The background features a blue grid with various business-related icons: a large white arrow pointing up, a blue line graph, a yellow bar chart, and silhouettes of people celebrating. One person is standing on a yellow bar, another is climbing a ladder, and a third is giving a thumbs up.

# How can semantic models help integrate the smart grid?

Steve Ray

Carnegie Mellon University Silicon Valley

in collaboration with  
Ralph Hodgson, TopQuadrant

# Goal

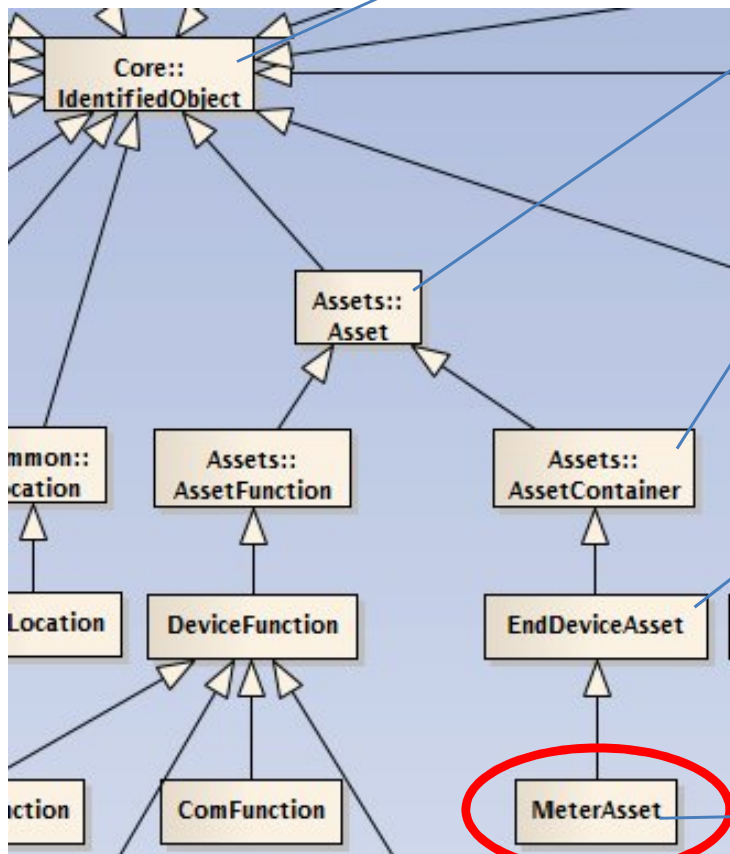
- To enable effective, error-free information exchange among all the players in the smart grid
  - The messages used to carry the information are based upon underlying information models

# Problem

- Many existing, and more emerging, information models reflect distinct industry sector perspectives on the electrical grid
- Legacy systems are already using these models
- The models:
  - Are at varying levels of abstraction
  - Partition concepts in different ways

# Example: Meter

IEC 61968



This is a root class to provide common naming attributes for all classes needing naming attributes

Tangible resource of the utility, including power system equipment, cabinets, buildings, etc. For electrical network equipment, the role of the asset is defined through **PowerSystemResource** and its subclasses, defined mainly in the Wires model (refer to IEC61970-301 and model package IEC61970::Wires). Asset description places emphasis on the physical characteristics of the equipment fulfilling that role.

Asset that is aggregation of other assets such as conductors, transformers, switchgear, land, fences, buildings, equipment, vehicles, etc.

AssetContainer that performs one or more end device functions. One type of **EndDeviceAsset** is a **MeterAsset** which can perform metering, load management, connect/disconnect, accounting functions, etc. Some **EndDeviceAssets**, such as ones monitoring and controlling air conditioner, refrigerator, pool pumps may be connected to a **MeterAsset**. All **EndDeviceAssets** may have communication capability defined by the associated **ComFunction(s)**. An **EndDeviceAsset** may be owned by a consumer, a service provider, utility or otherwise.

There may be a related end device function that identifies a sensor or control point within a metering application or communications systems (e.g., water, gas, electricity). Some devices may use an optical port that conforms to the ANSI C12.18 standard for communications.

Physical asset that performs the metering role of the **ServiceDeliveryPoint**. Used for measuring consumption and detection of events.

# IEC 61968

## Meter Attributes

MeterAsset
+ formNumber: String [0..1]
+ kWh: Float [0..1]
+ kR: Float [0..1]
<i>::EndDeviceAsset</i>
+ amrSystem: String [0..1]
+ demandResponse: Boolean [0..1]
+ disconnect: Boolean [0..1]
+ dstEnabled: Boolean [0..1]
+ loadControl: Boolean [0..1]
+ metrology: Boolean [0..1]
+ outageReport: Boolean [0..1]
+ readRequest: Boolean [0..1]
+ relayCapable: Boolean [0..1]
+ reverseFlowHandling: Boolean [0..1]
+ timeZoneOffset: Minutes [0..1]
<i>::Asset</i>
+ acceptanceTest: AcceptanceTest [0..1]
+ application: String [0..1]
+ category: String [0..1]
+ corporateCode: String [0..1]
+ critical: Boolean [0..1]
+ electronicAddress: ElectronicAddress [0..1]
+ initialCondition: String [0..1]
+ initialLossOfLife: PerCent [0..1]
+ installationDate: AbsoluteDate [0..1]
+ lotNumber: String [0..1]
+ manufacturedDate: AbsoluteDate [0..1]
+ purchasePrice: Money [0..1]
+ serialNumber: String [0..1]
+ status: Status [0..1]
+ utcNumber: String [0..1]
<i>::IdentifiedObject</i>
+ aliasName: String [0..1]
+ description: String [0..1]
+ localName: String [0..1]
+ mRID: String [0..1]
+ name: String [0..1]
+ pathName: String [0..1]

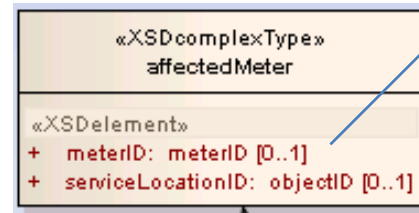
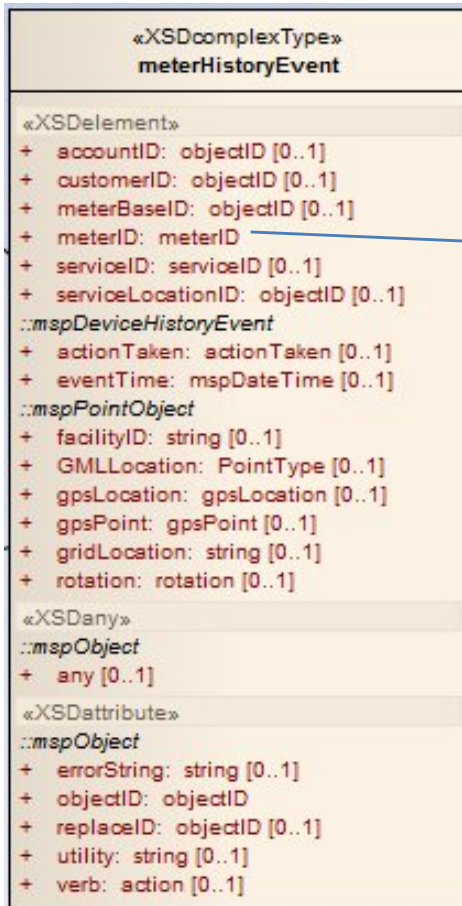
Generally speaking...

These are about what a device can do  
(from a control perspective)

These support the tracking and inventory of assets  
(from an owner's perspective)

These just help identify any object

# Multispeak V4.1 Meter Attributes

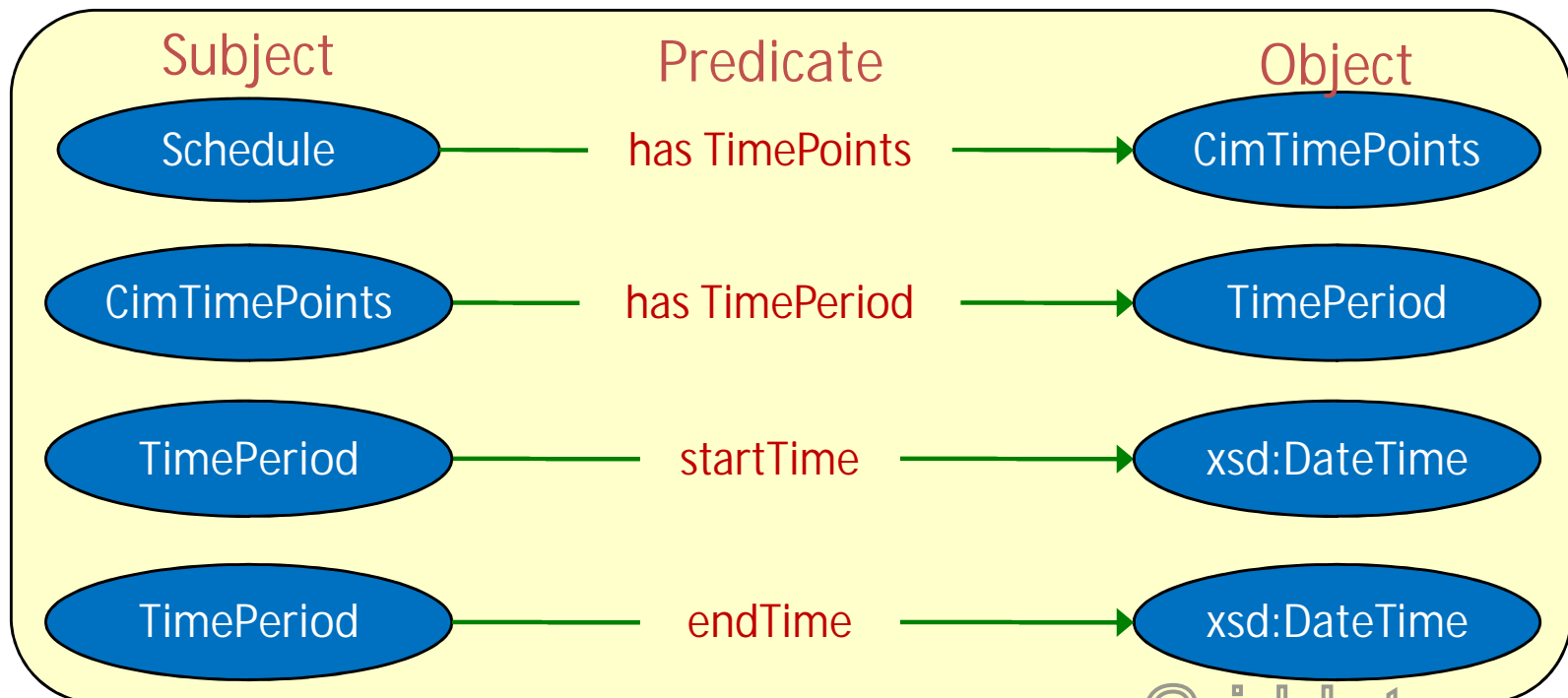


Emphasis here is on the actions  
taken on a meter.  
Meter is not modeled explicitly,  
and is simply identified via a  
meterID string  
(user's perspective)

# What is Semantic Technology?

“Semantic Technology allows the meaning of and associations between information to be represented precisely and, optionally, consulted and processed at execution time. ”

Semantic Technology represents information in a neutral form called Triples that consist of a Subject, a Predicate and an Object. Each of these can have properties that infer new information. Each has a unique identity, enabling merging of different datasets.



# Semantic Web Technologies provide increased expressiveness for Information Models

Query Language

S

OWL

P

A

RDFS

R

Q

RDF

L

XML

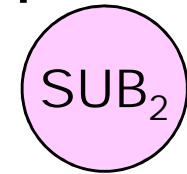
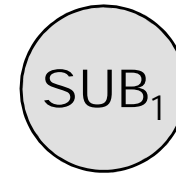
More Model Formalism + Reasoning

Classes

Relationships

Structures

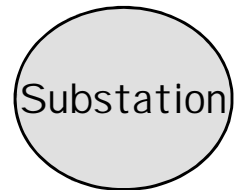
+ Proof



Rules

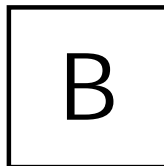
+ Trust

is a

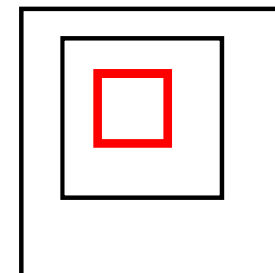


A

has Substation



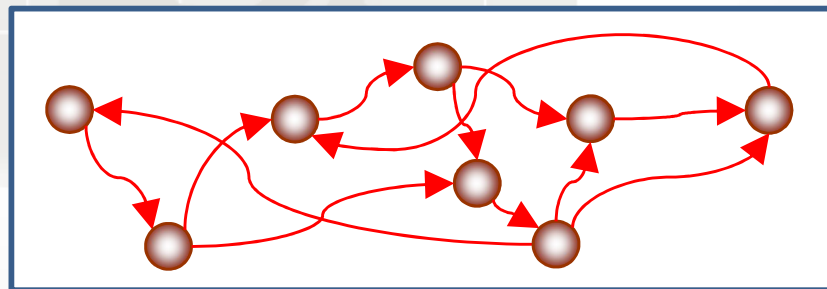
A





# What Problems can be Solved by Semantic Technology

- Alignment of Terms and relationships
- Generation of consistent XML Schemas
  - Component-Based
- Generation of controlled vocabularies
- Aggregation of data from disparate sources
- “Smart Data” - Inference of new information from data



# Recommended Steps

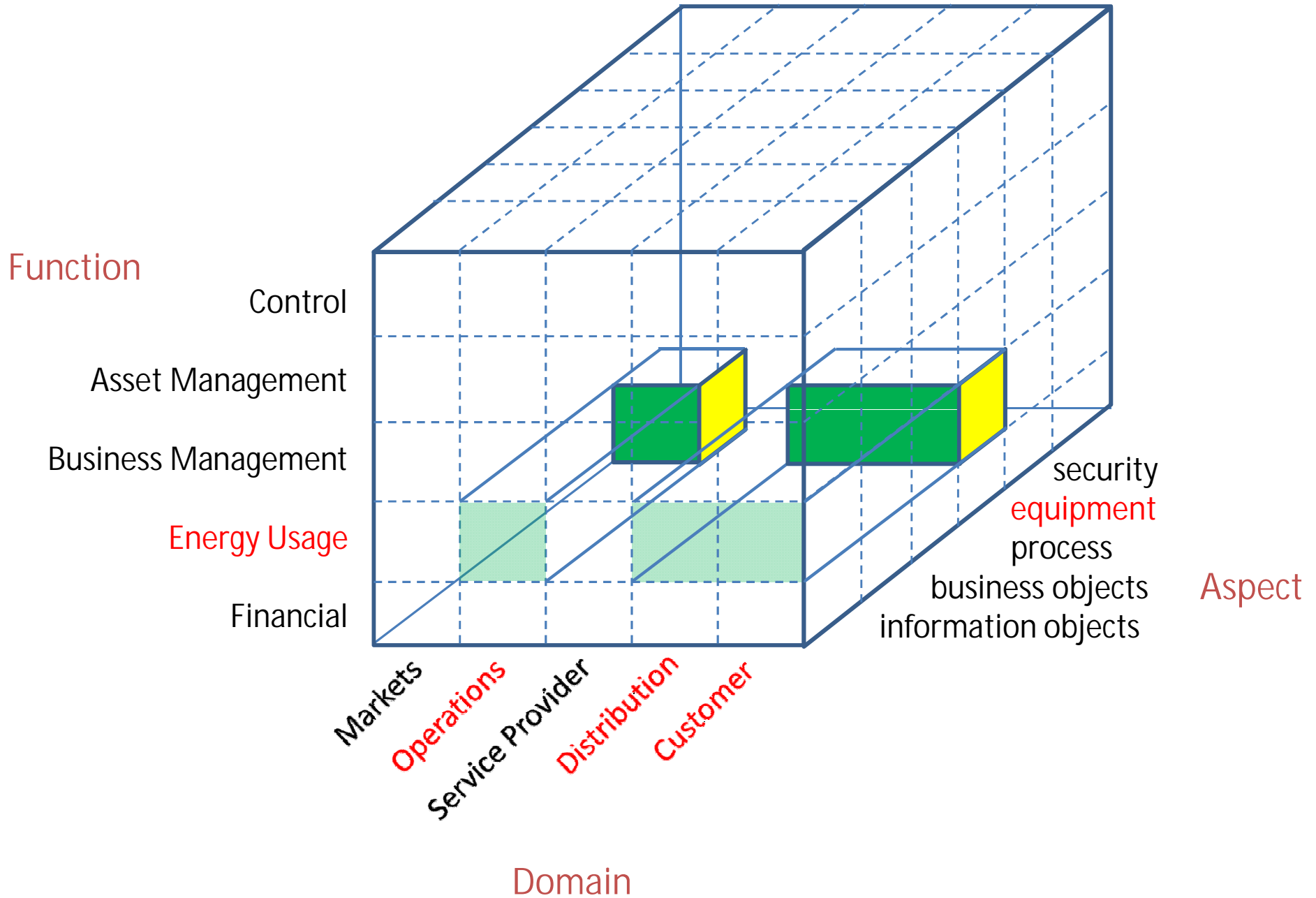
- Define a semantic framework in which we can position an abstract semantic model, and the component standards
- Migrate selected UML models to OWL Specifications
- Determine opportunities and approach for standardizing vocabularies
- Verify and Validate the OWL Models
- Coordinate Revisions with other CIM efforts
- Demonstrate creation of UN/CEFACT CCTS Schemas
- Define pilot assessment strategy

# The Semantic Framework (i.e. Context)

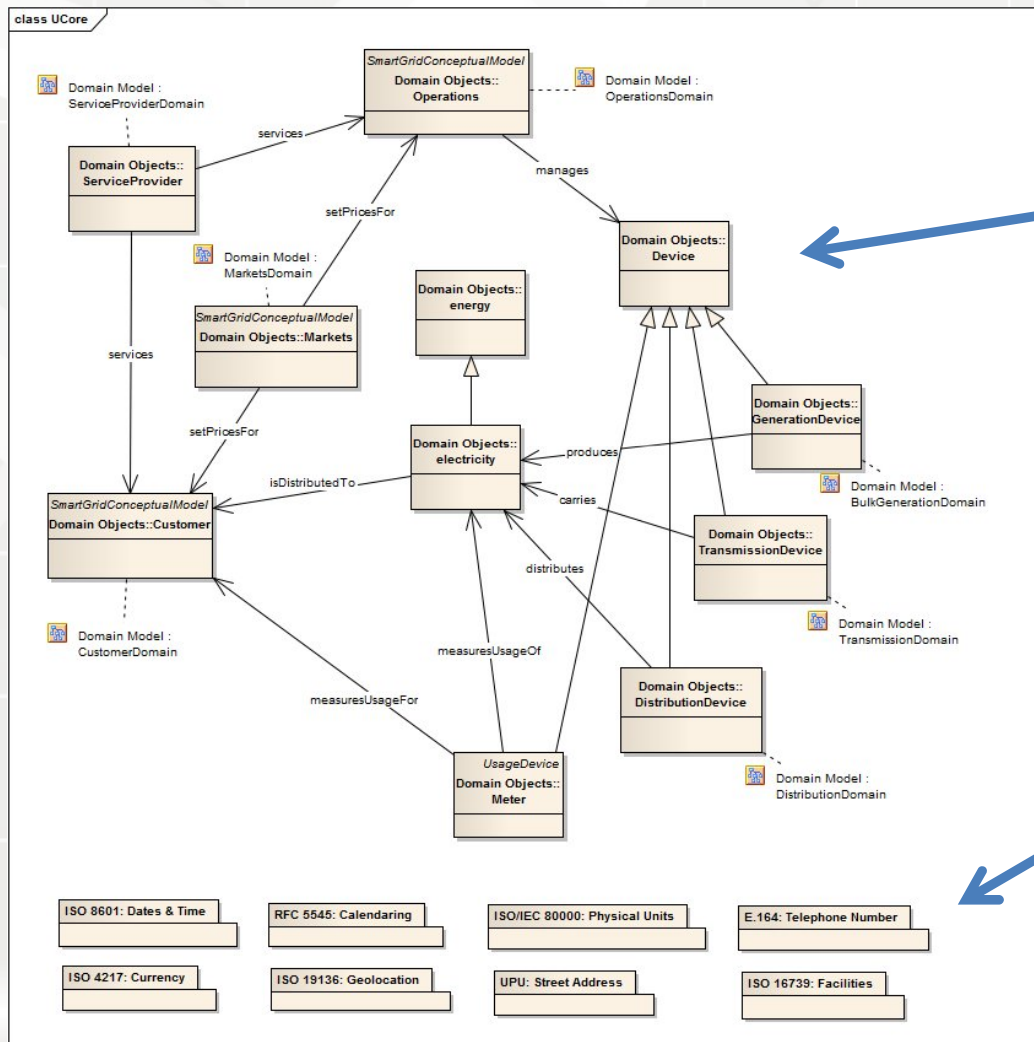
Semantic models (or model fragments) need to be positioned in terms of:

- What Function is it fulfilling (grid control, inventory & maintenance, business management, billing?)
- What Aspect is it modeling (security, equipment, process, business objects, information objects?)
- What Domain perspective within the smart grid community does it reflect (operations, customer, transmission?)
- What Specificity does it capture (individual appliance consumption, regional demand?)
- Is it Time Invariant information (snapshot) or Time Variant (recording history)?

# e.g. IEC 61968, Part 9



# Abstract Information Model



- The abstract layer relates the concepts found in all the smart grid standards
- It also contains links to standards outside the smart grid that can be incorporated

# Using RDF/OWL for Specifications

- An OWL Class specifies the data type properties (attributes) and object properties (associations) of an Information Object
- Multiple inheritance allows common properties to be factored out into reusable classes
- OWL models can be transformed to other representations using mappings
- Examples of transformations are:
  - OWL to XML Schema and XML Vocabularies at NASA
  - OWL to UN/CEFACT CCTS (Core Component Technical Specifications) at the Netherlands Ministry of Justice

# Work To-date on SmartGrid OWL Models

- Conversions of CIM and IEC Models to OWL
  - IEC61968
  - IEC61970
  - EUI Core
  - Multispeak V41
- UN/CEFACT approach to generating Schemas

# XML Schema (XSD) Constructs are converted to OWL Constructs

XSD	OWL	Notes
Complex Type	Class	
Scalar Attribute	Datatype Property	Values are xsd datatypes
Reference to Complex Type	Object Property	Expressed as a restriction on the OWL Class
Simple Type	Two Classes	One class is for the Simple Type and another subclasses an xsd data type. An rdf:value property is used to connect the first class to the xsd datatype subclass.
Complex Type extending Simple Type	Two Classes	One class is for the Simple Type and another subclasses an xsd data type. An rdf:value property is used to connect the first class to the xsd data type subclass.
Enumeration	Class with an additional class for members	OWL oneOf is used for enumerations with less than 100 members. This could be configured.
Substitution Group	Class	The Substitution Group Class is inherited at each occurrence of use
(Element) Group	Class	The Element Group Class is inherited at each occurrence of use
Attribute Group	Class	The Attribute Group Class is inherited at each occurrence of use



# Example of EUI OWL Model

Classes

Phase Configuration Kind is an Enumeration

Properties

The screenshot displays the Protege OWL editor interface. On the left, the 'Classes' pane shows a tree of classes, with 'PhaseConfigurationKind (8)' selected. The main editor window shows the definition for 'PhaseConfigurationKind' in the 'EUI\_2010\_06.rdf' file. The definition includes an annotation property 'comment' and several other properties: 'oneOf' (listing various PCK values), 'subClassOf Enumeration', 'rdf.value only xsd:string', and 'type Class [owl:Class]'. The 'Incoming References' section is also visible. On the right, the 'Properties' pane shows a list of properties grouped by namespace, including 'hasBillingPeriod', 'hasCurrency', 'hasCurrentDayLastYearNetConsumption', etc. Red arrows point from the 'Classes' and 'Properties' labels to their respective panes, and a red arrow points from the 'Phase Configuration Kind is an Enumeration' label to the 'subClassOf Enumeration' property.

**Classes**

- rdfs:Resource (591)
- owl:Ontology (2)
- owl:Thing (113)
- CurrentFlow
- CustomerAccount
- CustomerAgreement
- CustomerAuthorisation
- DateTimeInterval
- EndDeviceEvent
- EnergyUsageInformation
- Enumeration (113)
- Currency (14)
- PhaseConfigurationKind (8)**
- QualityOfReading (7)
- ReadingKind (22)
- ServiceKind (11)
- SourceKind (3)
- SupplierKind (3)
- UnitMultiplier (11)
- UnitSymbol (34)
- IntervalReading
- MeterAsset
- MeterReading
- owl:Nothing
- PowerQualitySummary
- Reading
- ReadingType
- ReadingQuality
- Seconds
- ServiceCategory
- ServiceDeliveryPoint
- ServiceSupplier
- Status
- SummaryMeasurement
- TariffProfile
- UsageSummary

**PhaseConfigurationKind**

**Annotation Properties**

- comment** Kind of phase configuration. Note that there is an enum Wires::WindingConnection with values {D, Y, Z, Yn, Zn}. However, there are many more phase configurations than delta (threePhaseThreeWire) and wye (threePhaseFourWire), which are defined here.

**Other Properties**

- oneOf** [PCK\_twoPhaseTwoWire, PCK\_onePhaseTwoWire, PCK\_onePhaseThreeWire, PCK\_twoPhaseThreeWire, PCK\_threePhaseTwoWire, PCK\_threePhaseThreeWire, PCK\_other, PCK\_threePhaseFourWire]
- subClassOf** Enumeration  
rdf.value only xsd:string
- type** Class [owl:Class]

**Incoming References**

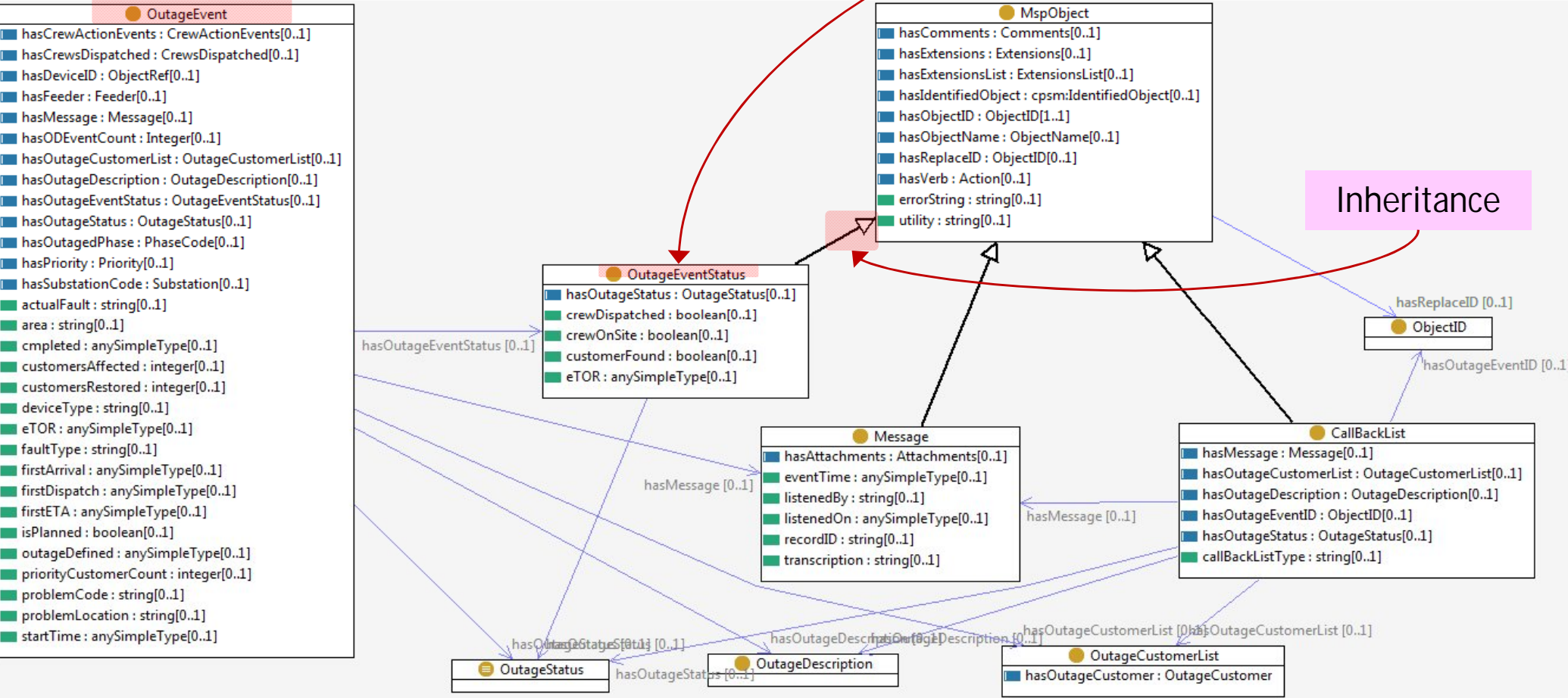
**Properties**

- <Default Namespace>
- hasBillingPeriod
- hasCurrency
- hasCurrentDayLastYearNetConsumption
- hasCurrentDayNetConsumption
- hasCurrentDayOverallConsumption
- hasCustomerAccount
- hasCustomerAgreement
- hasCustomerAuthorisation
- hasEndDeviceEvent
- hasEstimatedLoad
- hasIntervalReading
- hasKind
- hasMeterAsset
- hasMeterReading
- hasMultiplier
- hasPeakDemand
- hasPhaseConfig
- hasPowerQualitySummary
- hasPreviousDayLastYearOverallConsumption
- hasPreviousDayNetConsumption
- hasPreviousDayOverallConsumption
- hasQualityOfReading
- hasRatchetDemand

# Example 1: Outage Event

Outage Event Class – showing object properties (associations) with blue icons and datatype properties (attributes) with green icons

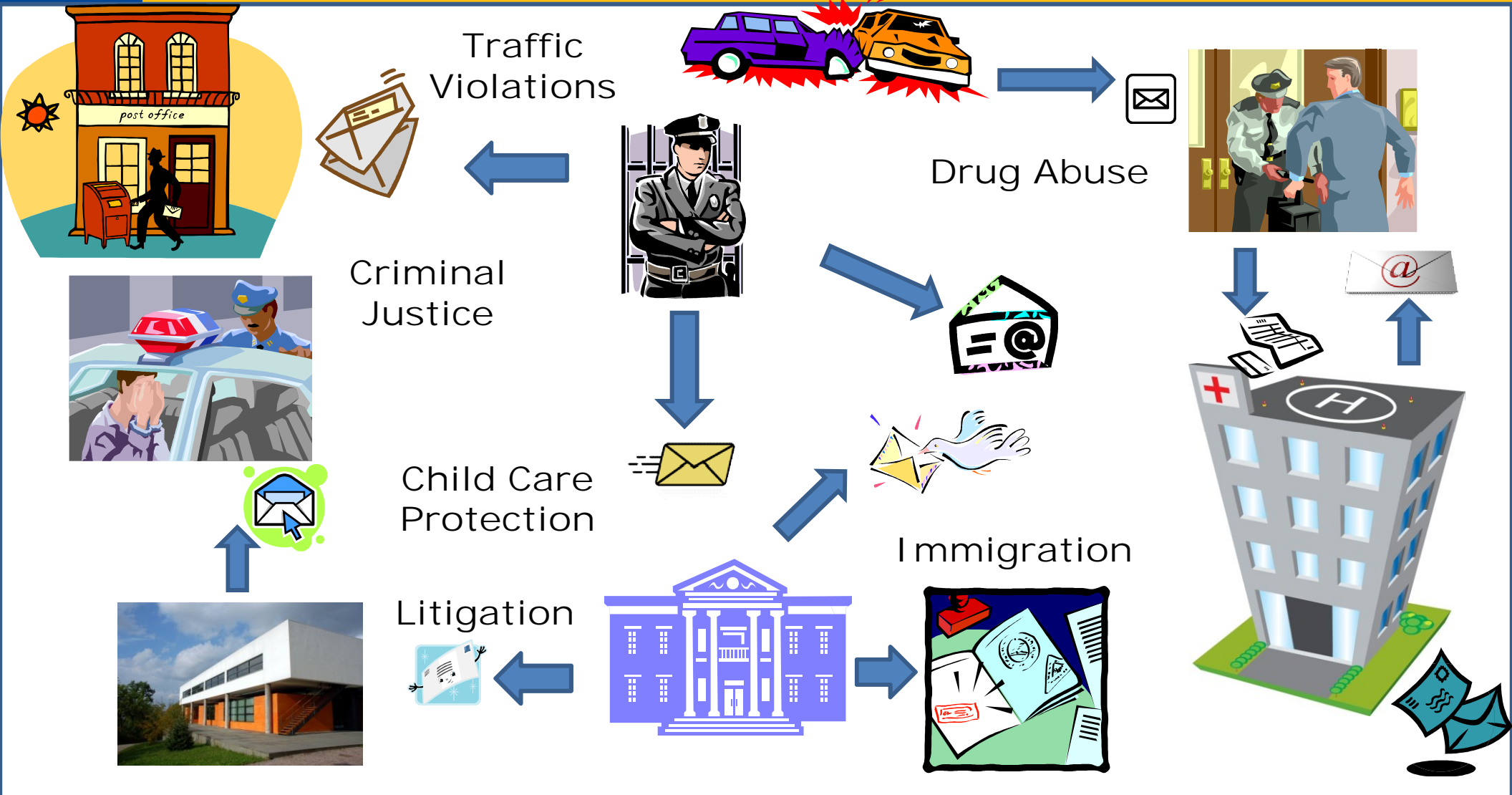
Outage Event Status – an example of an object association



Inheritance

# The Netherlands Ministry of Justice Message Builder

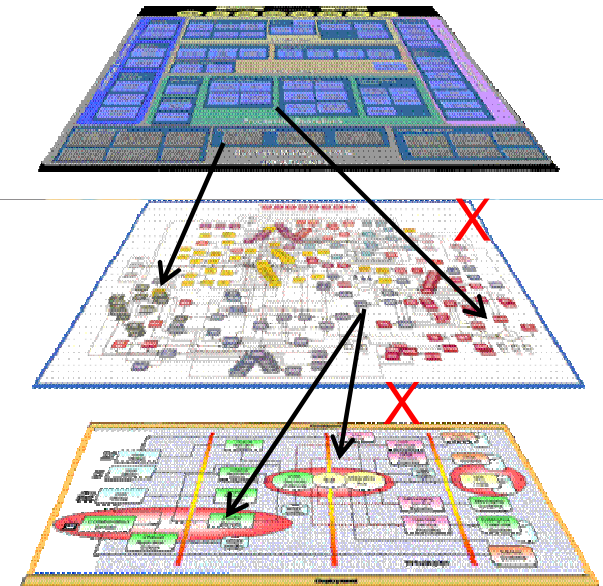
# Justice Data Exchange covers many Domains of Law



Seamless data exchanges are challenging:  
Unique data requirements exist at courts, police, hospitals, border control, motor vehicle, local and federal offices.

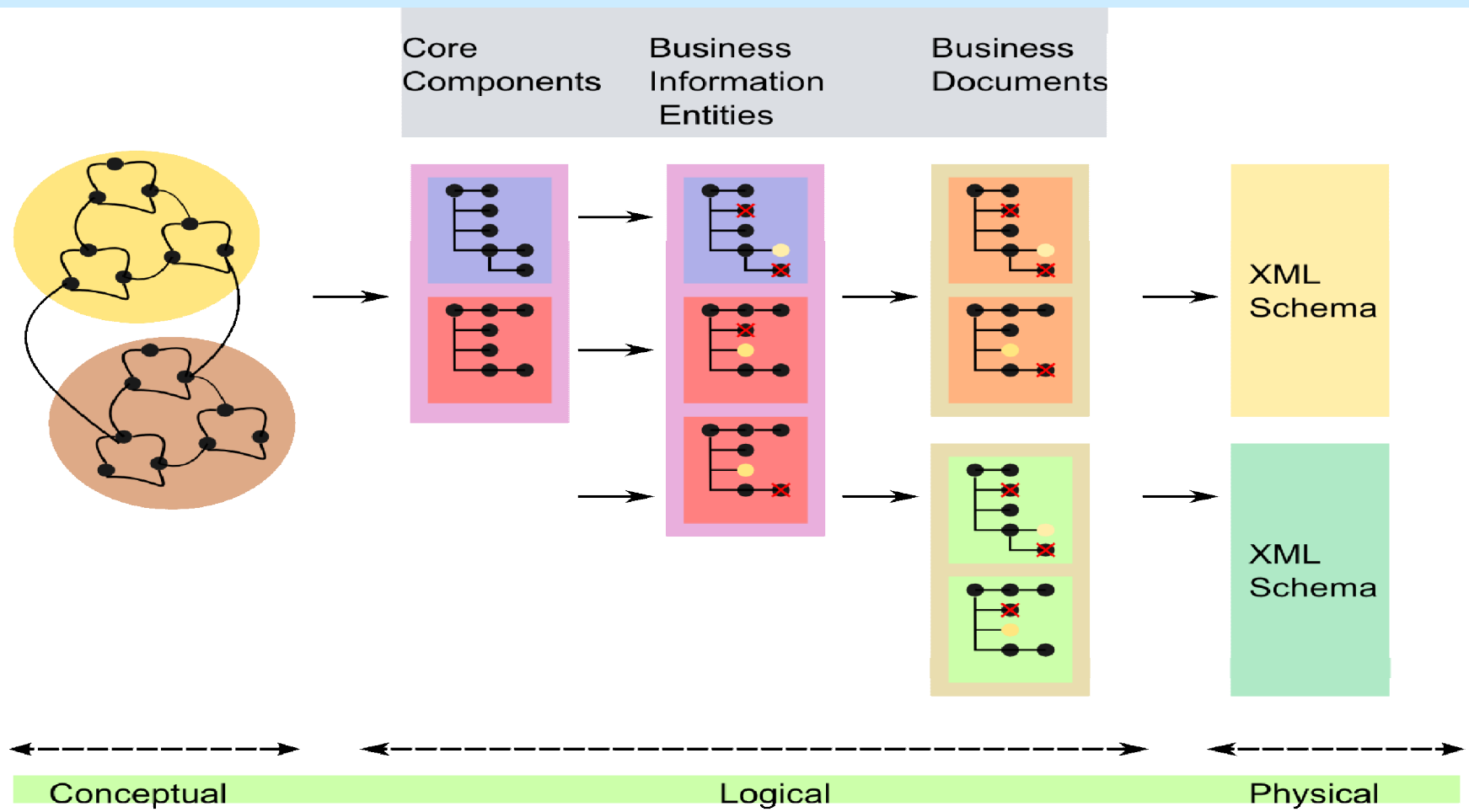
# “Living in the XML Ecology”: Challenges

- Addressing the "standards dilemma"
  - Too many exchange mechanisms and standards
  - Lack of conformance to XML (and OWL) Naming and Design Rules
- Brittleness in data models:
  - Often very complex,
  - Often incomprehensible
  - Sometimes non-implementable.

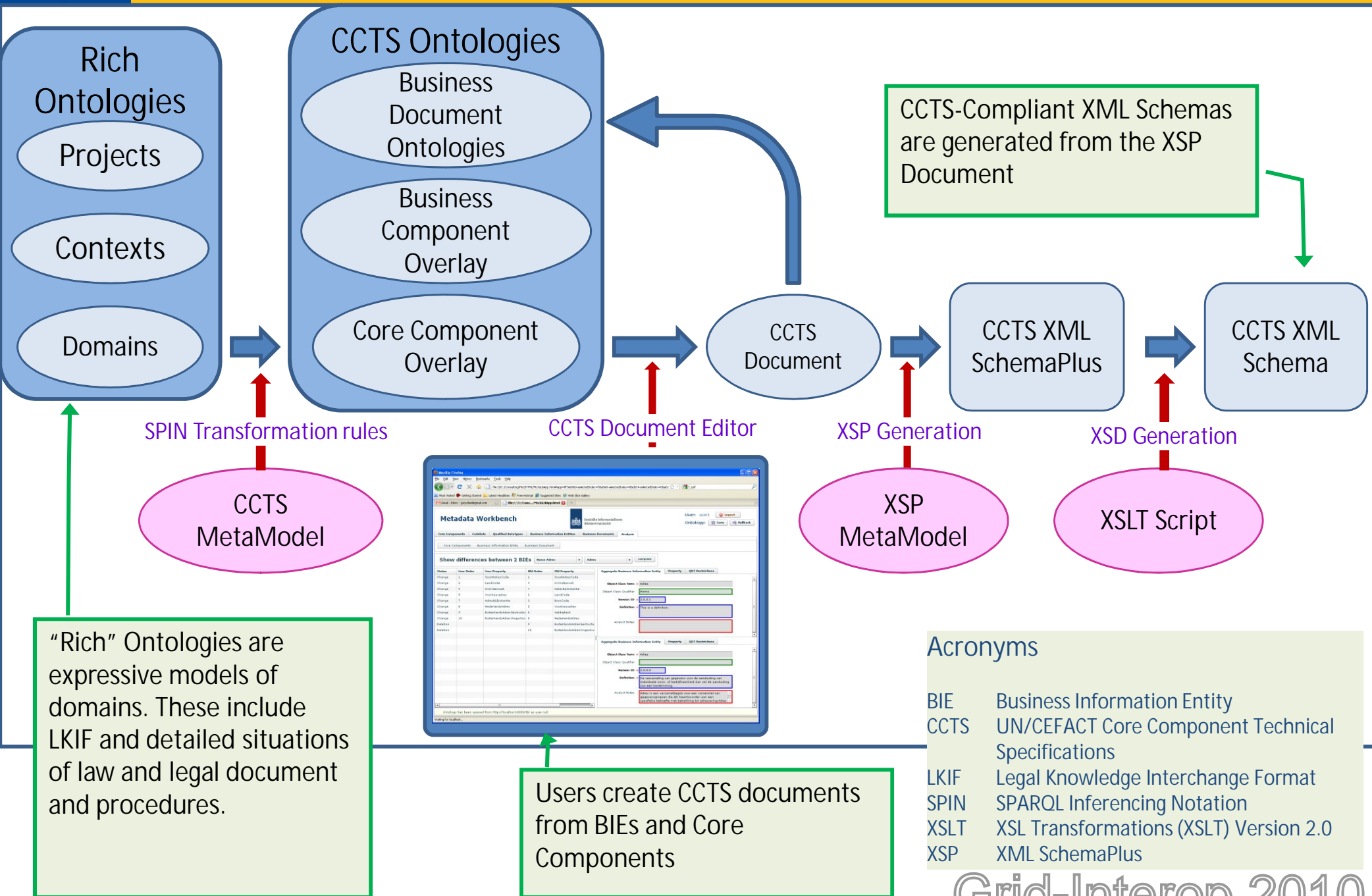


# Ontology-Driven Approach to Message Design for Interoperability

Solution: Ontology-Based Metadata Workbench:  
 Transform Domain Models into UN/CEFACT CCTS compliant representation and allow Business Analysts to assemble business documents for electronic messages from Component Parts.



# Creation of XML Message Schemas



# The Netherlands Ministry of Justice Metadata Workbench

**Metadata Workbench** (1.0b4) User: john.doe [Logout](#) [Refresh](#)

Justitiële Informatiedienst  
Ministerie van Justitie

Core Components | **Codelists** | Qualified datatypes | Business Information Entities | Business Documents | Analysis

[Add BD](#) [Generate schema](#)

Business Document	Order	Qualifier	Property	Representation	Car...
ChangeOfAddress	▼ 1	Old	Address	Old_Address	1..1
	1		Address Type Code	Code	0..1
	2		Country Code	Code	0..1
	3		Preferred Address	Indicator	0..1
	▼ 4		Validity	Validity	0..1
	1		Start Date	Date	0..1
	2		End Date	Date	0..1
	▼ 5	Foreign Address Structured	Foreign Address Structured	0..1	
	1	Location Description	Text	0..1	
	2	Street	Name	0..1	
	3	Number	Numeric	0..1	
	4	Number Addition	Text	0..1	
	5	Zip code	Text	0..1	
	6	Place Name	Name	0..1	
	7	State or region	Name	0..1	
	▼ 2	New	Address	New_Address	1..1
	1		Address Type Code	Code	0..1
	2		Country Code	Code	0..1
	3		Preferred Address	Indicator	0..1
	▶ 4		Validity	Validity	0..1
	▶ 5	Foreign Address Structured	Foreign Address Structured	0..1	
▶ 3	Addressee	Person Specification	Persoonskenmerk	1..1	

Business Document | **Property**

**Object Class Term** \* ChangeOfAddress

Object Class Qualifier

**Version ID** \* 0.0.0.1

**Definition** \* Change of Address Form.

Analyst Notes

[Edit](#) [Save](#) [Cancel](#)

**Business Document**

**Business Information Entities that make up the Business Document**

**Metadata about the Business Document**

Filter versions [Add Property](#)

Search

Example of a User Composing a Business Document from the CCTS Core Vocabulary.



# Benefits of the MoJ Solution

- Business Benefits

- Accurate and rapid Information Sharing between Organizations
- Agility in response to Legislation Changes
- Data Quality is guaranteed
- Reduced Costs of Message Schema Development

- Technical Benefits

- Direct and flexible Reuse of Data Components
- Full Automation of XML Schema creation
- Semantic Consistency is preserved and confirmed
- Linked Data / traceability
- Version Management



See Enterprise Data Journal Article for more details on this solution:

- ❖ <http://www.enterprisedatajournal.com/article/netherlands-ministry-justice-metadata-workbench-composing-xml-message-schemas-owl-models.htm>