

Issue 214: Separating Identification and Naming from Administration

Project #: 1.32.15.03.03.00
Standard #: 11179-3 edition 3
WG2 doc #: WG2 N0870 WD 11179-3 Edition 3 version 2
Submitter/Source: Ray Gates
Date Submitted: 2006-03-23
Severity: 1 Major Technical
Status: Open
Date Last Updated:
Updated by:

Problem Description:

There is a need to be able to both **identify** and **name** items in a registry without necessarily providing full administrative information. This arises in two ways:

1. there are some classes in the existing model which are not administered items, but have names (e.g. Unit_of_Measure, Datatype, Classification_Scheme_Item) or identifiers (e.g. Value_Meaning)
2. it is desirable to provide organizations with more flexibility over the items for which they provide administrative information.

Proposal:

Separate out **Identification** and **Designation** from Administered_Item, allowing items to be identified and named without the overhead of a full administration record.

Kevin Keck has provided:

- a revised Figure 2, which introduces a class 'Designated_Item' as a super-type of not only Administered_Item, but also other classes which are not currently administered items;
- a new Figure E, which introduces both 'Designatable_Item' and 'Identified_Item', where the latter is a specialization of the former, and Administered_Item is a specialization of the latter.

Issues:

- Is the use of both 'Designated_Item' and 'Designatable_Item' deliberate? Are they supposed to be the same or different? If the same, which is the better term?
- ebXML version 3.0 uses 'Identifiable' and 'RegistryObject', where the latter is close to a 11179-3 Administered_Item.
- Does it make sense to have items with names but no identifiers, or identifiers but no names? The latter seems more likely than the former, but the former is what the proposed Figure E supports.

References:

-
-
-

Discussion/Notes:

--

Committee Proposed Resolution:

Draft Resolution

Draft Resolution Date

Draft Resolution Source

Status of implementation

Final Resolution

Final Resolution Date

Final Resolution Source

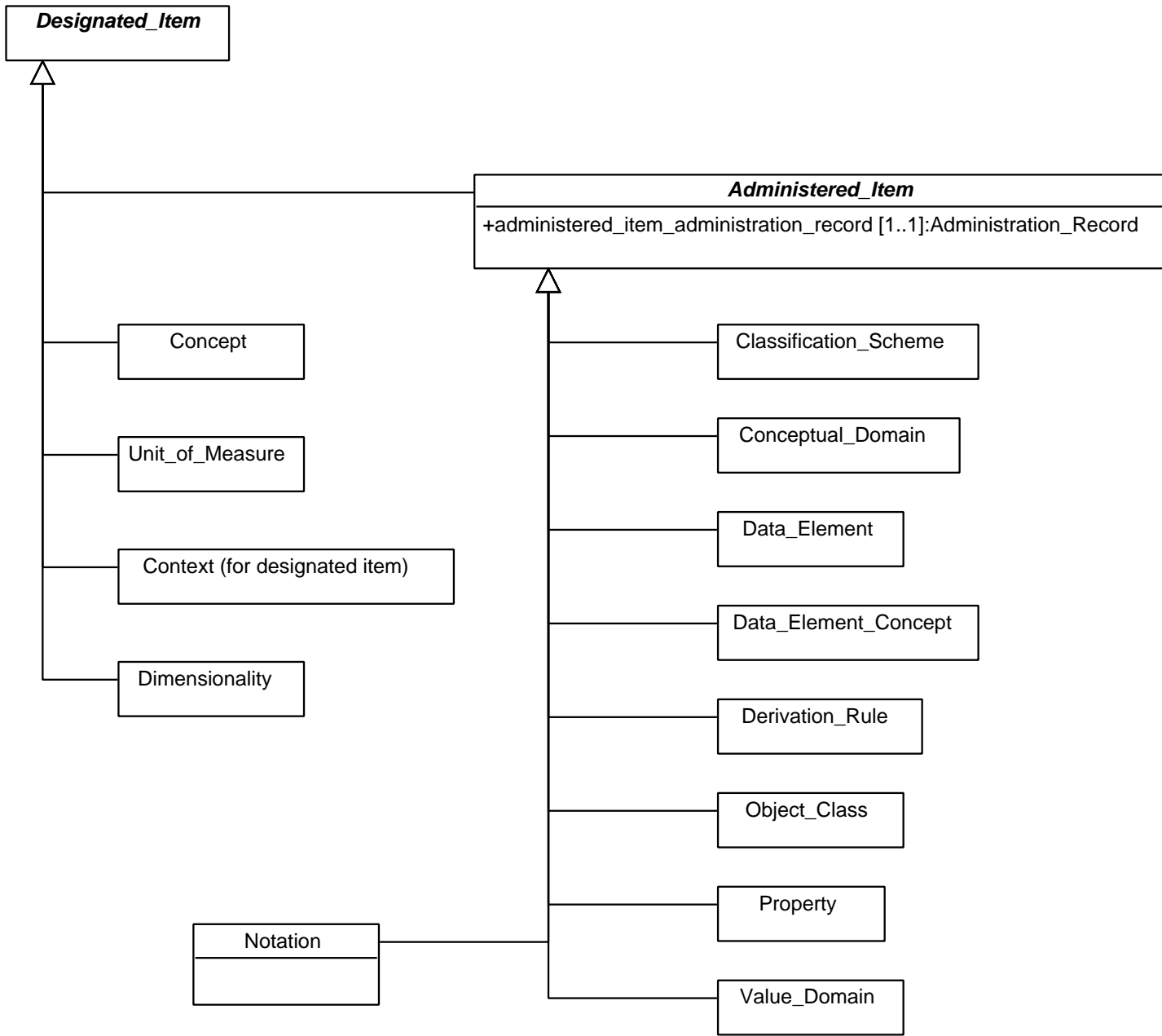
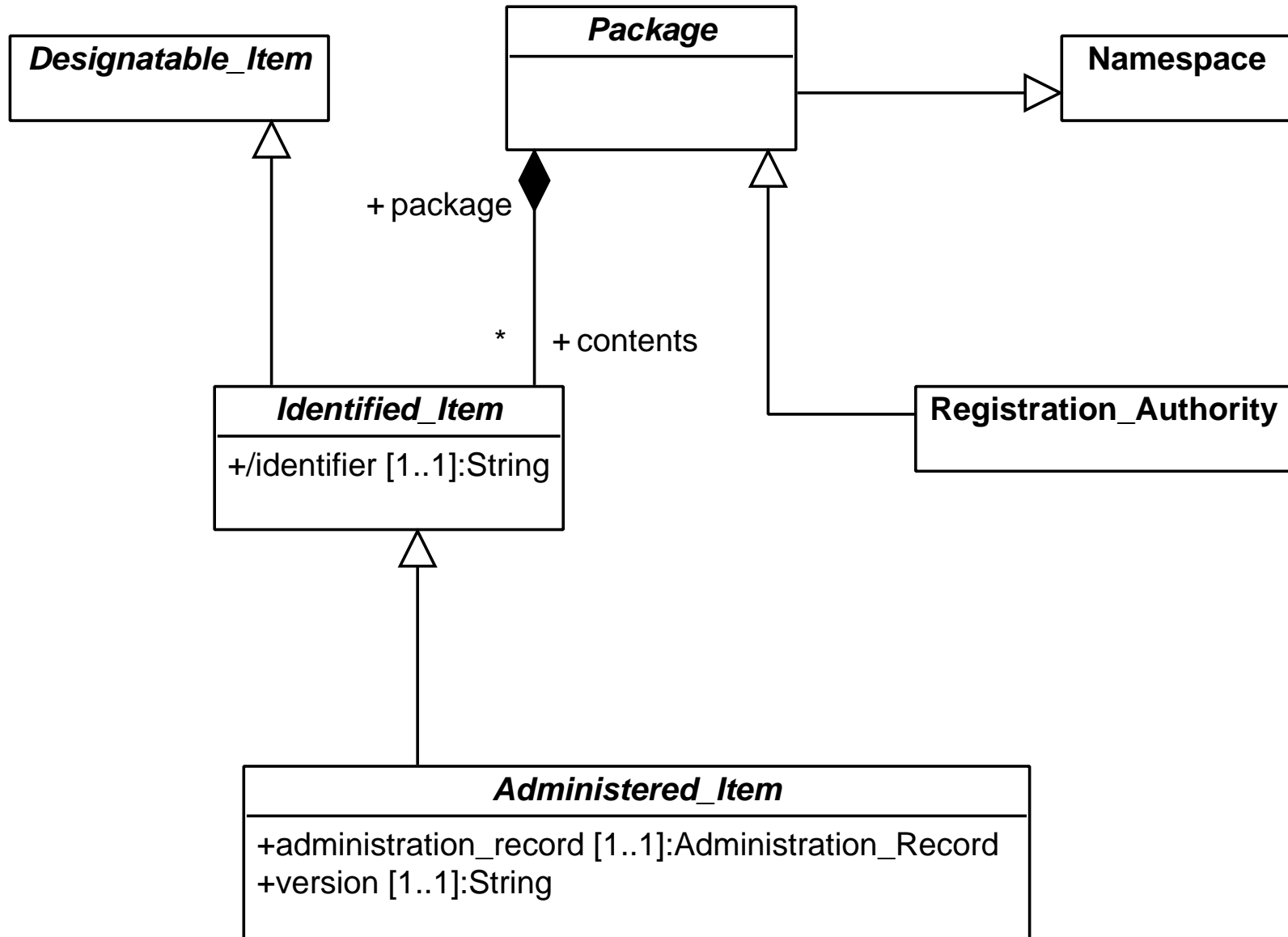


Figure 2 - Types of Administered Items
(modified - Keck, 2005-09-16)

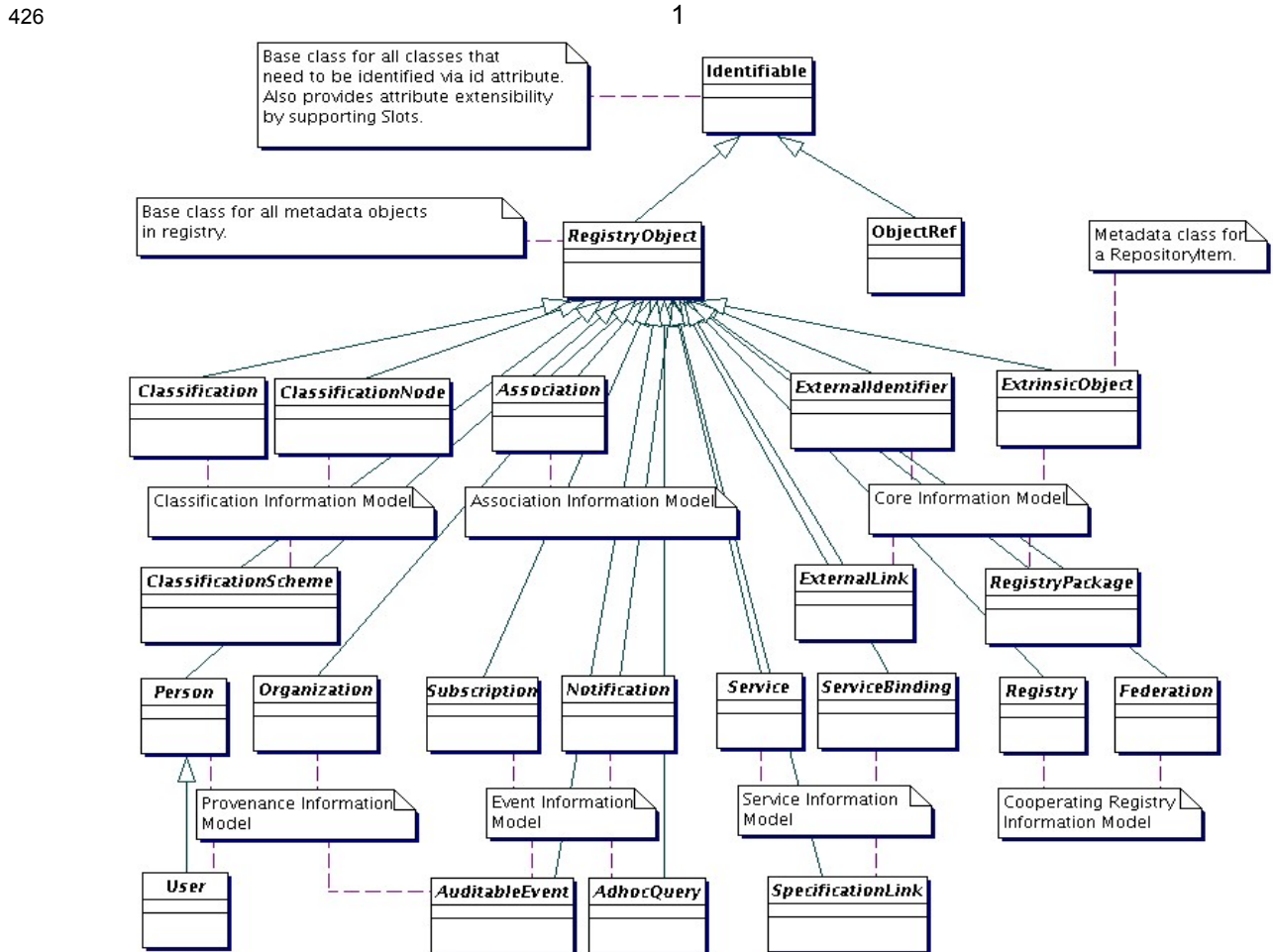


[Figure E Identification metamodel region]
 (Keck, 2006-02-16)

420 Detailed description of attributes of each class will be displayed in tabular form within the detailed
 421 description of each class.

422 1.7.2.1 Class Identifiable

423 The RegistryObject class and some other classes in RIM are derived from a class called *Identifiable*.
 424 This class provides the ability to identify objects by an id attribute and also provides attribute
 425 extensibility by allowing dynamic, instance-specific attributes called Slots.



427 **Figure 2: Information Model Inheritance View**

428

429 The RegistryObject sub-classes are shown in related groups as follows:

- 430 • Core Information Model: Defines core metadata classes in the model including the common base
- 431 classes.
- 432 • Association Information Model: Defines classes that enable RegistryObject instances to be
- 433 associated with each other.
- 434 • Classification Information Model: Defines classes that enable RegistryObjects to be classified.
- 435 • Provenance Information Model: Defines classes that enable the description of provenance or source
- 436 information about a RegistryObject.
- 437 • Service Information Model: Defines classes that enable service description.
- 438 • Event Information Model: Defines classes that enable the event subscription and notification feature
- 439 defined in [eBRS].