Dictionary Driven Exchange Content Assembly Blueprints

Concepts, Procedures and Techniques

Author:

David RR Webber

Chair OASIS CAM TC

January, 2010

http://www.oasis-open.org/committees/cam

(CAM – Content Assembly Mechanism Specification)

Agenda

- Today’s XSD Schema-based Exchanges
  - Current accepted practice – pitfalls and challenges
  - How to do this faster, simpler, more reliably?
  - Accelerated process overview
- Blueprints and Dictionaries
  - Constructing your exchange with Blueprint templates
  - Leveraging re-use – standard domain dictionaries
  - Creating your own domain dictionary from XSD or UML
- Generating Exchange Artifacts
  - NDR evaluation, Exchange schema, mapping crosswalk, XML instances, realistic data use, business rules documentation
- Summary

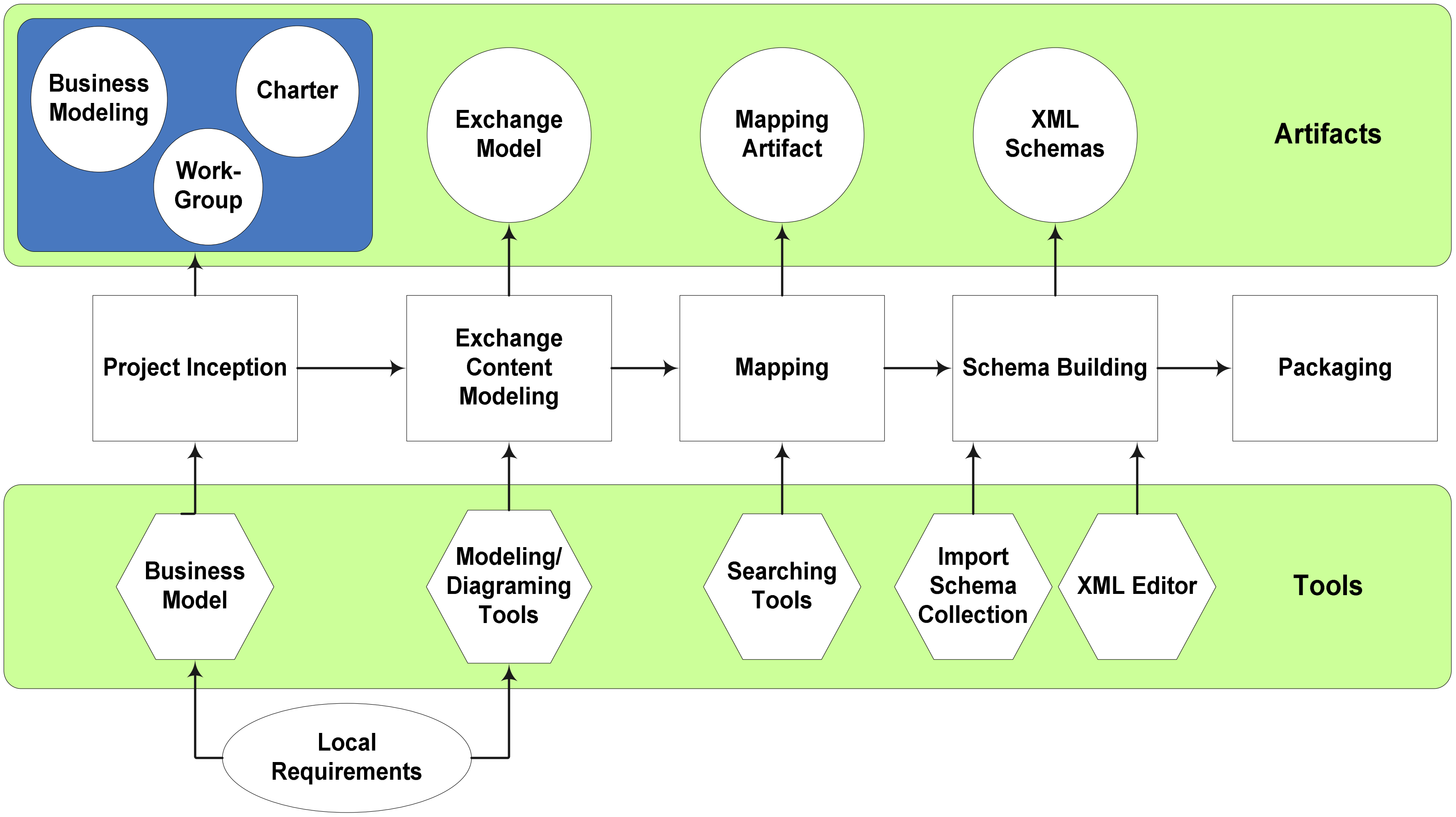
Today's XSD Schema-based Exchanges

Current Practice – Pitfalls and Challenges

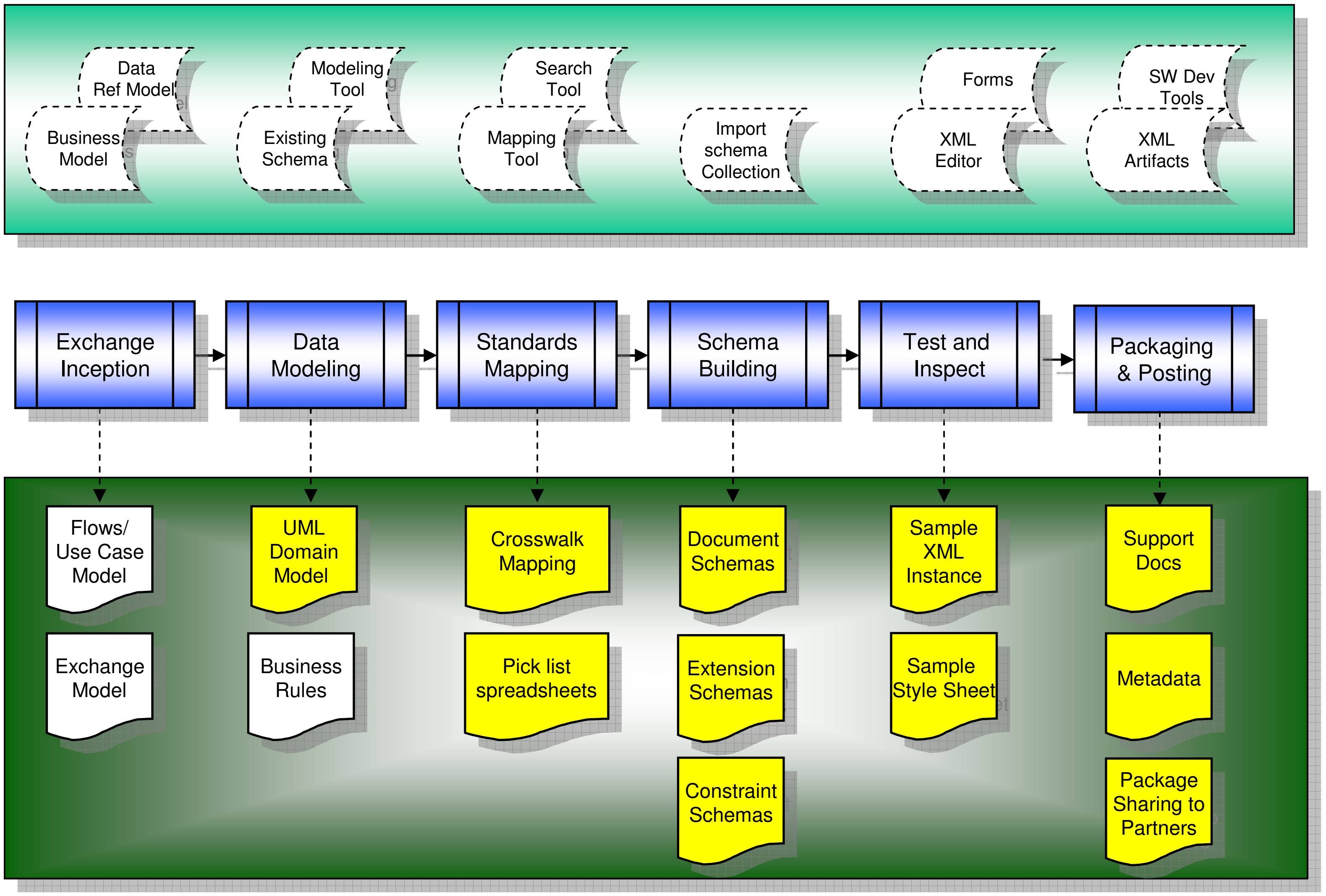
How to do this faster, simpler, more reliably?

Accelerated Process Overview

Current Practice – Conceptual



Current Practice - Mechanics



Current Practice – Team Matrix

FTE = full time equivalent

Skillset	Qualifications	Experience	Resource
Exchange practitioner / project lead	- Proficient in modelling methodology and exchange development requirements	Prior Information Exchange project work	1 FTE
W3C XSD schema expert	- Proficient in complex XSD syntax writing. Familiar with developer support tooling and constraints	One to two years actively writing XSD schema	1 FTE
Domain business analyst	- Familiar with project requirements and business applications and also developing XML based exchanges	One year or more in application area	1 FTE
UML/ data modelling practitioner	- Use of UML diagramming and models. Information modelling	Prior UML based modelling	1 FTE
SW dev tooling user	- Knowledge of SW tools available for target environment.	SW tools training and XML development	1 FTE
XML testing and development	- Creating test environments, working with XML test cases, test data generation	Data analysis and XML content creation	1 FTE
Documentation resources	- Writing documentation and spreadsheets	Technical writer	1 FTE

Pitfalls and Challenges

- Significant amount of manual labor needed to develop all the exchange documenting artifacts and XML related end products
- Multi-discipline team and supporting cast of exchange / XML savvy developers needed
- Disconnect between the software delivery teams' schedule and process and the exchange development team and process; production system not matching what the delivery doc says it does
- Alignment to existing domain Enterprise Data Model (EDM)
- Varying quality of hand checked results and no consistency of technical approach to schema development techniques and re-use of domain components
- Process not repeatable and predictable
- Scalability - differing production XML details across teams, often incompatible across implementations and platforms

Delivery Level of Effort Estimates

Component	Tasks	Timings	Constraints
Collect exchange needs	Model information needs	Weeks	Spiral analysis
Perform XSD schema development with EDM alignment	XSD syntax writing	Weeks	Complex with steep learning curve and limited practitioners.
Documentation of each element	Excel spreadsheet	Weeks	Manual preparation and review
Document domain dictionary mapping (pick list)	Excel spreadsheet	2 to 5 days	Manual preparation and review
Create test cases and examples	Sets of XML instances	Weeks	Manual hand editing of XML from XSD
Perform interoperability testing	Build test environments	Weeks	Test harnesses vary
Create exchange documentation	Word documentation	Weeks	Manual preparation

Currently 800+ hour process for 300+ node exchange

Improving the Process

- Resolving the issues and challenges
- Ensuring consistent results that can be easily reviewed
- Leverage existing dictionary work and repositories of components that the enterprise already has
- Reduce the learning curve and need for specialized skills
- Business analysts not excluded from design, review and implementation by technical barriers
- Lock-step the development process to the exchange
- Customizable and configurable so can adapt to changing requirements

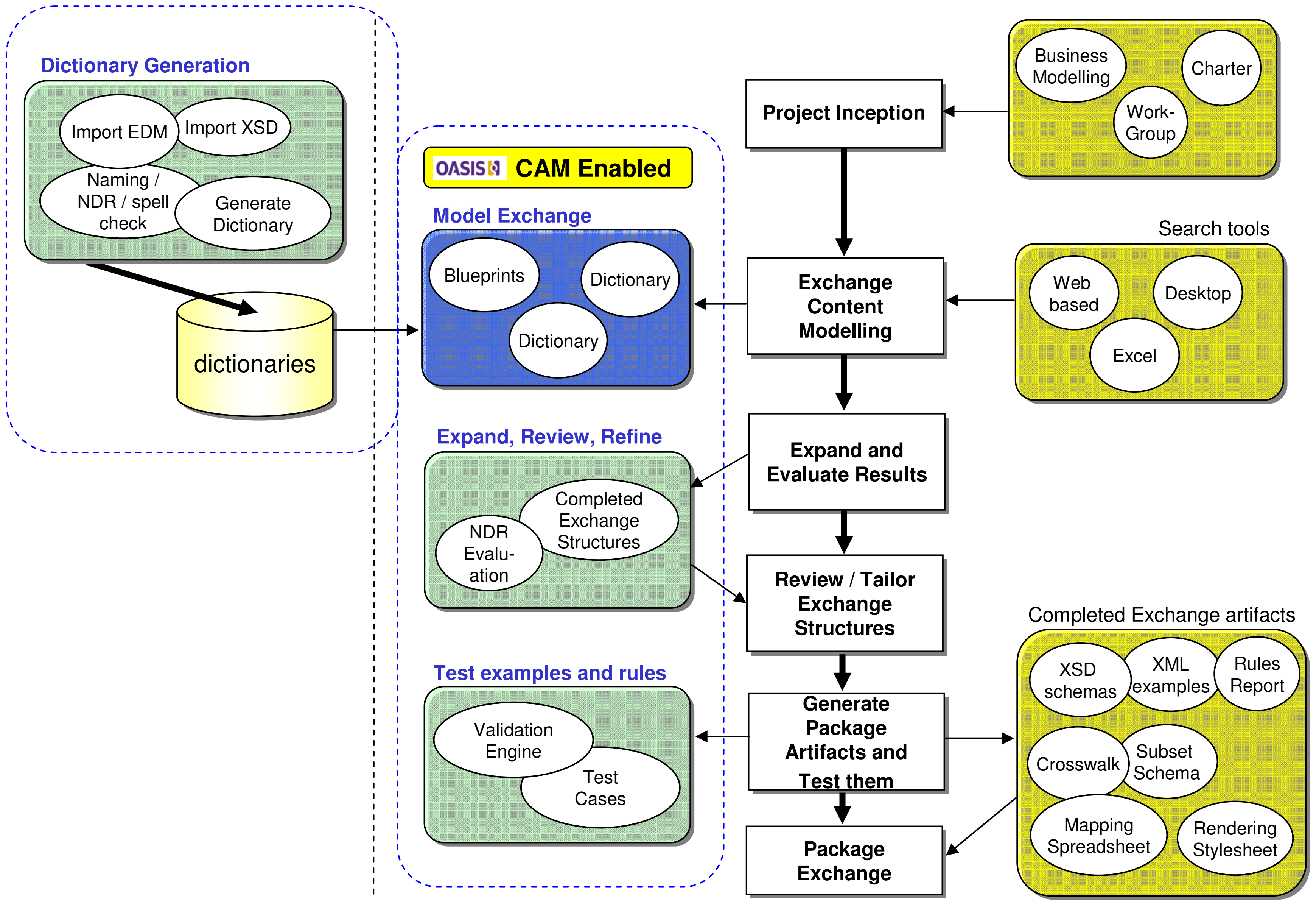
Faster, Simpler, Predictable

- Tooling automates much of the manual tasks; ensures predictable quality of results
- Reduce need for specialized technical knowledge of XSD and XML
- Provide consistent approach that leverages best-practice techniques with built-in smarts and knowledge
- Tooling checks for common pitfalls, applies NDR checks
- Allow business analyst to complete much of the design work and crosscheck application details
- Leverage reuse of domain component dictionaries and blueprints
- Lockstep development to exchange artifacts and their delivery
- Accelerate development tasks (test cases, testing, schema writing)
- Produce result that are neutral to developer tooling platforms
- Process repeatable and replicatable when requirements / versions change

Using Dictionaries & Blueprints

- **Dictionaries** provide reference sets of components to be used in exchanges; three possible sources:
  - Dictionaries imported from existing industry schema
  - Domain dictionary built from an Enterprise Data Model schema
  - Reverse engineered out from existing exchange schema
- **Blueprint**
  - Is the outline of the structure components to be used in an exchange schema
  - Can import components from one or more domain dictionary collections
  - Sketches out the desired information exchange with re-use of existing exchange component structures, plus any local additions / extensions / exclusions
- **Expander** tool reads the blueprint, references the dictionary, and constructs the complete exchange schema

Accelerated Process Overview



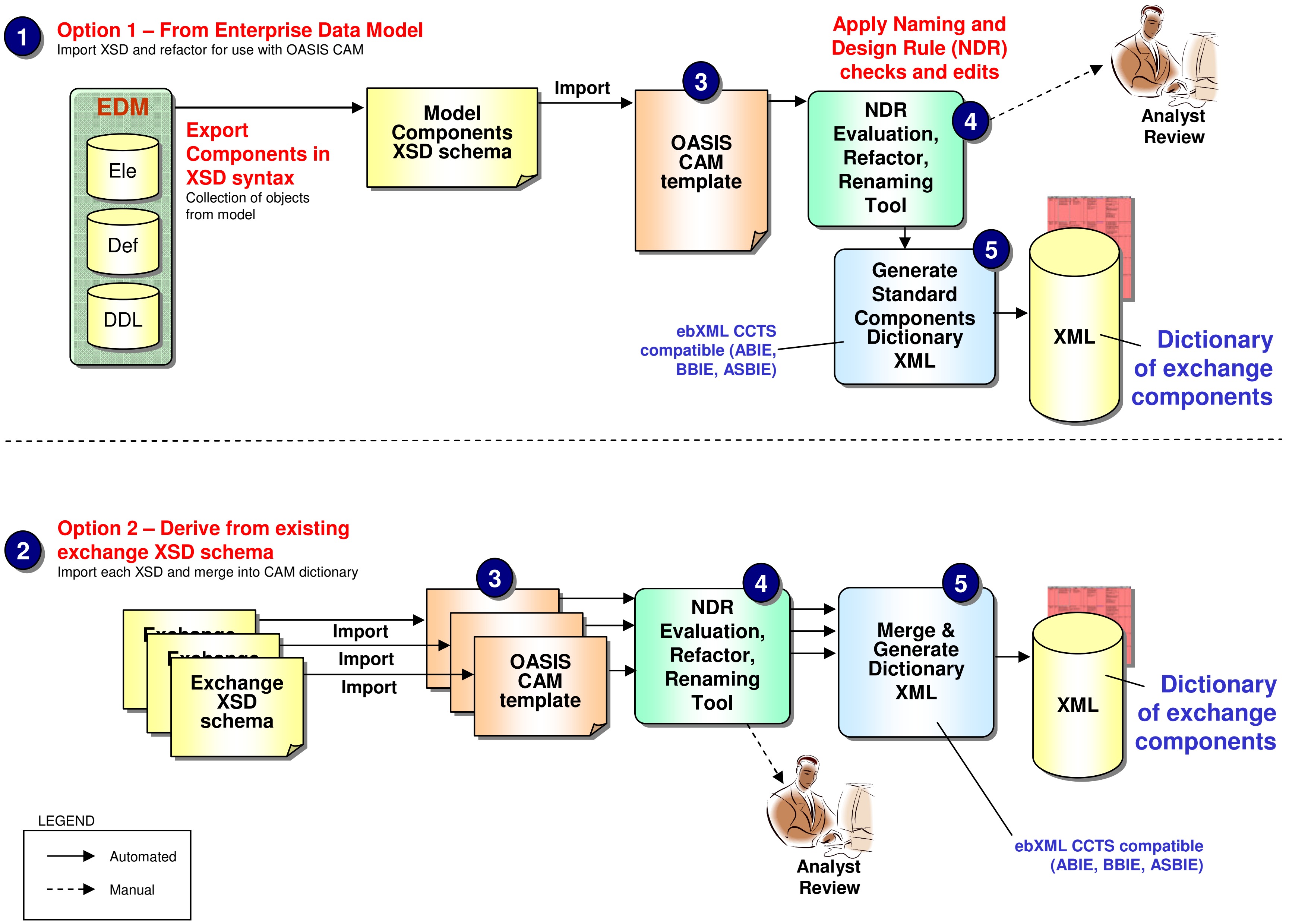
Blueprints and Dictionaries

Leveraging re-use – dictionaries from industry standards

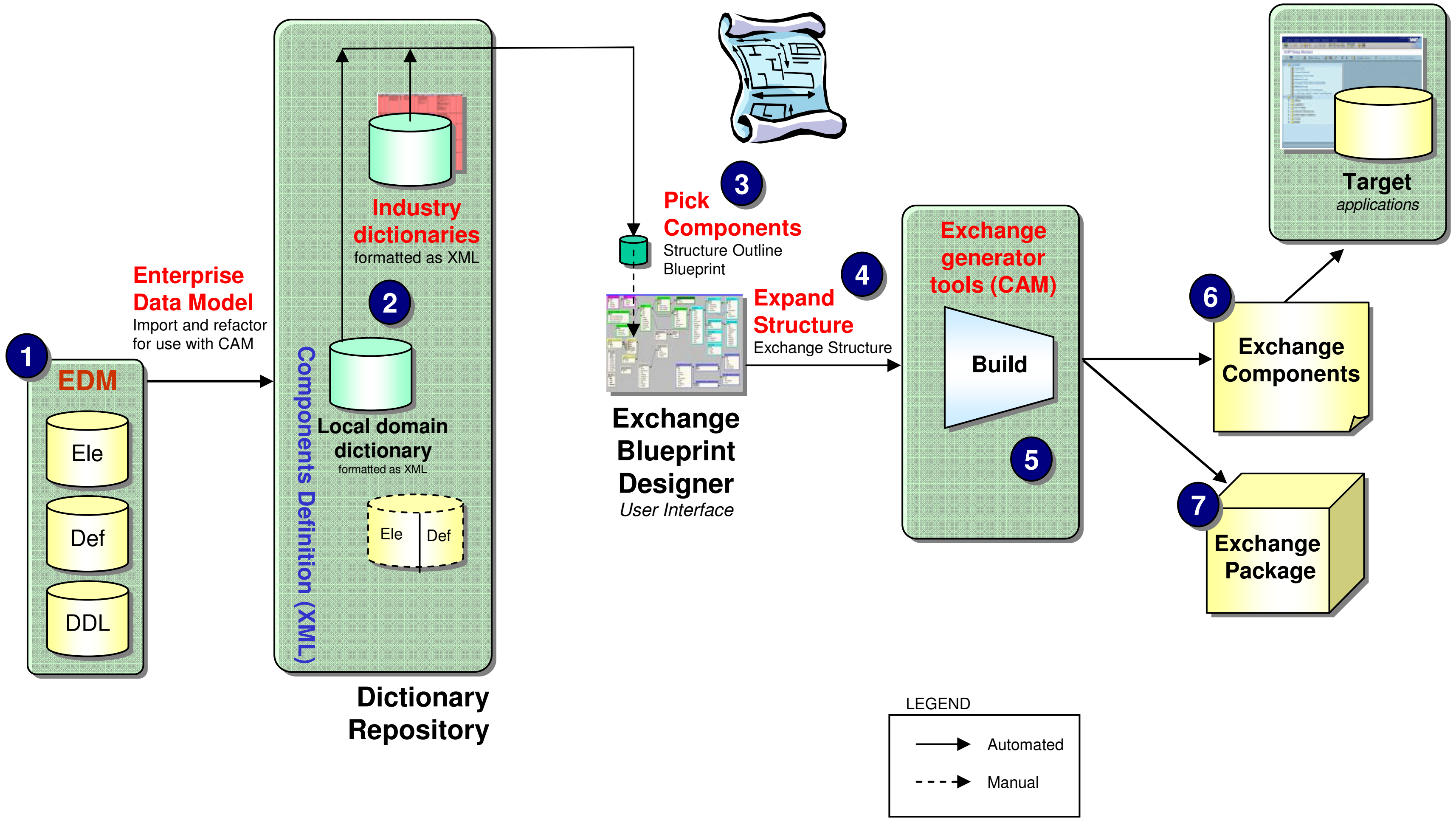
Creating your own domain dictionary from XSD or UML

Constructing your exchange and blueprints

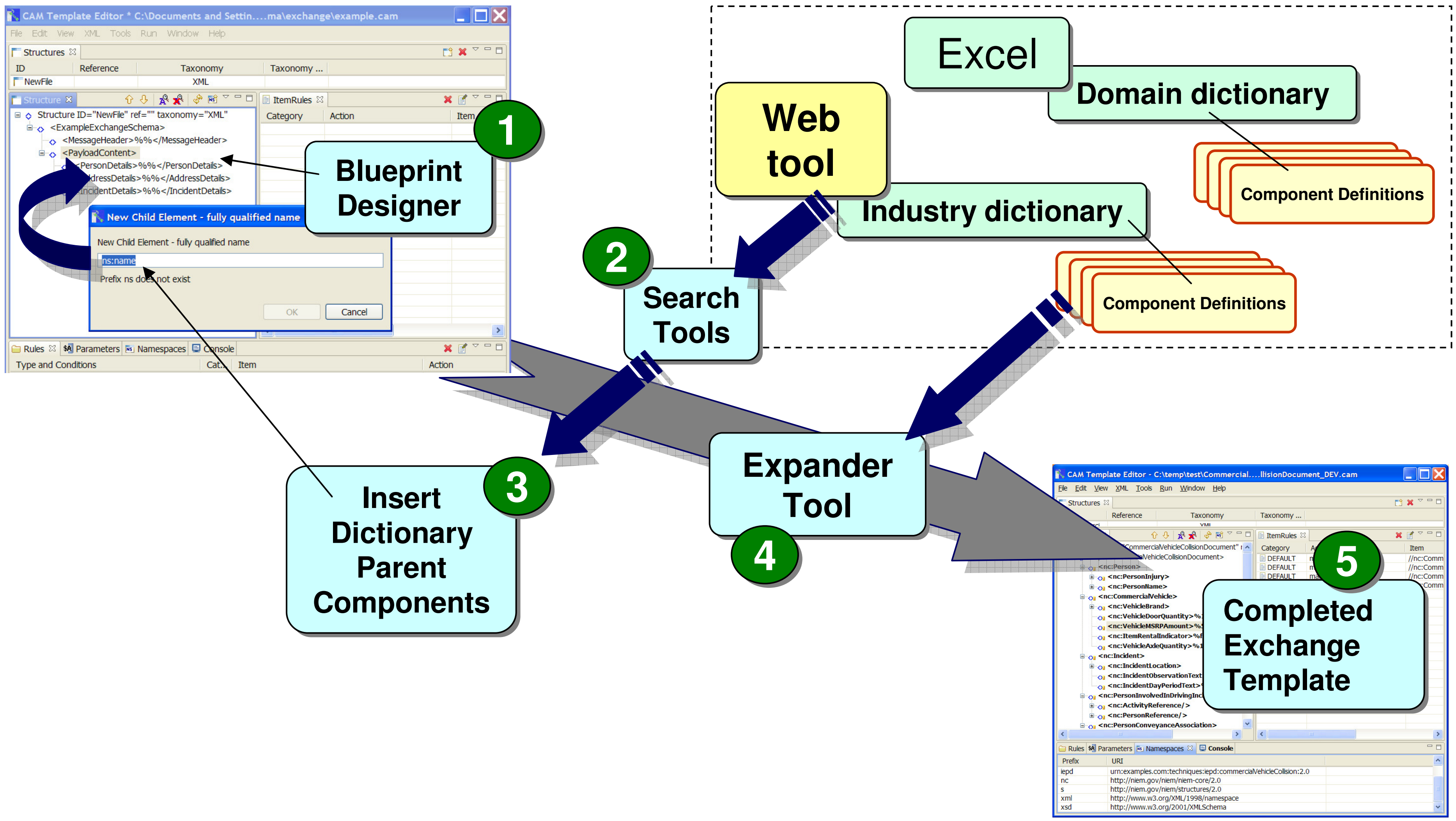
Building Domain Dictionaries


1
Option 1 – From Enterprise Data Model
Import XSD and refactor for use with OASIS CAM
EDM
Ele
Def
DDL
Export Components in XSD syntax
Collection of objects from model
Model Components XSD schema
Import
3
OASIS CAM template
Apply Naming and Design Rule (NDR) checks and edits
NDR Evaluation, Refactor, Renaming Tool
4
Analyst Review
5
Generate Standard Components Dictionary XML
ebXML CCTS compatible (ABIE, BBIE, ASBIE)
XML
Dictionary of exchange components
2
Option 2 – Derive from existing exchange XSD schema
Import each XSD and merge into CAM dictionary
Exchange XSD schema
Import
Import
Import
3
OASIS CAM template
4
NDR Evaluation, Refactor, Renaming Tool
Analyst Review
5
Merge & Generate Dictionary XML
XML
Dictionary of exchange components
ebXML CCTS compatible (ABIE, BBIE, ASBIE)
LEGEND
Automated
Manual

Blueprint Approach Overview



Blueprint Development Tools

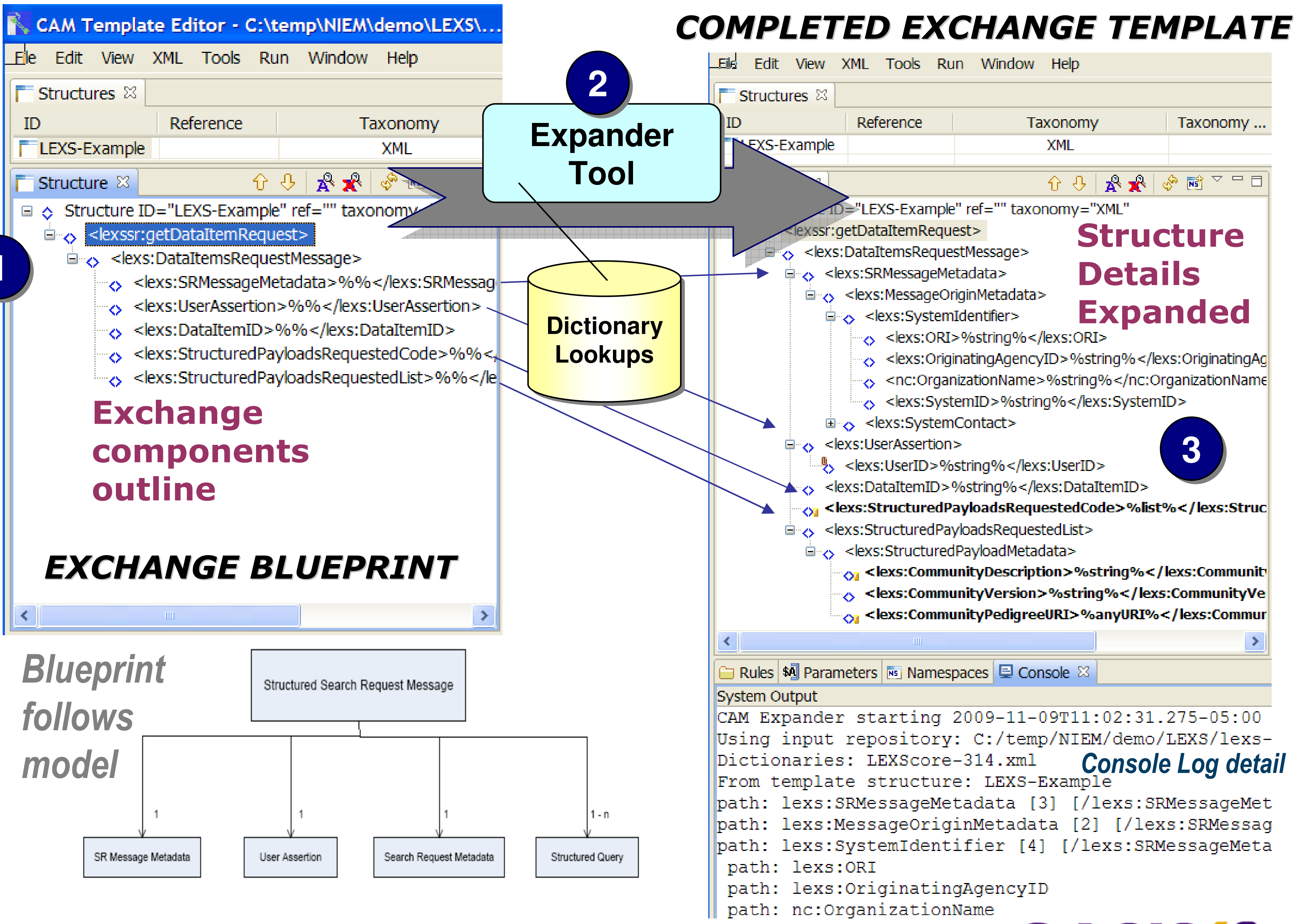

CAM Template Editor * C:\Documents and Settin....ma\exchange\example.cam
File Edit View XML Tools Run Window Help
Structures
ID Reference Taxonomy Taxonomy ...
NewFile XML
Structure
ItemRules
Category Action Item
New Child Element - fully qualified name
ns:name
Prefix ns does not exist
OK
Cancel
Rules Parameters Namespaces Console
Type and Conditions Cat. Item Action
1
Blueprint Designer
2
Search Tools
3
Insert Dictionary Parent Components
4
Expander Tool
5
Completed Exchange Template
Excel
Domain dictionary
Component Definitions
Web tool
Industry dictionary
Component Definitions
CAM Template Editor - C:\temp\test\Commercial....llisionDocument_DEV.cam

```
Structure ID="NewFile" ref="" taxonomy="XML"
  <ExampleExchangeSchema>
    <MessageHeader>%%</MessageHeader>
    <PayloadContent>
      <PersonDetails>%%</PersonDetails>
      <AddressDetails>%%</AddressDetails>
      <IncidentDetails>%%</IncidentDetails>
```

```
<nc:Person>
  <nc:PersonInjury>
  <nc:PersonName>
<nc:CommercialVehicle>
  <nc:VehicleBrand>
  <nc:VehicleDoorQuantity>%1
  <nc:VehicleMSRPAmount>%
  <nc:ItemRentalIndicator>%f
  <nc:VehicleAxleQuantity>%1
<nc:Incident>
  <nc:IncidentLocation>
  <nc:IncidentObservationText
  <nc:IncidentDayPeriodText>
<nc:PersonInvolvedInDrivingInc
  <nc:ActivityReference/>
  <nc:PersonReference/>
<nc:PersonConveyanceAssociation>
```

Prefix	URI
iepd	urn:examples.com:techniques:iepd:commercialVehicleCollision:2.0
nc	http://niem.gov/niem/niem-core/2.0
s	http://niem.gov/niem/structures/2.0
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema

Blueprint Expander Example



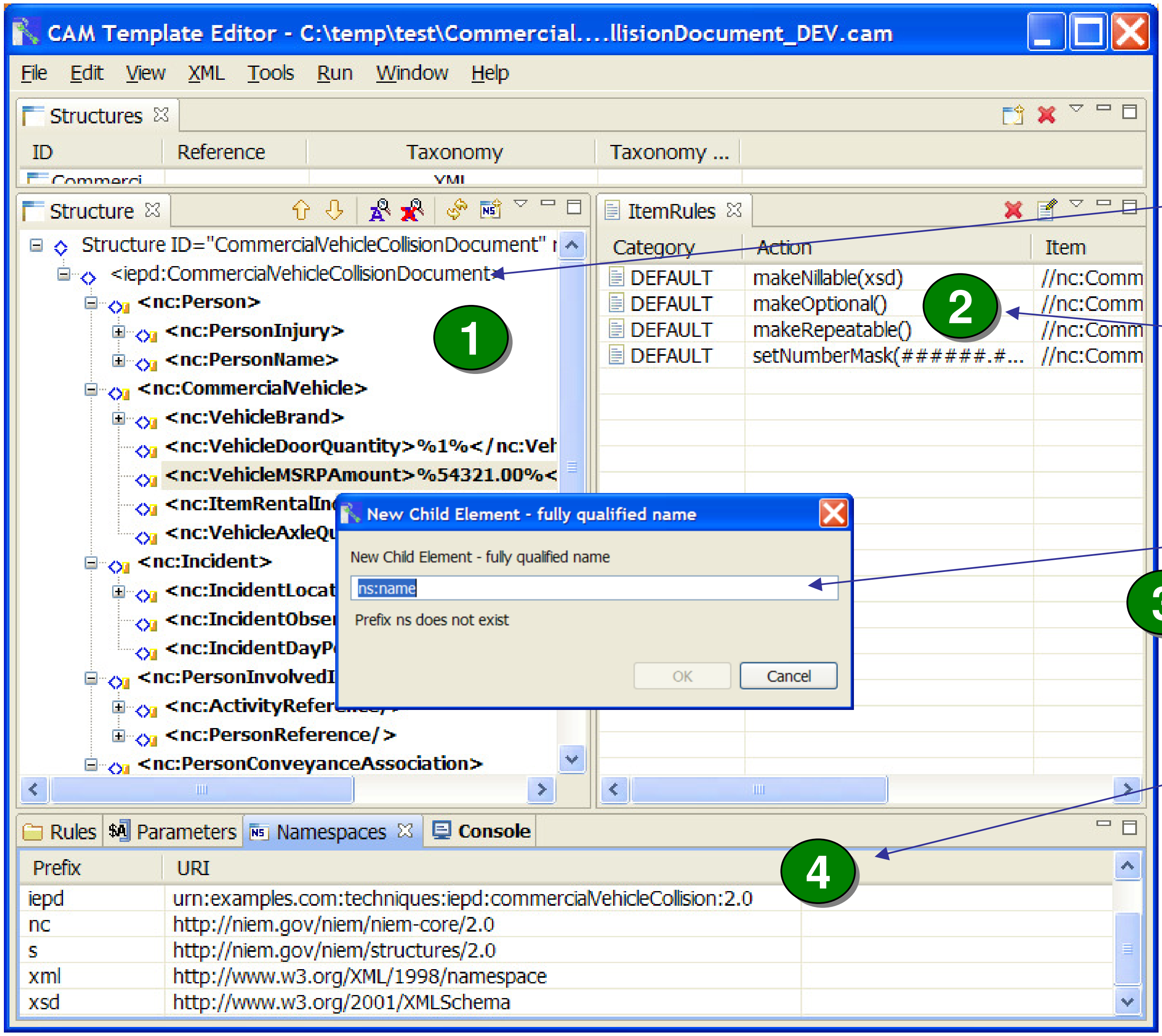
CAM Template Editor - C:\temp\NIEM\demo\LEXS\...
File Edit View XML Tools Run Window Help
Structures
ID Reference Taxonomy
LEXS-Example XML
Structure
1
2
Expander Tool
Dictionary Lookups
3
COMPLETED EXCHANGE TEMPLATE
Structure Details Expanded
Exchange components outline
EXCHANGE BLUEPRINT
Console Log detail
Rules Parameters Namespaces Console
Blueprint follows model
Structured Search Request Message
SR Message Metadata
User Assertion
Search Request Metadata
Structured Query
1
1
1
1 - n

```
Structure ID="LEXS-Example" ref="" taxonomy
  <lexssr:getDataItemRequest>
    <lexs:DataItemsRequestMessage>
      <lexs:SRMessageMetadata>%%</lexs:SRMessag
      <lexs:UserAssertion>%%</lexs:UserAssertion>
      <lexs:DataItemID>%%</lexs:DataItemID>
      <lexs:StructuredPayloadsRequestedCode>%%<
      <lexs:StructuredPayloadsRequestedList>%%</le
```

```
ID="LEXS-Example" ref="" taxonomy="XML"
<lexssr:getDataItemRequest>
  <lexs:DataItemsRequestMessage>
    <lexs:SRMessageMetadata>
      <lexs:MessageOriginMetadata>
        <lexs:SystemIdentifier>
          <lexs:ORI>%string%</lexs:ORI>
          <lexs:OriginatingAgencyID>%string%</lexs:OriginatingAg
          <nc:OrganizationName>%string%</nc:OrganizationName
          <lexs:SystemID>%string%</lexs:SystemID>
        <lexs:SystemContact>
    <lexs:UserAssertion>
      <lexs:UserID>%string%</lexs:UserID>
    <lexs:DataItemID>%string%</lexs:DataItemID>
    <lexs:StructuredPayloadsRequestedCode>%list%</lexs:Struc
    <lexs:StructuredPayloadsRequestedList>
      <lexs:StructuredPayloadMetadata>
        <lexs:CommunityDescription>%string%</lexs:Communit
        <lexs:CommunityVersion>%string%</lexs:CommunityVe
        <lexs:CommunityPedigreeURI>%anyURI%</lexs:Commur
```

```
System Output
CAM Expander starting 2009-11-09T11:02:31.275-05:00
Using input repository: C:/temp/NIEM/demo/LEXS/lexs-
Dictionaries: LEXScore-314.xml
From template structure: LEXS-Example
path: lexs:SRMessageMetadata [3] [/lexs:SRMessageMet
path: lexs:MessageOriginMetadata [2] [/lexs:SRMessag
path: lexs:SystemIdentifier [4] [/lexs:SRMessageMeta
 path: lexs:ORI
 path: lexs:OriginatingAgencyID
 path: nc:OrganizationName
```

Exchange Template Editor



CAM Template Editor - C:\temp\test\Commercial....llisionDocument_DEV.cam

File Edit View XML Tools Run Window Help
Structures
Structure
ItemRules
1
2
3
4
Rules Parameters Namespaces Console

ID	Reference	Taxonomy	Taxonomy ...
Commerci		XML	

```
Structure ID="CommercialVehicleCollisionDocument"
  <iepd:CommercialVehicleCollisionDocument>
    <nc:Person>
      <nc:PersonInjury>
      <nc:PersonName>
    <nc:CommercialVehicle>
      <nc:VehicleBrand>
      <nc:VehicleDoorQuantity>%1%</nc:Veh
      <nc:VehicleMSRPAmount>%54321.00%<
      <nc:ItemRentalIn
      <nc:VehicleAxleQu
    <nc:Incident>
      <nc:IncidentLocat
      <nc:IncidentObser
      <nc:IncidentDayP
    <nc:PersonInvolvedI
      <nc:ActivityRefer
      <nc:PersonReference/>
    <nc:PersonConveyanceAssociation>
```

Category	Action	Item
DEFAULT	makeNillable(xsd)	//nc:Comm
DEFAULT	makeOptional()	//nc:Comm
DEFAULT	makeRepeatable()	//nc:Comm
DEFAULT	setNumberMask(######.#...	//nc:Comm

New Child Element - fully qualified name

New Child Element - fully qualified name

ns:name

Prefix ns does not exist

OK Cancel

Prefix	URI
iepd	urn:examples.com:techniques:iepd:commercialVehicleCollision:2.0
nc	http://niem.gov/niem/niem-core/2.0
s	http://niem.gov/niem/structures/2.0
xml	http://www.w3.org/XML/1998/namespace
xsd	http://www.w3.org/2001/XMLSchema

Exchange Structure items

Rules for each item

Can add new custom domain Items

Domain Components Namespaces

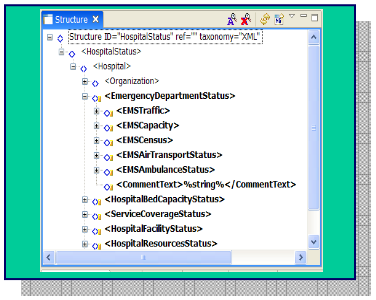
Generating Exchange Artifacts

NDR evaluation, crosswalk mapping, Exchange Schema, Subset schema, XML instances, business rules documentation

Exchange Generation Steps

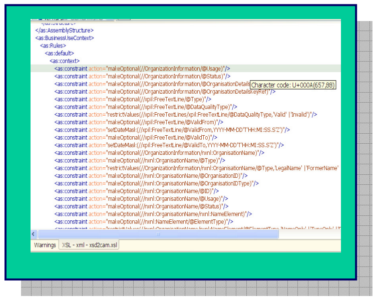
EXCHANGE TEMPLATE

Structure

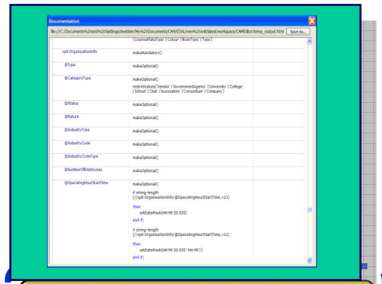


```
Structure ID="HospitalStatus" ref="" taxonomy="XML"
  <HospitalStatus>
    <Hospital>
      <Organization>
      <EmergencyDepartmentStatus>
        <EMSTraffic>
        <EMSCapacity>
        <EMSCensus>
        <EMSAirTransportStatus>
        <EMSAmbulanceStatus>
        <CommentText>%string%</CommentText>
      <HospitalBedCapacityStatus>
      <ServiceCoverageStatus>
      <HospitalFacilityStatus>
      <HospitalResourcesStatus>
```

Rules



Documentation

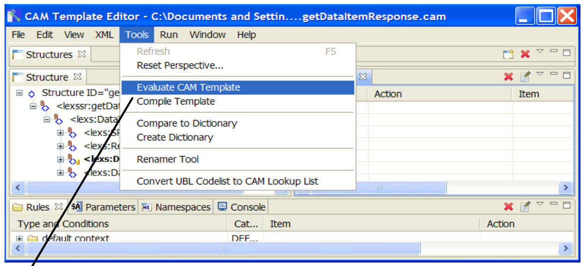


OASIS CAM Template

Toolkit for exchange artifacts generation

Suite of menu options and tools in desktop CAM toolkit editor

(each run custom xslt scripts on the exchange CAM template xml)



1 **Run template Evaluation Report**

2 **Compare to industry dictionary**

- create mapping spreadsheet

- create crosswalk xml

3 **Generate business rules report**

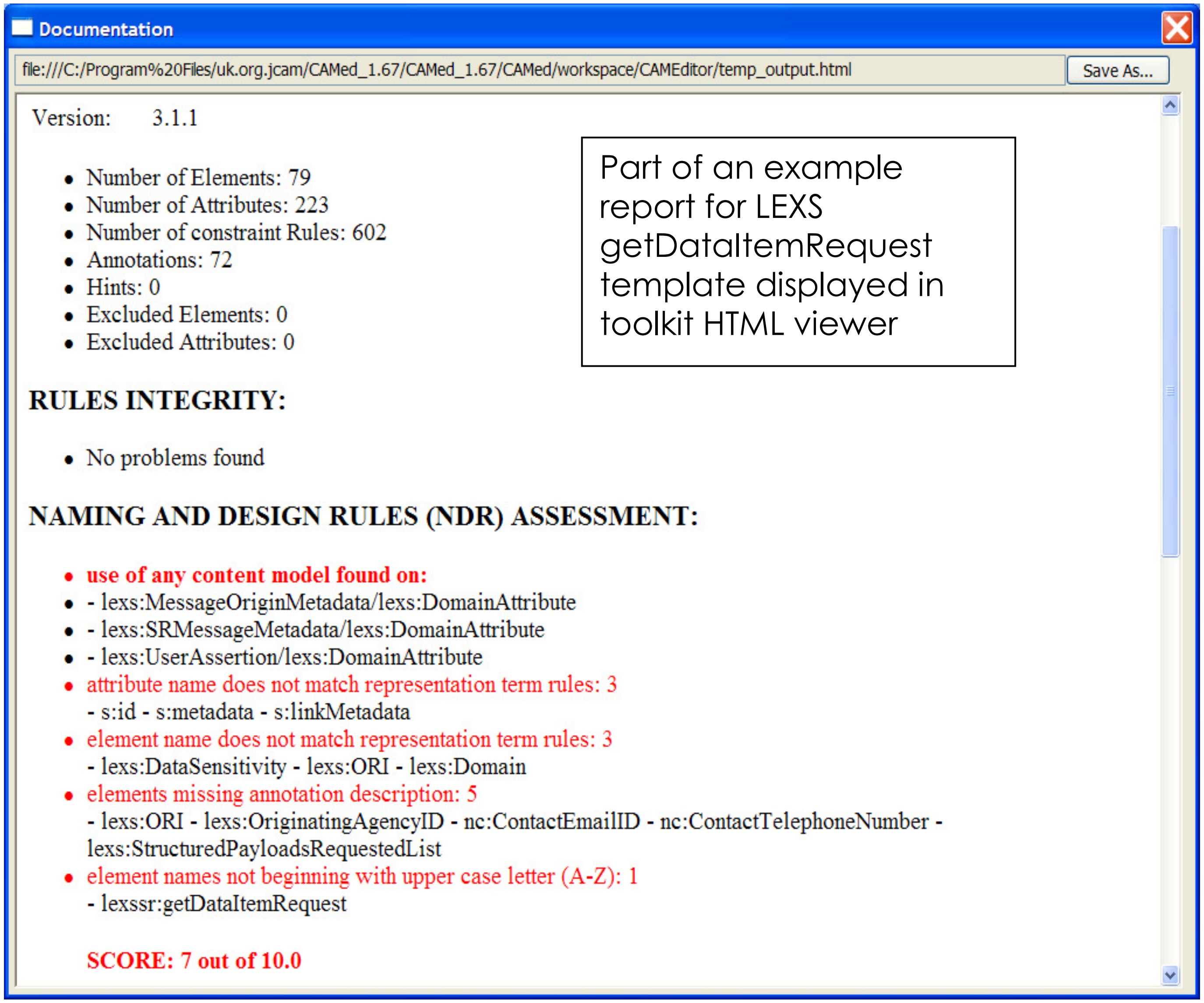
4 **Generate exchange XSD schema**

5 **Generate XML test instance(s)**

NDR Evaluation Report

- Provides scoring, alerts, warnings and potential issues including:
  - Naming and content model conventions
  - Naming and restriction consistency checks
  - Interoperability enablers/inhibitors checks
  - Rules integrity and duplicates
  - Statistics on exchange size
  - Spell checking on component names

Evaluation Report (NDR) example


Documentation
file:///C:/Program%20Files/uk.org.jcam/CAMed_1.67/CAMed_1.67/CAMed/workspace/CAMEditor/temp_output.html
Save As...

Version: 3.1.1

- Number of Elements: 79
- Number of Attributes: 223
- Number of constraint Rules: 602
- Annotations: 72
- Hints: 0
- Excluded Elements: 0
- Excluded Attributes: 0

RULES INTEGRITY:

- No problems found

NAMING AND DESIGN RULES (NDR) ASSESSMENT:

- **use of any content model found on:**
- - lexs:MessageOriginMetadata/lexs:DomainAttribute
- - lexs:SRMessageMetadata/lexs:DomainAttribute
- - lexs:UserAssertion/lexs:DomainAttribute
- attribute name does not match representation term rules: 3
  - s:id - s:metadata - s:linkMetadata
- element name does not match representation term rules: 3
  - lexs:DataSensitivity - lexs:ORI - lexs:Domain
- elements missing annotation description: 5
  - lexs:ORI - lexs:OriginatingAgencyID - nc:ContactEmailID - nc:ContactTelephoneNumber - lexs:StructuredPayloadsRequestedList
- element names not beginning with upper case letter (A-Z): 1
  - lexssr:getDataItemRequest

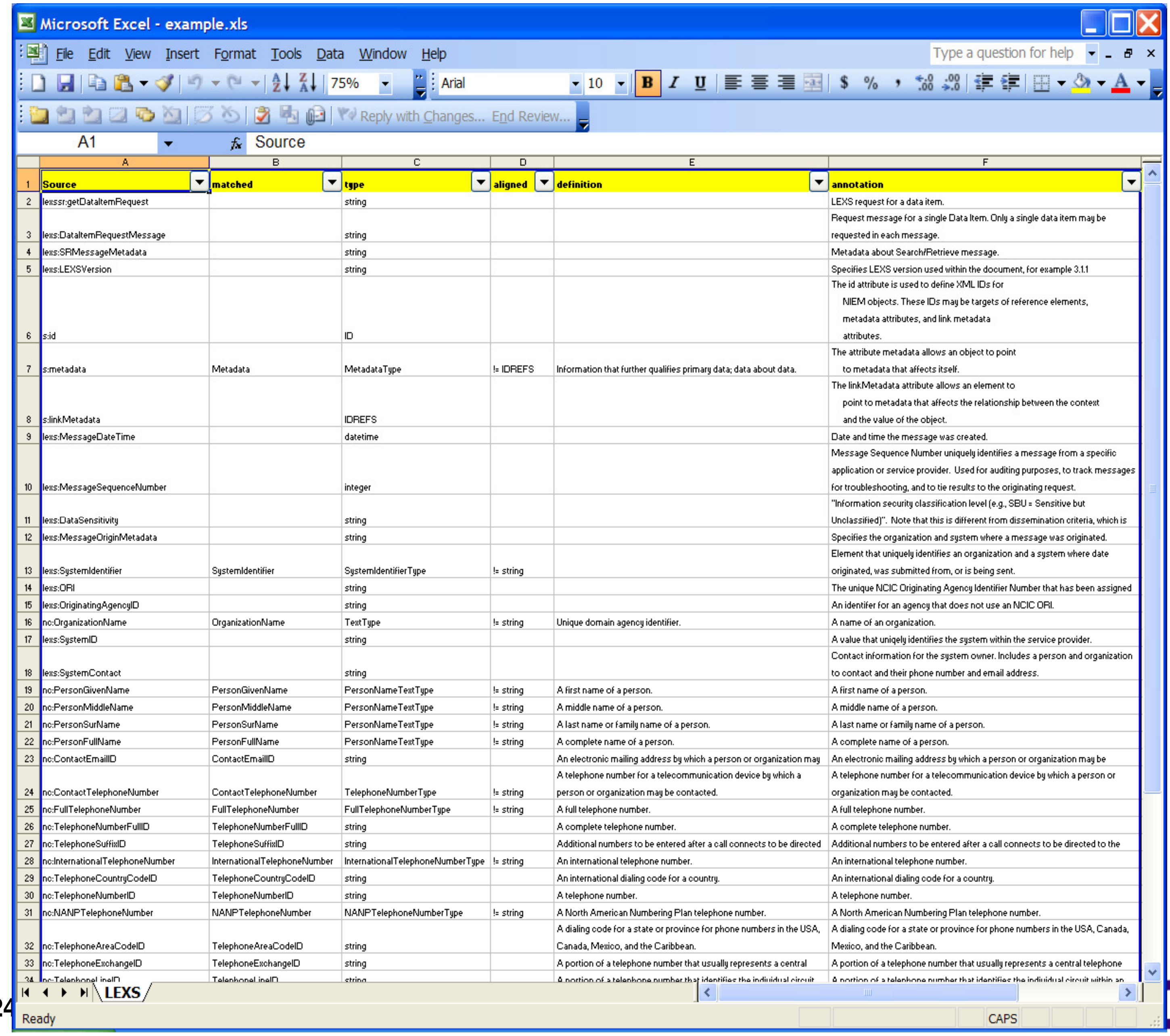
SCORE: 7 out of 10.0

Part of an example report for LEXS getDataItemRequest template displayed in toolkit HTML viewer

Compare to industry dictionary

- References industry dictionary of names and properties
- Matches on physical names
- Reports mapping details
- Compatible with Microsoft Excel
- Report can be used to do spell checking
- Generates crosswalk xml file

Example cross-reference spreadsheet


Microsoft Excel - example.xls

A1 Source

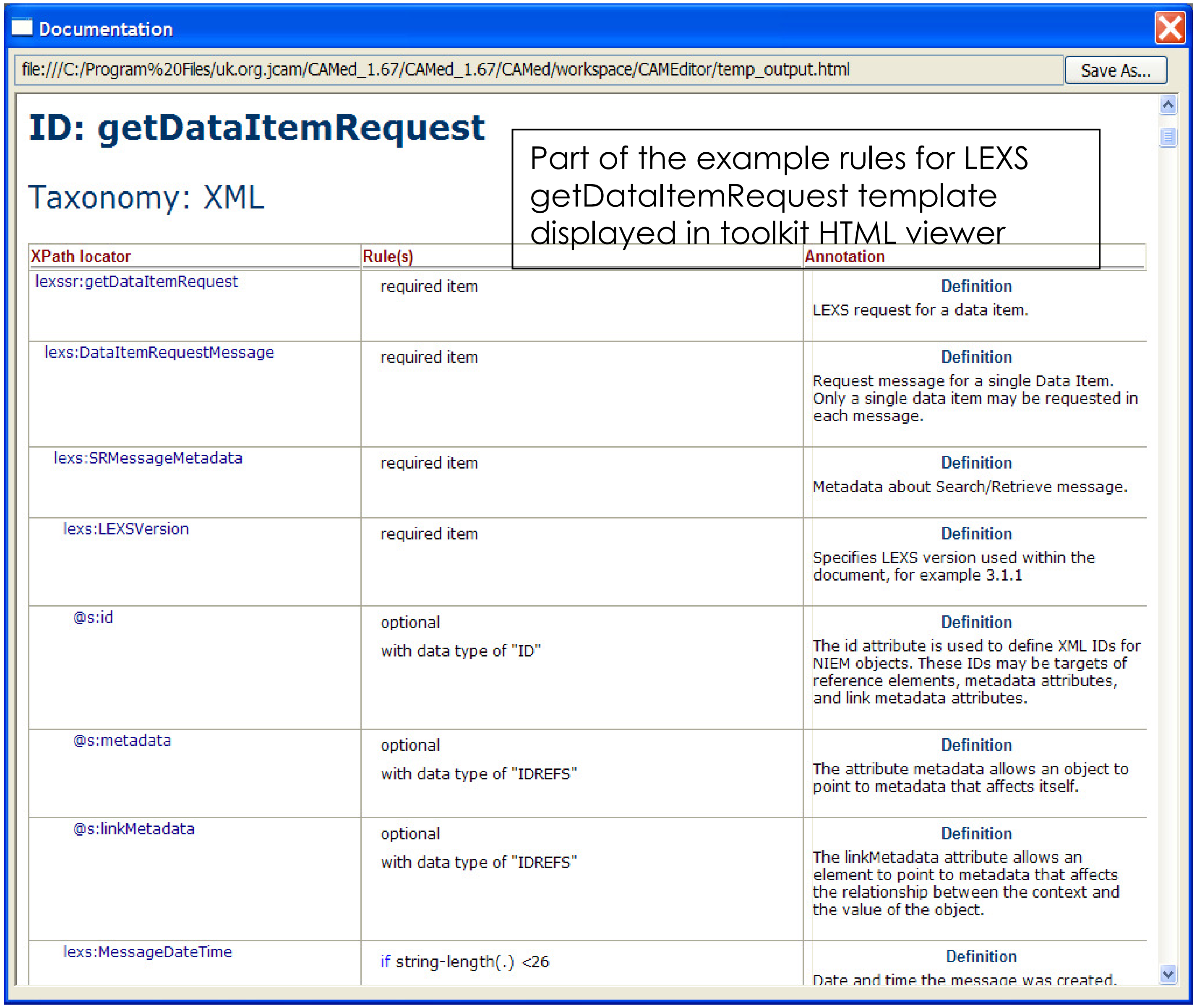
	Source	matched	type	aligned	definition	annotation
2	lexssr:getDataItemRequest		string			LEXS request for a data item.
3	lexs:DataItemRequestMessage		string			Request message for a single Data Item. Only a single data item may be requested in each message.
4	lexs:SRMessageMetadata		string			Metadata about Search/Retrieve message.
5	lexs:LEXSVersion		string			Specifies LEXS version used within the document, for example 3.1.1
6	s:id		ID			The id attribute is used to define XML IDs for NIEM objects. These IDs may be targets of reference elements, metadata attributes, and link metadata attributes.
7	s:metadata	Metadata	MetadataType	!= IDREFS	Information that further qualifies primary data; data about data.	The attribute metadata allows an object to point to metadata that affects itself.
8	s:linkMetadata		IDREFS			The linkMetadata attribute allows an element to point to metadata that affects the relationship between the context and the value of the object.
9	lexs:MessageDateTime		datetime			Date and time the message was created.
10	lexs:MessageSequenceNumber		integer			Message Sequence Number uniquely identifies a message from a specific application or service provider. Used for auditing purposes, to track messages for troubleshooting, and to tie results to the originating request.
11	lexs:DataSensitivity		string			"Information security classification level (e.g., SBU = Sensitive but Unclassified)". Note that this is different from dissemination criteria, which is
12	lexs:MessageOriginMetadata		string			Specifies the organization and system where a message was originated.
13	lexs:SystemIdentifier	SystemIdentifier	SystemIdentifierType	!= string		Element that uniquely identifies an organization and a system where date originated, was submitted from, or is being sent.
14	lexs:ORI		string			The unique NCIC Originating Agency Identifier Number that has been assigned
15	lexs:OriginatingAgencyID		string			An identifer for an agency that does not use an NCIC ORI.
16	nc:OrganizationName	OrganizationName	TextType	!= string	Unique domain agency identifier.	A name of an organization.
17	lexs:SystemID		string			A value that uniqely identifies the system within the service provider.
18	lexs:SystemContact		string			Contact information for the system owner. Includes a person and organization to contact and their phone number and email address.
19	nc:PersonGivenName	PersonGivenName	PersonNameTextType	!= string	A first name of a person.	A first name of a person.
20	nc:PersonMiddleName	PersonMiddleName	PersonNameTextType	!= string	A middle name of a person.	A middle name of a person.
21	nc:PersonSurName	PersonSurName	PersonNameTextType	!= string	A last name or family name of a person.	A last name or family name of a person.
22	nc:PersonFullName	PersonFullName	PersonNameTextType	!= string	A complete name of a person.	A complete name of a person.
23	nc:ContactEmailID	ContactEmailID	string		An electronic mailing address by which a person or organization may	An electronic mailing address by which a person or organization may be
24	nc:ContactTelephoneNumber	ContactTelephoneNumber	TelephoneNumberType	!= string	A telephone number for a telecommunication device by which a person or organization may be contacted.	A telephone number for a telecommunication device by which a person or organization may be contacted.
25	nc:FullTelephoneNumber	FullTelephoneNumber	FullTelephoneNumberType	!= string	A full telephone number.	A full telephone number.
26	nc:TelephoneNumberFullID	TelephoneNumberFullID	string		A complete telephone number.	A complete telephone number.
27	nc:TelephoneSuffixID	TelephoneSuffixID	string		Additional numbers to be entered after a call connects to be directed	Additional numbers to be entered after a call connects to be directed to the
28	nc:InternationalTelephoneNumber	InternationalTelephoneNumber	InternationalTelephoneNumberType	!= string	An international telephone number.	An international telephone number.
29	nc:TelephoneCountryCodeID	TelephoneCountryCodeID	string		An international dialing code for a country.	An international dialing code for a country.
30	nc:TelephoneNumberID	TelephoneNumberID	string		A telephone number.	A telephone number.
31	nc:NANPTelephoneNumber	NANPTelephoneNumber	NANPTelephoneNumberType	!= string	A North American Numbering Plan telephone number.	A North American Numbering Plan telephone number.
32	nc:TelephoneAreaCodeID	TelephoneAreaCodeID	string		A dialing code for a state or province for phone numbers in the USA, Canada, Mexico, and the Caribbean.	A dialing code for a state or province for phone numbers in the USA, Canada, Mexico, and the Caribbean.
33	nc:TelephoneExchangeID	TelephoneExchangeID	string		A portion of a telephone number that usually represents a central	A portion of a telephone number that usually represents a central telephone
34	nc:TelephoneLineID	TelephoneLineID	string		A portion of a telephone number that identifies the individual circuit	A portion of a telephone number that identifies the individual circuit within an

Formatted view in Microsoft Excel of import of cross-reference report details (from generated XML file)

Generate Documentation and Schema

- Documentation:
  - Create HTML report of exchange schema details and associated content and business rules
  - Report layout and content designed to be reviewed by business analysts
- Schema:
  - Generate XSD schema for exchange
  - Customizable exchange folder layout management by namespace for extension, subset and exchange schema components
  - Writes XSD schema in syntax that is clear, simple and compatible with deployment tooling environments

Business Rules Documentation



Documentation

file:///C:/Program%20Files/uk.org.jcam/CAMed_1.67/CAMed_1.67/CAMed/workspace/CAMEditor/temp_output.html

Save As...

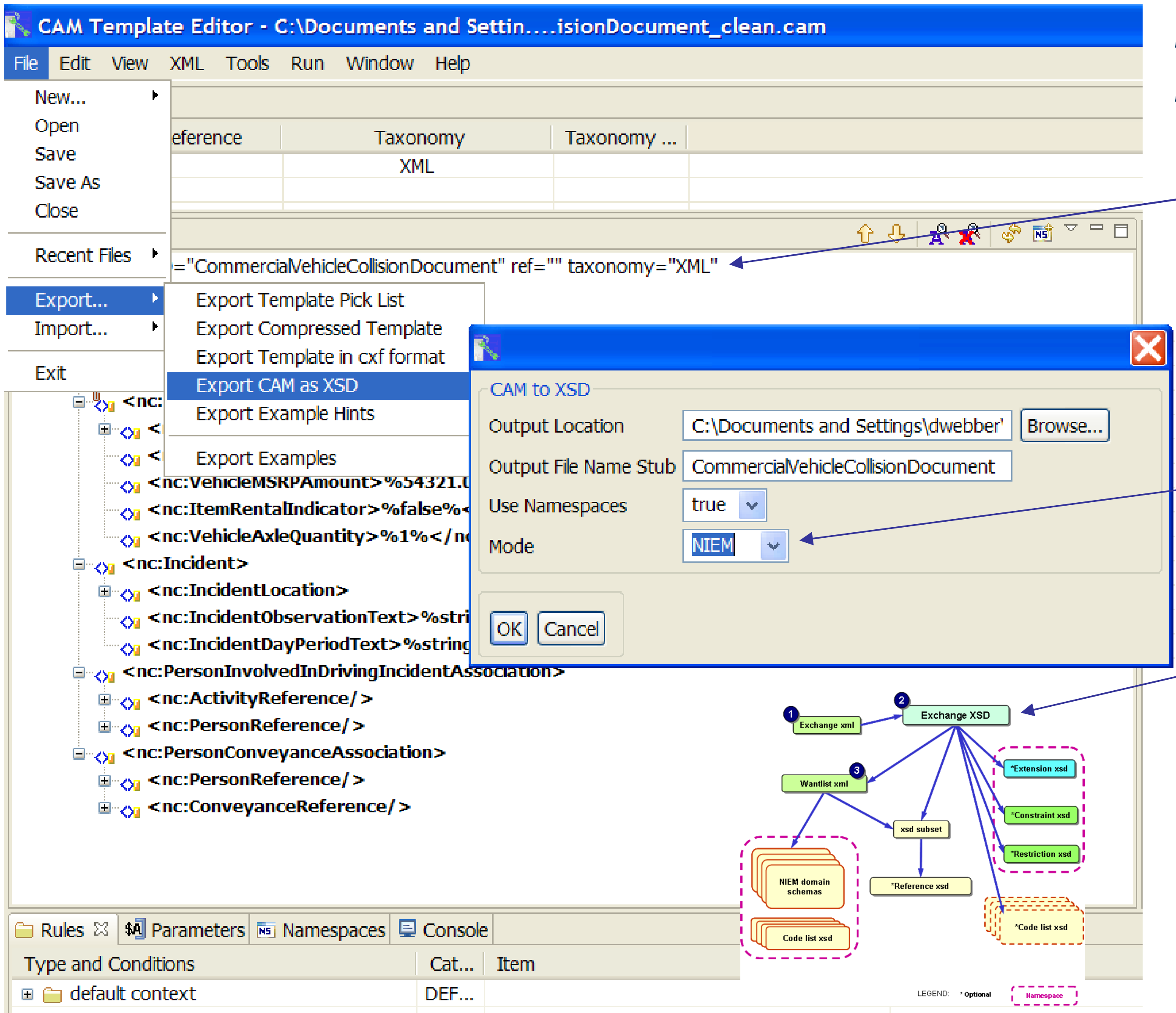
ID: getDataItemRequest

Taxonomy: XML

Part of the example rules for LEXS getDataItemRequest template displayed in toolkit HTML viewer

XPath locator	Rule(s)	Annotation
lexssr:getDataItemRequest	required item	**Definition** LEXS request for a data item.
lexs:DataItemRequestMessage	required item	**Definition** Request message for a single Data Item. Only a single data item may be requested in each message.
lexs:SRMessageMetadata	required item	**Definition** Metadata about Search/Retrieve message.
lexs:LEXSVersion	required item	**Definition** Specifies LEXS version used within the document, for example 3.1.1
@s:id	optional with data type of "ID"	**Definition** The id attribute is used to define XML IDs for NIEM objects. These IDs may be targets of reference elements, metadata attributes, and link metadata attributes.
@s:metadata	optional with data type of "IDREFS"	**Definition** The attribute metadata allows an object to point to metadata that affects itself.
@s:linkMetadata	optional with data type of "IDREFS"	**Definition** The linkMetadata attribute allows an element to point to metadata that affects the relationship between the context and the value of the object.
lexs:MessageDateTime	if string-length(.) <26	**Definition** Date and time the message was created.

Export Exchange to XSD Schema



eference	Taxonomy	Taxonomy ...
	XML	

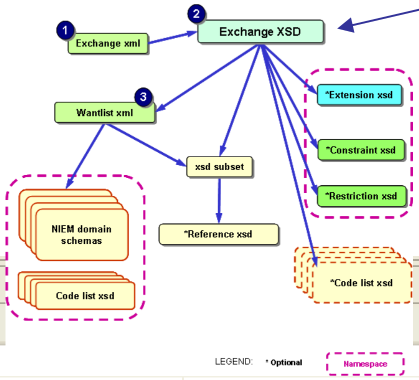
="CommercialVehicleCollisionDocument" ref="" taxonomy="XML"

```
<nc:
  <
  <
  <nc:VehicleMSRPAmount>%54321.0
  <nc:ItemRentalIndicator>%false%<
  <nc:VehicleAxleQuantity>%1%</nc
<nc:Incident>
  <nc:IncidentLocation>
  <nc:IncidentObservationText>%stri
  <nc:IncidentDayPeriodText>%string
<nc:PersonInvolvedInDrivingIncidentAssociation>
  <nc:ActivityReference/>
  <nc:PersonReference/>
<nc:PersonConveyanceAssociation>
  <nc:PersonReference/>
  <nc:ConveyanceReference/>
```

CAM to XSD

Output Location	C:\Documents and Settings\dwebber'
Output File Name Stub	CommercialVehicleCollisionDocument
Use Namespaces	true
Mode	NIEM

Type and Conditions	Cat...	Item
default context	DEF...	



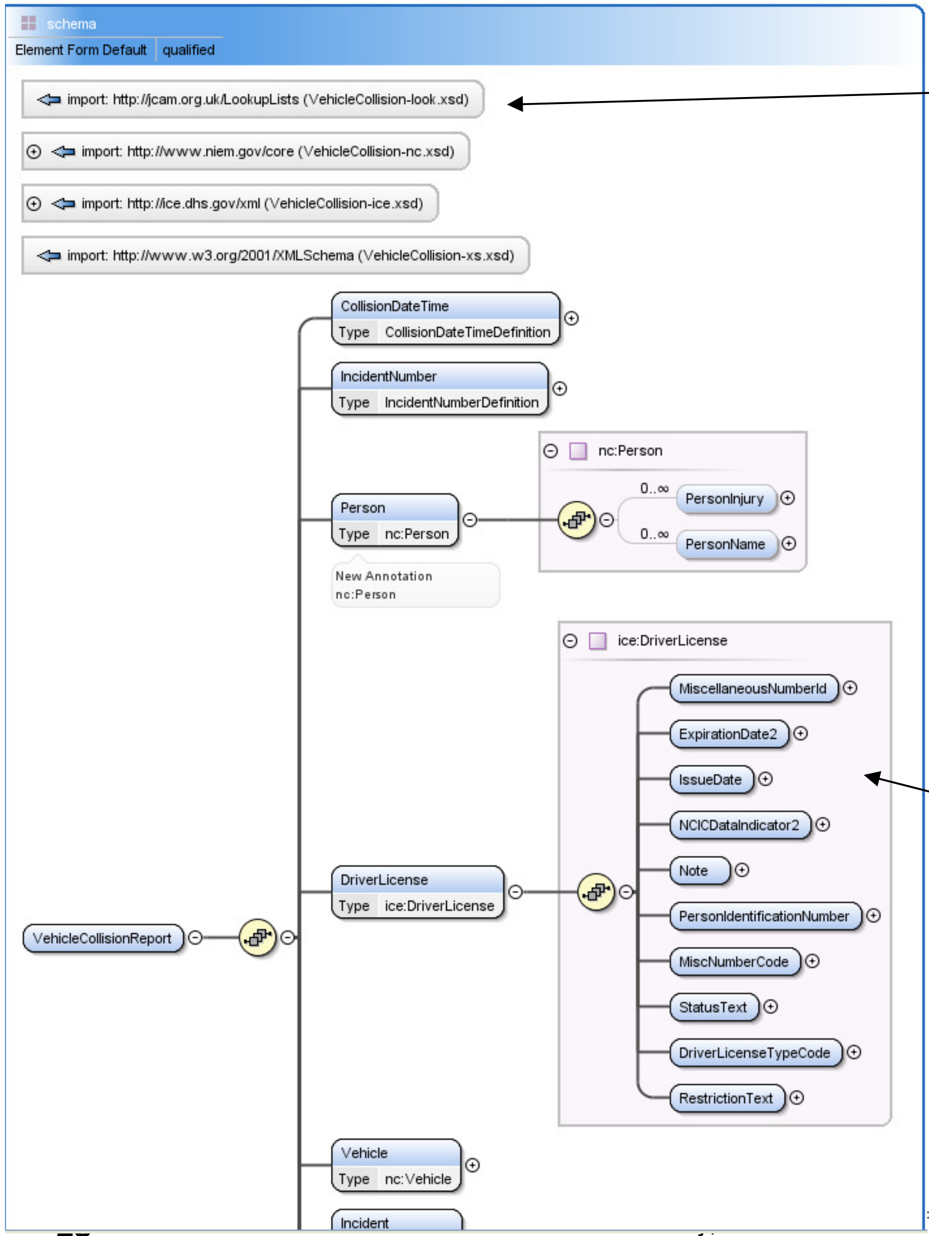
Export Template to Exchange XSD:

Completed Exchange Structure

Set Exchange Options

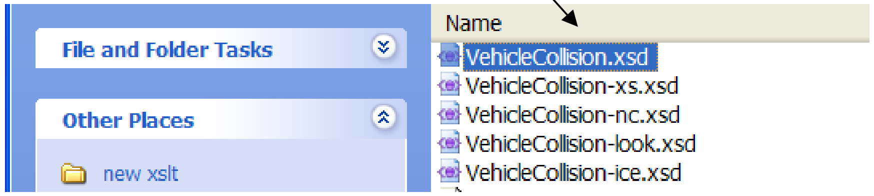
Complete set of exchange schemas generated

Exchange Schema Generated


schema
Element Form Default qualified
import: http://jcam.org.uk/LookupLists (VehicleCollision-look.xsd)
import: http://www.niem.gov/core (VehicleCollision-nc.xsd)
import: http://ice.dhs.gov/xml (VehicleCollision-ice.xsd)
import: http://www.w3.org/2001/XMLSchema (VehicleCollision-xs.xsd)
CollisionDateTime
Type CollisionDateTimeDefinition
IncidentNumber
Type IncidentNumberDefinition
nc:Person
0..∞
PersonInjury
0..∞
PersonName
Person
Type nc:Person
New Annotation
nc:Person
ice:DriverLicense
MiscellaneousNumberId
ExpirationDate2
IssueDate
NCICDataIndicator2
Note
PersonIdentificationNumber
MiscNumberCode
StatusText
DriverLicenseTypeCode
RestrictionText
DriverLicense
Type ice:DriverLicense
VehicleCollisionReport
Vehicle
Type nc:Vehicle
Incident

Each namespace file is import for those specific type definitions

Set of XSD files with filename and namespace suffix

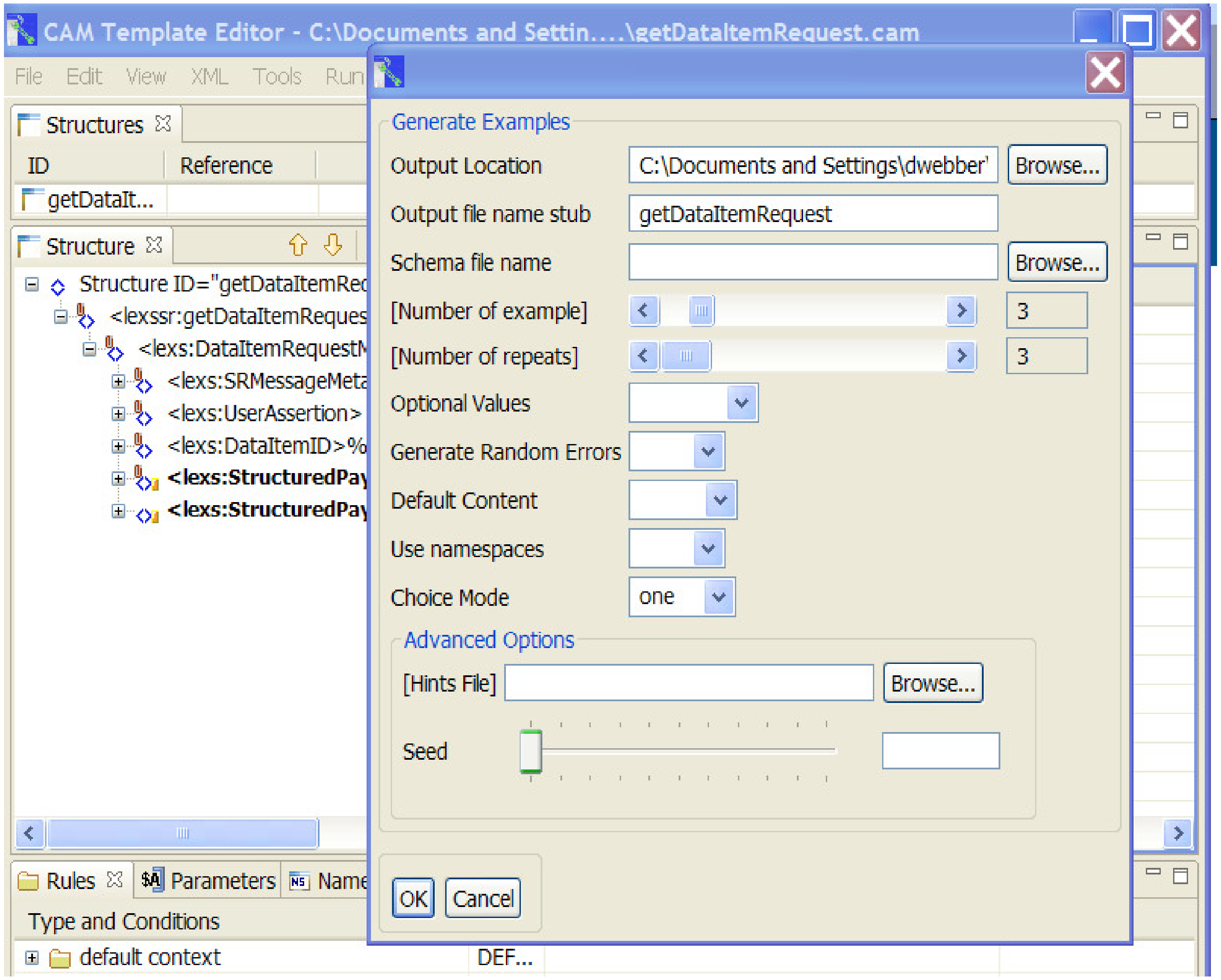


Reviewing XSD results in a schema editor tool

XML Testing Examples Generation

- Support for software development testing process
- Designed to allow creation of concrete realistic examples not just random value based
- Hinting system allows insertion of actual test system values into XML examples
- Can create both valid and invalid examples to support unit testing of application software
- Exclude capability allows generator to create examples that contain only a portion of the entire exchange
- Control over random seed value used allows re-generation of identical test cases

XML example generation wizard



CAM Template Editor - C:\Documents and Settin....\getDataItemRequest.cam

File Edit View XML Tools Run

Structures

ID	Reference
getDataIt...	

Structure

```
⊟ Structure ID="getDataItemRec
  ⊟ <lexssr:getDataItemReques
    ⊟ <lexs:DataItemRequestM
      ⊞ <lexs:SRMessageMeta
      ⊞ <lexs:UserAssertion>
      ⊞ <lexs:DataItemID>%
      ⊞ <lexs:StructuredPay
      ⊞ <lexs:StructuredPay
```

Rules | Parameters | Name

Type and Conditions	
⊞ default context	DEF...

Generate Examples

Output Location	C:\Documents and Settings\dwebber'
Output file name stub	getDataItemRequest
Schema file name	
[Number of example]	3
[Number of repeats]	3
Optional Values	
Generate Random Errors	
Default Content	
Use namespaces	
Choice Mode	one

Advanced Options

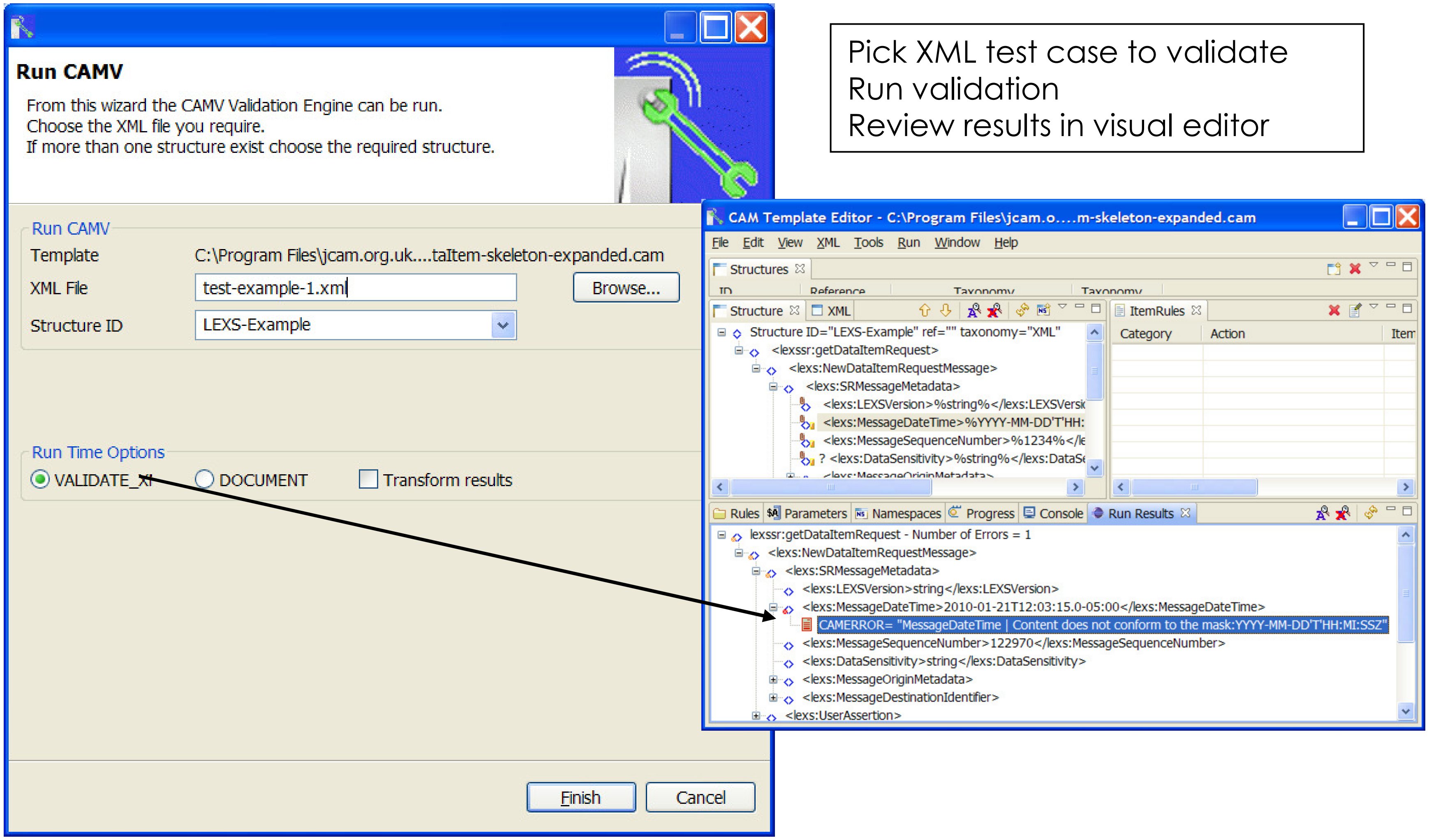
[Hints File]	
Seed	

View of CAM toolkit with LEXS getDataItemRequest and dialogue for XML test example generator tool

Running validation rules tests

- Built-in CAM validation engine allows testing of XML instances against actual exchange rules (CAMV).
- Critical to ensure that the exchange validates actual live production example scenarios correctly
- Allows deployed solution to match exchange schema details
- Errors can be reviewed interactively in exchange visual interface
- Post-processing of validation results allows unit regression tests to be created with reporting of errors, warnings and information level notes

Run Exchange Template



Run CAMV

From this wizard the CAMV Validation Engine can be run.
Choose the XML file you require.
If more than one structure exist choose the required structure.

Run CAMV

Template	C:\Program Files\jcam.org.uk....taItem-skeleton-expanded.cam
XML File	test-example-1.xml
Structure ID	LEXS-Example

Run Time Options

[✓ VALIDATE_XI] DOCUMENT ☐ Transform results

CAM Template Editor - C:\Program Files\jcam.o....m-skeleton-expanded.cam

```
Structure ID="LEXS-Example" ref="" taxonomy="XML"
  <lexssr:getDataItemRequest>
    <lexs:NewDataItemRequestMessage>
      <lexs:SRMessageMetadata>
        <lexs:LEXSVersion>%string%</lexs:LEXSVersi
        <lexs:MessageDateTime>%YYYY-MM-DD'T'HH:
        <lexs:MessageSequenceNumber>%1234%</le
        ? <lexs:DataSensitivity>%string%</lexs:DataSe
```

```
lexssr:getDataItemRequest - Number of Errors = 1
  <lexs:NewDataItemRequestMessage>
    <lexs:SRMessageMetadata>
      <lexs:LEXSVersion>string</lexs:LEXSVersion>
      <lexs:MessageDateTime>2010-01-21T12:03:15.0-05:00</lexs:MessageDateTime>
        CAMERROR= "MessageDateTime | Content does not conform to the mask:YYYY-MM-DD'T'HH:MI:SSZ"
      <lexs:MessageSequenceNumber>122970</lexs:MessageSequenceNumber>
      <lexs:DataSensitivity>string</lexs:DataSensitivity>
      <lexs:MessageOriginMetadata>
      <lexs:MessageDestinationIdentifier>
    <lexs:UserAssertion>
```

Pick XML test case to validate
Run validation
Review results in visual editor

Example Exchange Packaging Details

Package Artefact	Description
Exchange Files	
Subset Schema	Subset of the full exchange schema—a reduced set of components only used in this exchange, not every possible component.
Crosswalk XML	Itemized list of each dictionary component element and attribute included in the exchange.
Exchange Schema	Base document schema that defines the full XML structure for the exchange and is generally named after the exchange itself.
Constraint Schema	Optional schema that includes additional constraints and code values for the main exchange schema
Extension Schema	Specification for extended components—separate local name-spaces of components not contained in dictionary
Sample XML Instance	Example instance(s) – may reference optional stylesheet.
Stylesheet	Example stylesheet for display of instance(s).
Documentation	
Master Documentation	The Master Document is the main document for which all of the context and details around the exchange are explained. This document includes, the overview, as well as details surrounding the exchange, business drivers and requirements
Exchange model	Exchange model in standard open format (xmi, vsd, zargo) and standard open graphic (jpg, pdf, etc.) preferably a Unified Modeling Language (UML) model.
Business rules	Business rules in one of the following formats: (1) plain or structured English, (2) written into master documentation, (3) generated by a development tool.
Mapping to Dictionary	Mapping of domain components, tagged with constraints (i.e., cardinality, etc.) to dictionary components as a spreadsheet.
Extended components	Components created because they were not in dictionary—may be part of mapping spreadsheet and include structure and definitions of new components.
Change log	Record of cumulative changes from previous exchange versions. The initial exchange simple records its creation date.
Catalog	
Catalog XML file	A machine-readable list of artifacts provided in this exchange package.
Metadata XML file	All metadata of owner and domain to be associated with the exchange.

Summary

Dictionary driven exchanges
Blueprint enabled reuse
Automated exchange package generation
Alignment to NDR Principles and Rules
Testing and validation support

Review

- **Top Down development**
  - Reference dictionary components
  - Create exchange blueprint
  - Run Expander tool
  - Refine desired structure in visual editor
- **NDR Principles and Rules**
  - Best practices for interoperability and schema techniques
- **Dictionary driven reuse**
  - Enterprise Data Model and industry components
  - Ensures consistency of definition and use
- **Automated exchange package generation**
  - Schemas, XML, documentation, mapping crosswalk
  - Test generated example XML with rules validation

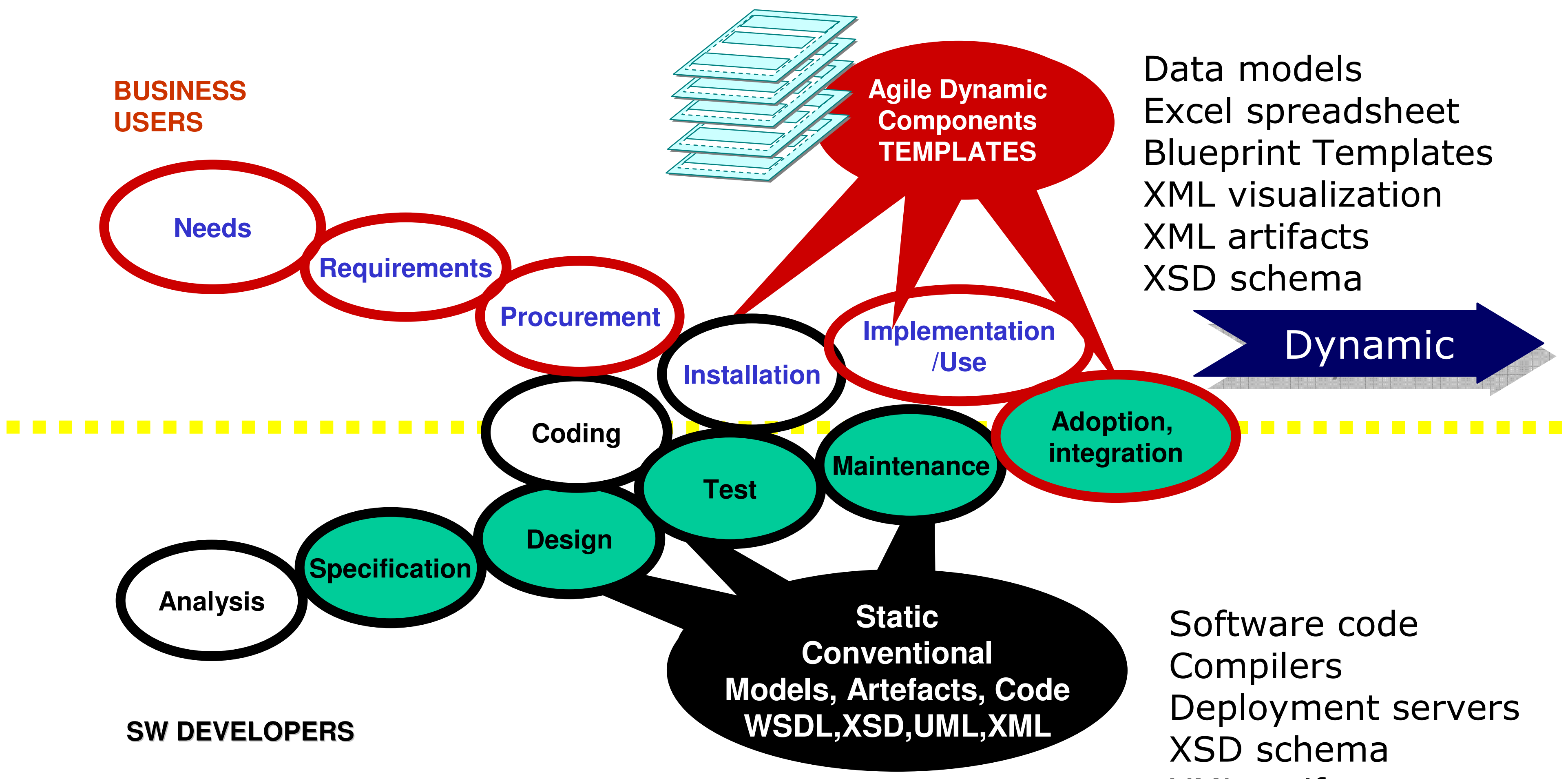
Reference Materials

References and Links

Links and Resources

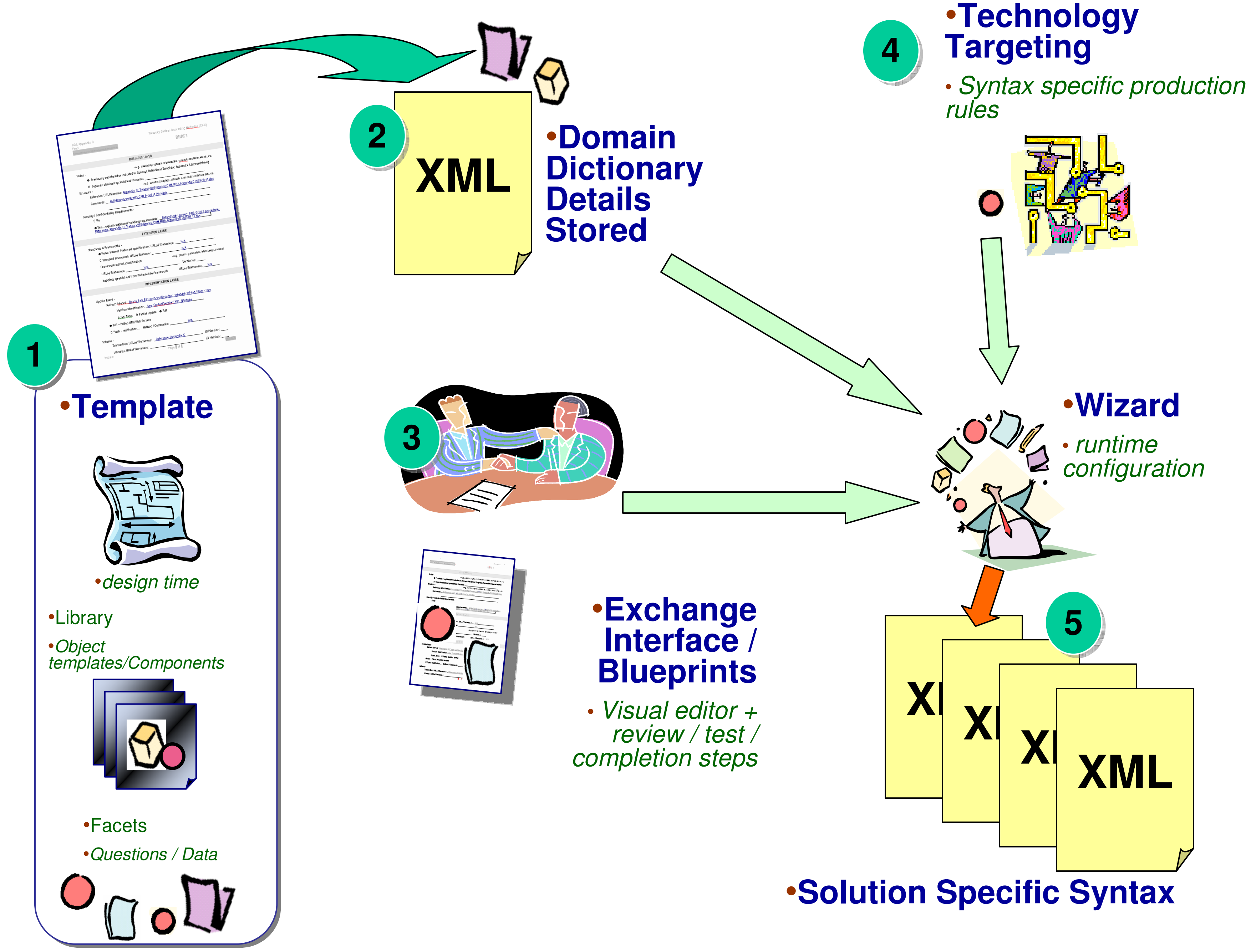
- **DOWNLOADS -**
  - CAM Toolkit download
    - https://sourceforge.net/projects/camprocessor
- **SUPPORTING MATERIALS -**
  - NIEM Naming and Design Rules (NDR) 1.3
    - http://www.niem.gov/pdf/NIEM-NDR-1-3.pdf
- **RESOURCES –**
  - Additional support slides (following)

Blueprint Driven Approach



Data models
Excel spreadsheet
Blueprint Templates
XML visualization
XML artifacts
XSD schema

Software code
Compilers
Deployment servers
XSD schema
XML artifacts

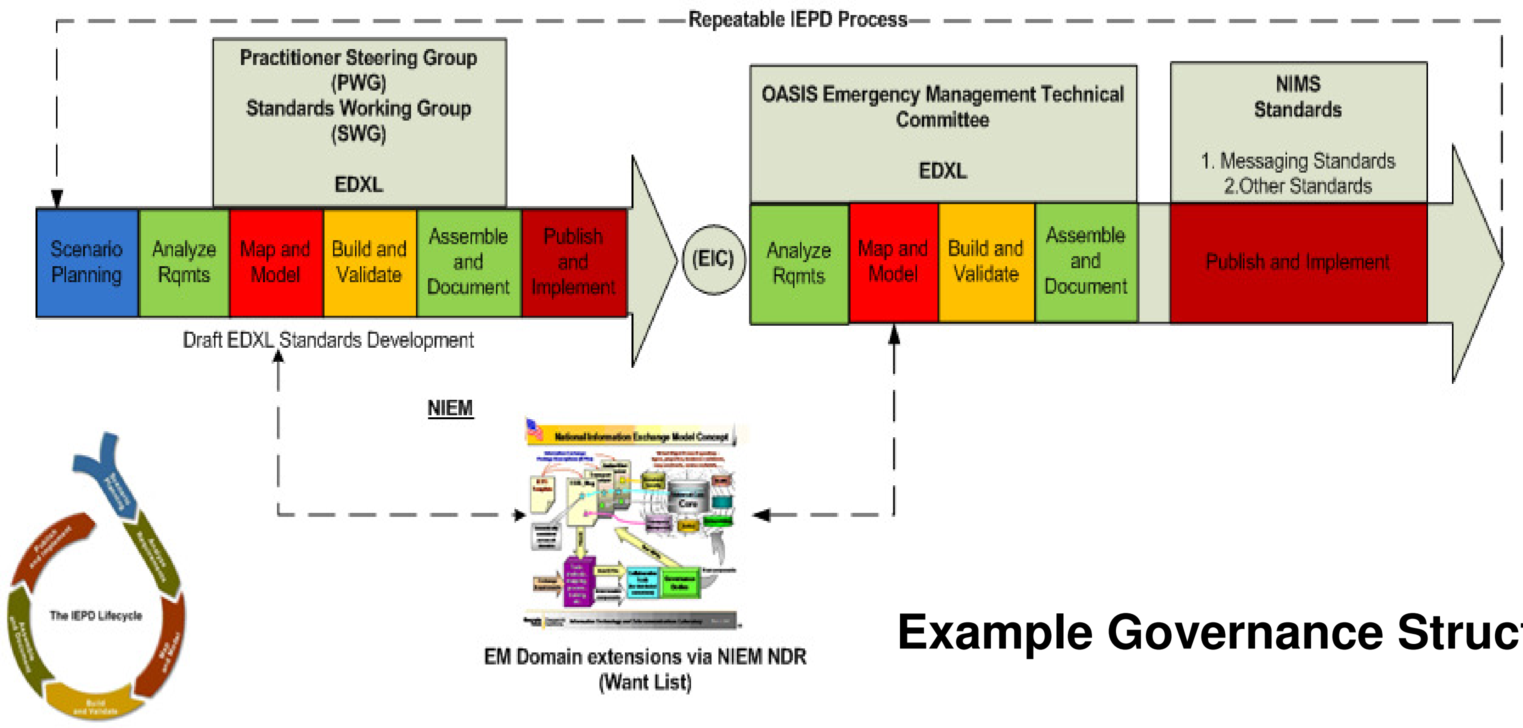

1
•Template
•design time
•Library
•Object templates/Components
•Facets
•Questions / Data
2
XML
•Domain Dictionary Details Stored
3
•Exchange Interface / Blueprints
• Visual editor + review / test / completion steps
4
•Technology Targeting
• Syntax specific production rules
•Wizard
• runtime configuration
5
XML
•Solution Specific Syntax

Domain Exchange Development Steps

- **Adopt formal Naming and Design Rules (NDR)**
    - UN/CEFACT – NDR
    - OASIS UBL – Universal Business Language
    - OASIS EML – Election Markup Language
    - NIEM – National Information Exchange Model approach (http://www.niem.gov)
    - OASIS EM - Emergency Management joint initiative with NIEM
- **Develop data models of core components for use in exchanges**
- **Build Dictionary of Core Components**
- **Provide Principles and Rules guidance to schema team**
    - Use namespaces, Yes / No?
    - Camel case naming convention?
    - Schema constructs and restrictions on use?
- **Information Exchange Package Documentation (IEPD)**
    - Describes formal exchange that conforms to NDR and principles and rules
    - Provides schema, example XML, supporting artifacts
    - Re-uses core components
    - Defines domain specific components

Example Governance Structure

DHS Disaster Management (DM) – EDXL Standards Development Process (DRAFT)



Required Actions

DM process & OASIS

1. Utilize NIEM as development Data Dictionary
2. Utilize NIEM NDR for element/attribute extension requests
3. Utilize and Update the NIEM EM Domain via governance established by NIEM

NIEM

1. Provide final Naming and Design Rules (NDR)
2. Provide governance for change management to include processes for addition/deletion/ modification of EM Domain elements and attributes
3. Provide IEPD Repository

NIMS

1. Evaluate proposed standards for adoption into "NIMS Approved" Messaging Standards
2. Require NIEM compliancy for all "NIMS Approved" Messaging standards.
3. Publish to NIEM IEPD Repository with XML Messages for community re-use.

OASIS Content Assembly Mechanism (CAM) & Integration Technologies Guide

W3C XSD Schema	OASIS CAM Templates	Domain Dictionaries NDR	Shared Semantics Registry	Ontology Classification Discovery	Machine Based Reasoning
- WHAT?	- HOW? WHY?	- WHO?	- WHERE?	- WHERE?	- WHEN?
Provides lexicon of information content Describes structure constructs Arranges groups of information Simple content typing Software tooling interfaces	Provides actual use patterns (templates) Supports context handling and rules Rendering outputs and documentation for verification Enables integration testing / certification Versioning	- Alignment of meaning and terms - Consistent domain definitions Modelling methods and practice Business information content building blocks	Shared resources of semantic definitions Code lists Dynamic rendering Distributed versioning control Role and access security management	Domain classification systems Ontology and reasoning definitions Associations and linkages Search and drilldown Modelling tools	Alerts Process control Workflow Automated interfacing Business Intelligence