



IPTC “Tower of Standards” Architecture – DEMO “abcML”

(DRAFT by MD)

Document history				[Document URN: urn:iptc:workdoc:dir:04010:1]
Revision	Issue Date	Pages	Author (revised by)	Remark
unrevised	2004-12-08		Michael Steidl	
Rev a.				

This document intends to summarise the current state of discussing how to build a (new) IPTC standard while abiding to the framework of the IPTC “Tower of Standards”.

Making of a standard (“abcML”), step by step – with Design Rules for building it first:

Design Rule 1: Any content is contained by an “item” structure. The basic “item” structure is agnostic to any content = can contain any type of content.

Design Rule 2: The basic characteristics of an item are:

- it has a globally unique identifier = makes it identifiable, an item can be referenced
- it has (basic) management facilities (versioning information, publish status ...)
- it contains zero or more “pieces” of content. If more than one is present these pieces are only allowed to be different (physical) renditions of the same content. (e.g.: thumbnail/full-res picture, audio of different sampling rates ...)
- it contains metadata about the (pieces of) content – which metadata is specified by an individual standard

Design Rule 3: The content itself could be of various types: XML content, text, image data, audio data, video data (inclusion by value or by reference)

Design Rule 4: A set of “Common Components” (CC) are available:

- they can be used as “plug-ins” to extend the metadata facilities of an “item” – see Design Rule 2.
- The basic characteristic of CCs is: they can be used by several standards = they are not specific to any type of content.

Design Rule 5: “Standard Specific Components” can be specified as “plug-ins” to extend the metadata facilities of an “item” – such components are specific to a certain content type.

Step 1: Specify the specific content (mark up) for this standard.

Step 2: Specify a content type identifier for this standard.

Step 3: Select a set of appropriate Common Components to be adopted by this standard.

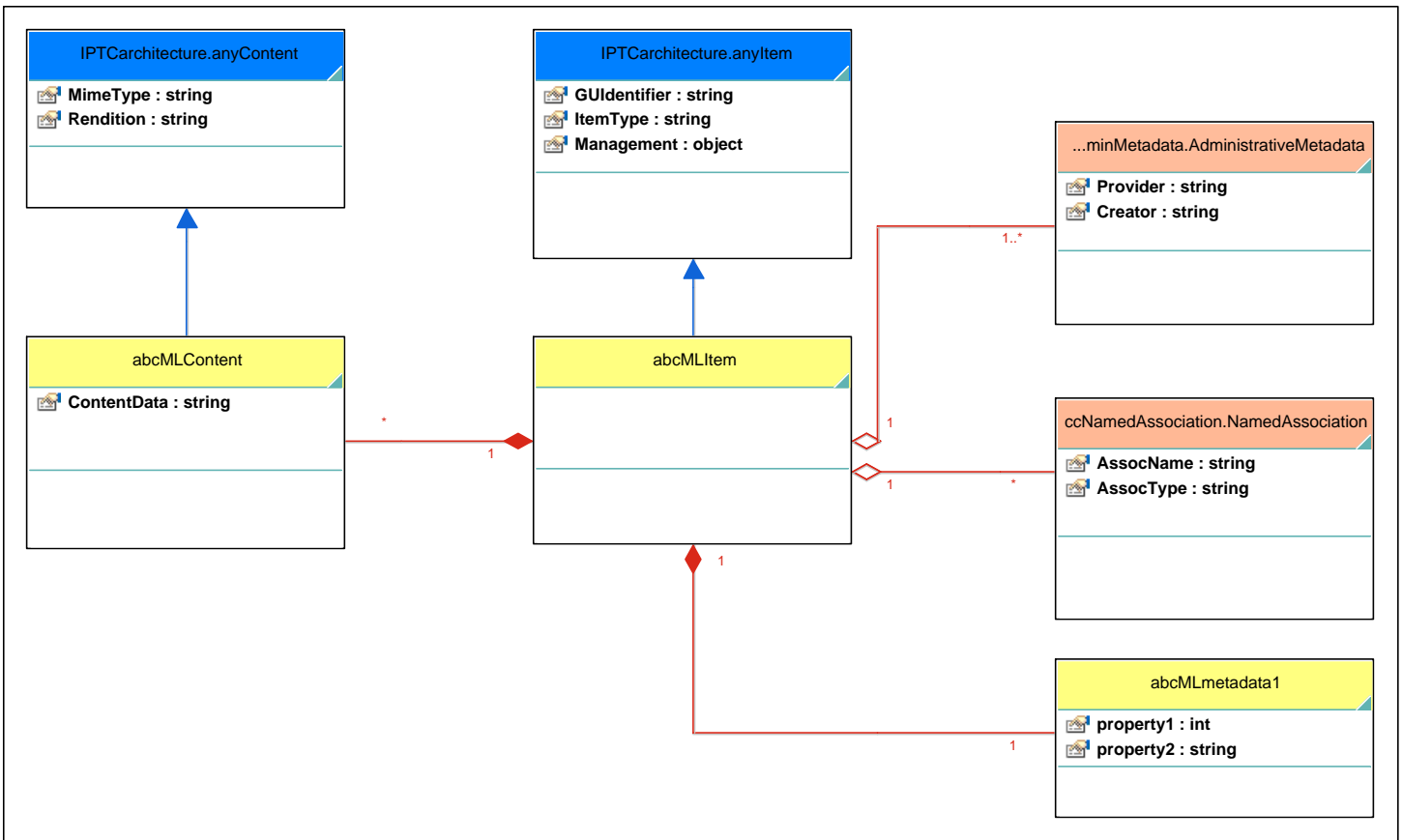
Step 4: Develop metadata components specific to this standard.

Step 5: Specify all these elements by extending an abstract “item” template to make it an “abcML-item”.

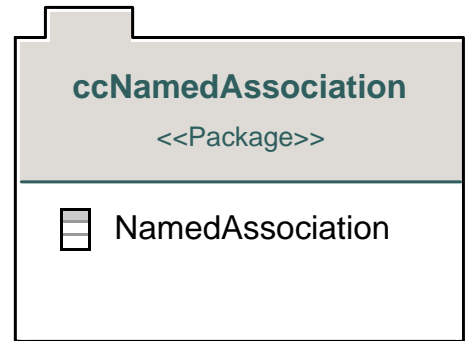
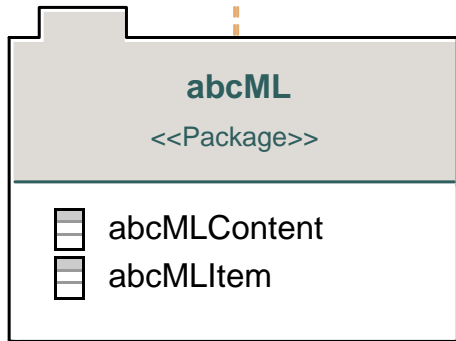
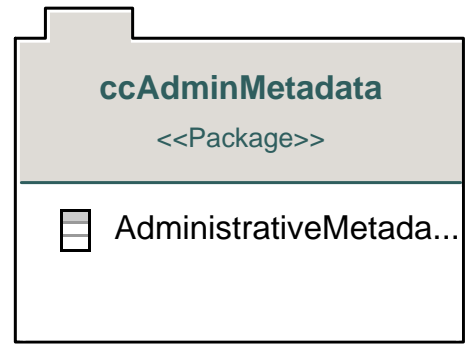
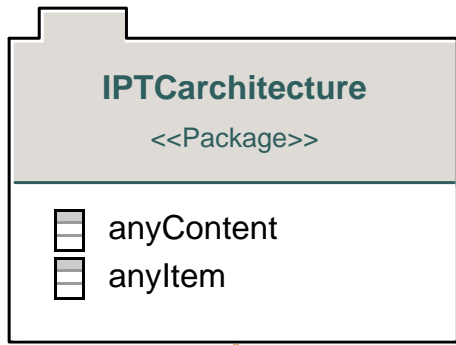
The next two pages illustrate the architecture of “abcML” by using UML for notation: (The details of the individual classes are only a sketch!!)

- First figure: UML class diagram to specify the class relationships: “blue” classes are the abstract templates, “apricot” classes are the Common Components, “yellow” classes are specific to “abcML”
- Second figure: four packages (= namespaces!). One for the abstract templates, one for each Common Component, and finally one for the “implementation” of “abcML”.

Continued →→→



abcML



«Use»