|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Types | Description of purpose | UML importance | XML importance | RDF importance | Comments |
| Primitive | Limited set of common constrained strings | Supports a limited set (Integer, Boolean, String, UnlimtedNatural, Real) | Currently uses [Primitive XML Data Types](about:blank), some [Derived XML Data Types](about:blank), and incorporates some [Data Type Facets](about:blank) | [RDF and OWL recommendations for use of simple types from XML schema](about:blank) | DDI currently uses and differentiates between [Primitive XML Data Types](about:blank)  primitives not used by UML |
| Regular Expression | Primitive with a DDI defined constraint based on a pattern |  | Supports verifiable constraints on content | Supports verifiable constraints on content | Currently in use |
| Enumeration | Primitive with a DDI defined constraint based on a list of allowed strings (terms) | Supports use of enumeration list | Supports use of enumeration list. Expressed as simple element | Supports use of enumeration list | Currently in use |
| Cardinality Types |  |  |  |  |  |
| Property type is primitive; cardinality not repeatable | Capture simple non-repeatable unstructured content conforming to a simple type (Cannot exist outside of parent; cannot be referenced) | Common UML property definition | Could be translated as attribute or in-line element. No identifier (agency, id, version) |  |  |
| Property type is primitive; cardinality repeatable | Capture simple unstructured content conforming to a simple type which may be repeated  (Cannot exist outside of parent; cannot be referenced) | Common UML property definition | Should be translated as an in-line element. No identifier (agency, id, version) |  |  |
| Class content |  |  |  |  |  |
| Properties only  (By number of additional levels contained:  0 levels: 74  1 level: 27  2 levels: 13  3 levels: 5  4 levels: 2 | Capture complex structured content conforming containing multiple properties which may be repeated; Bundles a set of properties within a parent  (Cannot exist outside of parent; cannot be referenced) | UML would handle this as a relationship  Note that UML provides class content information for enumerations and data types but is there an issue with visualizing content with multiple levels | Should be translated as an in-line element. No identifier (agency, id, version). Do the multiple levels conflict with the goal of flattening the structure. |  | Supports consistent use of a set of properties across several object types; example a bundle of text strings; components of an address) |
| Properties and/or relationship  (56 classes) | Capture a complex structure that contains at least one simple relationship (Cannot exist outside of parent)  Is a component part of a whole but can only be a component part of a single whole. May need to support the ability to reference it in relationship to its role in the composite whole. | UML Composition relationship. Could have identification (agency, id, version) if it requires a another class to be able to point to a specific instance within the parent object | Should be translated as an in-line element. |  | Supports the use of independent objects within a parent class where the combination of the independent object(s) and/or specified properties is unique to the parent class (example: indexed order of a classification within a classification set). Can contain an identification (agency, id, version) without requiring versioning support. |
| Class Independence |  |  |  |  |  |
|  | Target is an Independent object that has a relationship other than partitive to the parent. Can exist independently. Requires full class management information (agency, id, version, ownership, citation, and change information) | Independently managed classes; assumption is that may be part of a whole but not dependent upon the whole for existence. |  |  |  |
| Class Membership |  |  |  |  |  |
|  | Target is an Independent object that is a part of a larger whole as in a component. Can be part of more than one whole. | Independently managed classes; assumption is that may be part of a whole but not dependent upon the whole for existence. |  |  |  |