

CBRN Data Model

**Presentation to:
Net-Ready Sensors: The Way Forward
Oak Ridge National Laboratory**

August, 2, 2006

Distribution authorized to U.S. Government agencies and their contractors only; administrative or operational use; July, 2006. Other requests shall be referred to the Data APM (NSA/JO), Naval Postgraduate School, 1411 Cunningham Road, Monterey, CA 93943.

**Professor Tom Johnson
Naval Postgraduate School
JPM IS Data APM
thjohnso@nps.edu**

Briefing Agenda

- **CBRN Data Model 101**
- **Why a Common CBRN Data Model?**
- **Model's Structure**
- **History**
 - Releases
 - Complexity
- **CBRN Data Model Products**
- **Model Structure**
 - Overview
 - CBRN Event
 - CB Agents
 - Material and Facility Representations
 - Sensor Representation
- **Service Oriented Architectures (SOA) & The CBRN DM**
- **Implementation Strategy**
- **Summary**

Why a Common Data Model?

- Enables Data Interoperability & Re-use.
- Data Model facilitates **Common CBRN Domain Representation**
- Data Model facilitates Interoperability:
 - Scalability and extensibility
 - Specifies meaning and structure of data
 - Specifies relationships among data
 - Provides open standard basis for Data Exchange
 - eXtensible Markup Language (XML)
- *Data Model is consistent with DoD Net-centric Data Strategy and SOA*

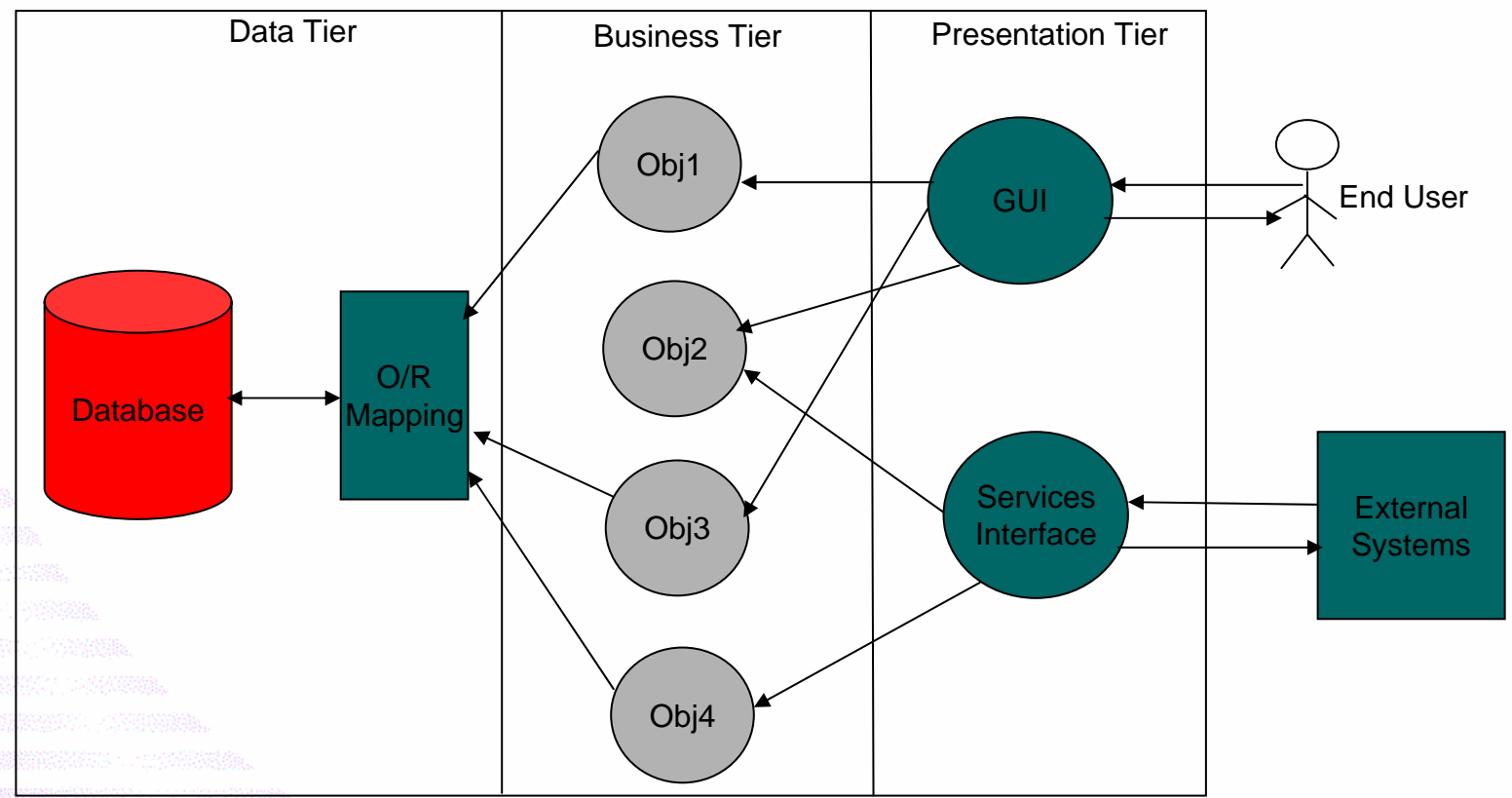
DATA MODEL REPRESENTS A CONCEPTUAL MODEL OF CBRN BATTLESPACE RELATIONSHIPS AND COMMON SEMANTICS AND SYNTAX. THE MODEL *DOES NOT* REPRESENT A CANNED SOFTWARE SOLUTION FOR SYSTEM INTEROPERABILITY.

Database to Data Model Equivalents

<u>Physical</u>	<u>Logical</u>
Database	Data Model
Table	Entity
Column	Attribute
Row	[doesn't go this low]

*Think of Data Model as a **Blueprint** for a Database
Same design principles apply to both...*

CBRN-IS Application Data Perspective



History of the CBRN Data Model, Part 1

- **2002: White Paper written on Common Data Representation.**
- **2003: Data Model Development begun.**
 - Several preliminary drafts released.
- **2004: Release 1.0 and 1.1 released**
 - Most major areas represented in Release 1.0.
 - **Release 1.1:**
 - Added Agent Simulant Knowledgebase (ASK) attributes.
 - Added remainder of ATP-45 attributes.
 - Added metadata entities and attributes.
 - Adopted UK spelling.
 - ATP-45 Panel endorsed data model as ‘Extended C2IEDM.’



History of the CBRN Data Model, Part 2

- **2005: Release 1.2 and 1.3 released**
 - **Release 1.2**
 - Added enhanced Configuration Management tracking.
 - Did significant remodeling of material properties.
 - Added CBRN equipment entities.
 - Added radiation exposure guidelines.
 - Added many HPAC variables.
 - Added tracing of requirements to entities.
 - Removed reserved words from physical model.
 - Began participating in MIP meetings to add ATP-45 attributes to the JC3IEDM (C2IEDM).
 - JRO issued JSAP Tasker to review the data model.
 - Resulted in 266 Change Proposals

History of the CBRN Data Model, Part 3

- **2005, cont.:**

- **Release 1.3**

- **Added over 600 additional transport & dispersion variables.**
 - **Significant work done for chemical sensors and biological collectors.**
 - **Hosted first technical review focusing on Sensor entities and attributes in November, 2005.**
 - **Added US Mission Oriented Protective Posture (MOPP) Levels.**
 - **Adopted new and revised JC3IEDM class words.**
 - **Remodeled CBRN measurements and control features.**
 - **Added support for capture of row-level metadata.**
 - **Added support for Population.**
 - **Extensive improvements to definitions.**

Data Model Size (Complexity?)

CBRN Data Model has grown significantly with each release:

Release Number	Number of Entitles	Number of Attributes	Number of Relationships
v1.1	201	1302	311
v1.2	329	1940	602
v1.3	441	3327	1200
v1.4	446	3611	1317

Growth from Release 1.3 to Release 1.4 was less dramatic due to a great deal of remodeling and consolidation that partially offset the new additions.

Release 1.4 Enhancements

- **Remodeling of CBRN Event Subtypes**
 - A NUCLEAR-FACILITY-INCIDENT is now a subtype of RADIOLOGICAL-EVENT.
 - Based on community input at January 2006 Technical Review.
 - NUCLEAR-WEAPON-EVENT has subtypes of NUCLEAR-WEAPON-ACCIDENT and NUCLEAR-WEAPON-DETONATION.
- **HPAC Integration**
 - Variables from Hazard Prediction Assessment Capability (HPAC) have been more tightly integrated into the model.
- **Medical Representation**
 - Vectors, Fomites and Disease Vehicles have been added to the model as CAPABILITYs.
 - These CAPABILITYs can apply to Types (species or class of object) or Items (specific person / animal / object).
 - MILITARY-EXPOSURE-GUIDELINES have also been added.

Release 1.4 Enhancements, cont.

- **Sensor Representation**
 - Radiation Portal Monitor (RPM) entities have been added.
 - Support has also been added for Remote Sensor Panels and Cameras that are used in conjunction with RPMs.
- **Geographic Feature**
 - Updated to align with the DIGEST standard, per JC3IEDM.
 - An URBAN-AREA entity was added at the request of the community.
- **Plants, Animals, and Military Working Animals**
 - Have been added to the data model.
- **Business Rules Document**
 - Defines permissible combinations of domain values for interdependent attributes.
 - Developed at the request of the community.

Future Direction: Release 1.5*

- **Material Properties**
 - Improve the representation of biological and radioactive materials.
- **Concentration**
 - Changes/ additions related to Canadian CP for JC3IEDM.
- **Sensor Generalization**
 - Remodel sensor section of data model to make it more generic, and less specific to individual sensors.
 - As a result, data model will support a greater variety of sensors.
 - Plan to also implement a methodology to capture sensor-specific data.
- **Radiation Sensors**
 - Add support for other types of radiation sensors (besides Radiation Portal Monitors).

**Note that this is not an exhaustive list of all additions that will be included in Release 1.5.*

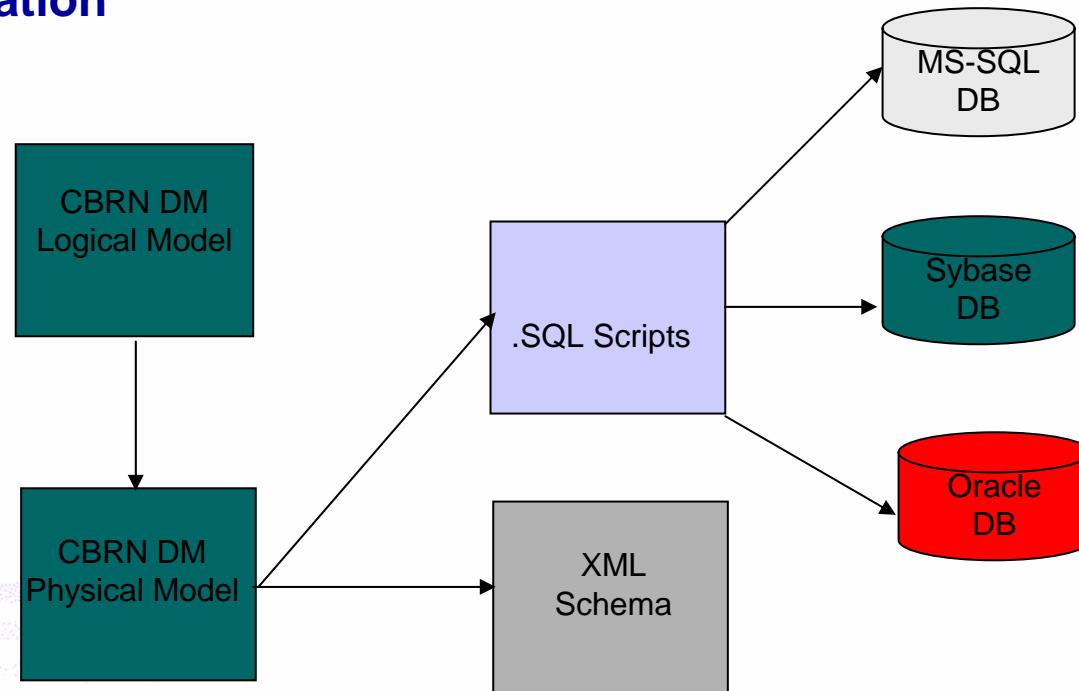
Future Direction: Release 1.5, cont.

- **Decontamination**
 - Add attributes for Decontamination and Restoration Materials and Equipment.
- **Medical**
 - Estimation of CBRN casualties.
 - Estimation of medical requirements.
 - Analysis of alternative medical courses of action.
- **Documents / Binary Objects**
 - Enhance support for description of documents or other binary objects stored in the database.
- **Representative Sample Data**
 - Based on use case(s).

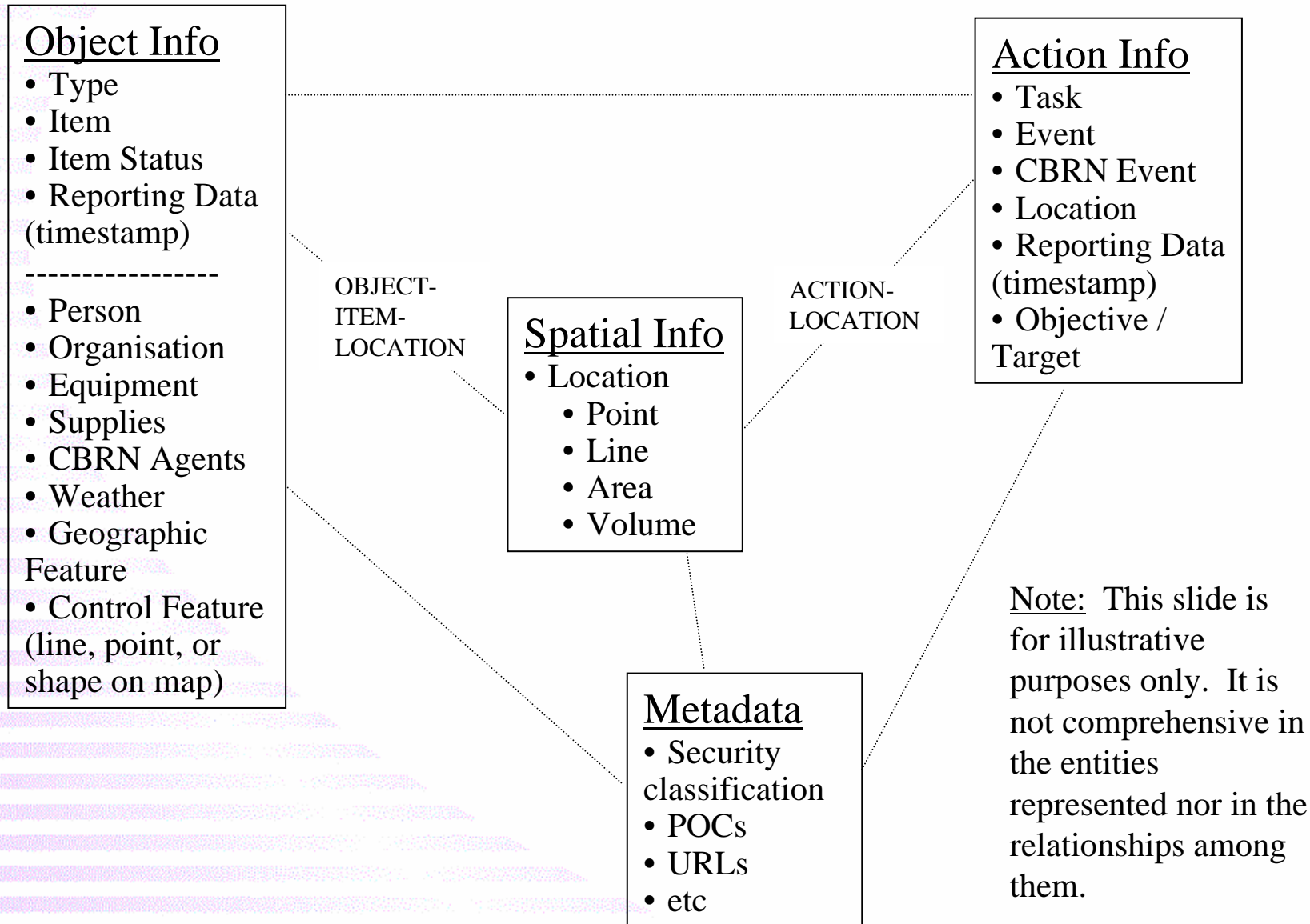
CBRN Data Products

CBRN Data Products:

- ERWIN Data Model
- CBRN XML Schema
- SQL Scripts
- Documentation



CBRN Data Model High-level Overview



CBRN Data Model OBJECT Overview

OBJECT-TYPE

- Information about the class.
- Information that is the same for each instance of the OBJECT.
- Example: Model Number

OBJECT-ITEM-TYPE

- Relates an OBJECT-ITEM to an OBJECT-TYPE..

OBJECT-ITEM

- Information about an instance of the OBJECT-TYPE.
- Example: Serial Number

OBJECT-ITEM-STATUS

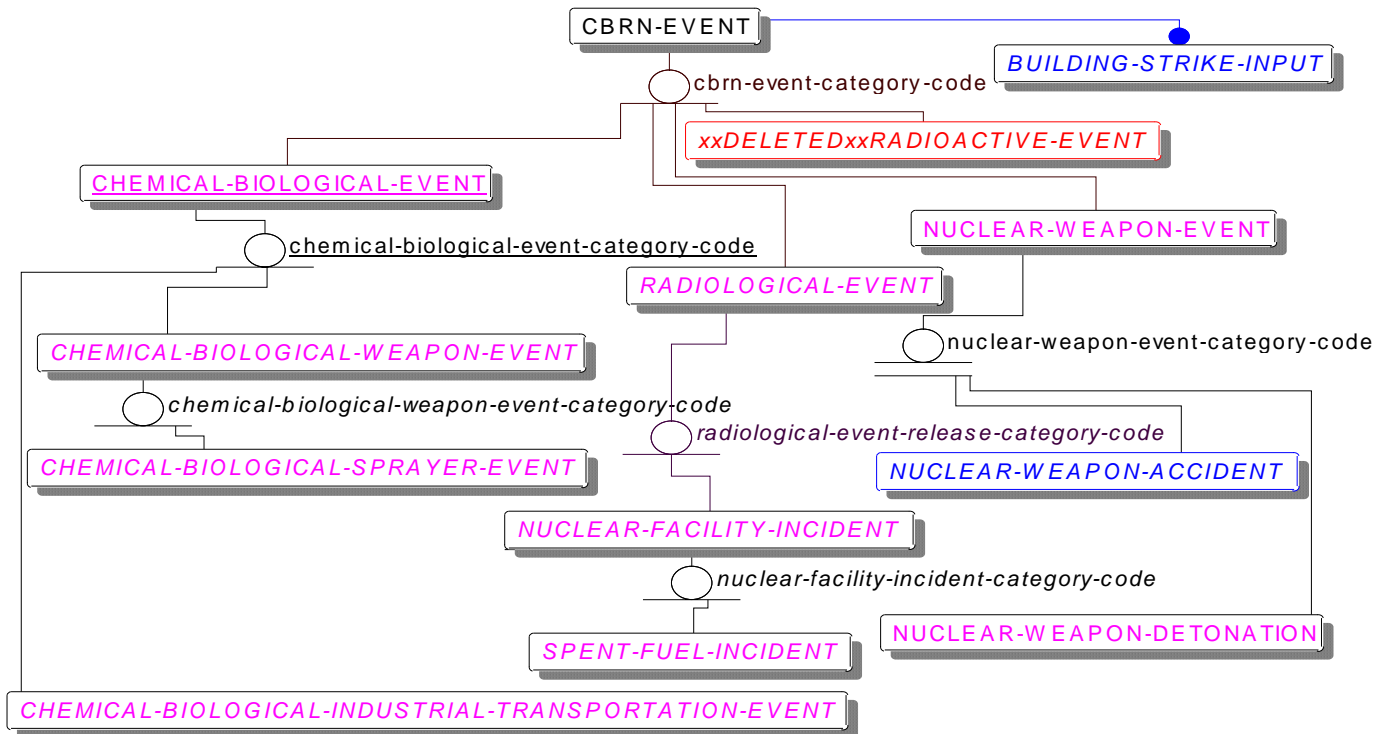
- Information about the current status of an OBJECT-ITEM.
- Example: Operational State

An OBJECT may be a:

- Person
- Organisation
- Equipment
- Supplies
- CBRN Agents
- Weather
- Geographic Feature
- Control Feature (line, point, or shape on map)

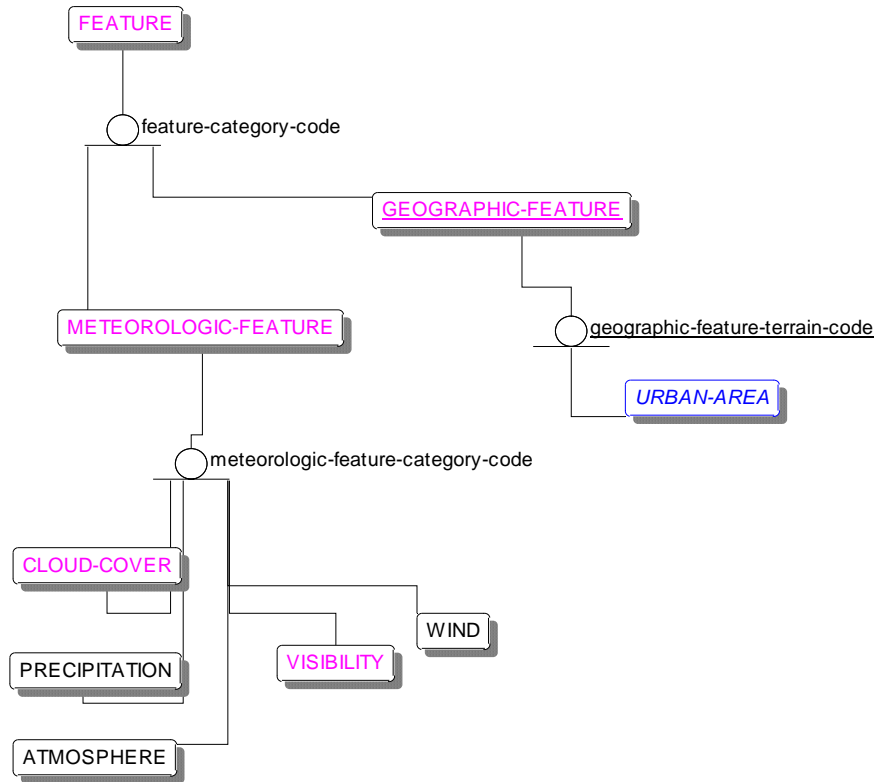
Note: This slide is for illustrative purposes only. It is not comprehensive in the entities represented nor in the relationships among them.

CBRN Event



Distribution authorized to U.S. Government agencies and their contractors only; administrative or operational use; January, 2006. Other requests shall be referred to the Data APM (NSA/JO), Naval Postgraduate School, 1411 Cunningham Road, Monterey, CA 93943.

Weather and Terrain Inputs



Distribution authorized to U.S. Government agencies and their contractors only; administrative or operational use; October, 2005. Other requests shall be referred to the Data APM, Naval Postgraduate School, 1411 Cunningham Road, Monterey, CA 93943.

Biological Materiel: Category and Subcategory

- Categories are from **FM 3-11.9**.
- **Biological-materiel-type-genetically-engineered-code.**
 - The specific value that indicates whether the **BIOLOGICAL-MATERIEL-TYPE** is the product of the “directed alteration or manipulation of genetic material.” [FM 3-11.9/ MCRP 3-37.1B/ NTRP 3-11.32/ AFTTP(I) 3-2.55, 10 January 2005, Glossary-15]

biological-materiel-type-category-code	biological-materiel-type-subcategory-code
Bioregulator	Not otherwise specified
	Not known
Pathogen	Bacterial
	Rickettsiae
	Viral
Prion	Not otherwise specified
	Not known
Toxin	Cytotoxin
	Neurotoxin
	Not known
	Not otherwise specified
Not known	[NULL]
Not otherwise specified	[NULL]

Structure for CHEMICAL-MATERIEL-TYPE

CHEMICAL-MATERIEL-TYPE

chemical-materiel-type-id (FK)

chemical-materiel-type-category-code

chemical-materiel-type-subcategory-code

chemical-materiel-type-chemical-abstracts-service-code

chemical-materiel-type-persistency-code

CHEMICAL-AGENT-TYPE

chemical-materiel-type-id (FK)

chemical-agent-type-category-code

chemical-agent-type-subcategory-code

MILITARY-CHEMICAL-COMPUND-TYPE

military-chemical-compund-type-id (FK)

- Entity: **MILITARY-CHEMICAL-COMPUND-TYPE**, defined as:
 - A CHEMICAL-MATERIEL-TYPE that is less toxic than chemical agents and is generally accepted for use in warfare. Military chemical compounds include materials such as respiratory irritant agents, riot control agents, smoke and obscurants, incendiary materials and military herbicides. The term excludes chemical agents.

Chemical Agent: Category and Subcategory Codes

chemical- materiel- type- category- code	chemical- agent-type- category- code (Mandatory)	chemical- agent-type- subcategory- code (Optional)	chemical- materiel- type- category- code	chemical- agent-type- category- code (Mandatory)	chemical- agent-type- subcategory -code (Optional)
<p><i>Categories and subcategories are from FM 3-11.9</i></p>	Blister agent	Arsenical blister agent	Chemical agent	Incapacitating agent	Deliriant agent
		Mustard agent			Depressant agent
		Urticant agent			Psychedelic agent
		Not known			Stimulant agent
		Not otherwise specified			Not known
	Blood agent	Cyanogen agent		Nerve agent	Not otherwise specified
		Not known			G-agent
		Not otherwise specified			V-agent
	Choking agent	Not otherwise specified			Not known

Military Chemical Compound: Category and Subcategory Codes

Categories and subcategories are from FM 3-11.9

chemical-materiel-type-category-code	military-chemical-compound-type-category-code (Mandatory)	military-chemical-compound-type-subcategory-code (Optional)
Military Chemical Compound	Incendiary agent	Hydrocarbon fuel incendiary
		Metal fuel incendiary
		Hydrocarbon-metal fuel incendiary
		Pyrophoric aluminium alkyl incendiary
		Not known
		Improvised incendiary
		Not otherwise specified
	Respiratory irritant agent	[NULL]
	Riot control agent	[NULL]
	Smoke and obscurant materiel	
Not otherwise specified		
Not known		

Sensor Representations

- Hyper link to Erwin Representation
- Hyper link to Data Dictionary

Attributes
✕

Entity: ▼ ...

Attribute ▲ ▼

- cbrn-event-id
- cbrn-event-category-code
- cbrn-event-subcategory-code
- cbrn-event-alarm-result-indicator-code
- cbrn-event-confirmation-test-indicator-cc
- cbrn-event-type-code
- cbrn-event-occurrence-qualifier-code
- cbrn-event-delivery-mechanism-code
- cbrn-event-initial-size-diameter-dimension
- cbrn-event-release-height-dimension
- cbrn-event-spill-size-code
- cbrn-event-arrhenius-constant-rate

General
Datatype
Definition
Note

✂
📄
📁
🔍

Definition:

The one-dimensional linear distance representing the diameter of the initial release/ effective range of the CBRN-EVENT. Unit of Measure = Metres.

Data Model Screen Shot– Datatype and Validation Rule

Example: cbrn-event-delivery-mechanism-code

Entity: CBRN-EVENT

Attribute: cbrn-event-delivery-mechanism-code

Datatype:* CHAR(6)

Valid: VR5031_deliv_mech

Default:* one-dash

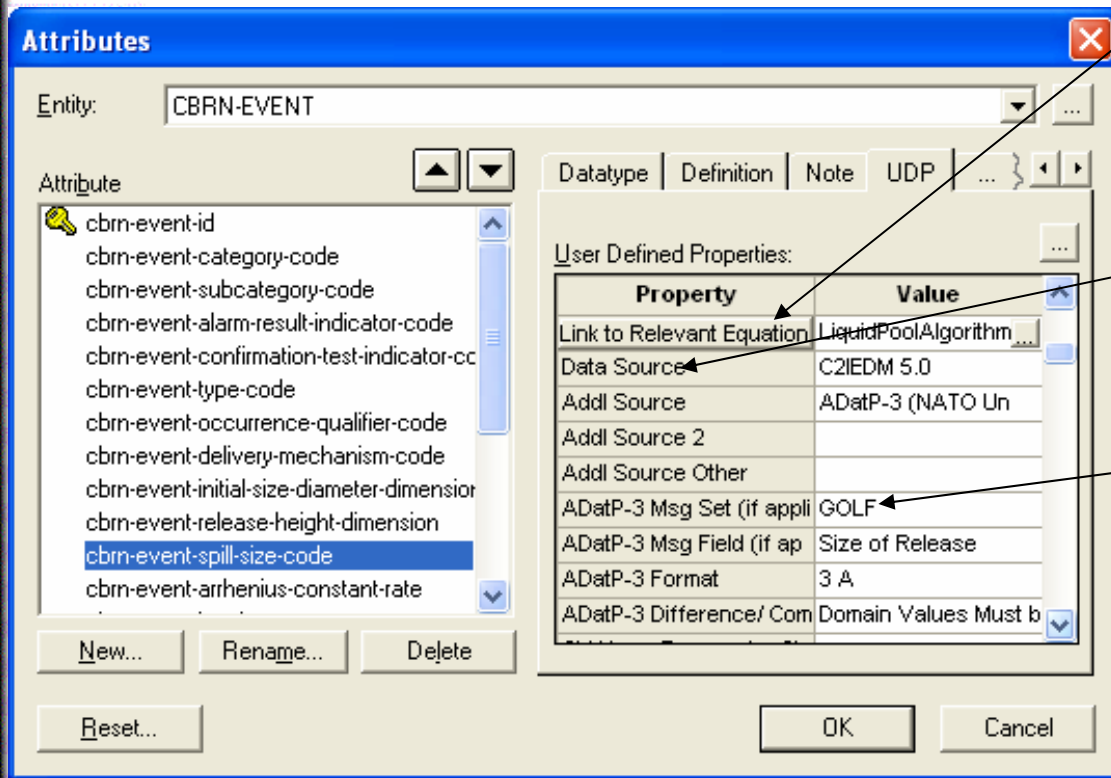
- Datatype is defined here
- Indicates that there is a validation rule associated with this attribute

Valid Value	Display Value	Definition
AIR	Aircraft	The delivery mechanism is an air
BOM	Bomb (delivering	The delivery mechanism is a bon
CAN	Cannon	The delivery mechanism is a car
MLR	Multiple Launche	The delivery mechanism is multi
MSL	Missile	The delivery mechanism is a mis
MOR	Mortar	The delivery mechanism is a mor

Valid Values List for Validation Rule for cbrn-event-delivery-mechanism-code

User-Defined Properties (UDP)

User-Defined Property (UDP) fields provide a great deal of flexibility to track information of interest to the user base



- **Link to Relevant Equation** will launch a web page of the equation(s) relevant to this attribute
- **Data Source** indicates the source of the data requirement (support exists for multi-source)
- **ADatP-3 (ATP-45) Msg Set, Field, Format, Difference** UDPs indicate that the spill size code can be found in the GOLF set where it is called Size of Release and consists of 3 alphabetic characters.

Equation Link for Mass Mean Diameter

Entity: CBRN-EVENT

Attribute: cbrn-event-spill-size-code

Property	Value
Link to Relevant Equation	LiquidPoolAlgorithm...
Data Source	C2IEDM 5.0
Addl Source	ADatP-3 (NATO Un)
Addl Source 2	
Addl Source Other	
ADatP-3 Msg Set (if appli)	GOLF
ADatP-3 Msg Field (if ap)	Size of Release
ADatP-3 Format	3 A
ADatP-3 Difference/ Com	Domain Values Must b

Click here (on *Link to Relevant Equation button*) to launch equation page

HPAC/SCIPUFF Liquid Pool Evaporation

Description

The evaporation material from of a liquid pool forms vapor puffs. The source parameters are dependent on the evaporation rate. Evaporation continues until the pool is depleted. The evaporation rate depends on the pool temperature.

HPAC Version

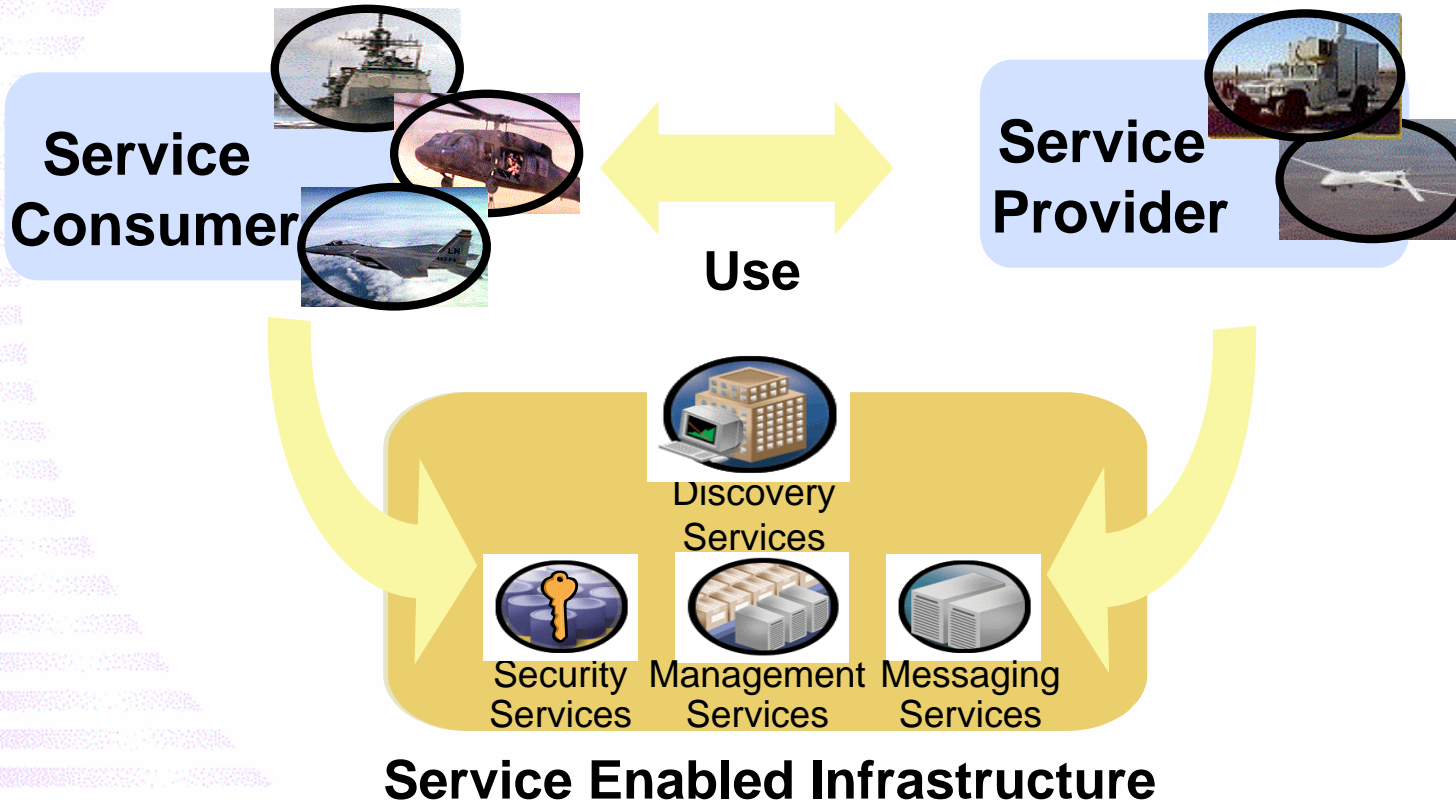
The spread parameters are given by:

$$W = \sqrt{A_{pool}}$$

(From SCIPUFF Technical Documentation Equation 7.11)(1)

Service Oriented Architectures (SOA) & The CBRN DM

Service Oriented Architecture (SOA)



SOA comprises the policies, practices, frameworks and standards that enable application functionality to be provided by a service provider and consumed as sets of services by a service consumer across a heterogeneous, networked environment.

SOA approach ...

Advantages:

- Allows war-fighters/customers direct access to the information, data they need to accomplish their job
- Enables developers to work independently & effectively by allowing them to work toward common set of interfaces without any knowledge of systems they are interacting with
- Take advantage of well know industry standards like UDDI, SOAP and XML
- Easily align with DOD's Net Centric strategy

Disadvantages:

- Performance may be impacted due to the distributed nature of SOA
- Standards continue to evolve and adaptation may be difficult for large programs with multiple projects

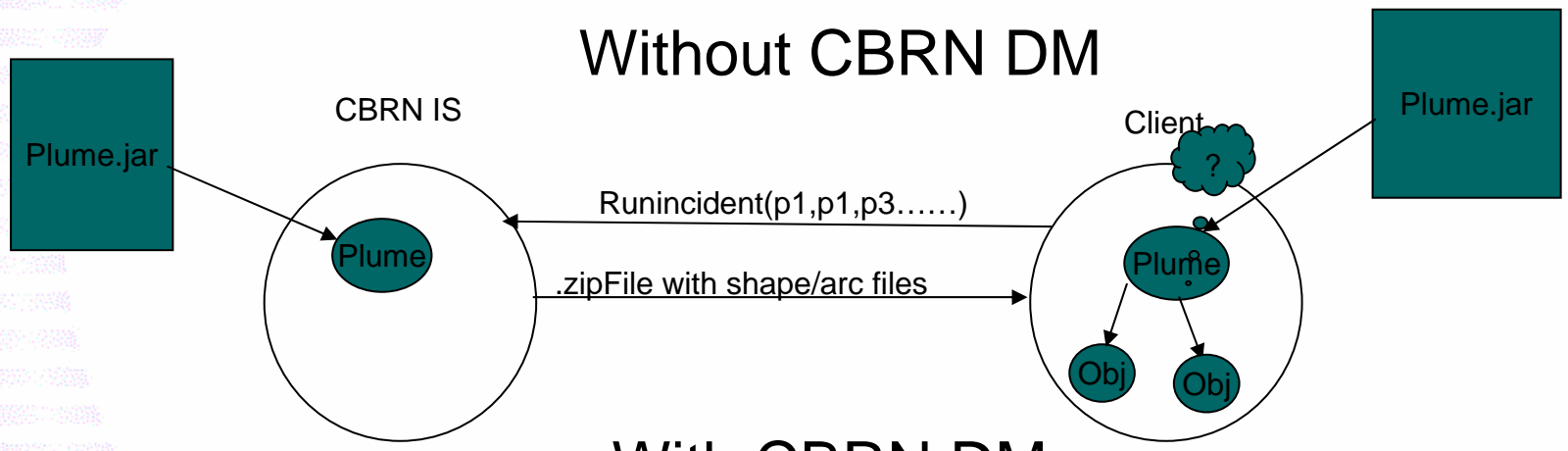
Data model in the SOA world

The nature of applications in SOA is distributed so that an application may access information distributed across many enterprise systems:

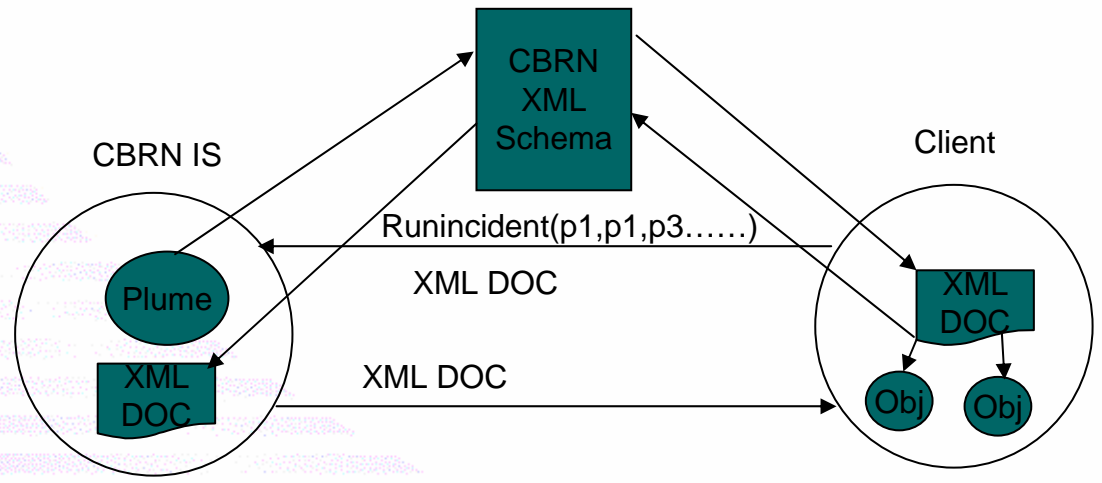
- 1. Inconsistencies in data format among different systems.**
- 2. Simply using XML schemas does not guarantee interoperability. For example English speaker and German speaker use the same alphabet but they still can't communicate with each other.**
- 3. Thus the need for a common semantic provided by a common data model to reduce cost by stressing loose coupling by providing standard interfaces among CBRN systems making it possible to easily integrate with future platforms like FCS, NCEs, and JC2.**

CBRN Data Product (XML Schema) in use

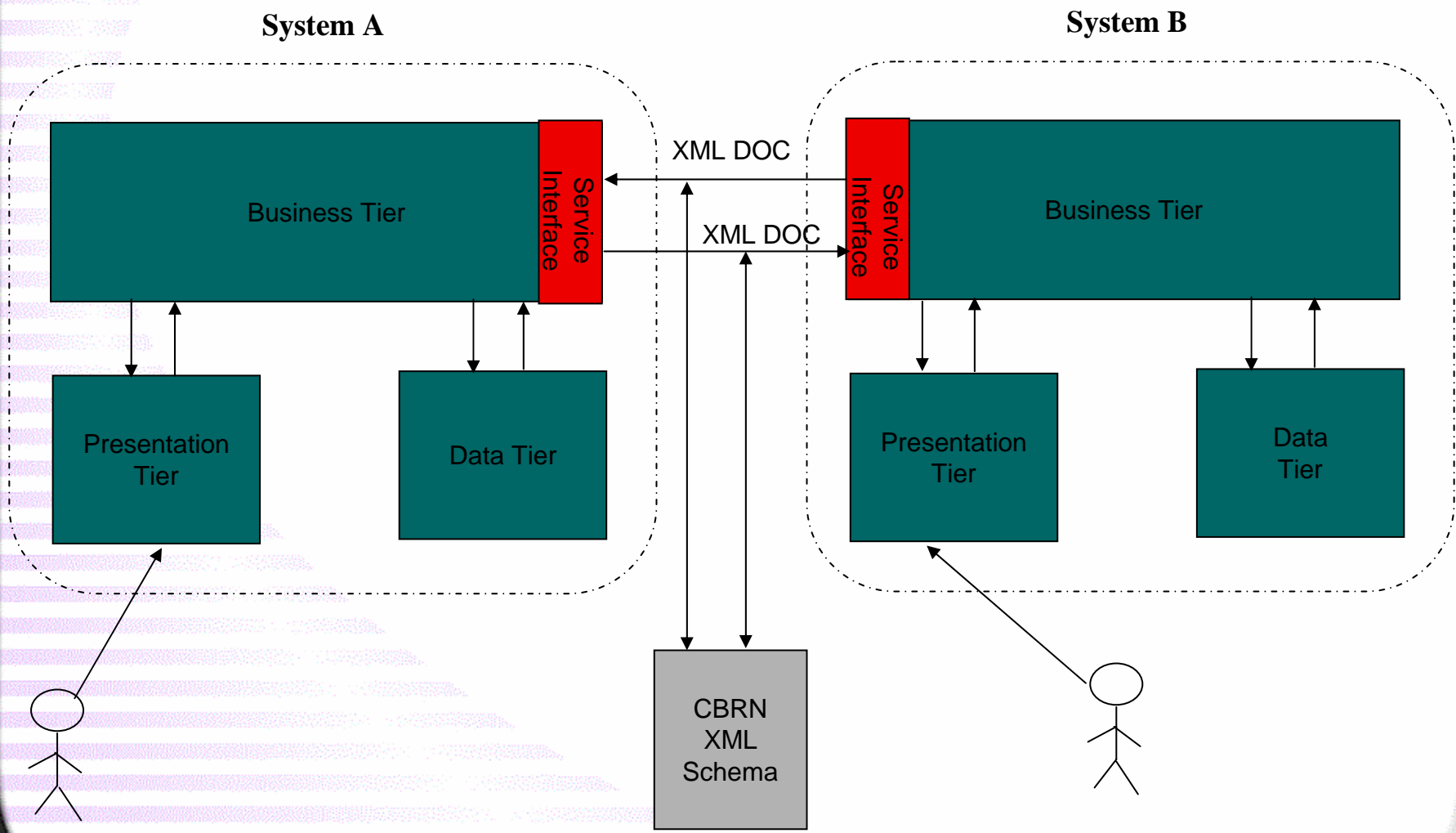
Without CBRN DM



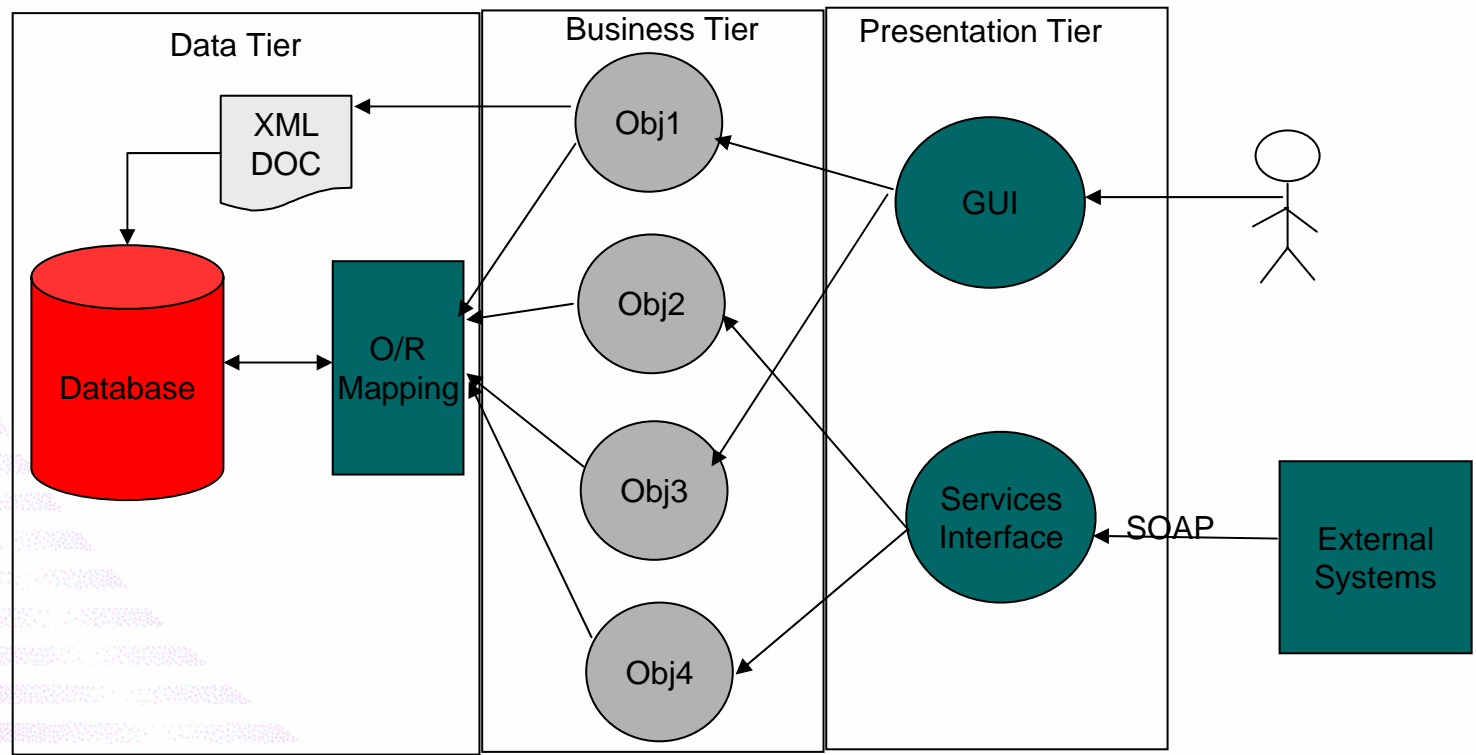
With CBRN DM



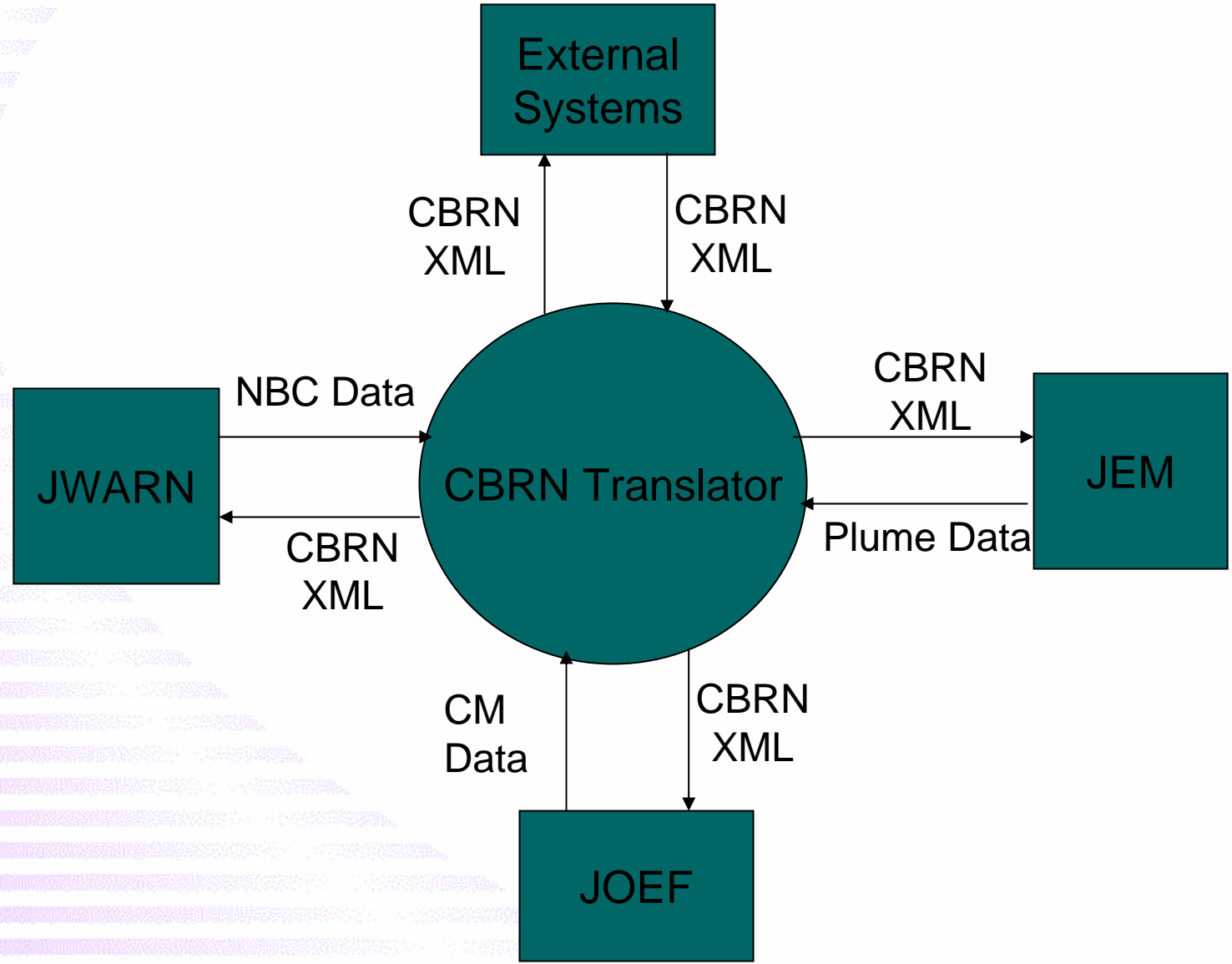
Data Exchange/Validation with XML Schema



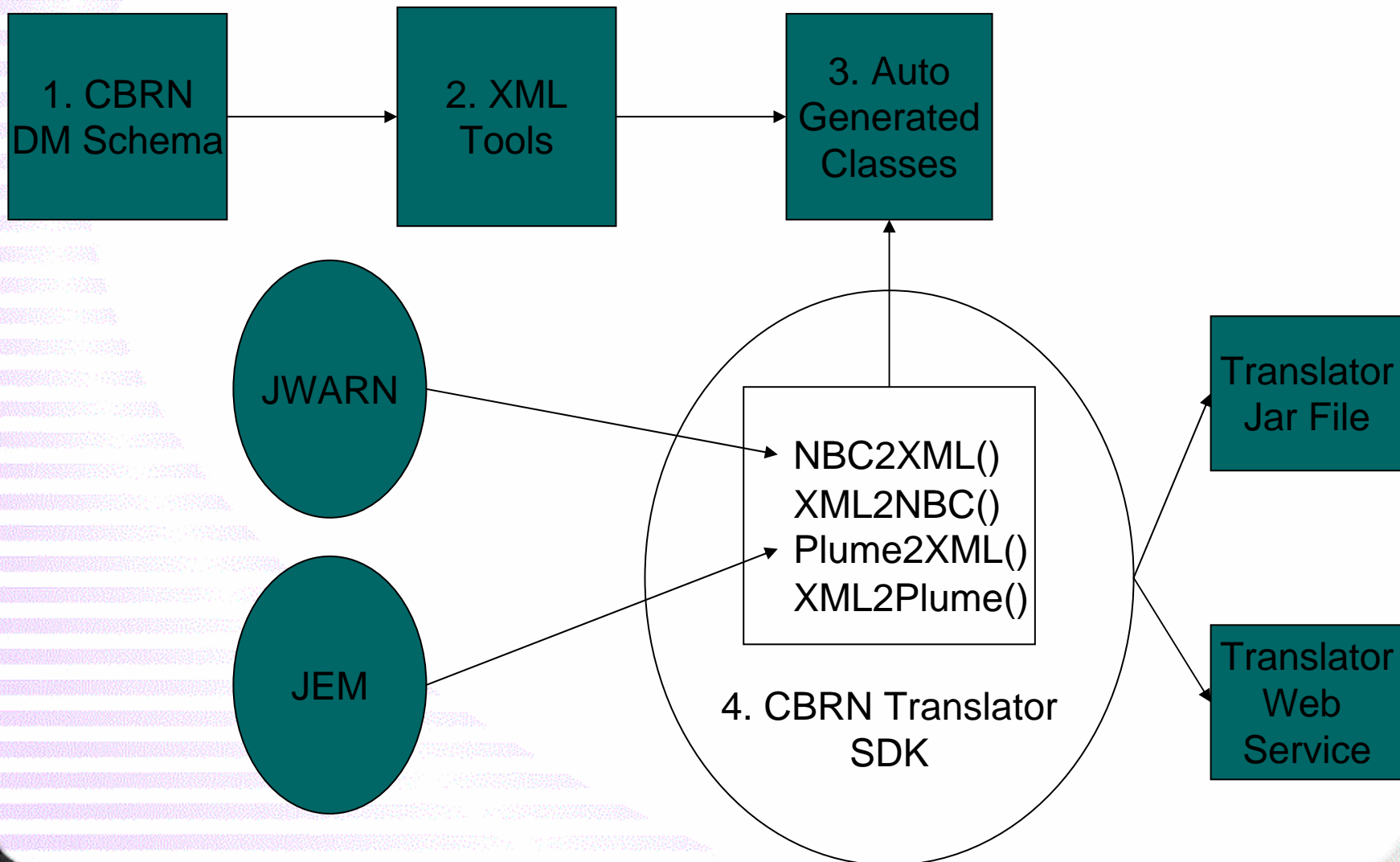
CBRN Database in Use



CBRN Data Exchange Translator



CBRN Data Exchange XML SDK



Summary

- **We view the CBRN data model as central to interoperability of our systems**
- **We are implementing using an incremental proactive approach**
 - Legacy systems: help build wrappers & transition to DM
 - New systems: encourage to use DM from start to exchange data
- **Supported by an implementation infrastructure**
 - Common tools and techniques
 - Implementation guidance and requirements
 - Collaboration environment
- **Focused on the exchange of common data with outreach to a wide variety of communities**

Additional Information??

**Professor Thomas H. Johnson
(NSA/JO)**

**Naval Postgraduate School
Monterey, CA 93943**

thjohnso@nps.edu

703-933-3301 (DC Office telephone, MSIAC)

793-933-3325 (DC fax)

831-656-3190 (NPS Office telephone)

831-656-2949 (NPS fax)

www.ccc.nps.navy.mil

BACKUP

Tools & Techniques

XML Binding Approaches

- **May start from:**
 - Schema
 - DTD (legacy)
 - one or more XML Documents
 - Existing Java classes
- **May generate Java classes based on XML document/DTD/schema**
- **Any way you slice it, you get marshalling (Java to XML), unmarshalling (XML to Java), modification, and validation**
- **Each tool offers value-added features**

The Castor Project

<http://www.castor.org/>

- Can map existing classes to XML
- XML documents can be unmarshalled directly into JavaBeans components without preconfigured mapping
- Likewise, any JavaBeans component can be marshalled to XML via introspection without preconfigured mapping
- Includes a tool to generate a schema from a DTD or existing XML documents
- XML documents can be unmarshalled into existing objects to save memory
- Reflection based, but able to call custom getters and setters as well

Castor, cont.

- **Generated classes are plain JavaBeans components, along with generated or manually provided compile-time descriptor classes**
- **Developer can customize generated source code via the binding file, including event support (namely, bound properties)**
- **Developer can customize the marshalled XML formatting via the mapping file**
- **Supports custom FieldHandlers for extensibility**
- **Supports optional validation including invoking the validator by hand**

XMLBeans

<http://xmlbeans.apache.org/>

- Originally a BEA tool, then incubated at Apache, now a top-level Apache project
- Version 1.03 just approved, Version 2 in development
- Supports all of XML Schema
- Provides full schema object model API
- Preserves the full XML Infoset
- Can validate after any change to the objects

XMLBeans, cont.

- **Generated code extends XML base classes with special XML/Schema features**
 - Provides access to XmlCursor API
 - Developer can switch between XML/Java technology features
- **Does not create Java objects at parse time**
 - Java objects created lazily as necessary
- **Upcoming version 2:**
 - Reduces JAR and memory footprint
 - Adds DOM Level II support over XML Store
 - Adds support for Java technology-centric approach
 - Ability to extend generated objects with custom functionality
 - Pre- and post- events on generated methods

<http://www.jibx.org/>

- **JiBX uses binding definitions**
 - You define how XML relates to Java objects
 - Allows structural changes going to and from XML
 - Start from schema, code, or both
- **Binding code compiled into your class files**
 - At build time, or on-the-fly at runtime
 - Allows compact and fast runtime
- **Fundamentally a *Java technology-centric* approach**
- **Current beta 3c release in widespread production use**

JiBX Binding Definition Flexibility

- **Current code supports:**
 - Complex types with simple content, mixed content, etc.
 - Input-output, input-only, or output-only bindings
 - Multiple bindings, even for the same classes
 - Easy extension hooks (pre-set, post-set, pre-get)
 - Custommarshallers / unmarshallers
 - Selective marshalling / unmarshalling, streaming
- **More features planned for future...**

XMLBeans Overview

- **Advantages**
 - Full schema support
 - Navigation & query of documents
 - Lots of metadata available
- **Disadvantages**
 - More overhead at runtime
 - Likes its generated JAR, doesn't like it so much if you just compile the code it generates
- **Overall**
 - My favorite when starting with complex schemas and generating code ahead of time

JAXB Overview

- **Advantages**
 - Standards-based (other implementations out there, or coming)
 - Supports DTDs, Relax-NG, etc.
 - Interfaces look quite nice
- **Disadvantages**
 - Incomplete schema support
 - Generated code required casts (JAXB 2 should be better with Generics, etc.)
- **Overall**
 - Not my first choice... yet.

Castor Overview

- **Advantages**
 - Can operate without the schema compiler step (using reflection only)
 - All generated code goes in one package, no class bloat
 - Castor offers O/R features too
- **Disadvantages**
 - Incomplete schema support
 - Required manual mapping; some rough edges
- **Overall**
 - First choice for for on-the-fly mapping

JiBX Overview

- **Advantages**
 - Small & fast
 - You write the code
- **Disadvantages**
 - You write the code – hopefully the *right* code
 - Bytecode mangling makes debugging hard
 - Still under active development (mapping file format changes between beta releases, etc.)
- **Overall**
 - Best for small schemas, existing Java code, version conflicts, etc.

Current Status

- Identified a process to instantiate a database from CBRN Data Model.
- Documented errors discovered during generation of a database from SQL scripts auto generated by ERWIN.
- Generated JAVA classes from CBRN Data Model using JAXB.
- Marshalled / Unmarshalled XML messages using JAVA classes generated by JAXB.
- Investigating different strategies to optimize XML message generation from XML Schema.
- Identifying data exchanges for each POR.
- Identifying persistent Object to Relational mapping using Hibernate.

Data Terms Important for ERwin

- **Tables, Columns, and Rows**

- Tables are 2-dimensional
 - Columns (fields)
 - Rows (records)
- Each table represents an object or a concept (Example: CBRN-EVENT).
- Each column represents a single characteristic that applies to that object.
- Each row represents a single instance of that object.

- **No duplicates allowed!** (for tables, columns, or rows)

- **Primary Keys**

- Every table must have a primary key defined.
- It consists of one or more columns that uniquely identify each row in the table.
- A primary key can never be null.

- **Non-key fields**

- Columns that are not part of the primary key are called non-key fields.

- **Foreign Keys**

- When the primary key of one table is inherited by another table, it is called a foreign key in the table to which it gets inherited.
- A foreign key can be non-key or part of the primary key of the table to which it is inherited.

Entities and Attributes in Logical Model

- ‘Box’ on diagram represents an **Entity**
 - In a physical implementation, this would be a **table**
 - **Square corners** indicate an **independent** entity
 - **Rounded corners** indicate a **dependent** entity
- Text label just outside of ‘Box’ is the **Entity Name**
 - Example: PRECIPITATION
- Text labels inside the box represent **Attributes**
 - In a physical implementation, these would be **columns**
 - Above the line indicates the attribute is part of the **primary key**
 - Example: precipitation-id
 - Below the line indicates the attribute is **non-key**
 - Example: precipitation-category-code, precipitation-rate, et al
- ‘**FK**’ at the end of an attribute name indicates that it is a **Foreign Key**, i.e. it has been inherited from a parent
 - Both primary key and non-key attributes can be foreign keys
 - Example (primary key): precipitation-id (FK)

PRECIPITATION

precipitation-id (FK)

precipitation-category-code

precipitation-rate

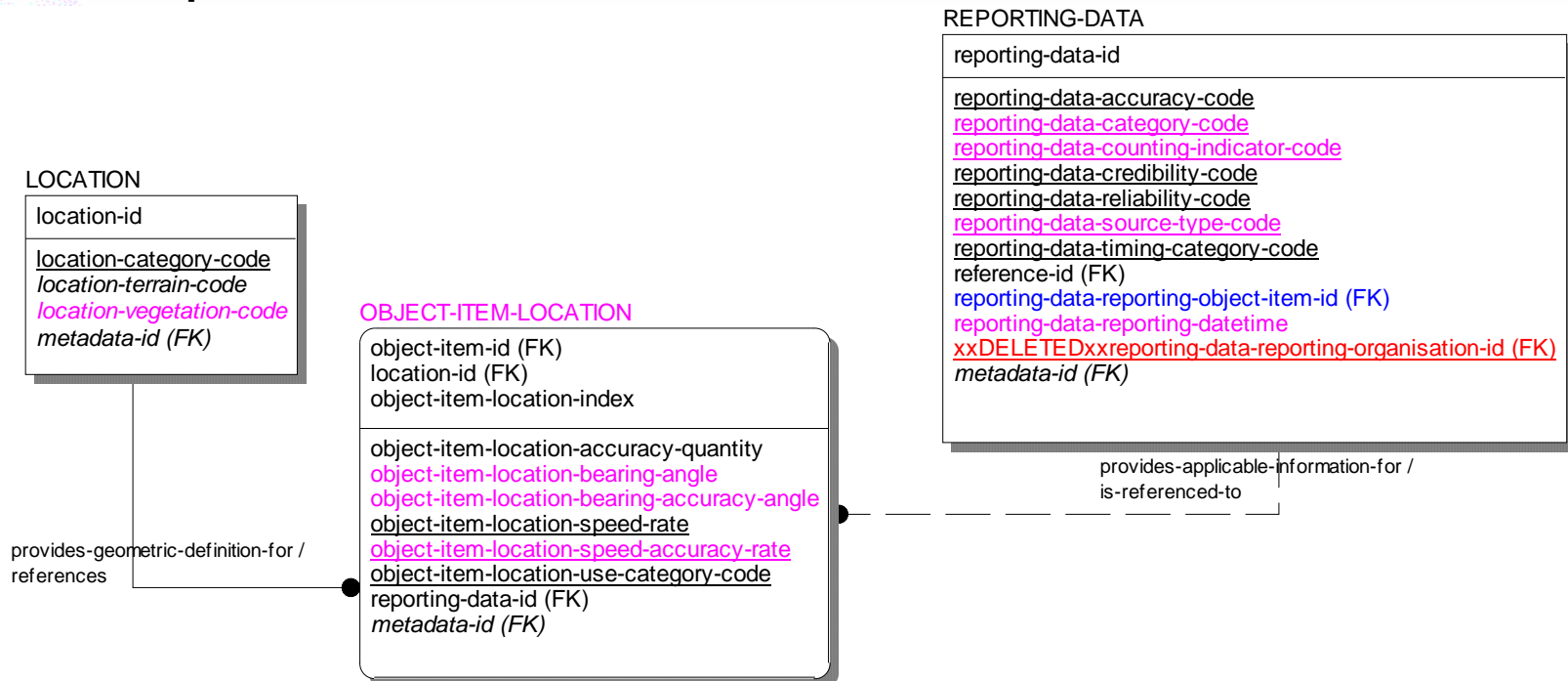
precipitation-class-name-code

precipitation-class-type-code

metadata-id (FK)

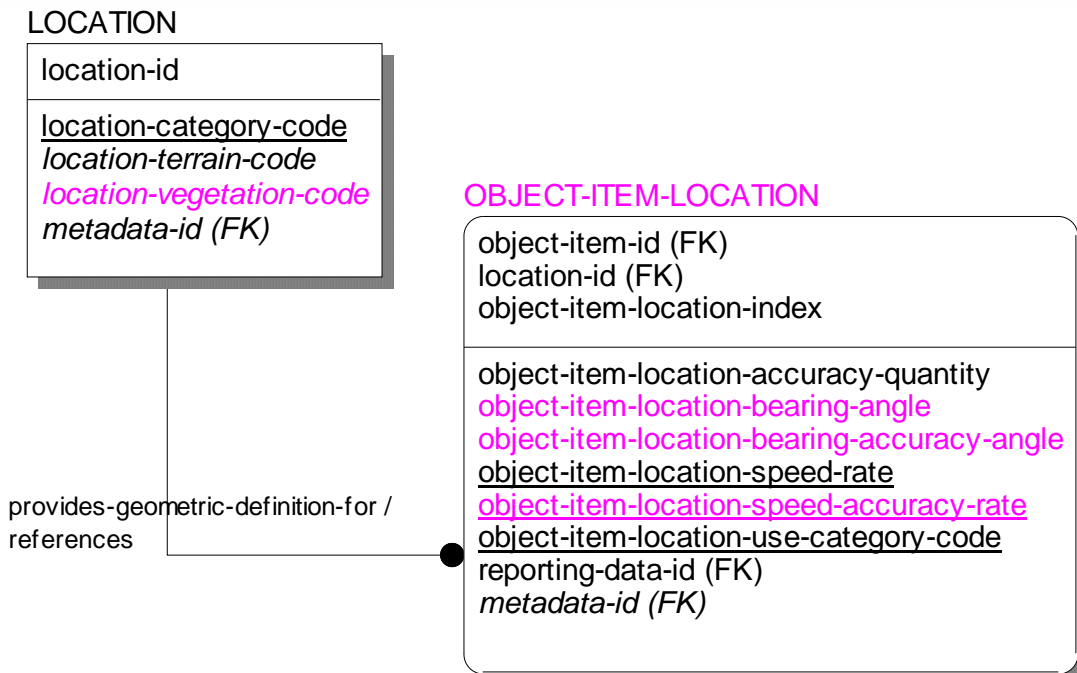
Relationship Lines in ERwin (IDEF1X)

- **Solid Line** _____ means relationship is **Identifying**.
 - Primary key of the parent will be inherited as part of the primary key of the child, i.e. child is **dependent** on parent.
 - Example: LOCATION to OBJECT-ITEM-LOCATION
- **Dashed Line** - - - - - means relationship is **Non-identifying**.
 - Primary key of the parent will be inherited as a foreign key attribute in the child (not part of the child's primary key), i.e. child is **independent** of parent.
 - Example: REPORTING-DATA to OBJECT-ITEM-LOCATION



Identifying Relationships Lines with/ without Dots

- **Solid Plain Line** — indicates the entity is the Parent in an identifying relationship
 - Example: LOCATION side of LOCATION to OBJECT-ITEM-LOCATION
- **Solid Line with Dot** —• indicates the entity is the Child in an identifying relationship
 - Example: OBJECT-ITEM-LOCATION side of LOCATION to OBJECT-ITEM-LOCATION



Non-Identifying Relationships: Lines and Dots or Diamonds

- Plain Dashed Line — — - indicates the entity is the Parent in a non-identifying relationship that **allows no nulls (is mandatory)**
 - Ex: REPORTING-DATA side of REPORTING-DATA to OBJECT-ITEM-LOCATION
- Dashed Line with Diamond — — ◇ indicates the entity is the Parent is a non-identifying relationship that **allows nulls (is optional)**
 - Ex: REPORTING-DATA side of REPORTING-DATA to ACTION-EVENT-STATUS
- Dashed Line with Dot — — ● indicates the entity is the Child in a non-identifying relationship
 - Ex: OBJECT-ITEM-LOCATION AND ACTION-EVENT-STATUS sides of their relationships to REPORTING-DATA

OBJECT-ITEM-LOCATION

object-item-id (FK)
location-id (FK)
object-item-location-index

object-item-location-accuracy-quantity
object-item-location-bearing-angle
object-item-location-bearing-accuracy-angle
object-item-location-speed-rate
object-item-location-speed-accuracy-rate
object-item-location-use-category-code
reporting-data-id (FK)
metadata-id (FK)

provides-applicable-information-for /
is-referenced-to

REPORTING-DATA

reporting-data-id

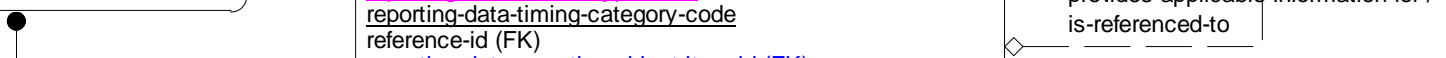
reporting-data-accuracy-code
reporting-data-category-code
reporting-data-counting-indicator-code
reporting-data-credibility-code
reporting-data-reliability-code
reporting-data-source-type-code
reporting-data-timing-category-code
reference-id (FK)
reporting-data-reporting-object-item-id (FK)
reporting-data-reporting-datetime
xxDELETEDxxreporting-data-reporting-organisation-id (FK)
metadata-id (FK)

ACTION-EVENT-STATUS

action-event-status-index
action-event-id (FK)

action-event-status-completion-ratio
reporting-data-id (FK)
metadata-id (FK)

provides-applicable-information-for /
is-referenced-to



The Letters P, Z, and Numbers in ERwin Relationships

- **P**

- A '**P**' next to the dot (child) end of a relationship indicates that the cardinality of the relationship is **'One to (at least) one or more.'**
 - Relationship is mandatory.
- The **lack of a single character** next to the dot generally indicates that the cardinality is **'One to zero, one, or more.'**
 - Relationship is optional.

- **Z**

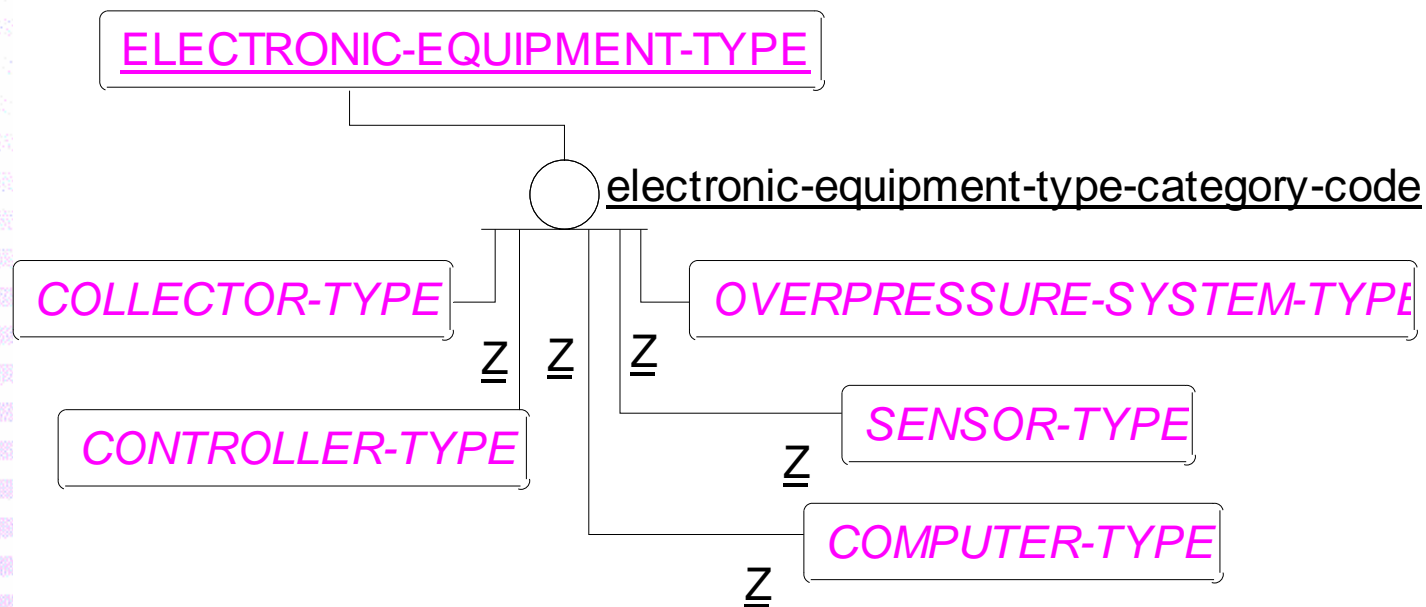
- A '**Z**' next to the dot (child) end of a relationship indicates that the cardinality of the relationship is **'One to zero or one only.'**
 - Hence is used for subtypes.

- **Number (n)**

- A number (**n**) next to the dot (child) end of a relationship indicates that the cardinality of the relationship is **'one to exactly n.'**
 - Example: a '1' next to the dot would indicate 'one to exactly one.'

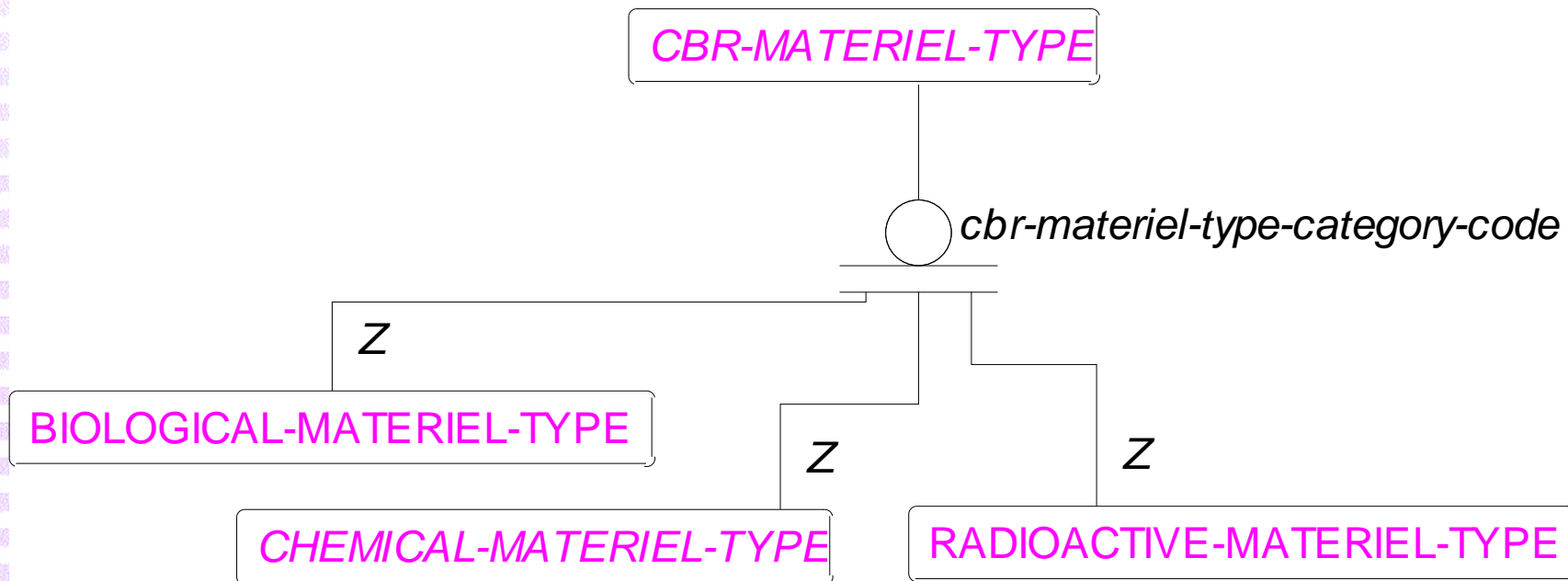
Partial Subtypes

- A circle with one bar under it $\underline{\bigcirc}$ indicates a Partial Subtype
 - I.e. some of the child subtypes of the parent are specified, but other subtypes also exist
 - Example: **ELECTRONIC-EQUIPMENT-TYPE** and its subtypes
 - There are other types of Electronic Equipment besides those pictured below.



Complete Subtypes

- A circle with two bars under it ○ indicates a Complete Subtype
 - I.e. all of the child subtypes of the parent are specified
 - Example: CBRNE-MATERIEL-TYPE and its subtypes
 - All CBRNE materiel will be either Biological, Chemical, or Radioactive
 - Chemical includes Explosives
 - Radioactive includes Nuclear and Radiological (material properties are same)



CBRN Data Model

Colors and Font Styles

- **Color scheme indicates whether an item was changed from the previous release**
 - **Blue = New Addition**
 - **Red = Deletion (will not appear in following release)**
 - **Pink = Changed from previous release**
 - **Black = No Change from previous release**
- **Font Style indicates the origin of an item**
 - **Normal = JC3IEDM item**
 - ***Italicized = CBRN COI item***
 - **Underlined = JC3IEDM item modified by the CBRN COI**

Quick Review

- **Methods to Review Data Model**

- For introduction, read Summary document issued with release
- ERwin Model Navigator (read-only version)
- HTML version of Data Model
- Data Dictionary (Excel file)
- XML Schema (.xsd file)
- PDFs

- **Corresponding Terms**

- Entity  Table
- Attribute  Column

- **Relationship Lines**

- Solid  Child is Dependent on Parent
- Dashed  Child can exist Independent of Parent

JPMIS Data Model Implementation Strategy

- Integrate DM implementation into program requirements
- Incremental proactive approach
 - Make it easy for programs to implement the data model
 - Research, tailor, recommend industry best practices
 - Develop infrastructure and user support
 - Work closely with programs to help ensure successful DM implementation
 - Legacy systems: help build wrappers & transition to DM
 - New systems: encourage to use DM from start to exchange data.
- Focus on *exchange of common data*