Justice XML Data Model

Technical Overview

Justice XML Structure Task Force

April 2003

Why Justice XML Data Model Version 3.0?

- Aligned with standards (some were not available to RDD)
- Model-based → consistent
- Requirements-based – data elements, processes, and documents
- Object-oriented → efficient extension and reuse
- Expanded domain (courts, corrections, and juvenile)
- Extensions to activity objects/processes
- Relationships (to improve exchange information context)
- Can evolve/advance with emerging technology (RDF/OWL)
- Model provides the basis for an XML component registry that can provide
  - Searching/browsing components and metadata
  - Assistance for schema development/generation
  - Reference/cache XML schemas for validation
  - Interface (via standard specs) to external XML registries

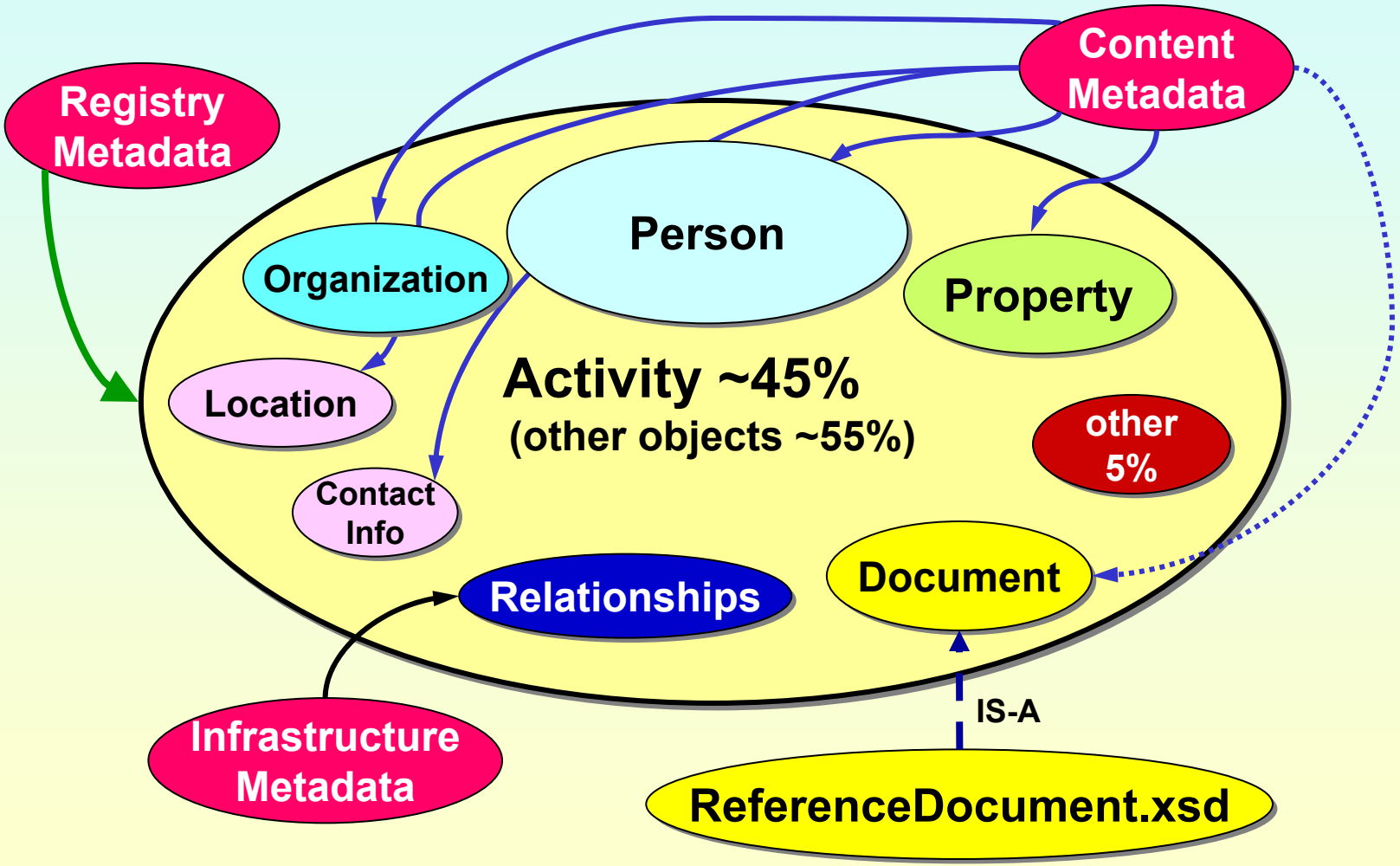
Design Principles

- Design and synthesize a common set of reusable, extensible data components for a Justice XML Data Dictionary (JXDD) that facilitates standard information exchange in XML.
- Generalize JXDD for the justice and public safety communities – do NOT target specific applications.
- Provide reference-able schema components primarily for schema developers.
- JXDD and schema will evolve and, therefore, facilitate change and extension.
- Best extension methods should minimize impact on prior schema and code investments.
- Implement and represent domain relationships so they are globally understood.
- Technical dependencies in requirements, solutions, and the time constraints of national priorities and demands will require rational compromises.

What Standards Apply?

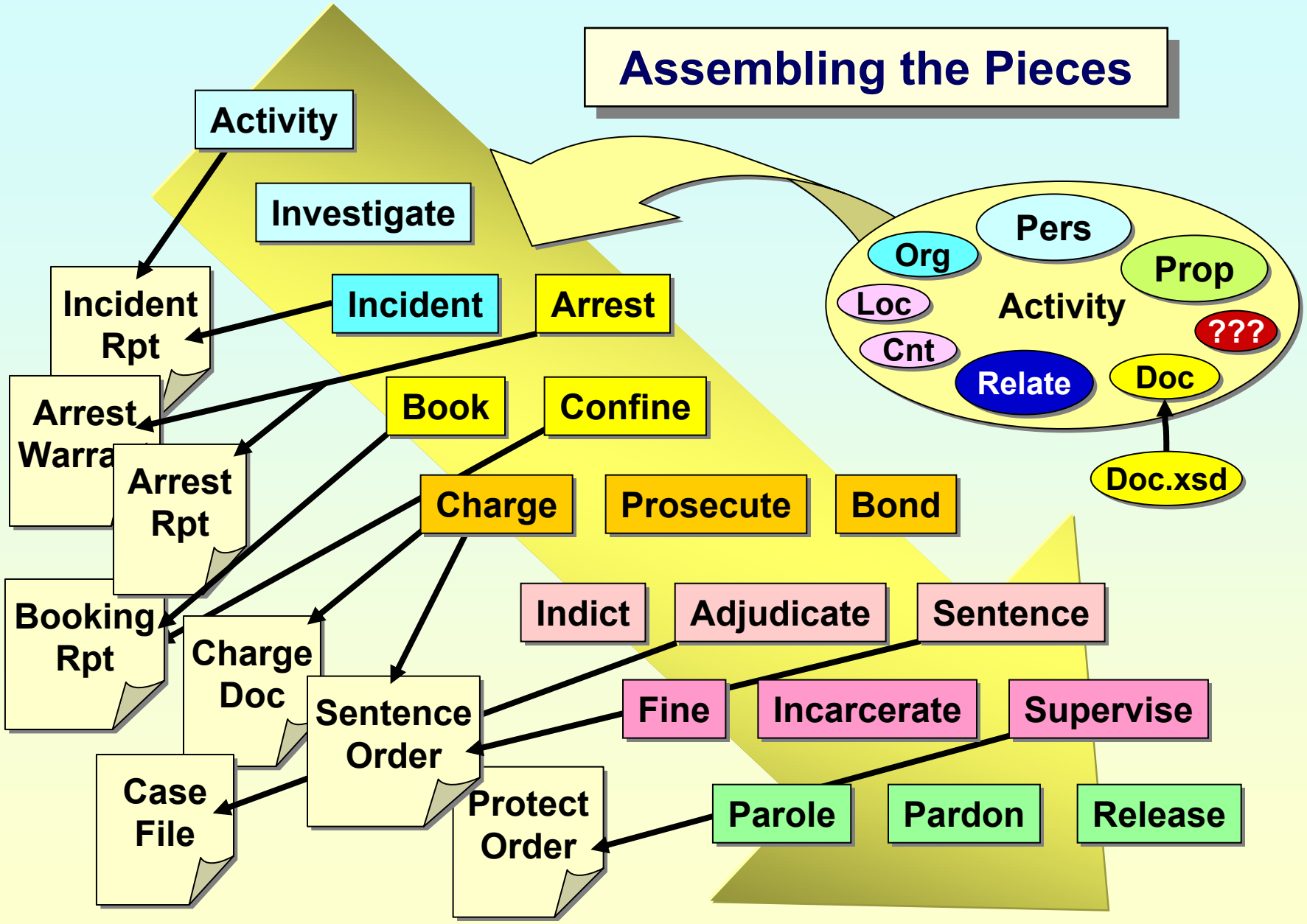
XML.gov	Draft Federal XML Schema Developer's Guide (04/02)
ISO / IEC	11179 Specification & Standardization of Data Elements
UN / CEFACT	ebXML Core Components Technical Spec 1.9 (12/02)
ASC X12	Reference Model for XML Design (10/02)
FBI	Electronic Fingerprint Transmission Spec v7 (01/99)
ANSI / NIST	Data Format for Interchange of Fingerprint, Facial, & SMT
OASIS	XML Common Biometrics Format Committee (09/02)
Dept of Navy	Draft XML Registry Requirements (09/02)
DoD	DoD 5015.2-STD Design Criteria Std for E-RMS Apps (06/02)
IC	Intelligence Community Metadata Language (ICML)
DC	Dublin Core metadata for documents
W3C	XML Schema Specification (05/01)
W3C	RDF and RDF Schema Specification (02/99)

Justice XML Data Dictionary Components

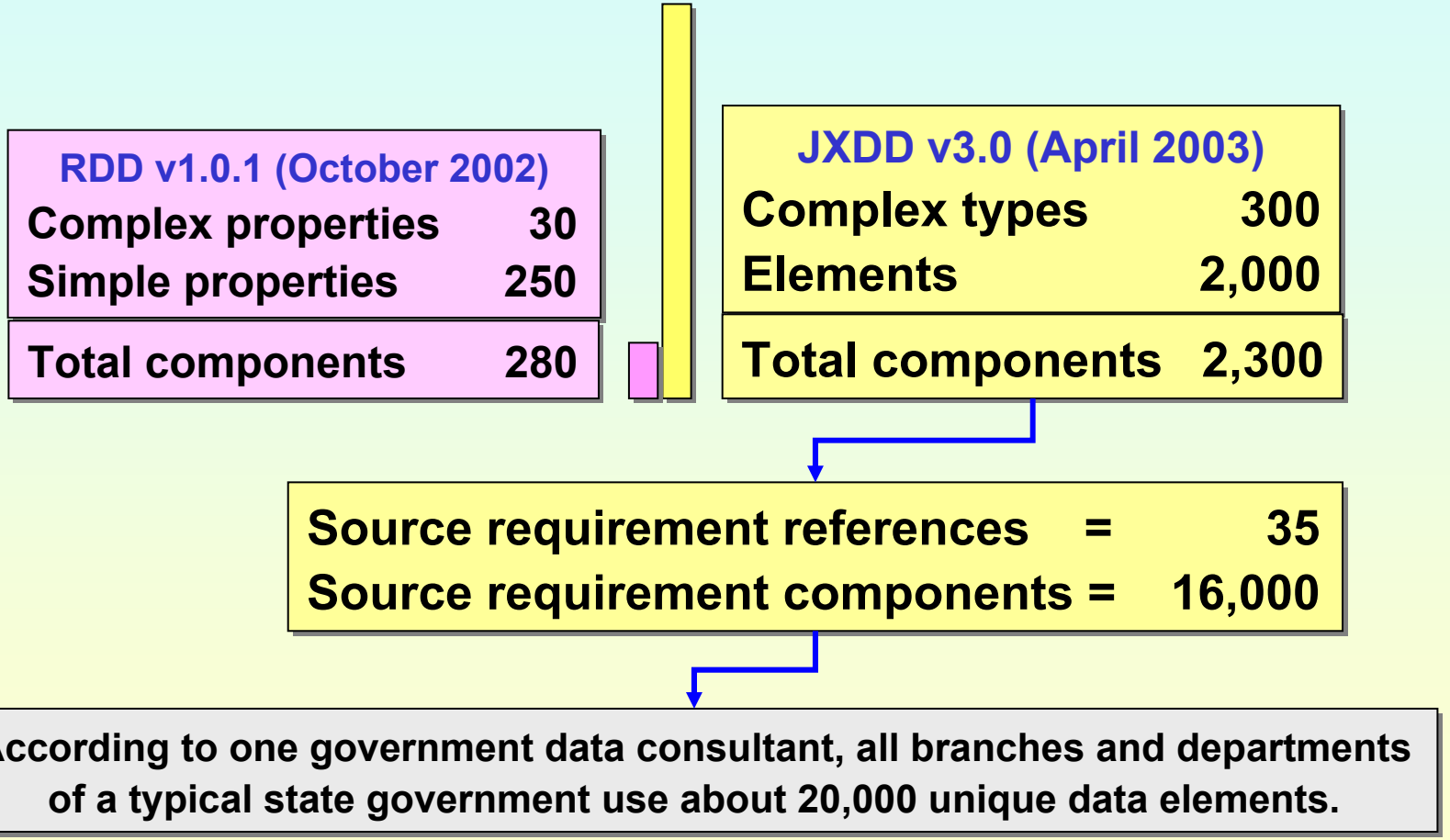


Justice XML Structure Task Force

April 2003


Assembling the Pieces
Activity
Investigate
Incident Rpt
Incident
Arrest
Arrest Warra
Arrest Rpt
Book
Confine
Charge
Prosecute
Bond
Booking Rpt
Charge Doc
Sentence Order
Case File
Protect Order
Indict
Adjudicate
Sentence
Fine
Incarcerate
Supervise
Parole
Pardon
Release
Pers
Org
Prop
Loc
Activity
???
Cnt
Relate
Doc
Doc.xsd
Justice XML Structure Task Force
April 2003

Justice XML Data Dictionary v3.0 Database Statistics



XML Namespace and Versioning

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema
   targetNamespace=  "http://www.it.ojp.gov/jxdd/prerelease/rap/2.3.0/rap.xsd"
   xmlns:xsd=        "http://www.w3.org/2001/XMLSchema"
   xmlns:jxdd=       "http://www.it.ojp.gov/jxdd/prerelease/3.0.0.0/jxdd.xsd"
   xmlns:rap=        "http://www.it.ojp.gov/jxdd/prerelease/rap/2.3.0/rap.xsd">

<xsd:import
   namespace=        "http://www.it.ojp.gov/jxdd/prerelease/3.0.0.0/jxdd.xsd"
   schemaLocation=   "http://www.it.ojp.gov/jxdd/prerelease/3.0.0.0/jxdd.xsd" />

   <xsd:element name="RapSheet">
      . . .
</xsd:schema>
```

Basic Concepts and Terminology

- XML Types define data structure
- XML Elements define data semantics

XML schema:	Concept:
• Type	• Object/Class
• Element	• Property

XML instance:	Reality:
• TagName	• Instance of Class
• Value of TagName	• Value of Property

Named Types *vs.* Elements

Question: Why define standard named types?

Answer: At times you will want to compare (similar) object instances with different semantic meanings but with same syntax and structure. If they are of the same type then you can easily compare or operate on them.

Example: *ArrestDate* and *ReleaseDate* have different semantic meaning. But it is easier to compute a time interval between them if they are both of the same data type (date).

Question: Why define standard elements?

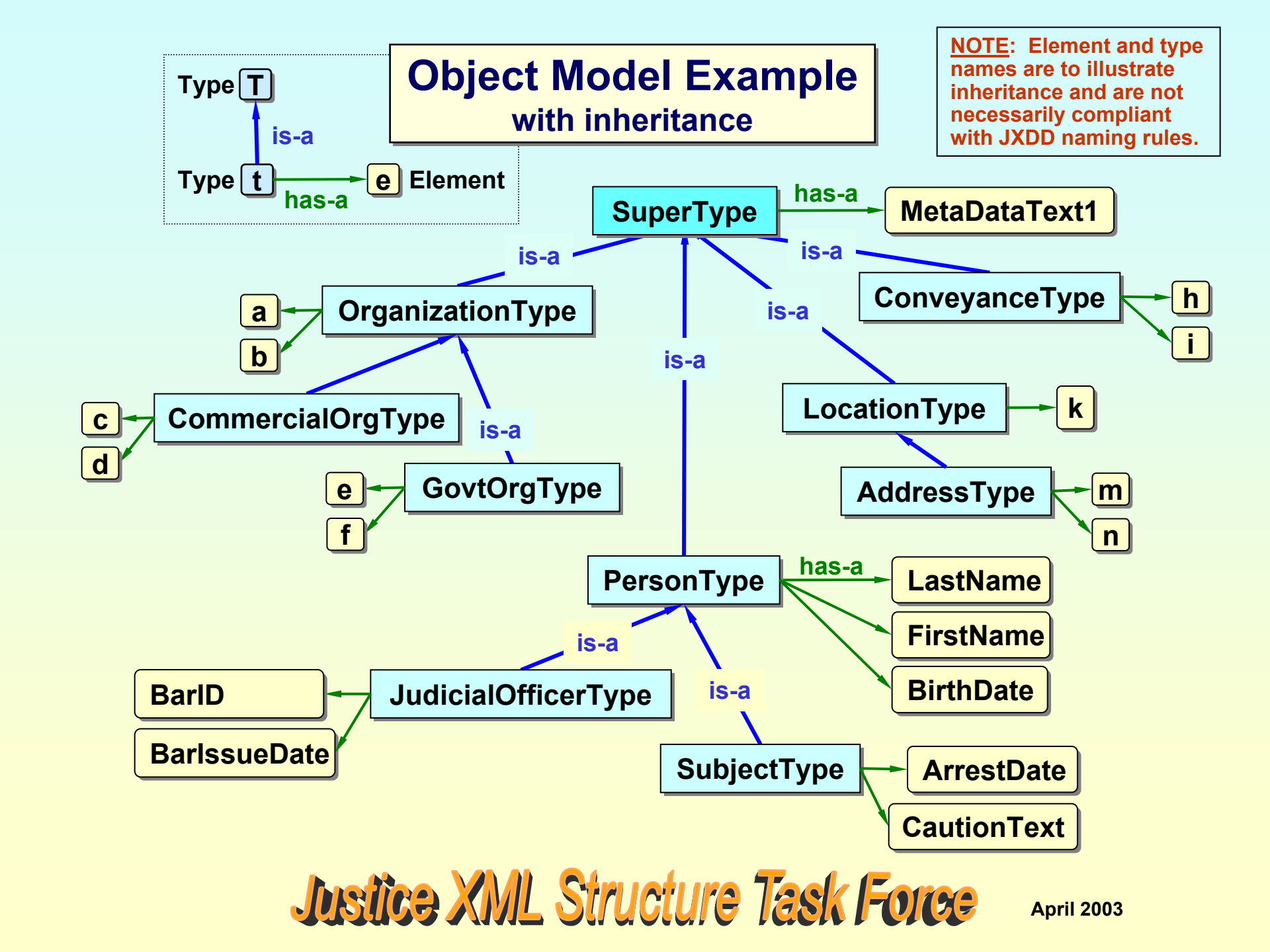
Answer: To discourage different element names for the same data concept (instance type). Users want to recognize semantically equivalent elements (that have same meaning). Also, enables users to define standard relationships and relate data objects more easily.

Examples: Are *SentencingOrder* and *DispositionOrder* the same?
How can software understand the difference between *OrganizationID* and *AgencyID*?

Question: Why is inheritance useful?

Answer: Organizes objects by their common properties (elements).
Allows software to treat objects with common properties in a uniform way.
Object types may share a common definition (eliminates duplicate definitions).
Extension mechanism for adding new properties is intuitive (the way we think).

Example: Can treat all "conveyances" (vehicles) in a uniform way.


Object Model Example
with inheritance
NOTE: Element and type names are to illustrate inheritance and are not necessarily compliant with JXDD naming rules.
Type T
is-a
Type t
has-a
e Element
SuperType
has-a
MetaDataText1
is-a
is-a
OrganizationType
a
b
ConveyanceType
h
i
is-a
is-a
CommercialOrgType
c
d
is-a
GovtOrgType
e
f
LocationType
k
AddressType
m
n
PersonType
has-a
LastName
FirstName
BirthDate
is-a
is-a
JudicialOfficerType
BarID
BarIssueDate
is-a
SubjectType
ArrestDate
CautionText
Justice XML Structure Task Force
April 2003

Rules for Object Model Extension

To ensure inheritance hierarchy integrity and consistency:

Webb's Rules

1. **A derived type may add (extension) additional fields (elements / attributes) to its base type.**
2. **A derived type may restrict one or more fields of its base type, but only so that a derived field is a subset of the field of the base type.**

   Examples: A derived type may …

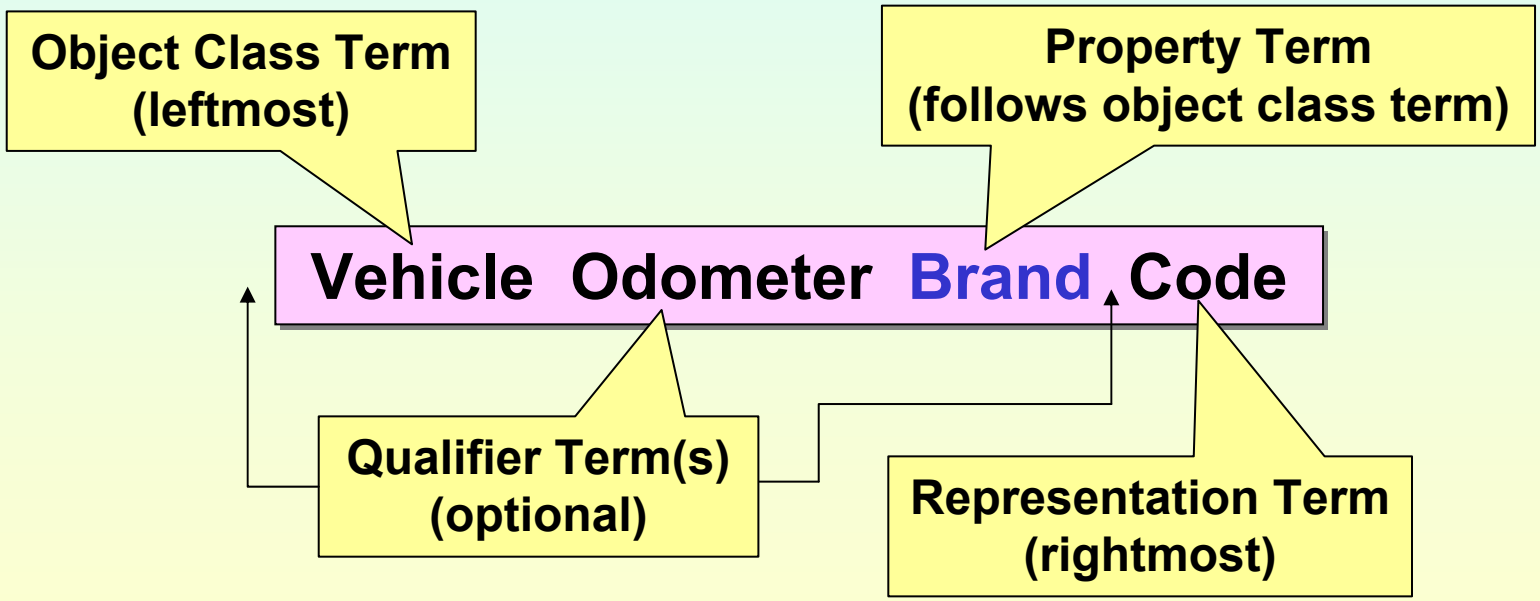
   - **restrict an enumeration from a large set of options to a smaller set of options, as long as every option in the derived set appears in the base set.**
   - **remove a field of the base type, but only if the field is optional in the base type.**
   - **require a field to appear, but only if the field is optional or required to appear in the base type.**
3. **A derived type may not modify a field of its base type such that it violates the constraints of its base type.**

   Examples: A derived type may NOT …

   - **add additional enumerations to a field.**
   - **remove a field that is required by its base type.**
   - **modify the type of a field of its base type.**

ISO Standard 11179
Data Element Naming Syntax

ISO 11179 standardizes data dictionary design, names, definitions.



Representation Terms
(ebXML Core Component Tech Spec v1.9)

1. Amount – Monetary value with units of currency.
2. BinaryObject – Set of finite-length sequences of binary octets.
   (secondary: Graphic, Picture, Sound, Video)
3. Code – Character string that for brevity represents a specific meaning.
4. DateTime – Date+time; point in time.
   (secondary: Date, Time)
5. Identifier – Character string used to establish identity of, and uniquely distinguish one instance of an object within an ID scheme.
6. Indicator – Boolean (exactly two mutually exclusive values).
7. Measure – Numeric value determined by measurement with units.
8. Numeric – Assigned or determined by calculation.
   (secondary: Value, Rate, Percent)
9. Quantity – Non-monetary numeric value or count with units.
10. Text – Character string generally in the form of words.
    (secondary: Name)

ISO Standard 11179 Element / Tagname Issues

Syntax: [Q] Object [Q] Property [Q] Representation

Strict adherence to 11179 results in

- **Longer data names (although easily translated by machines).**
- **Potential need for compression or translation (through tables).**
- **More elements (more for humans to remember, apps to understand).**
- **Precise element semantics (for a large-scale, multi-use standard).**
- **Reduced element dependence on context in instances.**
- **Elements are understood in isolatation or within context (e.g. in a document or complex element container)**

XML.gov Federal Guideline requires 11179-compliant tagnames.

If object term is omitted:

- **XML nesting (or XPath names) can provide context in many instances.**
- **Additional reuse of tagnames (they are independent of object context)**
- **Length savings is minimal (object term is usually a single word).**
- **UBL moves tagname context (object term) into schema definitions.**

ISO Standard 11179 – Example (1)

Syntax: [Q] Object [Q] Property [Q] Representation

```
<Person>
    <PersonName>
        <PersonLastNameText>Kindl</PersonLastNameText>
        <PersonFirstNameText>Mark</PersonFirstNameText>
    </PersonName>
    <PersonEyeColorCode>BRN</PersonEyeColorCode>
    <PersonTaxID>222334444</PersonTaxID>
</Person>
```

```
<Person>
    <Name>
        <LastNameText>Kindl</LastNameText>
        <FirstNameText>Mark</FirstNameText>
    </Name>
    <EyeColorCode>BRN</EyeColorCode>
    <TaxID>222334444</TaxID>
</Person>
```

XPath: /Person/Name/FirstNameText

ISO Standard 11179 – Example (2)

Syntax: [Q] Object [Q] Property [Q] Representation

```
<ContainerElement>
    <PersonName>
        <PersonLastNameText>Kindl</PersonLastNameText>
        <PersonFirstNameText>Mark</PersonFirstNameText>
    </PersonName>
    <PersonEyeColorCode>BRN</PersonEyeColorCode>
    <PersonTaxID>222334444</PersonTaxID>
</ContainerElement>
```

```
<ContainerElement>
    <Name>
        <LastNameText>Kindl</LastNameText>
        <FirstNameText>Mark</FirstNameText>
    </Name>
    <EyeColorCode>BRN</EyeColorCode>
    <TaxID>222334444</TaxID>
</ContainerElement>
```

Xpath: /ContainerElement/Name/FirstNameText

Person Context Missing

External Enumerations (codes)

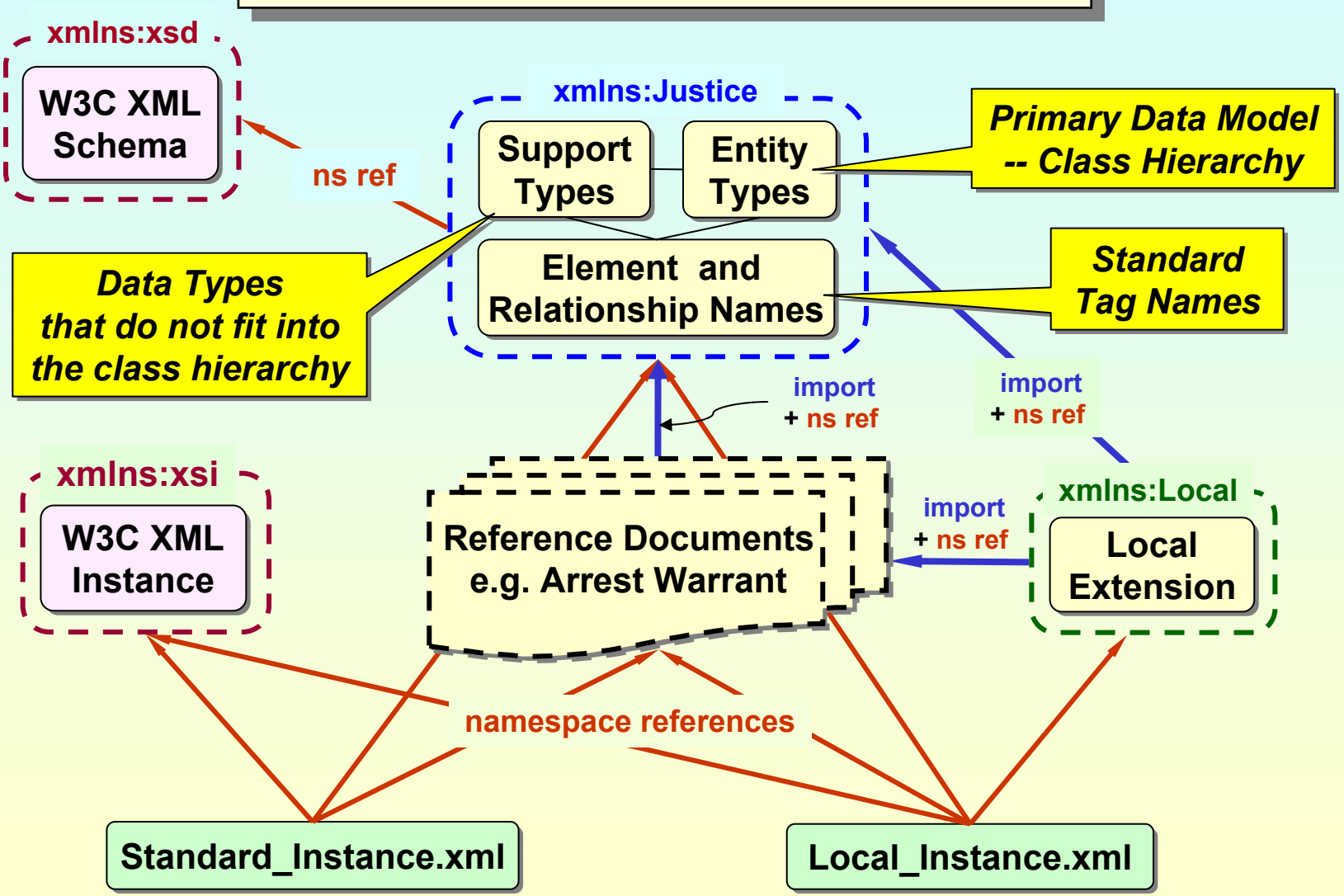
XML Instance

```
<?xml version="1.0" encoding="UTF-8"?>
<InstanceRootElement
    xmlns="http://www.it.ojp.gov/jxdd/beta/3.0.0"
    xmlns:ncic="http://www.it.ojp.gov/jxdd/beta/ncic/1.0.0"
    xmlns:aamva="http://www.it.ojp.gov/jxdd/beta/aamva/1.0.0"
    xmlns: . . .  [ reference additional tables as needed ]
    ... >
    <BoatHullMaterialCode>
        <ncic:HUL>ML</ncic:HUL>
        <aamva:BHMC>Fe</amvaa:BHMC>
    </BoatHullMaterialCode>
    <BoatHullMaterialText>
        Rusted Iron
    </BoatHullMaterialText>
...
</InstanceRootElement>
```

(1) Avoids maintenance of numerous code tables
(2) Allows use and validation of any code table
(3) Preserves code ownership
(4) Provides option for literal
(5) Similar to UBL method

BUT ... must have the codes!

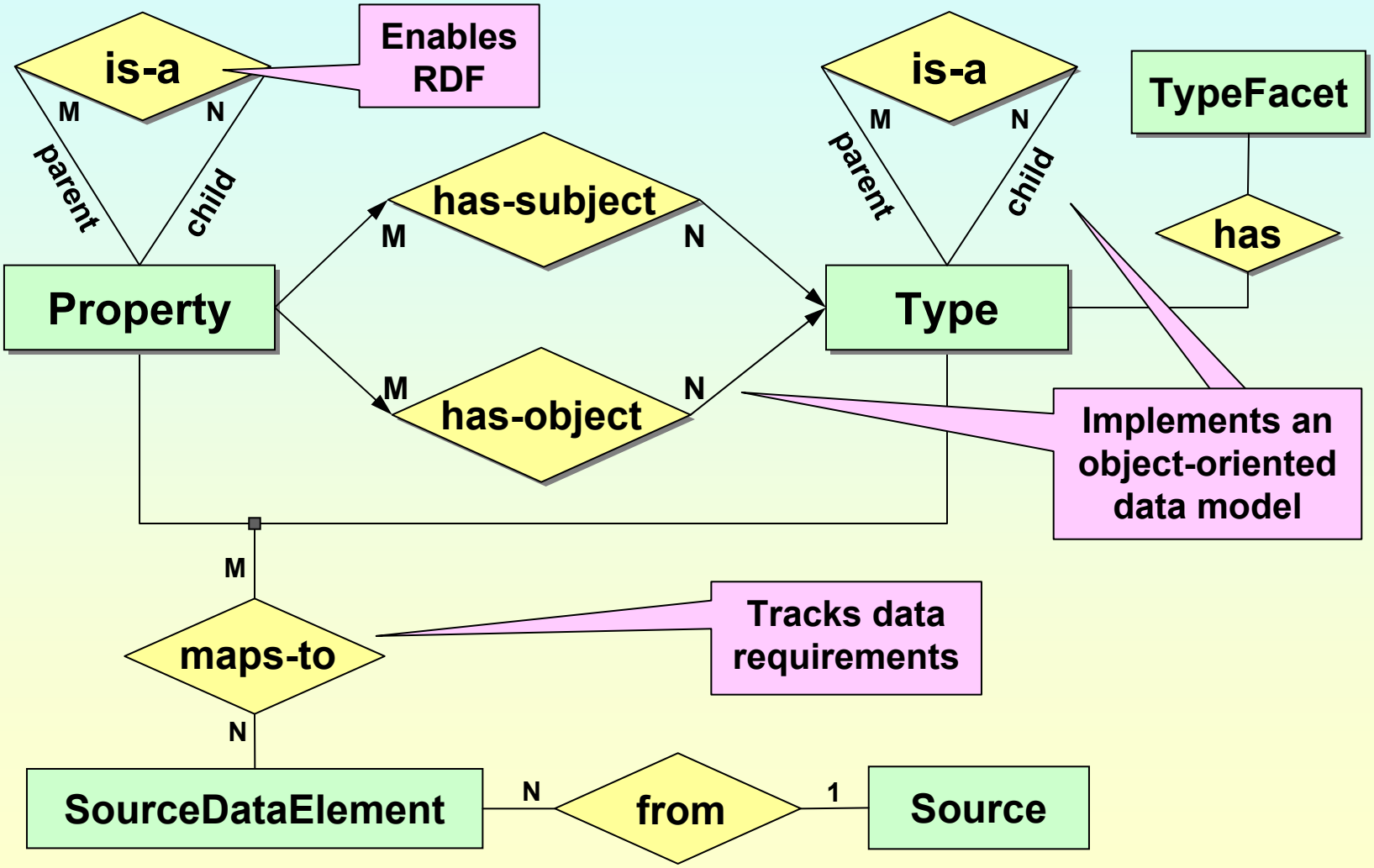
Schema Reference Architecture



Justice XML Data Model
Functions and Capabilities

- Represents an object-oriented XML data model.
- Enables forms-based maintenance / reconfiguration.
- Automatically generates XML Schema for the JXDDS.
- Will automatically generate equivalent RDF Schema.
- Stores and maps data element requirements; this enables
  - Tracing and tracking to source data components.
  - Measurement by source data requirements.
- Maintains metadata for XML data dictionary registry.
- Provides search filters, maintenance forms, and tools.

Entity-Relationship Diagram for Justice XML Model

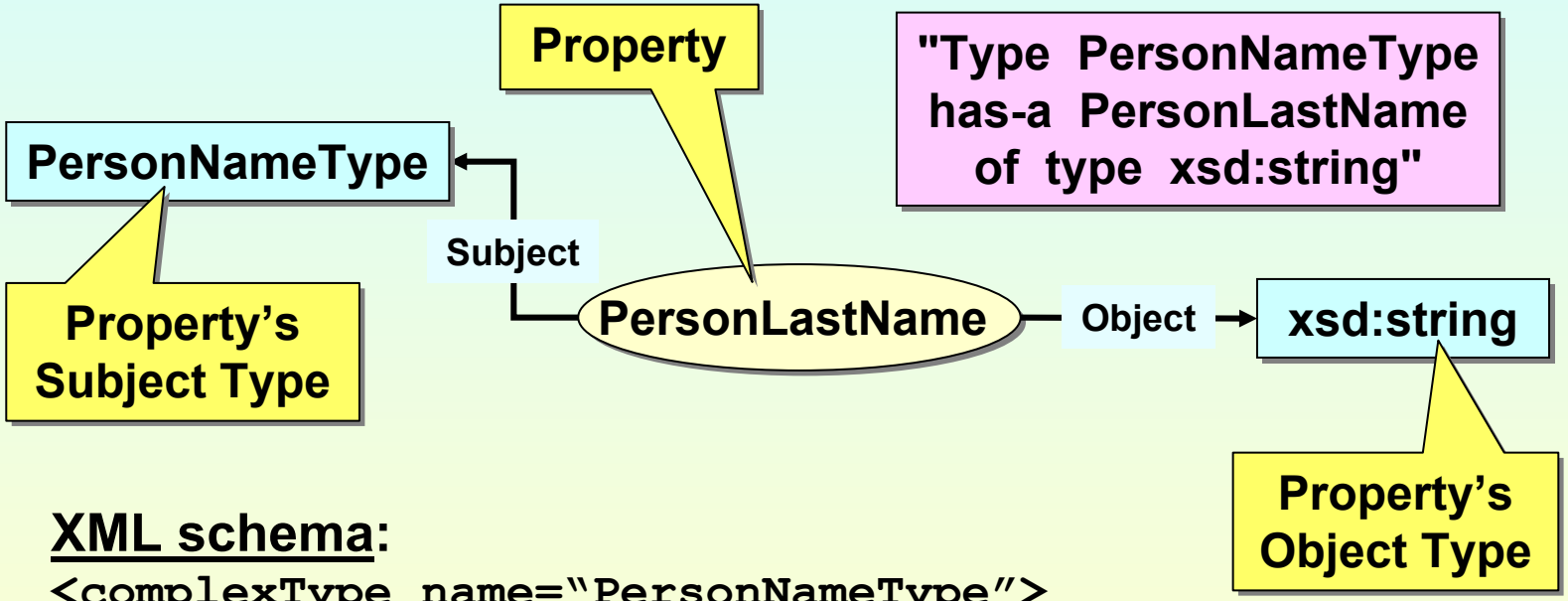


Justice XML Structure Task Force

April 2003

Conceptual Basis for the Justice XML Data Model

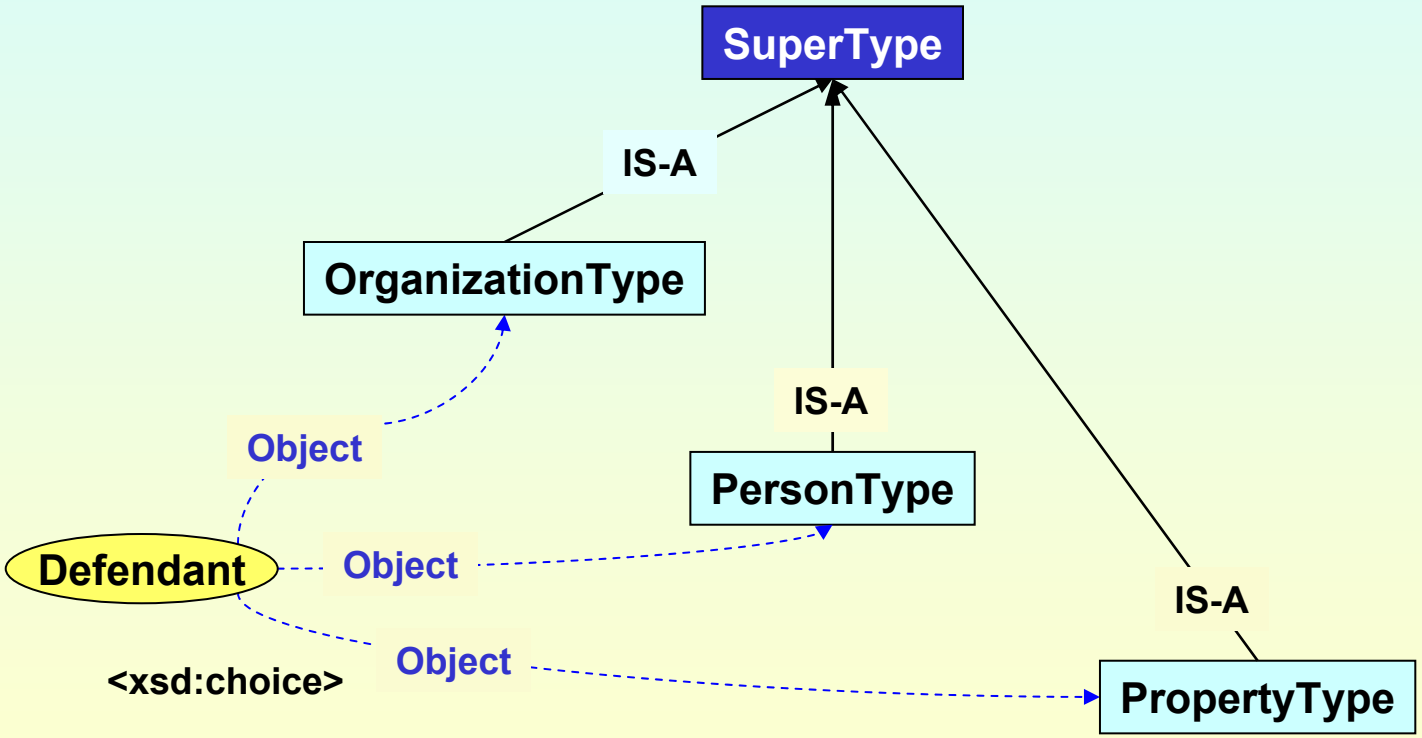
A simple example



XML schema:

```
<complexType name="PersonNameType">
  <sequence>
    <element name="PersonLastName" type="xsd:string"/>
  </sequence>
</complexType>
```

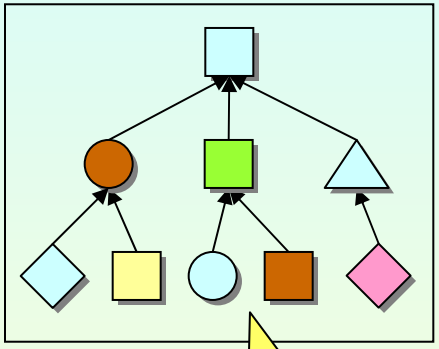
A Property (Defendant) with Multiple Object Types
(This example implements Court Filing *Actor*)



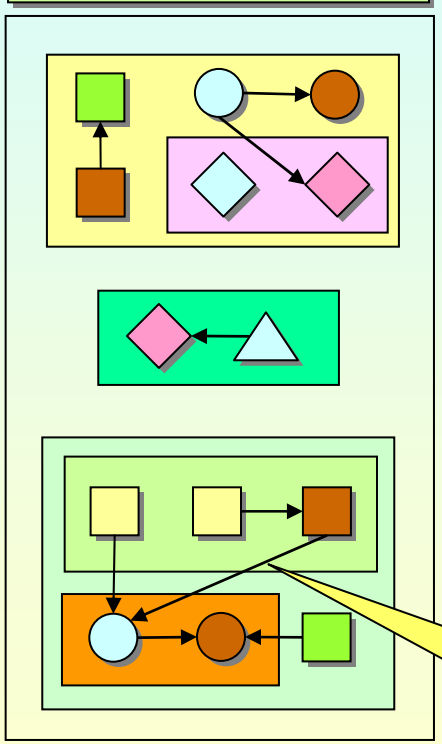
Primary and Secondary Relationships

Primary Relationships

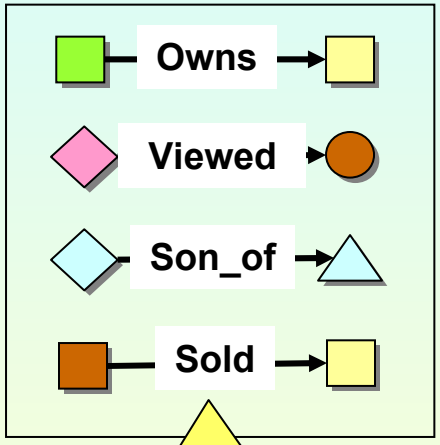
Data Objects



Containers



Secondary Relationships



Object class hierarchy built from "is-a" and "has-a" primary relationships

Explicitly defined; globally understood

Documents & transactions create local relationships

Secondary Relationships

Why are they necessary?

When should they be used?

Use secondary relationship form of a property when:

- **Convenient; do not want to "pollute" an object with properties that are not inherently associated with it.**
- **Passing a set of standard objects (such as documents) within a container and user must explicitly relate imbedded objects between documents.**
- **Impossible to create primary property relationship for a given situation.**

Primary Relationships (1)

1

Two object instances:

```
<Person>
    <Name>Bill</Name>
</Person>

<Person>
   <Name>Jane</Name>
</Person>
```

2

Primary property, by nesting:

```
<Person>
    <Name>Bill</Name>
    <Sister>
        <Name>Jane</Name>
    </Sister>
</Person>
```

Primary Relationships (2)

3

Primary, by reference:

```
<Person id="bill">
    <Name>Bill</Name>
</Person>

<Person>

   <Name>Jane</Name>
    <Brother ref="bill"/>
</Person>
```

4

Primary, by reference (*ad nauseam*):

```
<Person id="bill">
    <Name>Bill</Name>
</Person>

<Person>
    <Name>Jane</Name>
    <Brother ref="bill"/>
    <Sister ref="sue"/>
    <Uncle ref="lou"/>
    <SecondCousinOnceRemoved
   ref="jill"/>
    . . .
</Person>
```

Secondary Relationships

5

Secondary relationship:

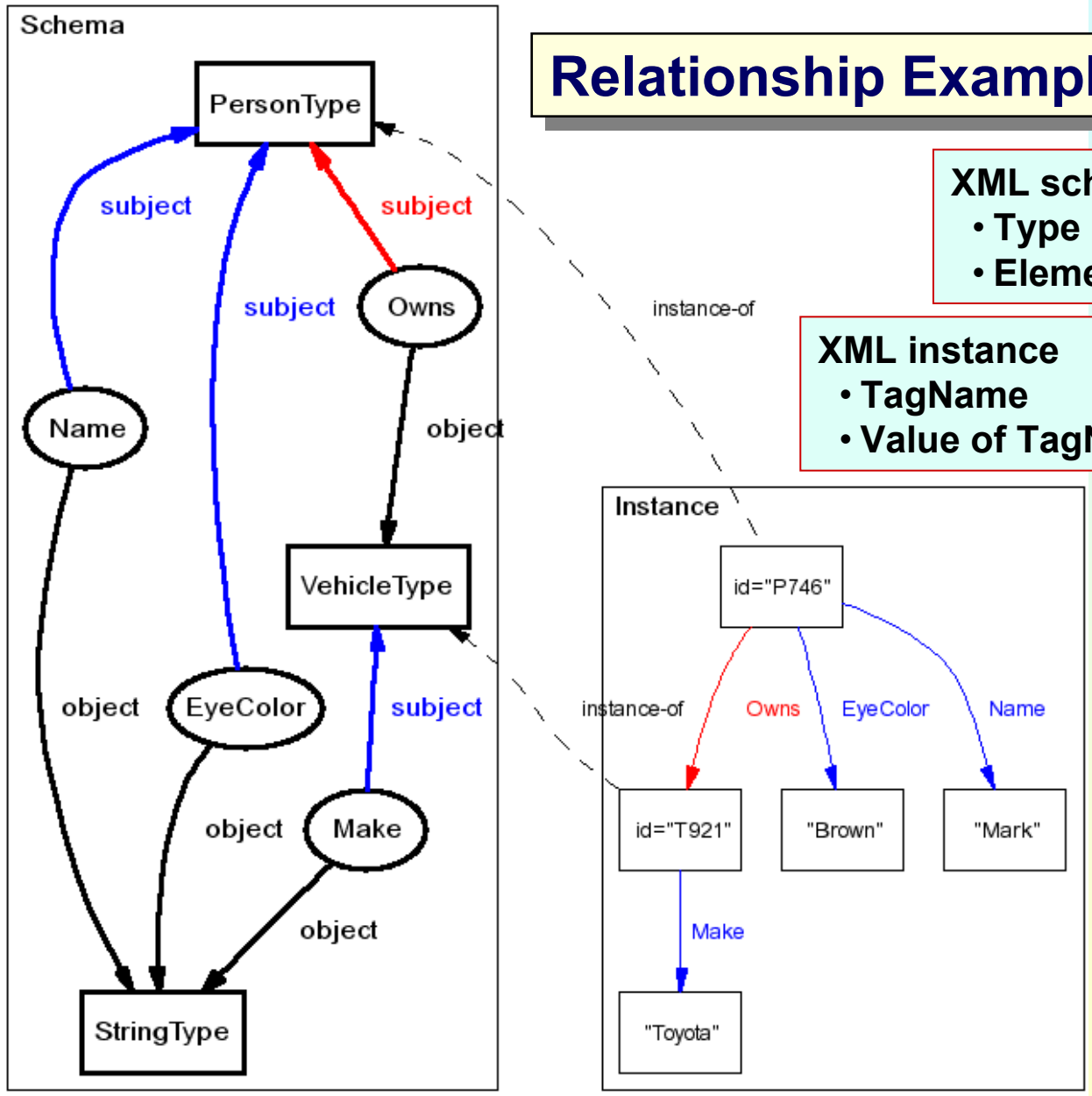
```
<Person id="bill">
   <Name>Bill</Name>
</Person>

<Person id="jane">
   <Name>Jane</Name>
</Person>

<Person id="sue">
   <Name>Sue</Name>
</Person>

<BrotherRelationship subject="jane" object="bill"/>
<SisterRelationship   subject="bill"  object="jane"/>
<SisterRelationship   subject="jane" object="sue"/>
```

Relationship Examples



XML schema (concept):
- **Type (Object/Class)**
- **Element (Property)**

XML instance (reality):
- **TagName (Instance of Class)**
- **Value of TagName (Value of Property)**

Primary
- Name
- EyeColor
- Make

Secondary
- Owns

Semantic Web
The 3rd Generation Internet

Agent Smith ("*The Matrix*" ©1999 Warner Bros)

Intelligent Software Agents → **decision + action**

3

- **OWL** → **Web Ontology Language (based on DAML+OIL)** → **semantics + inferencing**
- **RDF / RDFS** → **Resource Description Framework and Schema** → **relationships + metadata**
- **XML** → **eXtensible Markup Language** → **syntax + content**

Knowledge

2

- **HTML** → **HyperText Markup Language**
- **HTTP** → **HyperText Transfer Protocol**

→ **presentation**

1

- **TCP / IP** → **Transmission Control Protocol / Internet Protocol** → **network transmission**

Justice XML Structure Task Force

April 2003

Resource Description Framework (RDF) Conceptual Model

The conceptual model for the database representing the object-oriented Justice Data Dictionary looks like RDF.

This is no coincidence!