OASIS UIMA Technical Committee Specification Overview

(DRAFT v0.3)

November 30, 2007

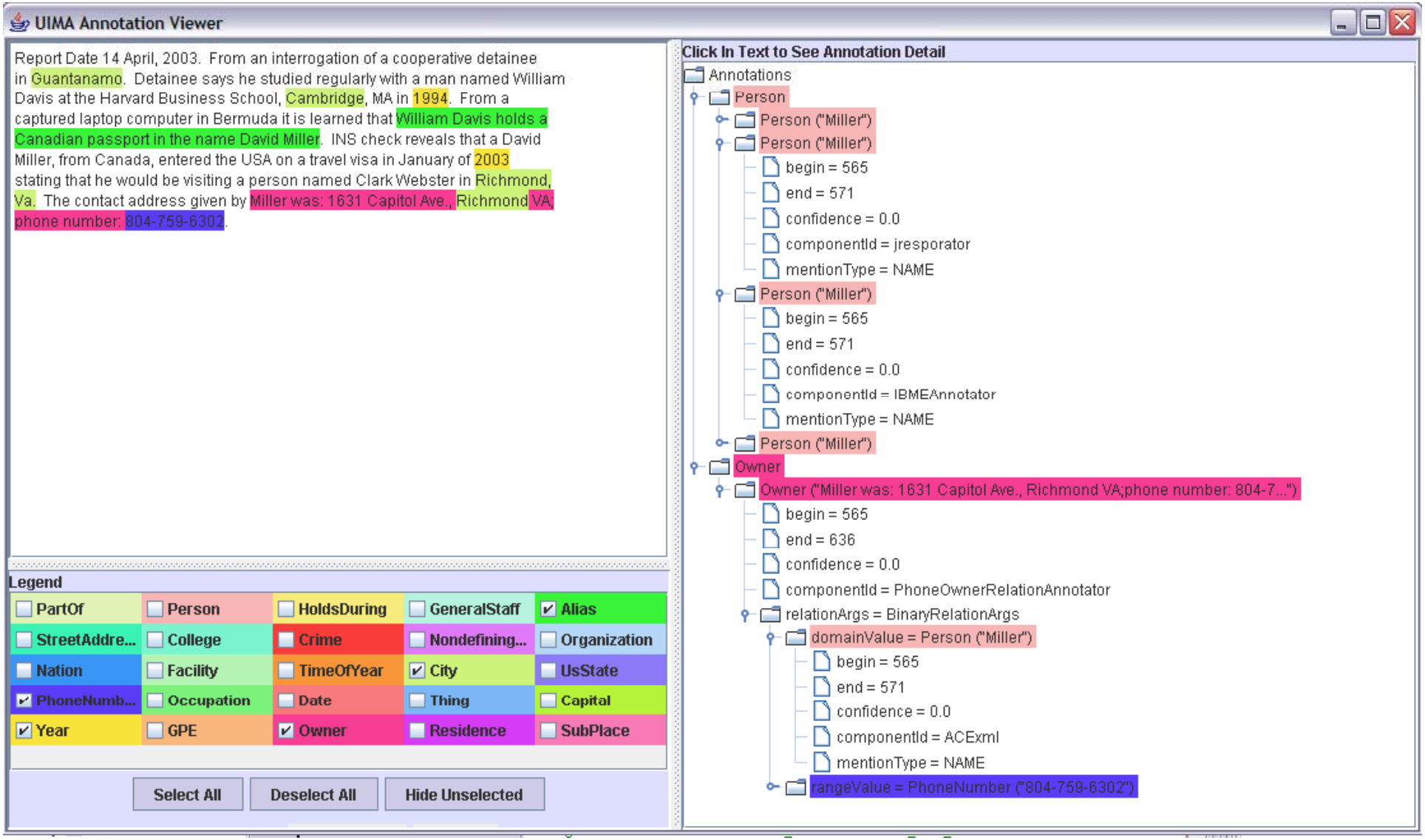
Outline

- Overview
- Status
- Design Goals
- Specification Elements
- Impact on Apache UIMA SDK

The UIMA Standard

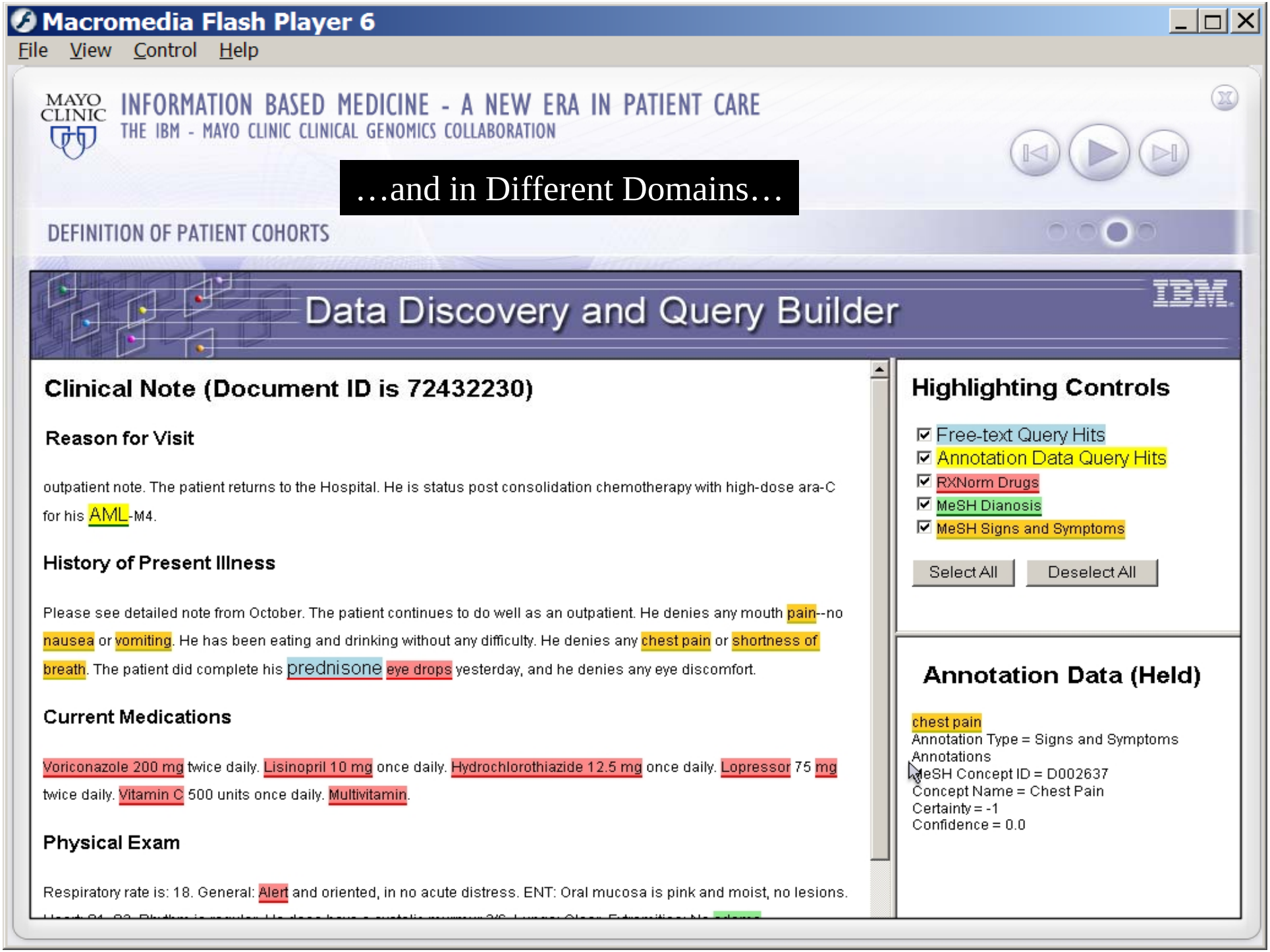
- Platform-Independent Data Representations & Interfaces for Text & Multi-modal Analytics
  - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uima
  - http://www.oasis-open.org/committees/uima/charter.php
- Enable Interoperability of Text & Multi-modal Analytics
  - Exchange Analysis Data
  - Exchange Analytic Metadata (Descriptions of what analytics do)
  - Interface with analysis applications at the Services Level

•Statement about existing adoption with UIMA – industry and government
•History with regard to what this is valuable – people use to roll their own with myriad of problems.
•Mention what we consider current weaknesses in general state of affairs AND in current Apache UIMA
•Apache UIMA is evidence not blue sky stuff and can be implemented

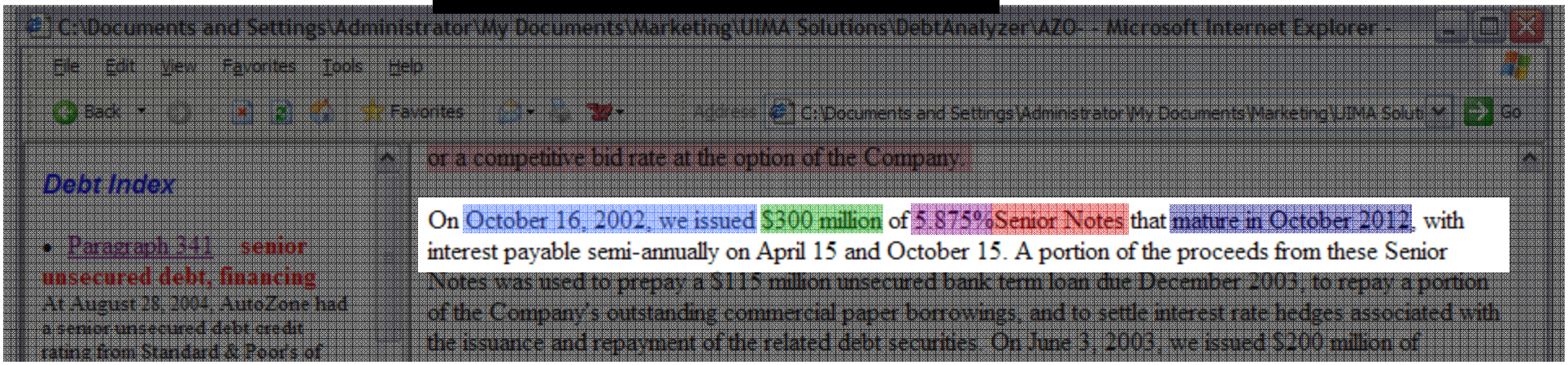


Analytics can detect a broad range of semantic types in text for example.


File Edit View Favorites Tools Help
Back Search Favorites
Address http://ikm001l.watson.ibm.com/cgi-bin/arm6/arm.cgi?stem=/homes/ikm001l/annotate/ar-mt05-Jan16/AFP20050119.0049&charset=utf-8&bgcolor=xEEFFEE&tblcolor=xFf
Go
Entities
New
Story /homes/ikm001l/annotate/ar-mt05-Jan16/AFP20050119.0049
Text Sentence Both Coref
UnDone
Help
Save changes at any time
Find
…in Different Languages…
Mention Types
AGE
ANIMAL
AREA
ATTRACTION
CARDINAL
COMPANYROLE
COUNTRY
DISEASE
EVENT
EVENTBUSINESS
EVENTCOMMUNICATION
EVENTCUSTODY
EVENTDEMONSTRATION
EVENTDISASTER
EVENTLEGAL
EVENTMEETING
EVENTPERFORMANCE
EVENTPERSONNEL
EVENTSPORTS
EVENTVIOLENCE
FACILITY
FOOD
GEOLOGICALOBJ
LAW
LOCATION
NAMED
OCCUPATION
ORDINAL
ORGAN
ORGANIZATION
PEOPLE
PERSONPEOPLE
PERSON
PLANT
PRODUCT
SALUTATION
SUBSTANCE
TITLEWORK
VEHICLE
WEAPON
WEATHER


Macromedia Flash Player 6
File View Control Help
MAYO CLINIC
INFORMATION BASED MEDICINE - A NEW ERA IN PATIENT CARE
THE IBM - MAYO CLINIC CLINICAL GENOMICS COLLABORATION
…and in Different Domains…
DEFINITION OF PATIENT COHORTS
Data Discovery and Query Builder
IBM
Clinical Note (Document ID is 72432230)
Reason for Visit
outpatient note. The patient returns to the Hospital. He is status post consolidation chemotherapy with high-dose ara-C for his AML-M4.
History of Present Illness
Please see detailed note from October. The patient continues to do well as an outpatient. He denies any mouth pain--no nausea or vomiting. He has been eating and drinking without any difficulty. He denies any chest pain or shortness of breath. The patient did complete his prednisone eye drops yesterday, and he denies any eye discomfort.
Current Medications
Voriconazole 200 mg twice daily. Lisinopril 10 mg once daily. Hydrochlorothiazide 12.5 mg once daily. Lopressor 75 mg twice daily. Vitamin C 500 units once daily. Multivitamin.
Physical Exam
Respiratory rate is: 18. General: Alert and oriented, in no acute distress. ENT: Oral mucosa is pink and moist, no lesions.
Highlighting Controls
Free-text Query Hits
Annotation Data Query Hits
RXNorm Drugs
MeSH Dianosis
MeSH Signs and Symptoms
Select All
Deselect All
Annotation Data (Held)
chest pain
Annotation Type = Signs and Symptoms
Annotations
MeSH Concept ID = D002637
Concept Name = Chest Pain
Certainty = -1
Confidence = 0.0

…and in Different Domains…



Debt Type: Senior Note
Issue Date: October 16, 2002
Maturity Date: October 2012
Dollar Amount: $300 million
Rate: 5.875%

- Find corporate debt buried within unstructured text of SEC filings
  - Identify all debt references, including types, dates, amounts, rates and other terms
- Highlight relevant concepts to simplify research
- Create index and table of contents of debt references for easier searching and navigation
- Create alerts to monitor for key indicators

…and in Different Modalities…

Need For A Standard

- Many Independently Developed Analytics
  - Parsers, Tokenizers, Entity Detectors, Topic Detectors, Summarizer, Classifiers, Speech Transcription, Video Analysis, Translation etc.
- Higher Level Applications need to mix and match
  - Business Intelligence, National Security, Healthcare, Customer Relationship Management, Web Self Service, Bioinformatics, Content Analytics etc.
- Support for interoperability is replicated over and over in industry, research and academia

To Review

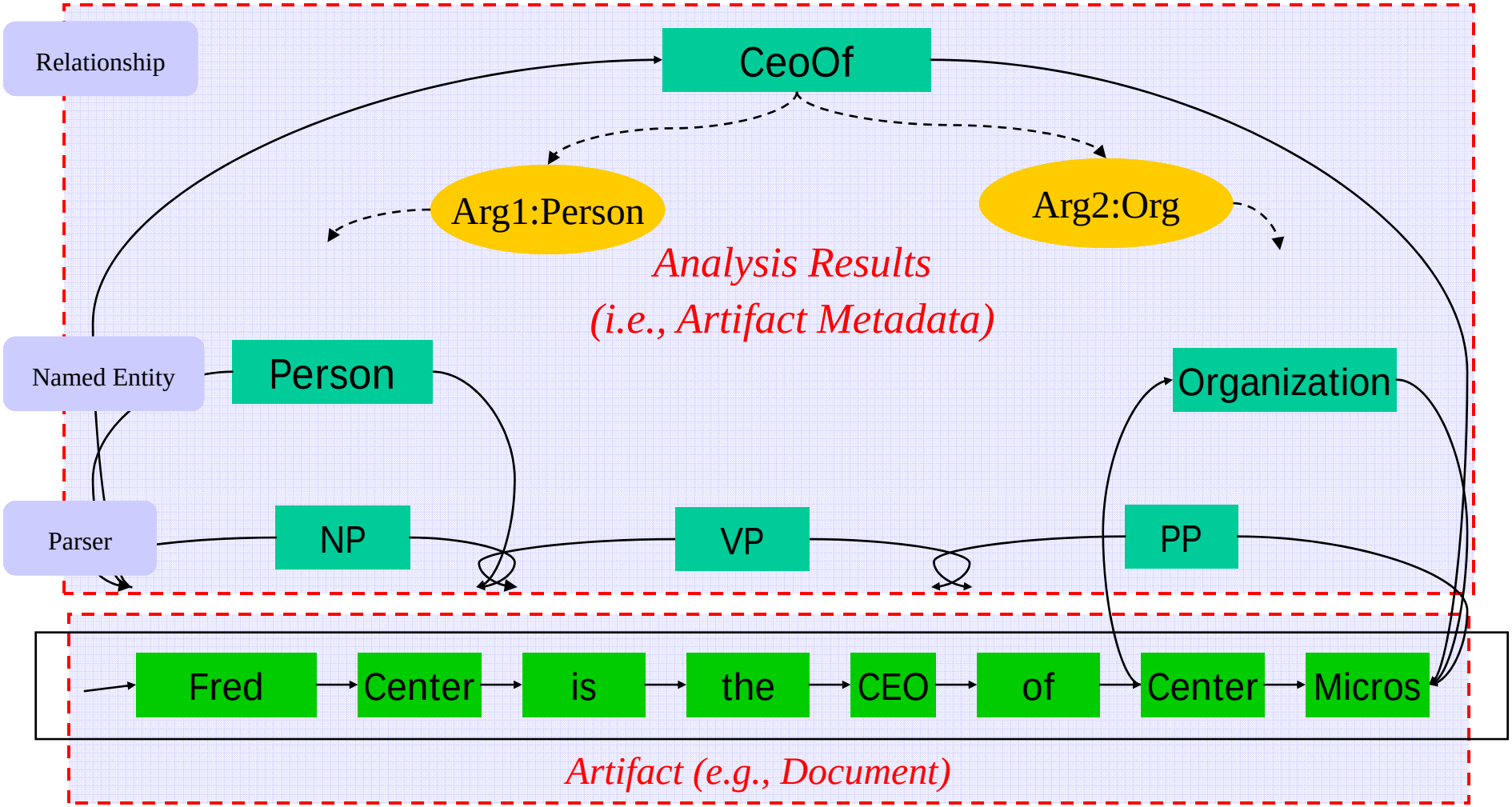
The Apache UIMA SDK

- Broadly Adopted Java Implementation
- Supports analytic development, composition and deployment
- Open Source (Apache Incubator: http://incubator.apache.org/uima/index.html)
- Informed UIMA Standard
- To support full compliance

Status Update

- The UIMA TC has met bi-weekly for 10 months and has completed a full review of the research report contributed by IBM as a initial proposal for a standard for interoperable text and multi-modal analytics based on UIMA.
- The UIMA TC will integrate all revisions gathered in review reports and meeting minutes into a formal specification draft by Feb 1, 2008.
- The UIMA TC will then conduct final votes on draft sections and any outstanding issues that remain.
- The UIMA TC will publish a final draft of the specification by end of March 2008.

Introduction to UIMA Standard



Example to introduce examples on next slide but not talk about CAS yet

Design Goals

- **Data Representation.** Support the common representation of *artifacts* and *artifact metadata* (analysis results) independently of *artifact modality* and *domain model*.

- **Data Modeling and Interchange.** Support the platform-independent interchange of *analysis data* in a form that facilitates a formal modeling approach and alignment with existing programming systems and standards.

- **Discovery, Reuse and Composition.** Support the discovery, reuse and composition of independently-developed *analytics.*

- **Service-Level Interoperability.** Support concrete interoperability of independently developed *analytics* based on a common service description and associated SOAP bindings.

Note: “Platform Independent Development” design goal in original spec draft was dropped as we have decided to focus on service-level interoperability only.

Have definitions in back-up slides and footnote this slide

Objectives Relative to Apache UIMA

- Maintain High-Degree of alignment
- Improve Known Representation Issues
- Better use of existing standards
  - XMI, UML, WSDL, OCL

Improve Known Representational Issues

- Discovery and Composition
  - Robust, Unambiguous Metadata
  - Describing Analytics' Input Requirements and Function
- Uniform treatment of data elements
  - Any item may be the subject of an analysis process
- More General "View" Concept
  - Views can contain any specific collection of data elements
- Improved Naming of Types
  - Better reflect their intent
- More Flexible way to connect meta-data to regions of artifact

Do we want to detail this here? Understanding depends on deeper treatment.

Better use of Existing Standards

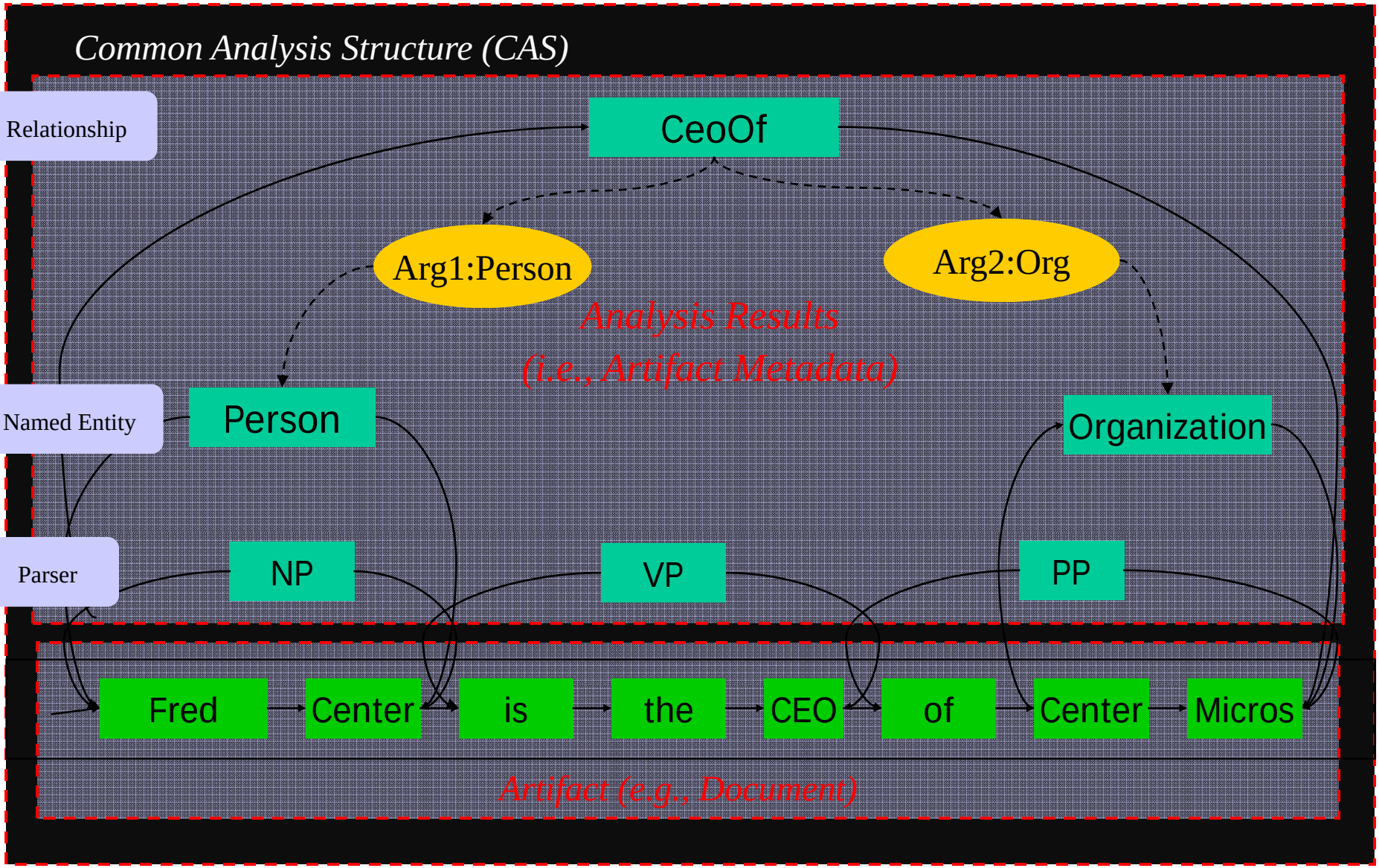
- XMI for object graphs
- UML/Ecore
- WSDL
- OCL

Specification Elements

1. Common Analysis Structure (CAS)
2. Type System Model
3. Base Type System
4. Abstract Interfaces
5. Behavioral Metadata
6. Processing Element Metadata
7. WSDL Service Descriptions

Common Analysis Structure (CAS)

- The common data structure **shared by all UIMA analytics**
- Supports interoperability by providing a common foundation for sharing data across analytics.
- A CAS Represents the
  - ***Artifact*:** the content being analyzed AND
  - ***Artifact Metadata*:** the metadata produced by the analytics (e.g., Annotations)
- The CAS is an Object Graph where
  - Objects are instances of Classes
  - Classes are Types in a ***type system***.
- Two fundamental types of objects in a CAS:
  - ***Subject of analysis (Sofa),*** holds the artifact
  - ***Annotation***, a type of artifact metadata that points to a region within a Sofa. An annotation annotates or labels the designated region in the artifact. This is an example of a *stand-off* annotation approach.



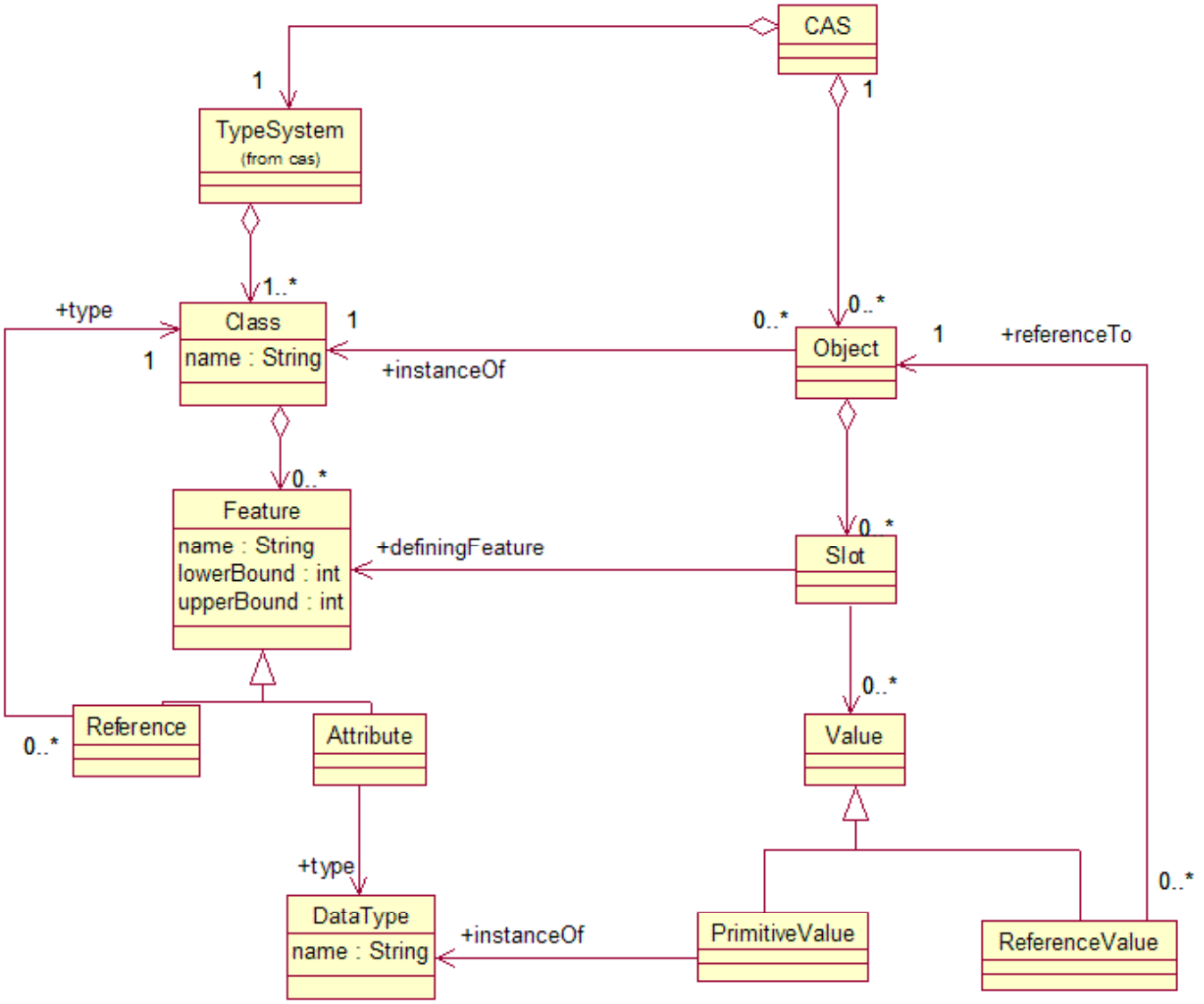
Before this dynamic slide – show a static example of text that calls out the features of the CAS, artifacts, Annotations, etc. and MUST show a non-text example

CAS Model

- General Object Graph
- Expressive Representational Power
- Aligned with UML

To Review

CAS UML


CAS
1
1
TypeSystem
(from cas)
1..*
+type
Class
name : String
1
1
+instanceOf
0..*
0..*
Object
1
+referenceTo
0..*
Feature
name : String
lowerBound : int
upperBound : int
+definingFeature
0..*
Slot
0..*
Reference
0..*
Attribute
0..*
Value
+type
DataType
name : String
+instanceOf
PrimitiveValue
ReferenceValue
0..*

CAS Data Representation

- An interchange format for the CAS
- Specified Using *XML Metadata Interchange (XMI)*
  - An OMG standard for representing object graphs in XML.
- Motivation for Using XMI
  - Established standard
  - Aligned with object-graph representation of CAS
  - Aligned with UML and with object-oriented programming
  - Supported by tooling such as the Eclipse Modeling Framework (EMF)

CAS XMI Example

Align with our running text example

```
<xmi:XMI xmi:version="2.0" xmlns:xmi=http://www.omg.org/XMI
   xmlns:ex="http:///org/example.ecore">
```

Header

```
  <ex:Quotation xmi:id="1"
    text="If we begin in certainties, we shall end with doubts;
   but if we begin with doubts and are patient with them, we
   shall end in certainties."
    author="Francis Bacon"/>
```

Artifact

```
  <cas:SofaReference xmi:id="2" sofaObject="1"
   sofaFeature="text"/>
```

Sofa Reference

```
  <ex:Clause sofa="2" begin="0" end="30"/>
  <ex:Pronoun sofa="2" begin="3" end="5"/>
  <ex:Pronoun sofa="2" begin="29" end="31"/>
```

Some Annotations

```
</xmi:XMI>
```

Close

Type System Model

- To support **data modeling and interchange**, a CAS must conform to a user-defined schema.
- We call this schema a *Type System*.
- Every object in a CAS must be associated with a *Type* defined by a *Type System*

Essential feature of a type system representation distinguish from required
And optional == Desired Features – addressed by next slide

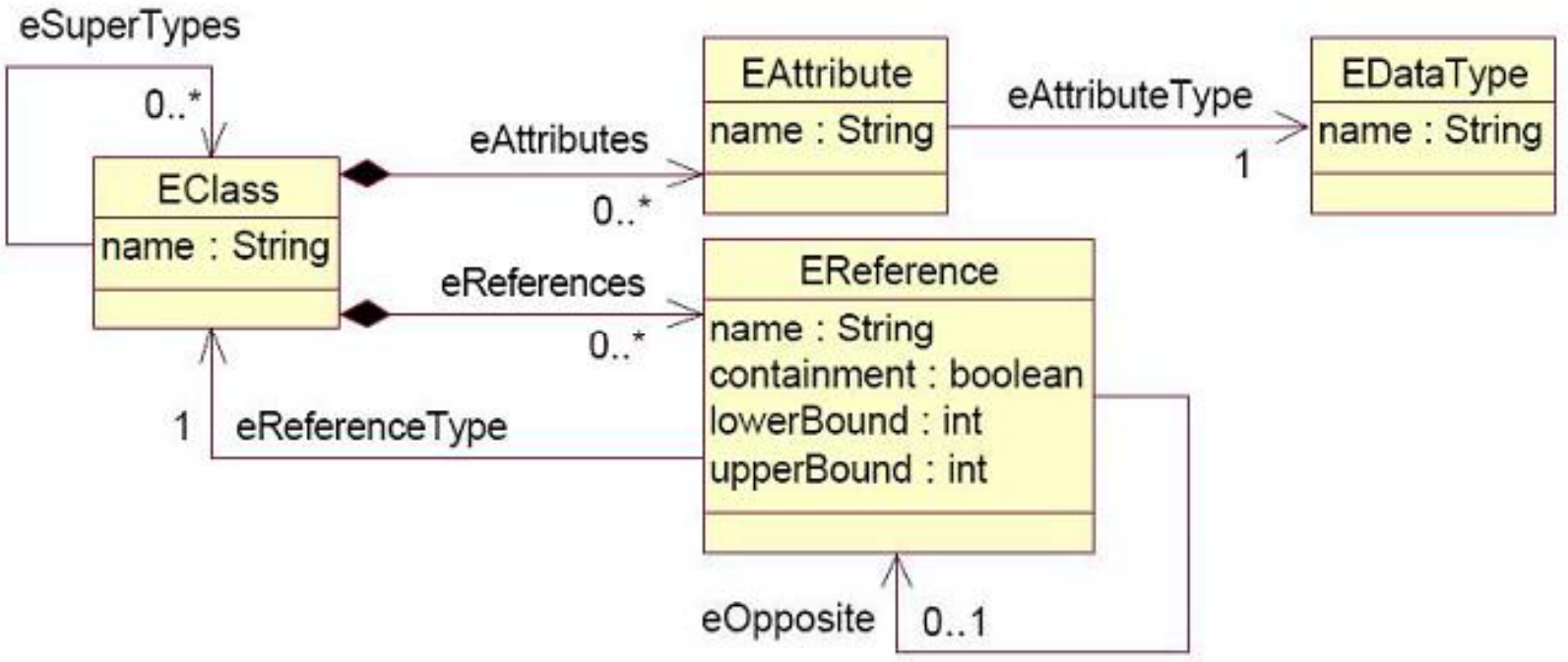
Desired Features of Type System Model

- Object-Oriented
- Inheritance
- Optional and Required Features
- Single and Multi-Valued
- Range Constraints on Features
- Aligned with UML standard
- Supported by Tooling

Type System Representation

- Possible choices considered for the type system representation
  - *EMOF* is an OMG Standard that is well aligned with UML and with Object Oriented Programming.
  - *Ecore* is the modeling language used by the Eclipse Modeling Framework (EMF).
  - *Ecore* provides equivalent modeling semantics to EMOF with minor syntactic differences.
- UIMA TC has chosen to adopt *Ecore* as its type system language, due to the availability of tooling provided by EMF.

Ecore “Kernel”



*need attribution (existing and managed by another intitustion)

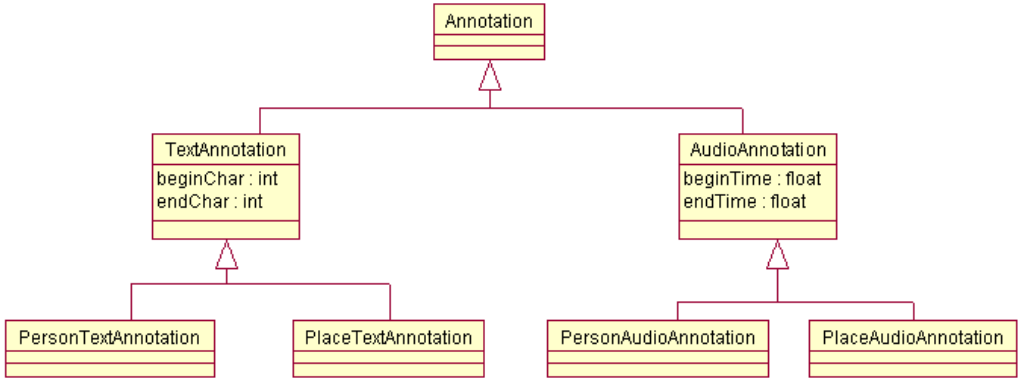
Base Type System

- Standard Definition of Basic, Domain-Independent types
- Supports Interoperability
- Includes
  - Primitive Types (defined by Ecore)
  - Annotation Model (how annotations are represented and linked to regions of Sofas)
  - Views (Specific collections of objects in a CAS. May be used to define specific interpretations or views of a *Sofa*.)
  - Other Commonly Used Types (e.g., Source Document Information)

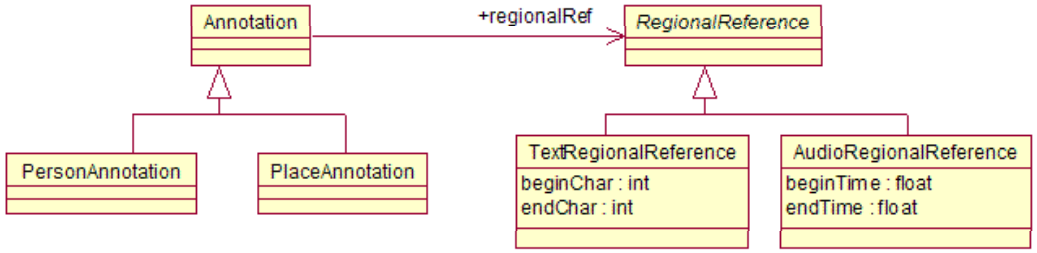
To Review – What is the compliance point if any? We expect Apache UIMA to use these definitions. Do we want to have detail slides for Primitive Types, Views etc? in Back up slides? Why focus On must annotation model, Views changes also.

Annotation Base Type-System – Two Alternatives

Annotation type includes offsets into Sofa (Apache UIMA style):



Separate RegionalReference object:



Annotation Base Type System – Two Alternatives

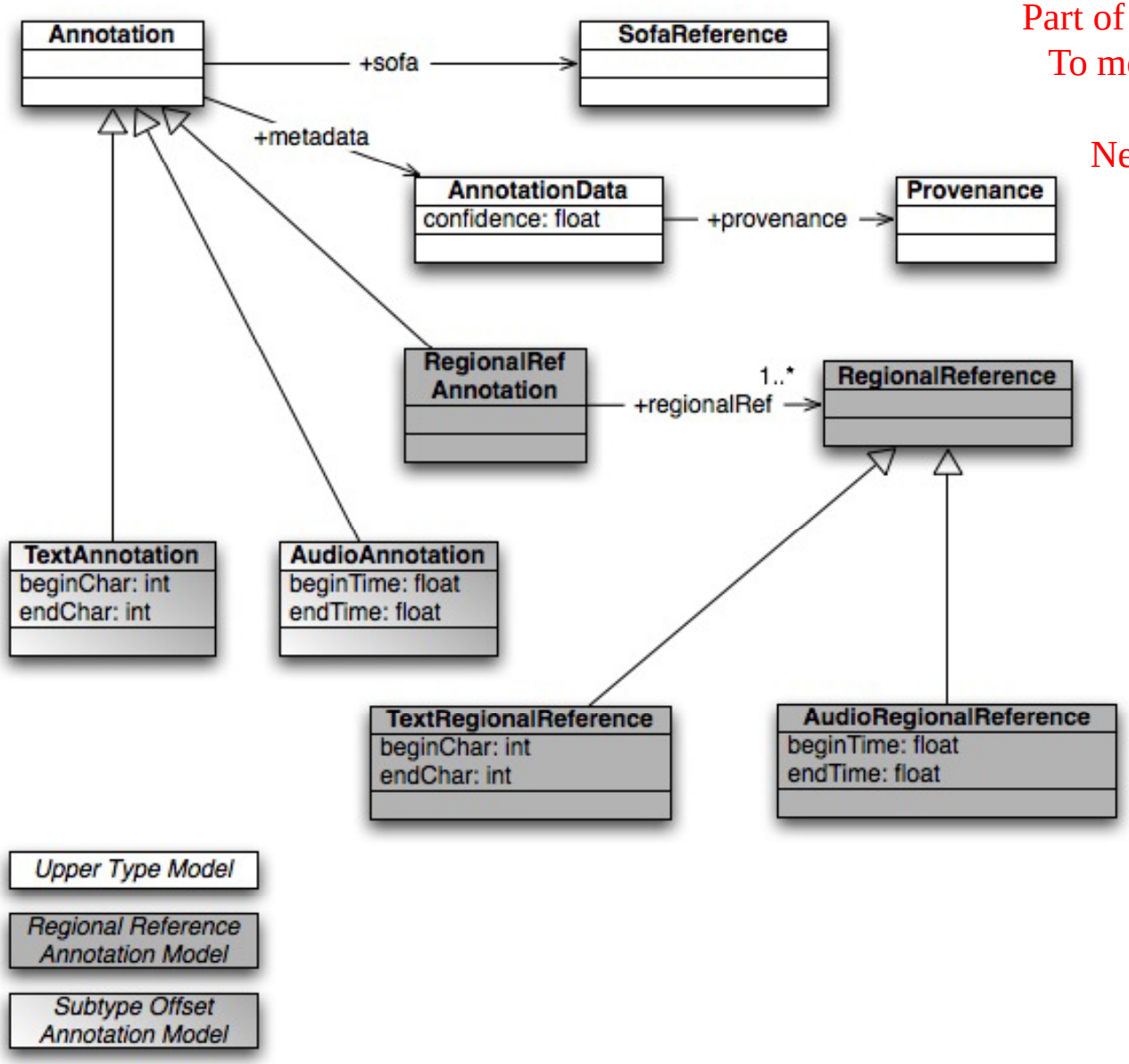
- Offsets in the annotation object may result in an explosion of annotation types.
  - For a new concept an annotation type for EACH modality must be added
  - e.g. Organization → OrgTextAnnotation + OrgAudioAnnotation
- Regional References introduce another level of indirection
  - Given a PersonAnnotation object, the modality must be known to retrieve the regional reference and then it's covered text
  - Requires two objects per annotation INSTANCE instead of one
- For these reasons, the UIMA Annotation Base Model will define type systems for *both* alternatives, and let users choose.

Clean up text but maybe a pros, cons table its best.
Also reconsider regional reference
Entity Type (Person) <-- Annotation=RegionReferences → Span

Annotation Base Model

Are confidence and provenance
Part of base model or just
To motivate example?

Need to look up.



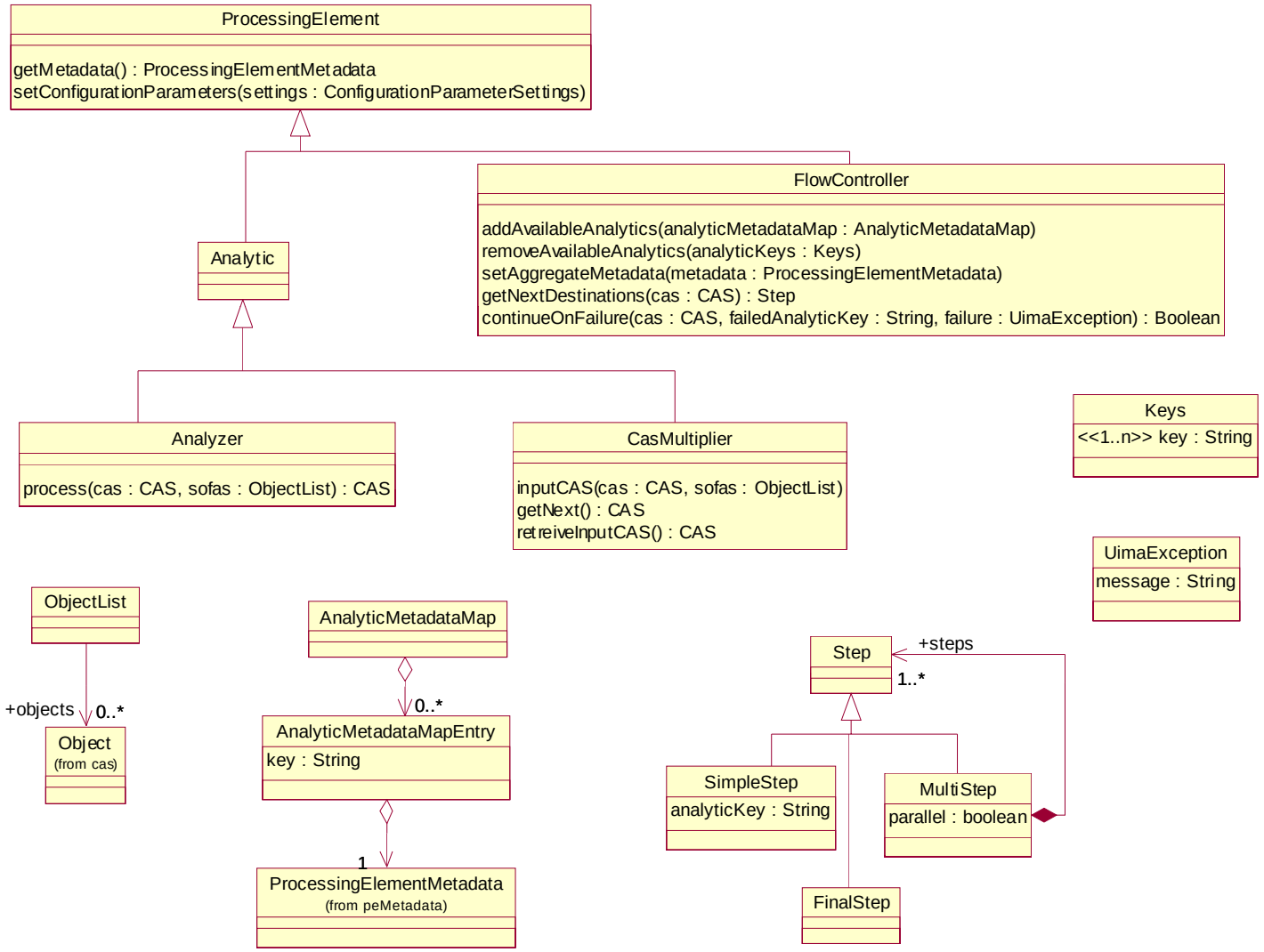
Abstract Interfaces

The goal of the Abstract Interfaces section is to provide a platform-independent model of the types of components that UIMA developers can implement and the operations supported by these components.

Types of Components

- ProcessingElement
  - The supertype of all components
- Analytic: performs analysis of CASes
  - *Analyzer*: Processes a CAS and possibly updates its contents
  - *CasMultiplier*: processes a CAS and possibly creates new CASes
- FlowController
  - Determines how CAS should be routed through multiple analytics

Abstract Interfaces UML



Behavioral Metadata

- Declaratively describes what a UIMA analytic does
  - What types of CASs it can process
  - What elements in a CAS it analyzes
  - What sorts of effects it may have on CAS contents as a result of its application.
- Supports
  - **Discovery:** Locate components that provide a particular function.
  - **Composition:** Help determine which components may be combined to produce a desired result
  - **Efficiency:** Efficient sharing of CAS content among the analytics in a combination based on knowledge of analytic requirements.

Elements of Behavioral Metadata

- Supporting Discovery:
  - Analyzes (Sofas that the analytic intends to produce annotations over)
  - Required Inputs
  - Optional Inputs
  - Creates
  - Modifies
  - Deletes
- Supporting Composition:
  - Precondition: Predicate that qualifies CASs that the analytic considers valid input
  - Postcondition: Predicate that is declared to be true of any CAS after having been processed by the analytic, assuming that the CAS satisfied the precondition when it was input to the analytic
- Supporting Efficiency:
  - Projection Condition: Predicate that evaluates to the set of objects that the Analytic declares it will consider to perform its function.

Ways of Expressing Behavioral Metadata

- Type Names
  - Simplest Expressions
- OCL Expressions
  - Formal standard and semantic foundation
  - May be used for more complex applications
  - Simple expressions can be captured as OCL
  - Other options possible
- Views
  - A convenient way to specify inputs and outputs that pertain to a particular *Sofa*.

Behavioral Metadata Example

```
<behavioralMetadata
  xmlns:org.example="http://docs.oasis-
  open.org/uima/org/example.ecore">
    <analyzes>
      <type name="org.example:TextDocument"/>
    </analyzes>
    <requiredInputs>
      <type name="org.example:Person"/>
      <type name="org.example:Place"/>
    </requiredInputs>
    <creates>
      <type name="org.example:At"/>
    </creates>
 </behavioralMetadata>
```

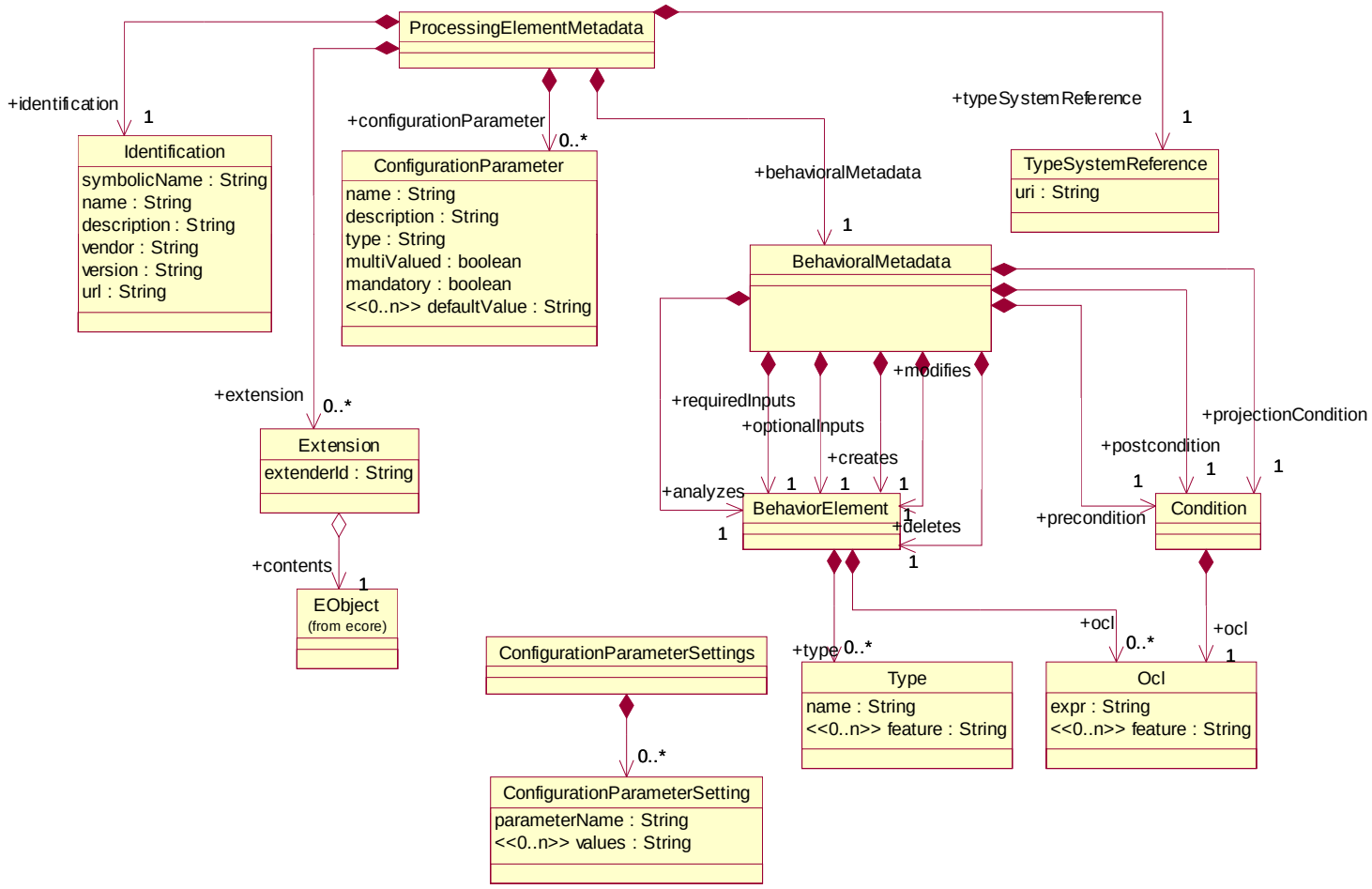
Generating OCL Expressions for Formal Semantic

- To give a formal meaning to the behavioral metadata, we can map each of the elements:
  - *analyzes*, *required inputs*, *optional inputs*, *creates*, *modifies*, and *deletes*
- To OCL preconditions, postconditions, and projection conditions.
- For example a *required input* of type Person becomes an OCL precondition that evaluates to true if and only if the CAS contains at least one instance of Person.

Processing Element Metadata

- Support Discovery and Composition
- All UIMA Processing Elements (PEs) must publish ***processing element metadata***
- Includes the Behavioral Metadata plus
  - **Identification Information.** Identifies the PE. It includes for example a symbolic/unique name, a descriptive name, vendor and version information.
  - **Configuration Parameters**. Declares the names of parameters used by the PE to affect its behavior, as well as the parameters' default values.
  - **Reference to a Type System**. Defines types referenced from the behavioral specification.
  - **Extensions**. Allows the PE metadata to contain additional elements, , the contents of which are not defined by the UIMA specification. This can be used by framework implementations to extend the PE metadata with additional information that may be meaningful only to that framework.

PE Metadata UML



WSDL Service Definitions

- Provides a WSDL document for the UIMA Processing Element Service Interfaces.
- Defines a binding to the SOAP protocol.
- This WSDL definition is an implementation of the Abstract Interfaces previously defined in the UIMA specification.
- This specification element intends to provide true out-of-the-box interoperability by specifying a concrete SOAP interface that compliant frameworks/services *must* implement.

Impact on Apache UIMA

November 9, 2007

Impact on Apache UIMA

- Well Aligned
  - CAS Data Representation: Apache UIMA already uses XMI
  - Abstract Interfaces: Very similar to the Apache UIMA interfaces
- Minor Differences
  - Type System Language Apache UIMA supports Ecore import/export, but still uses its own "native" type system language, and there are a few minor mismatches in semantics. Many constraints expressible in Ecore are not enforced.
  - Processing Element Metadata: Apache UIMA has some things not in the proposed standard:
    - Indexes
    - Configuration Parameter Groups
  - SOAP Interfaces: Would need to be implemented as new Apache UIMA service adapters. Apache UIMA is designed to make this relatively easy to do.
- Major Differences
  - Type System Base Model: significant differences
    - Apache UIMA Annotation Base Model is different from proposed standard
    - Apache UIMA Views are 1-1 with Sofas
  - Behavioral Metadata: Apache UIMA has limited behavioral metadata and it lacks precise semantics (and therefore can't be automatically converted to the proposed standard format).

Type System Base Model Differences

- Naming differences
  - Apache: uima.tcas.Annotation
  - OASIS: org.oasis-open.uima.TextAnnotation
- Differences in begin, end offsets?
  - Apache: UTF-16 code units (works well for Java)
  - OASIS: Unicode characters? (better for interoperability across platforms)
- Differences in what can be a Sofa
  - Apache: Annotation points to an object of type Sofa, which contains the data
  - OASIS: Annotation points to an object of type SofaReference, which then points to another object in the CAS that holds the actual Sofa data
- OASIS allows separate Regional Reference object.
- Differences in Views
  - Apache: Every Sofa has exactly one View, and every View must have a Sofa.
  - OASIS: Views are general collections of objects. A View *may* be linked to a Sofa, but this is not required. More than one View may be linked to the same Sofa.

Behavioral Metadata Differences

- Apache UIMA semantics are not well-defined
  - Allows specifying inputs and output but…
  - Not clear whether an input is required or optional
  - Not clear how input and output types relate to input & output Sofas
- Apache UIMA allows specifying multiple sets of capabilities
  - Allows specifying that different outputs may be produced depending on what inputs are received
  - Rarely Used

Service-Level Compliance

- Apache UIMA service adapters
  - Apache UIMA Analysis Engines can be deployed as UIMA-Standard-Compliant services.
- Implementing *getMetadata* operation
  - Type System converted to Ecore using existing converters (also need to convert base type system!)
  - PE Metadata can be serialized to standard-compliant format
    - Some things such as configuration groups would not be supported
  - Issue
    - How Can Apache UIMA Capabilities be published as OASIS UIMA Behavioral Metadata?

Service-Level Compliance

- Implementing *process* operation – need to map OASIS base type system. Not trivial but possible.
  - Convert Type Names
  - Convert character offsets
  - Create a Sofa object for everything that's pointed to by a SofaReference
  - Separate RegionalReference objects might not be supported (I think the spec may not require them to be)
  - For non-anchored Views could create a "dummy" Sofa
  - **Not Clear:** What to do with an incoming CAS where more than one View points to the same Sofa??

Deeper Compliance

- Over time Apache UIMA could “internalize” more of the UIMA standard representations.
  - Type System could natively use Ecore, and provide additional enforcement of Ecore constraints.
  - Descriptor formats could support UIMA-standard metadata XML.
  - Behavioral Metadata could move to the OASIS standard and get away from the underspecified representation currently supported.
  - The standard Type System Base Model could be supported natively in the CAS.

Backup Slides

Out Takes

Dropped Design Goal

- Original Specification Draft contained the design goal:
  - **Platform Independent Development.** Facilitate the compliance of existing applications or the development of new applications on different platforms and in different programming languages.
- This seems out of place now since the specification is only defining services interfaces for UIMA. We do not address programming language bindings at all. APACHE UIMA defines Java Bindings.

Other Notes from 10/26 telecon

- Administrative API:
  - Pascal: Administrative API useful. Perhaps just a method on Abstract Interfaces that returns a log?
  - Adam: If there's no standardization of log file content this may not be very useful.
  - We could try to define some standard kinds of log messages, but this would require more thought/discussions.
- Examples

Design Goal Overview

Provide a standard specification for text and multi-modal analysis that supports *data and service level interoperability* to facilitate the rapid combination and deployment of analytics in the development of UIM applications

Redundant – Remove or Reword and not make as specific with regard to UIM applications

(REMOVED)

UIMA Framework Adoption

- Gartner Report Quote
- Universities
- Government
- Healthcare
- Business Intelligence
- 10’s of thousands of downloads
- Ported or Wrapped Analytics
  - OpenNLP
  - NetOWL
  - Julie