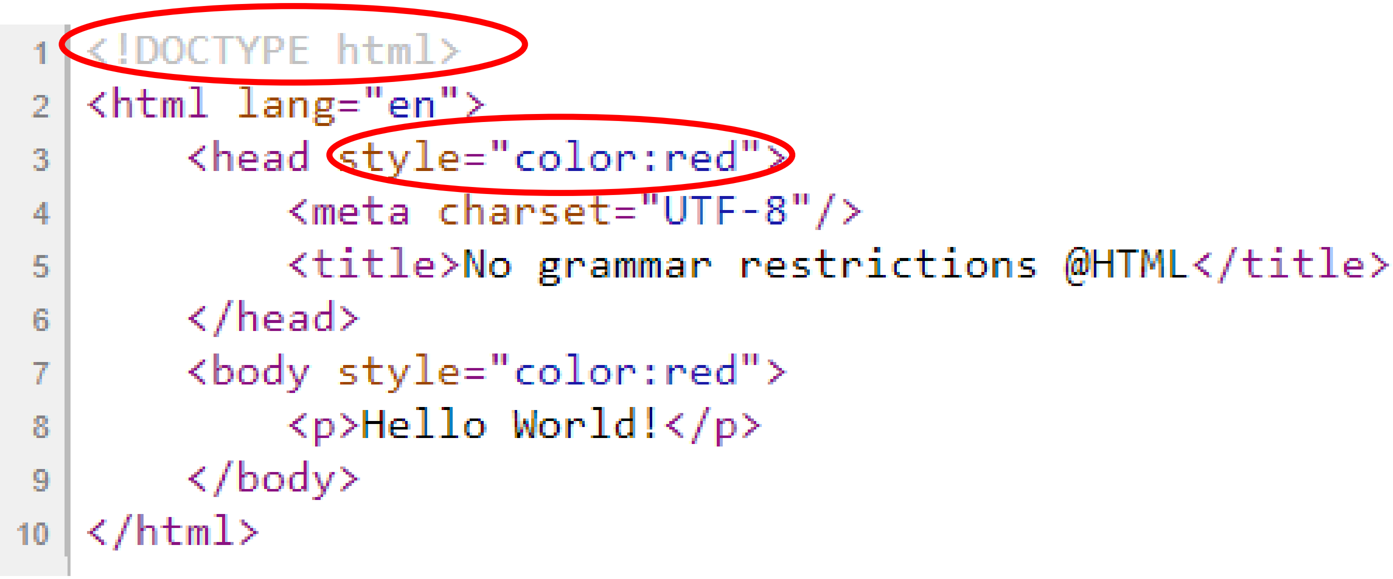
- **Summary:** Fix/Simplify XML for new implementations!

Simplify Tooling

- Why do we need >3 XML grammars?
  - **RelaxNG** (rng) used by ODF
  - **W3C Schema** (xsd) used by
    - OOXML
    - UBL
    - UN/CEFACT Cross-Industry Invoice (CII) XML
  - **DTD** explicitly not used by HTML 5

Valid HTML 5

W3 Validation without warning



```html
<!DOCTYPE html>
<html lang="en">
    <head style="color:red">
        <meta charset="UTF-8"/>
        <title>No grammar restrictions @HTML</title>
    </head>
    <body style="color:red">
        <p>Hello World!</p>
    </body>
</html>
```

- DTD unset as not sufficient enough
- HTML 5 allows styles everywhere

The Power of XML grammar

- Most expressive: RelaxNG
- Less expressive: W3C Schema
- Least expressive: DTD (should be dropped)

See Taxonomy of XML Schema Languages Using Formal Language Theory (2005)

XML Grammar Tools

- Sun's Multi-Schema-Validator (MSV)
  - Internal RelaxNG Model (most expressive)
  - Loads DTD, W3C Schema & RelaxNG
- Trang Converter from James Clark
  - W3C XML Schema only as output! (why?)
- How to generate scanners/validators from grammar: Regular-Expression Derivatives (paper from 1964)

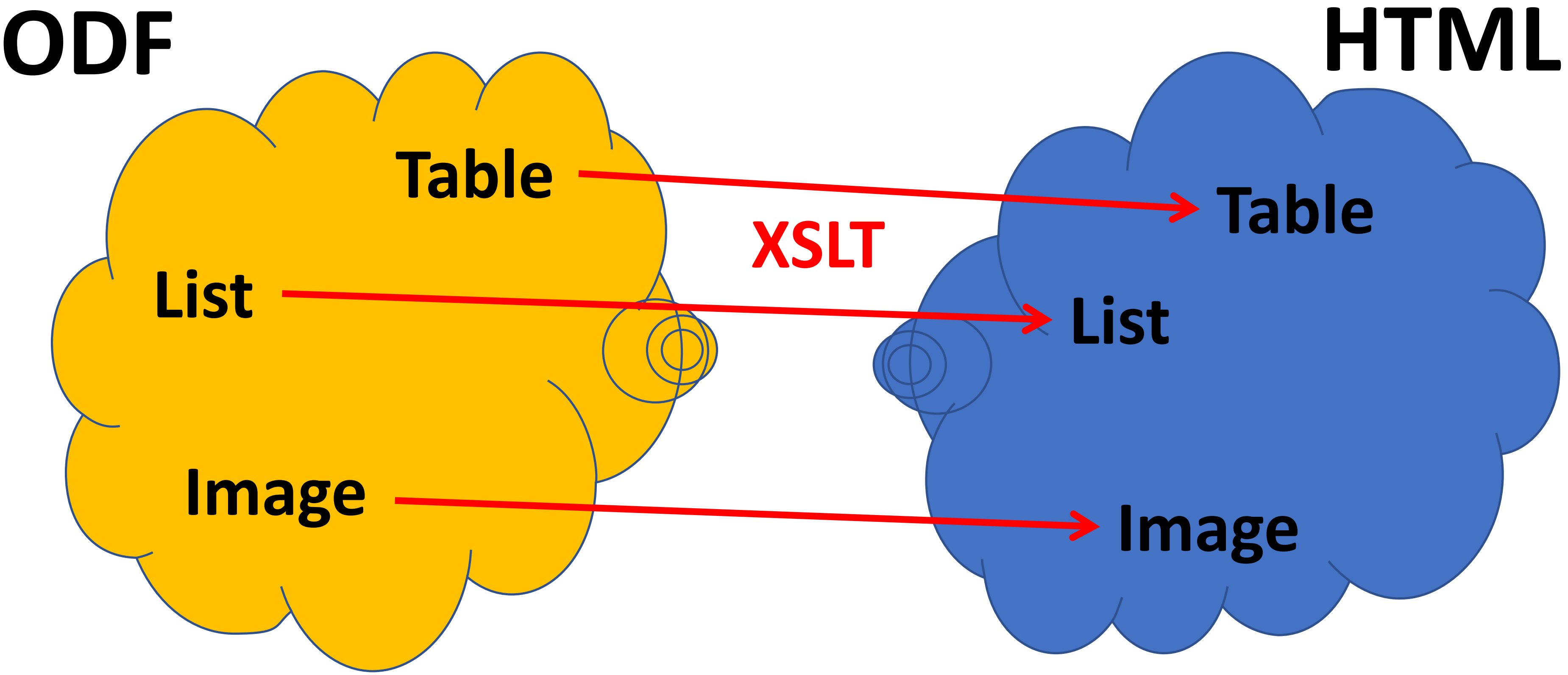
Journey into my Past

5 User Requirements

Example Requirements (1/5) - Mapping

Sun's Web Office (1999) – ODF Viewer

- Web Viewer of StarOffice (later ODF) Documents



Svante Schubert

Example Requirements (1/5) - Mapping

Sun’s Web Office (1999) – ODF Viewer

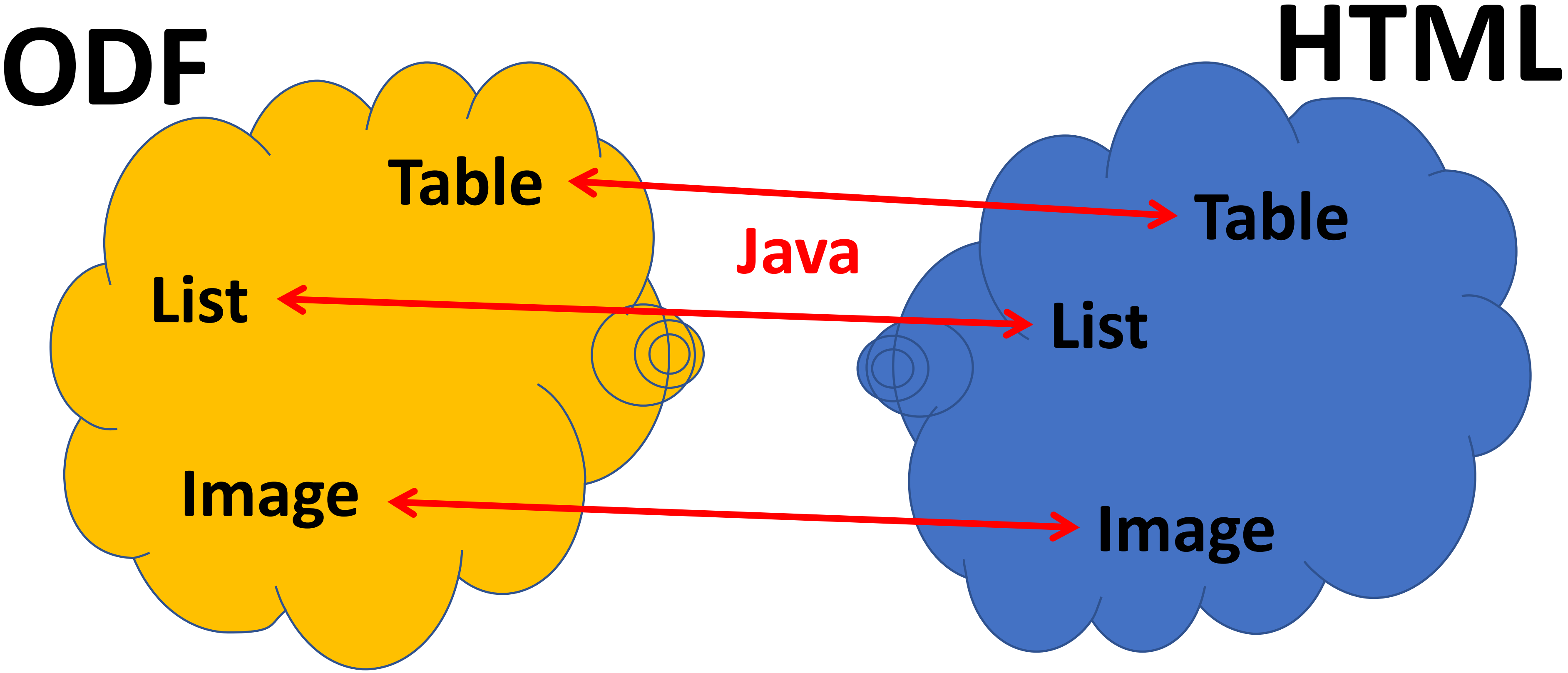
- Management & myself were often asking:
  - How much is done? When are we ready?
    (Grammar can tell coverage! – Saxon feature?)
  - Where is the bug? In my mapping or my XSLT?
  - Reuse some work for a Java DOM / SAX approach?

Example Requirements (2/5) – Two-Way-Mapping

Web Office 2nd Edition (till 2011)

Elaborate a mapping based on the Grammar

- **Web Editor ODF Documents – both way mapping**



Example Requirements (3/5) – XML Run Time Model

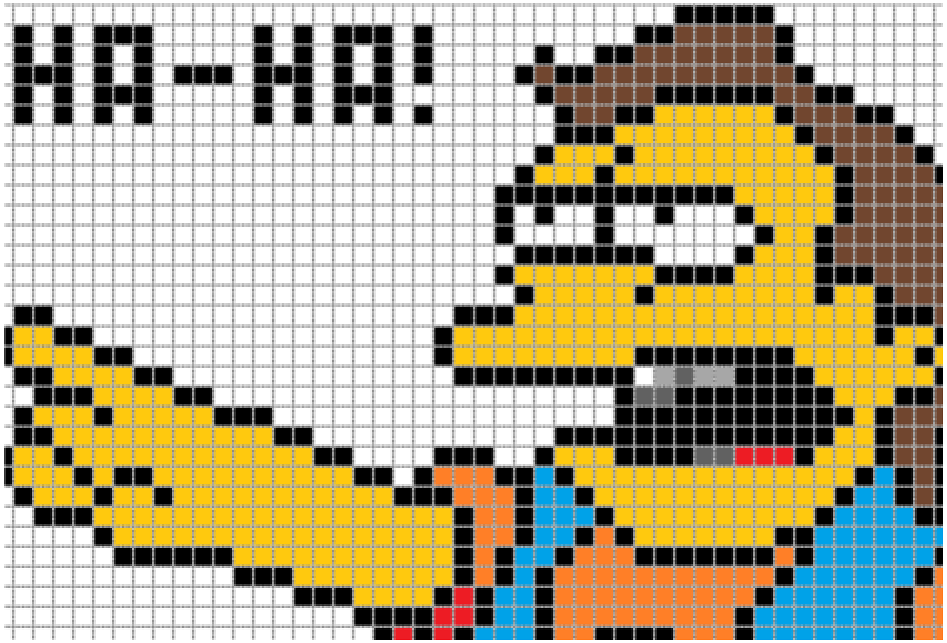
Run Time Model from RelaxNG Grammar

- Java XML Binding worked only for W3C Schema
- Load, Edit and Save ODF
- **ODF Runtime Environment** – ODFDOM (Java)
  - Loaded RelaxNG into MultiSchemaValidator (MSV)
  - Using MSV API to fill templates by (Apache Velocity)
  - Created from grammar Java (typed) DOM classes:
    <draw:image> → DrawImageElement class

Example Requirements (3/5) – XML Run Time Model

Run Time Model from RelaxNG Grammar

- XML DOM not succinct data structure
- Not well for huge documents
- Not streaming XML by Netflix
- Spreadsheet Pixel-Art problematic:



Example Requirements (4/5) – RTM close Semantic

Syntax is an implementation detail, Semantic counts!

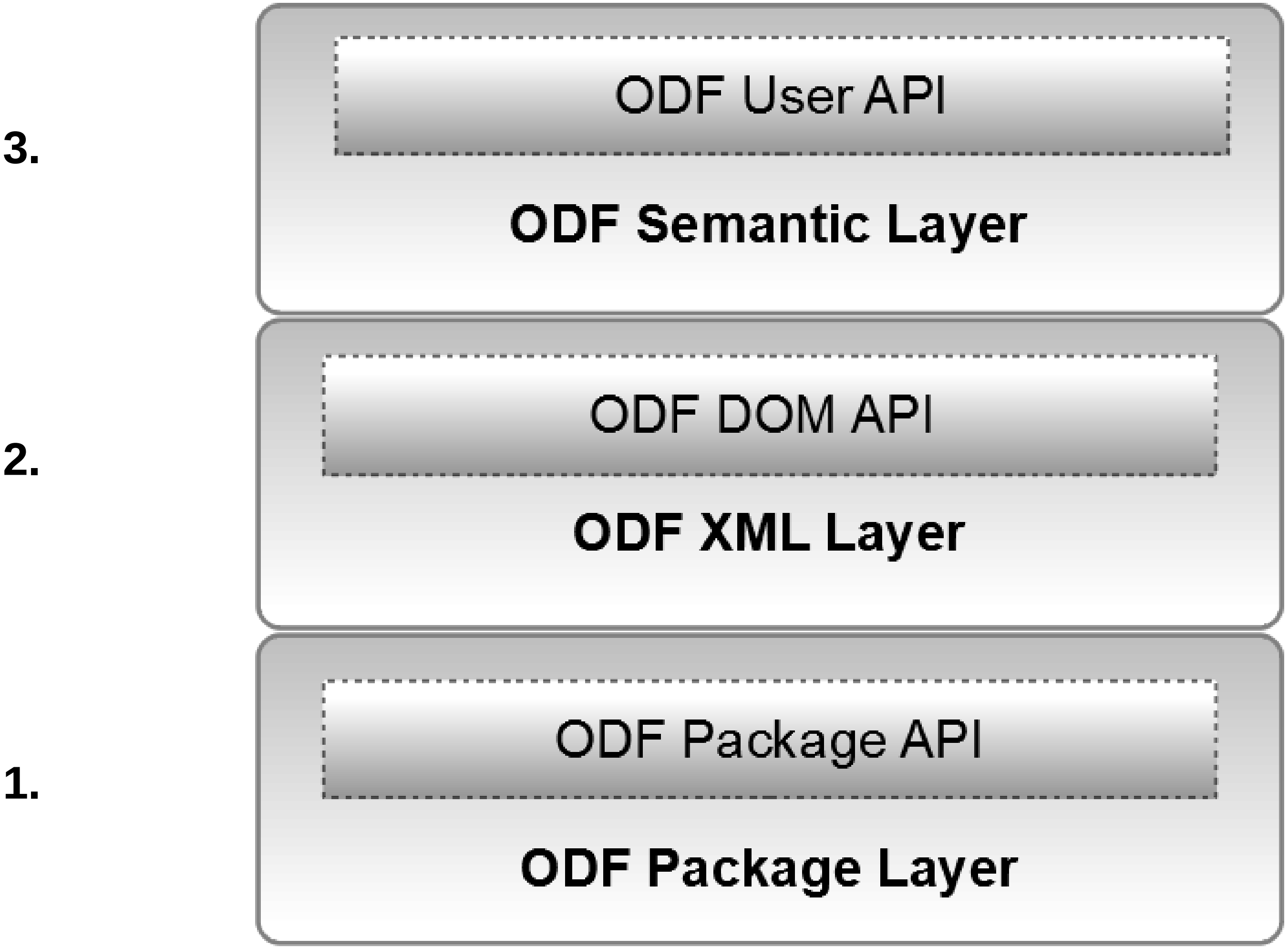
- I’ve been so proud of ODF XML..
- J David Eisenberg once said in a conf call:

“Leave me alone with your XML details, I want easy ***semantics****: Open Text doc, add paragraph with ‘HelloWorld!’ No implementation details desired!”*

- ***JAXB*** *created RunTimeModels overtaking XML syntax as semantic (improvable by abstracting XML details)*

Example Requirements (5/5) – Generate Everything

Generate as much as possible!



Example Requirements (5/5) – Generate Everything

Generation is based on Grammar

- ODF RelaxNG grammar is a text file of > 18.000 lines
- Hard to extract information by ODF users
- Infos missing: Semantic User Entities consisting of multiple XML like a table not yet defined in grammar!

ODF GRAMMAR

HARD TO ANSWER

Can a
paragraph **<text:p>**
be nested
in a valid document?
🤔🙄

ODF 1.2 XML:

- **598 XML Elements**
- **1300 XML Attributes**

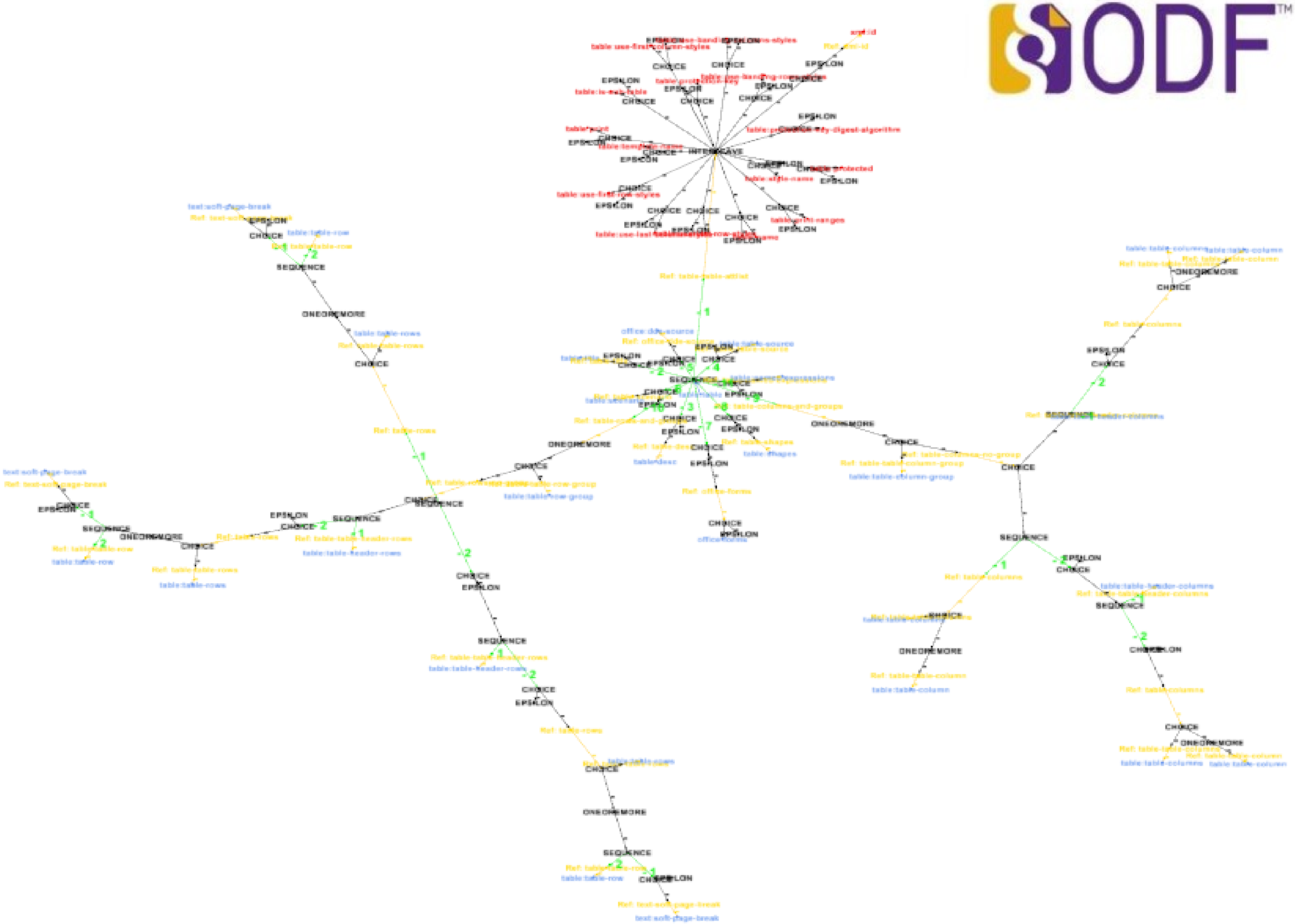
>18k lines

Example Requirements (5/5) – Generate Everything

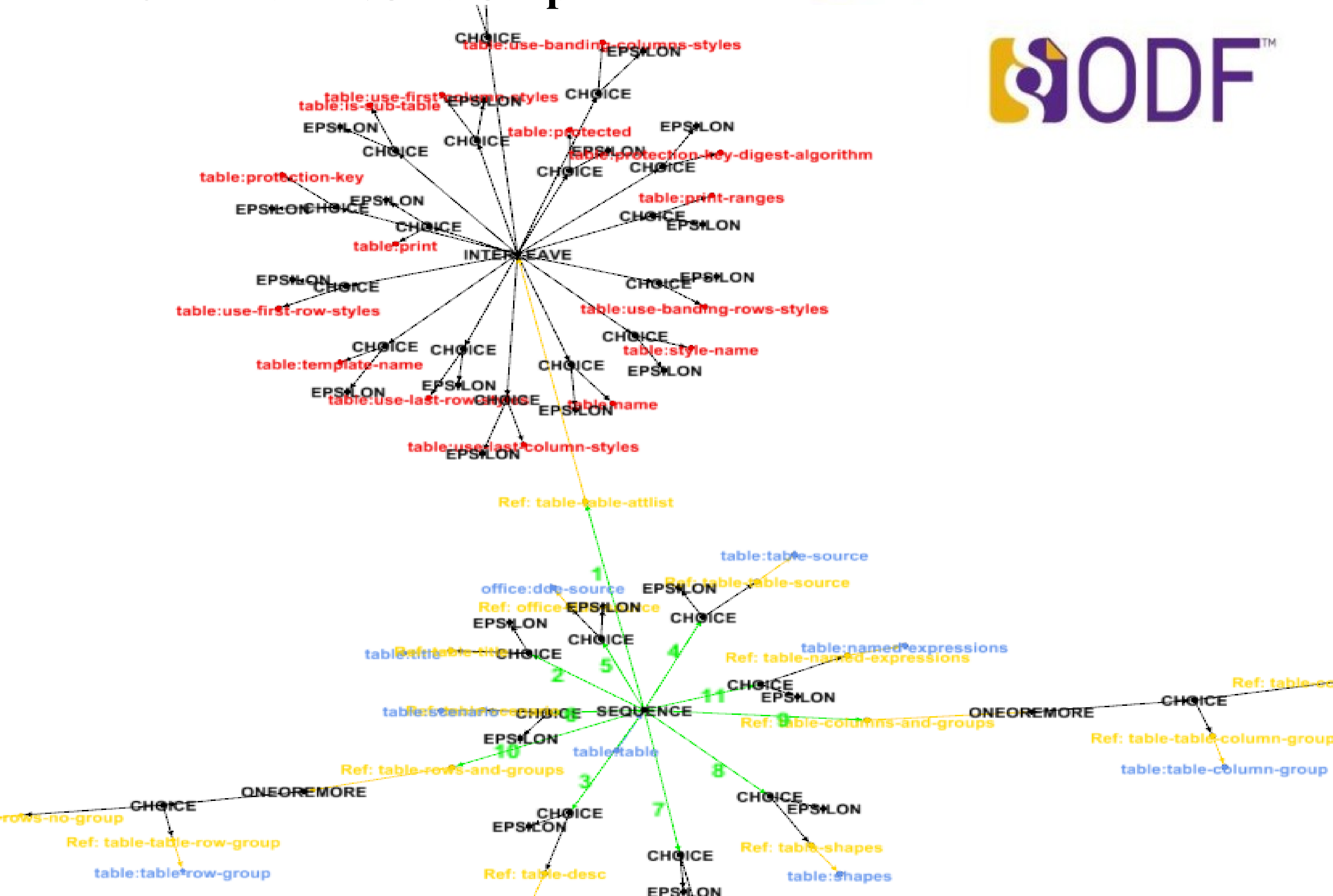
Generation based as RelaxNG Graph

- Solvable by traversing a (tree-ish) graph…
- Based on idea: Loading source code for analysis as graphs
- Loaded ODF RelaxNG from MSV into Apache Tinkerpop
- Saved Subgraphs from element to element as GraphML
- The ***<table:table>*** element with all its child elements
- Table GraphML rendered in Gephi

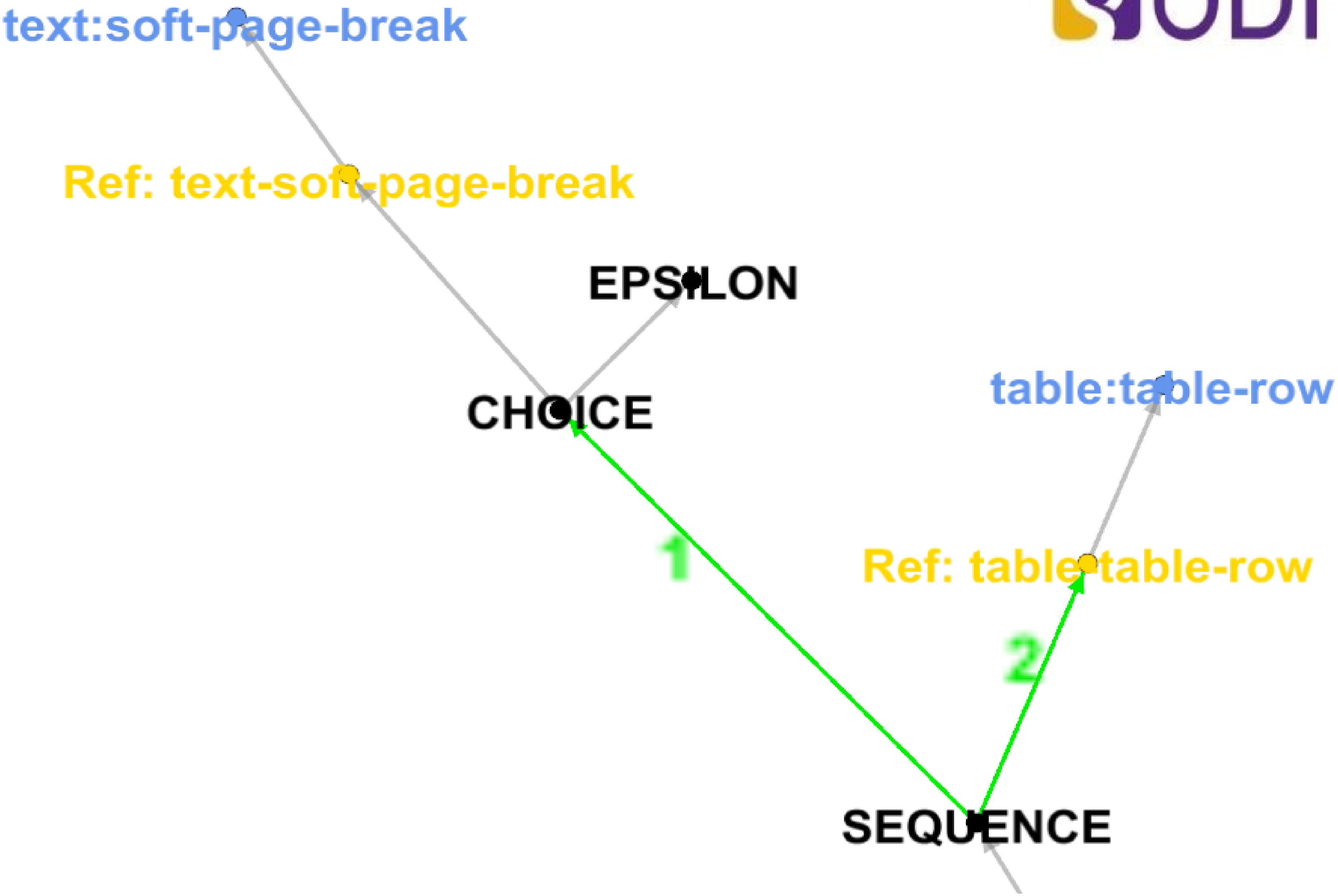
ODF RelaxNG as Graph



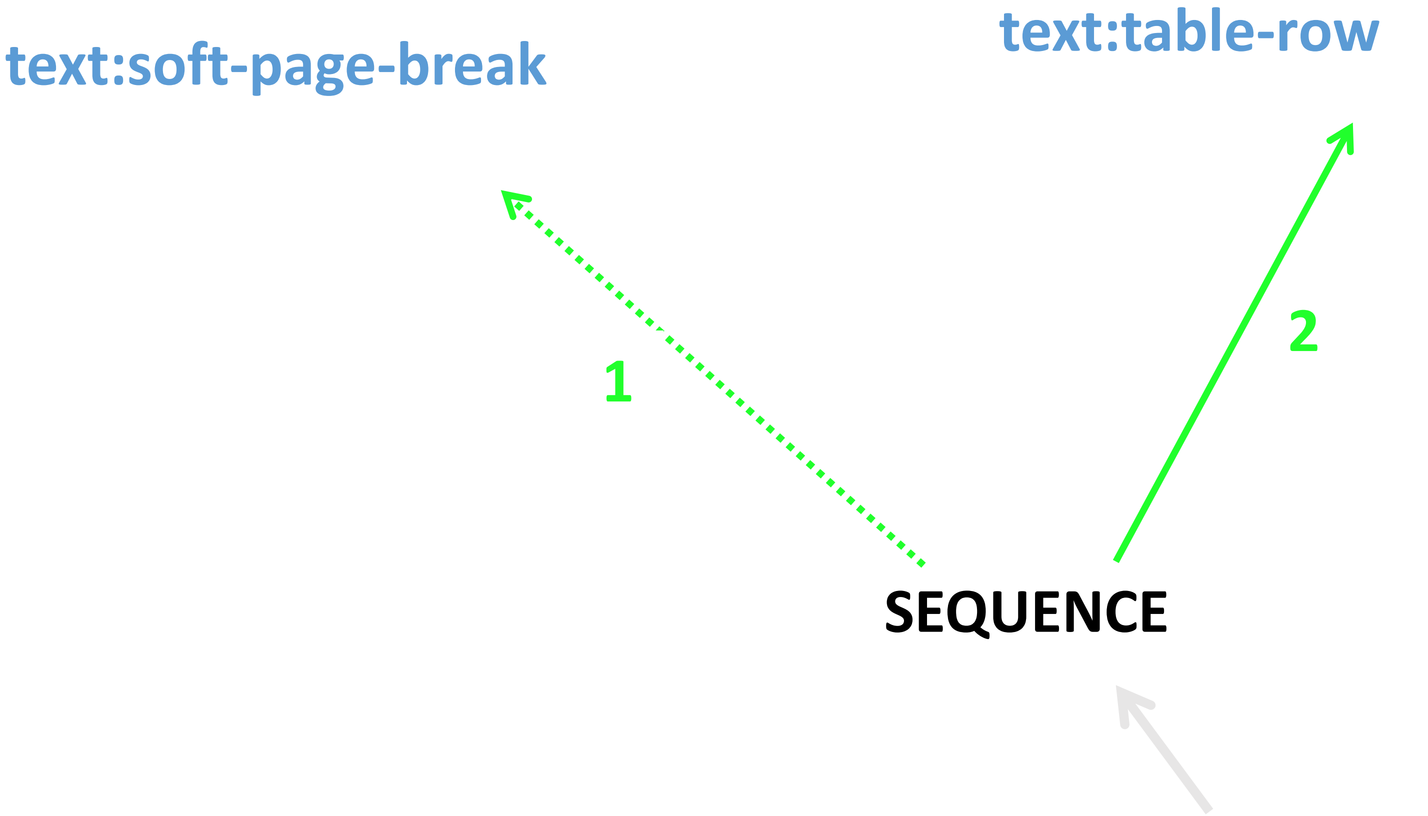
ODF RelaxNG as Graph



ODF RelaxNG as Graph



ODF RelaxNG as Graph



text:soft-page-break

text:table-row

1

2

SEQUENCE

Example Requirements (5/5) –Generate Everything

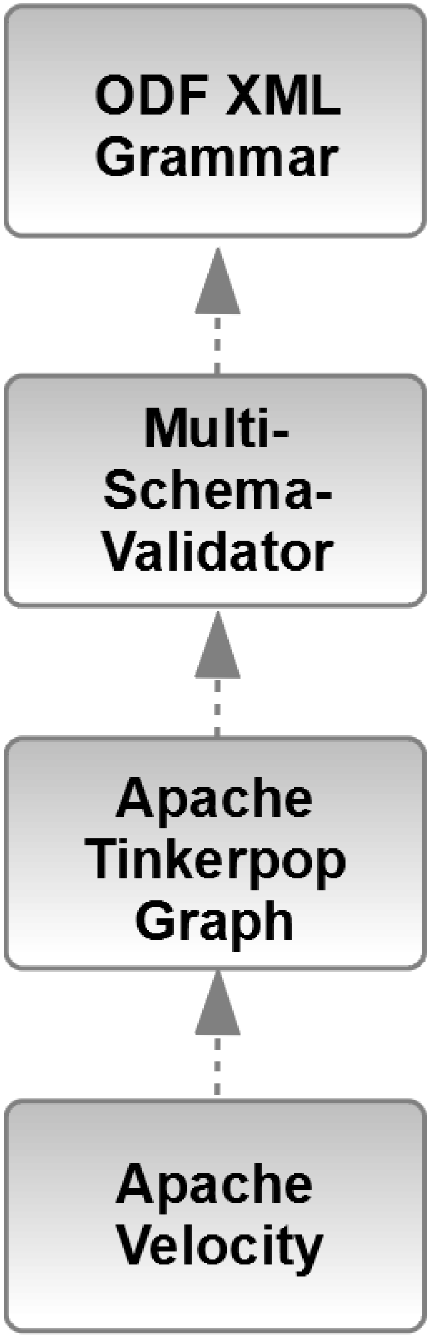
Improved RunTimeModel from XML Grammar

- JSON is far easier usable as RunTimeModel than XML
- Some Grammar to RunTimeModel features are still missing:
  - Insertion handling for multiple optional child elements: A,B,C
  - Parent with multiple similar children with ID
    Parent should become a lazy map (created on demand)
  - General missing XML grammar annotation/reverse-engineering Tool

ODF Toolkit

Source Code Generator

Architecture



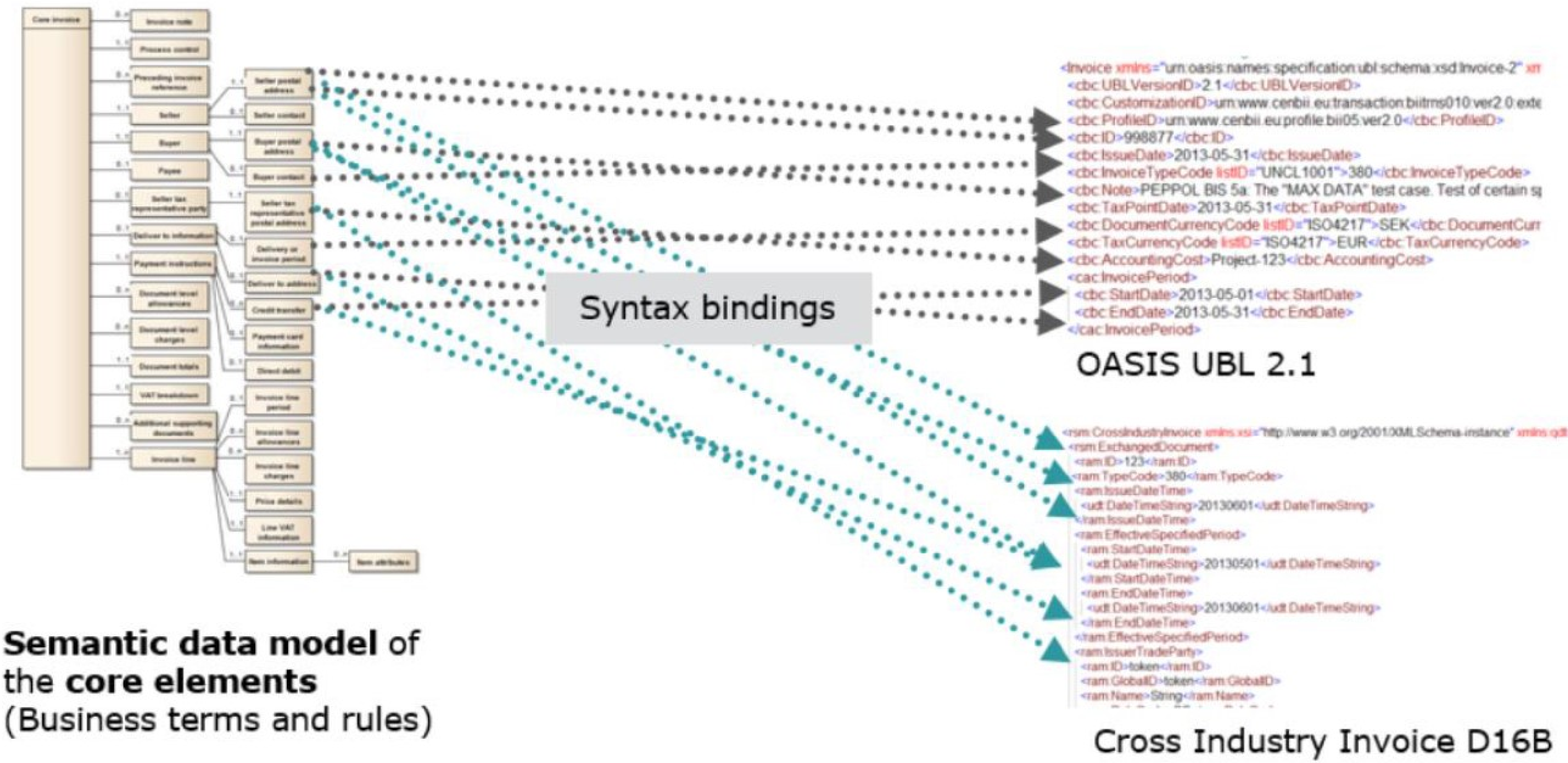
RelaxNG grammar:
~18000 text lines
~600 XML elements
~1200 XML attributes

XML Valdiator -
reads many
XML grammars

Graph of
XML grammar
(since ODF TOOLKIT 1.0.0)

template engine
- generating sources
by text templates with context
with Java access

Semantic Model
as Glue between XML Formats



Syntax bindings

```
<Invoice xmlns="urn:oasis:names:specification:ubl:schema:xsd:Invoice-2" xm
  <cbc:UBLVersionID>2.1</cbc:UBLVersionID>
  <cbc:CustomizationID>urn:www.cenbii.eu:transaction:biitrns010:ver2.0:exte
  <cbc:ProfileID>urn:www.cenbii.eu:profile:bii05:ver2.0</cbc:ProfileID>
  <cbc:ID>998877</cbc:ID>
  <cbc:IssueDate>2013-05-31</cbc:IssueDate>
  <cbc:InvoiceTypeCode listID="UNCL1001">380</cbc:InvoiceTypeCode>
  <cbc:Note>PEPPOL BIS 5a: The "MAX DATA" test case. Test of certain sp
  <cbc:TaxPointDate>2013-05-31</cbc:TaxPointDate>
  <cbc:DocumentCurrencyCode listID="ISO4217">SEK</cbc:DocumentCurr
  <cbc:TaxCurrencyCode listID="ISO4217">EUR</cbc:TaxCurrencyCode>
  <cbc:AccountingCost>Project-123</cbc:AccountingCost>
  <cac:InvoicePeriod>
   <cbc:StartDate>2013-05-01</cbc:StartDate>
   <cbc:EndDate>2013-05-31</cbc:EndDate>
  </cac:InvoicePeriod>
```

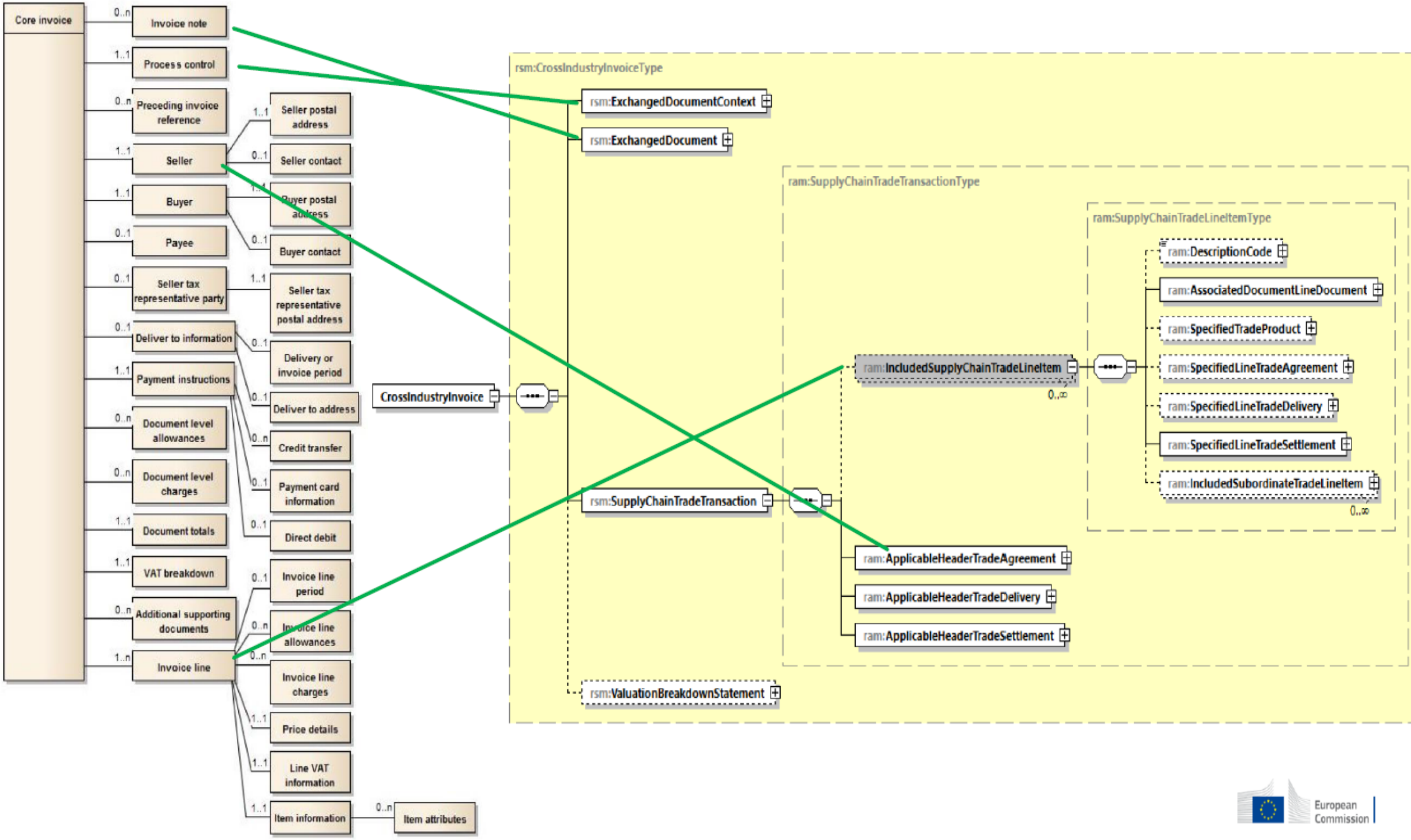
OASIS UBL 2.1

```
<rsm:CrossIndustryInvoice xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:qdt
 <rsm:ExchangedDocument>
  <ram:ID>123</ram:ID>
 <ram:TypeCode>380</ram:TypeCode>
  <ram:IssueDateTime>
   <udt:DateTimeString>20130601</udt:DateTimeString>
 </ram:IssueDateTime>
  <ram:EffectiveSpecifiedPeriod>
   <ram:StartDateTime>
    <udt:DateTimeString>20130501</udt:DateTimeString>
   </ram:StartDateTime>
   <ram:EndDateTime>
    <udt:DateTimeString>20130601</udt:DateTimeString>
   </ram:EndDateTime>
  </ram:EffectiveSpecifiedPeriod>
  <ram:IssuerTradeParty>
   <ram:ID>token</ram:ID>
   <ram:GlobalID>token</ram:GlobalID>
   <ram:Name>String</ram:Name>
```

Cross Industry Invoice D16B

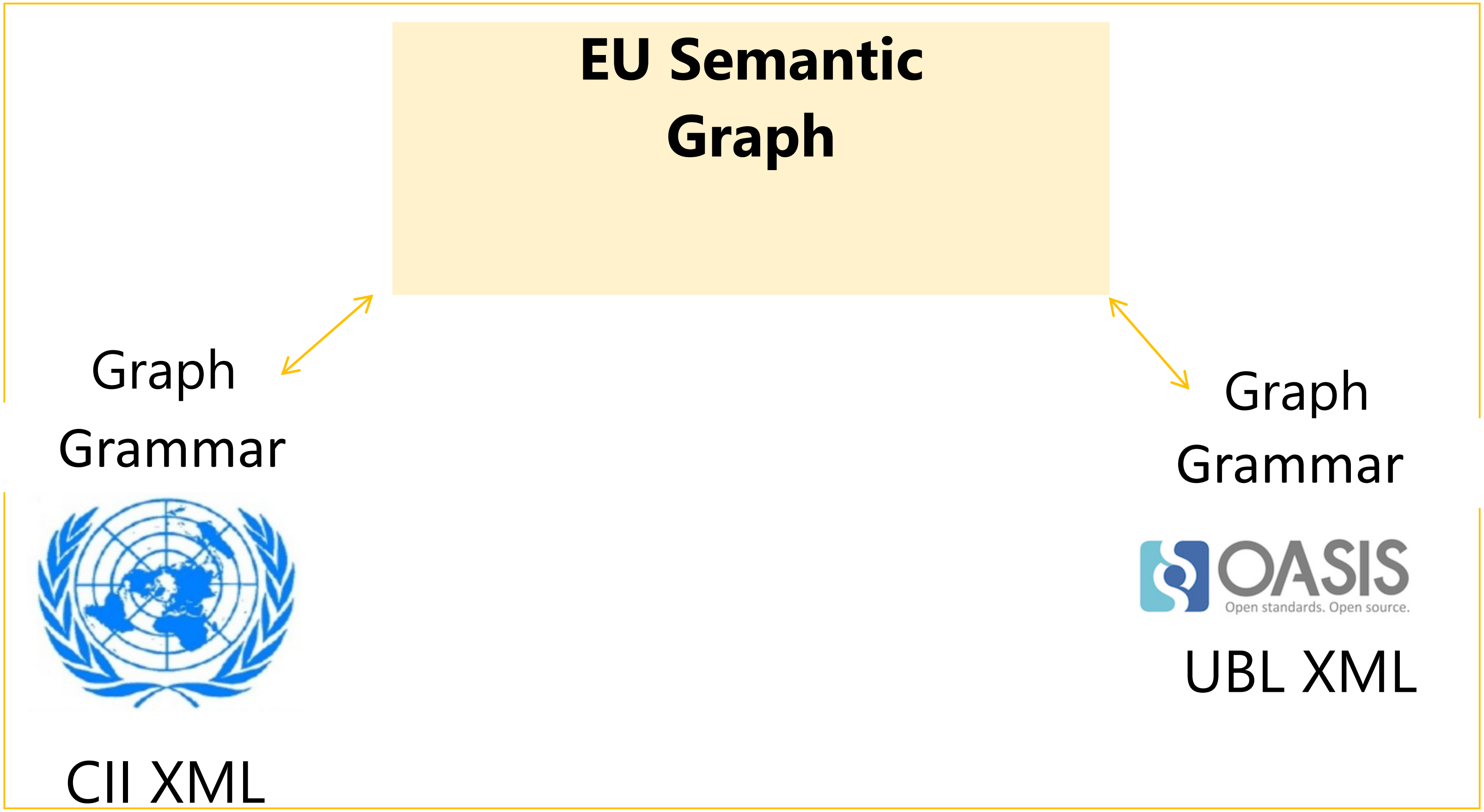
Semantic data model of
the **core elements**
(Business terms and rules)

Semantic Model
to CII XML Format


Core invoice
0..n
Invoice note
1..1
Process control
0..n
Preceding invoice reference
1..1
Seller
1..1
Seller postal address
0..1
Seller contact
1..1
Buyer
1..1
Buyer postal address
0..1
Buyer contact
0..1
Payee
0..1
Seller tax representative party
1..1
Seller tax representative postal address
0..1
Deliver to information
0..1
Delivery or invoice period
0..1
Deliver to address
1..1
Payment instructions
0..n
Credit transfer
0..1
Payment card information
0..1
Direct debit
0..n
Document level allowances
0..n
Document level charges
1..1
Document totals
1..1
VAT breakdown
0..n
Additional supporting documents
1..n
Invoice line
0..1
Invoice line period
0..n
Invoice line allowances
0..n
Invoice line charges
1..1
Price details
1..1
Line VAT information
1..1
Item information
0..n
Item attributes
CrossIndustryInvoice
rsm:CrossIndustryInvoiceType
rsm:ExchangedDocumentContext
rsm:ExchangedDocument
rsm:SupplyChainTradeTransaction
rsm:ValuationBreakdownStatement
ram:SupplyChainTradeTransactionType
ram:IncludedSupplyChainTradeLineItem
0..∞
ram:ApplicableHeaderTradeAgreement
ram:ApplicableHeaderTradeDelivery
ram:ApplicableHeaderTradeSettlement
ram:SupplyChainTradeLineItemType
ram:DescriptionCode
ram:AssociatedDocumentLineDocument
ram:SpecifiedTradeProduct
ram:SpecifiedLineTradeAgreement
ram:SpecifiedLineTradeDelivery
ram:SpecifiedLineTradeSettlement
ram:IncludedSubordinateTradeLineItem
0..∞

Graph as Data Model

Data from EU Specification as Graph
(Sum of 3 connected Graphs)



EU Semantic Graph

Graph Grammar

CII XML

Graph Grammar

UBL XML

Take Away

- XML should be simplified for newcomers
- XML Grammar are far more powerful than expected
- Too little Grammar conversion & tooling
- **Focus** should be more **on Semantic** instead of Syntax

To be continued…