

Introduction to Alfresco Content Models

Classifying Alfresco content for a specific domain

Overview of Alfresco Content Models

- One of the main differences between a network drive and a content management system (CMS) such as Alfresco is that the latter provides extra **classification** features
- If we look at any CMS behind the scenes, we can see that everything in the repository is typically a **node** or an object
- **Properties** are then set on the nodes so they become folders, files, categories, rules, forum posts, web pages, e-mails, people, groups, and so on
- Basically, the nodes are **classified** so it is **easier to search** for them, so we know how to display them in the user interface, what they can be used for, and so on

Overview of Alfresco Content Models continued

- These properties cannot be just any properties, as then the CMS would not know what each one of these properties represents/means
- So a CMS comes with a **definition language/metamodel** so properties can be defined in a structured way
- Alfresco comes preconfigured with a number of properties that can be used to classify content in the repository directly after it has been installed
- These properties are usually organized into groups that are called either **Types** or **Aspects**
- Furthermore, nodes do not live by themselves in the repository. They are related to one another in different ways with so-called **Associations**

Overview of Alfresco Content Models continued

- Types, Aspects, Properties, and Associations are in turn organized into models that we call **Content Models**
- A content model is used to describe the data being stored in the repository, or we can also say that it is used to **classify** the data being stored
- The content model is critical - without it, Alfresco would be little more than a file system

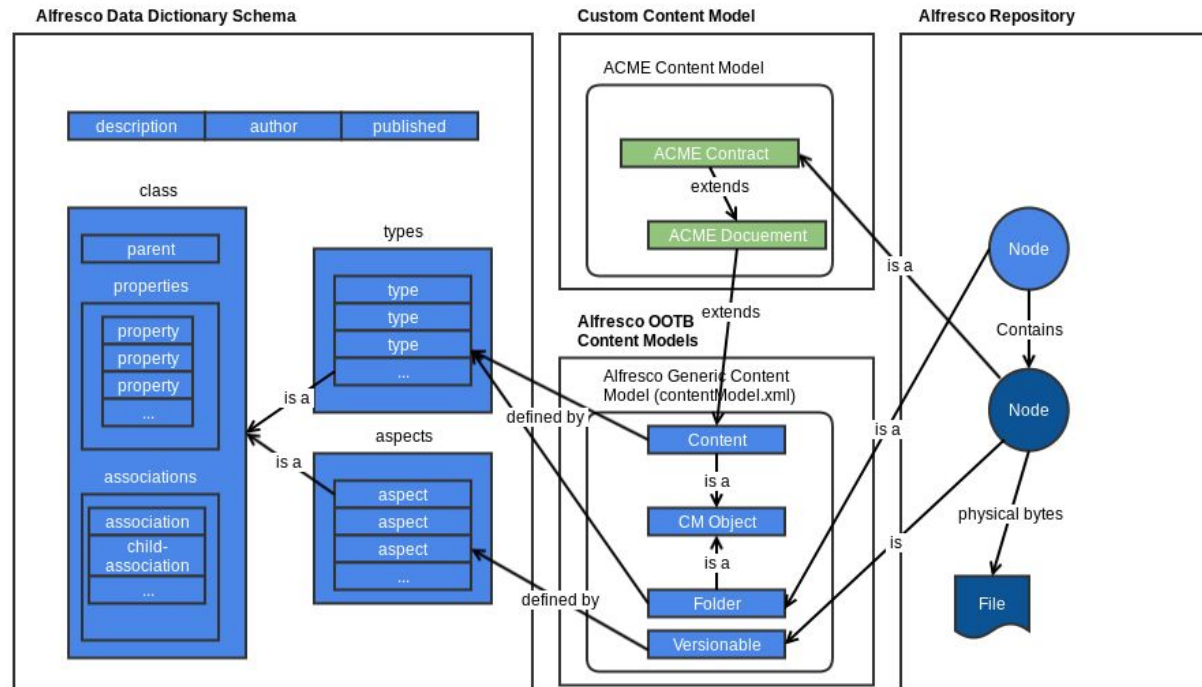
What does a Content Model provide to Alfresco Repository?

- Here is a list of key information that the content model provides to Alfresco Repository:
 - **Fundamental data types** and how those data types should be persisted to the database. For example, without a content model, Alfresco wouldn't know the difference between a "String" and a "Date"
 - **Higher order data types** like "Document" and "Folder" as well as custom content types like "White paper", "HR Document", or "Contract"
 - **Out-of-the-box aspects** (secondary types) like "versionable" and "taggable" as well as custom aspects like "rateable" or "commentable"
 - **Properties** (metadata) specific to each type and aspect
 - How to **index** content for searching
 - **Relationships** between types

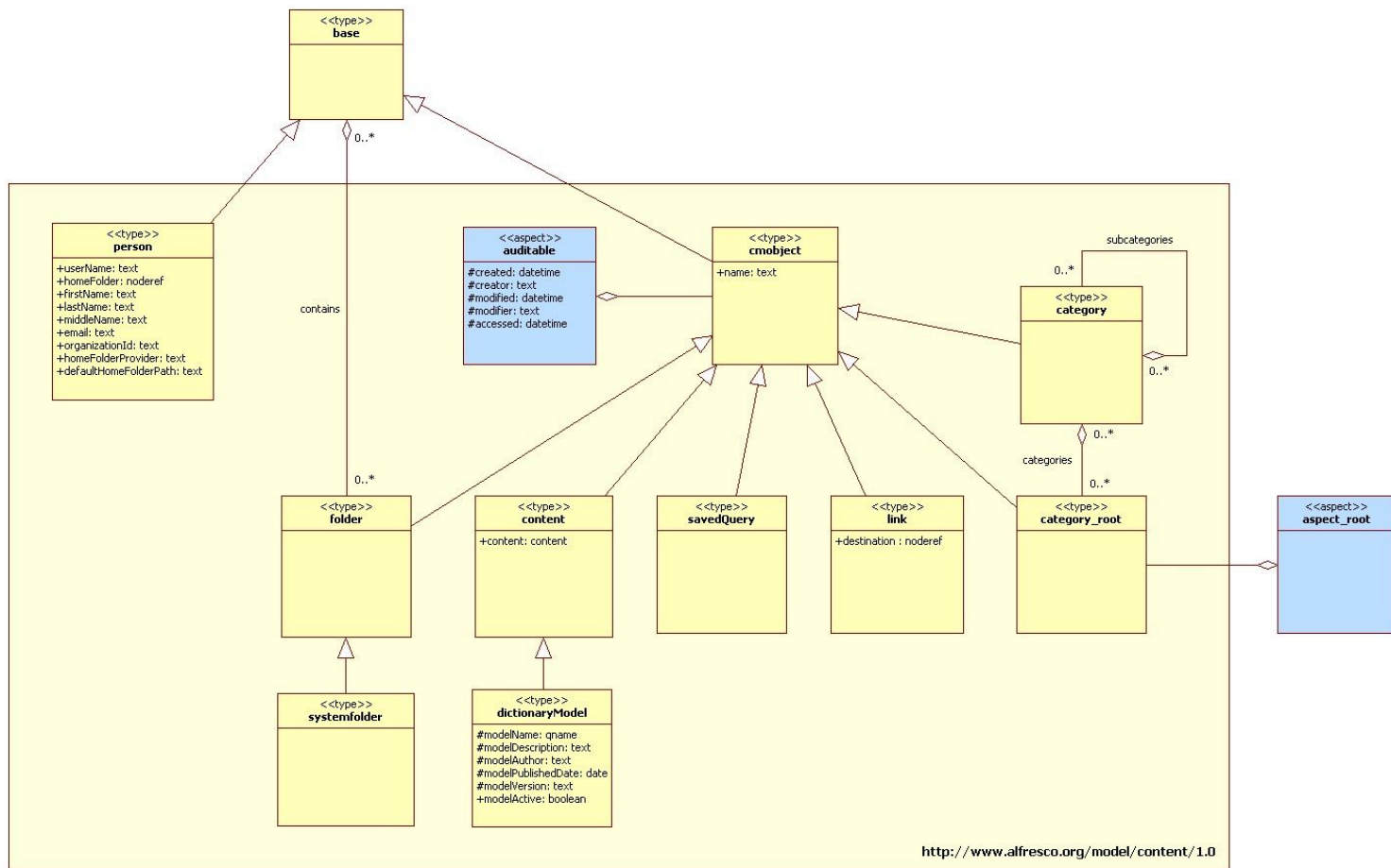
Content Model Definition Language

- Content Models are defined in XML based on an XSD meta model
 - But we are going to use the UI Design tool called the **Model Manager** (so no need to mess around with XML at this point)
- A generic content model is available out-of-the-box for folders and documents (i.e. files)
- When defining content models two classification concepts are used:
 - **Types** – a content node can have one type applied
 - **Aspects** – also called secondary types and a content node/object can have multiple aspects applied

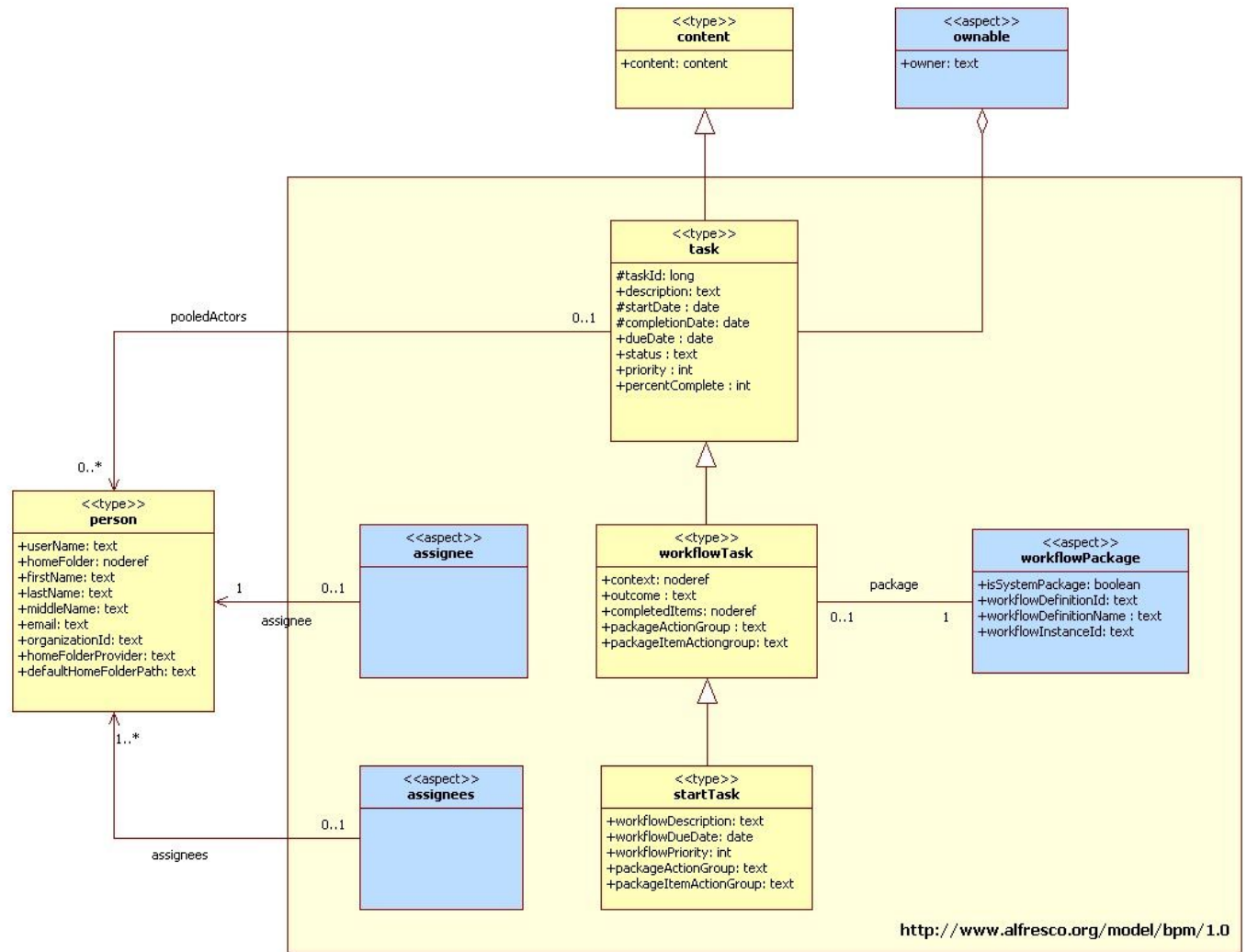
Relationship between Metamodel, Out-of-the-box Model, and Custom Model



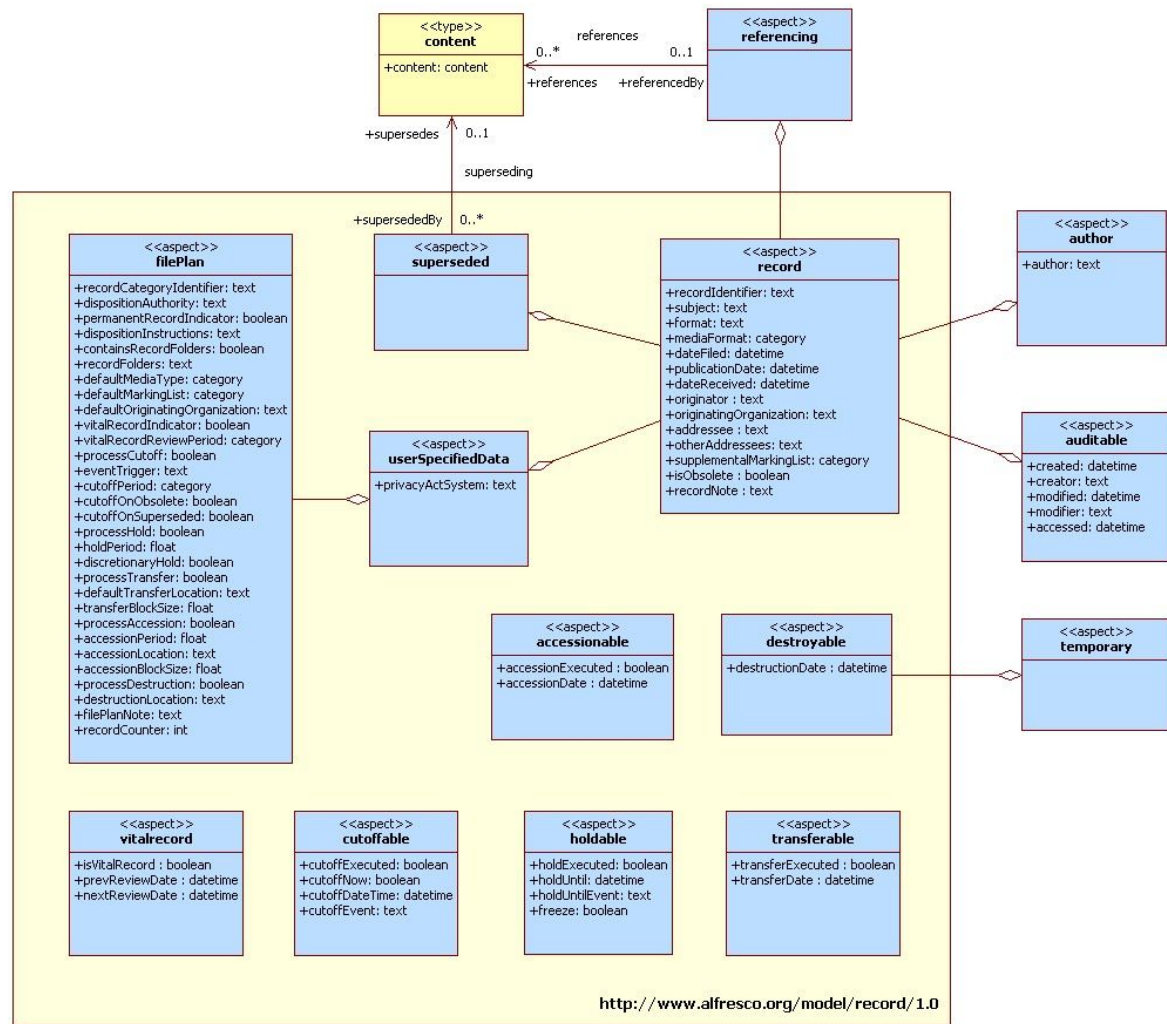
Out-of-the-box Content Models (Types): Folders and Files Model



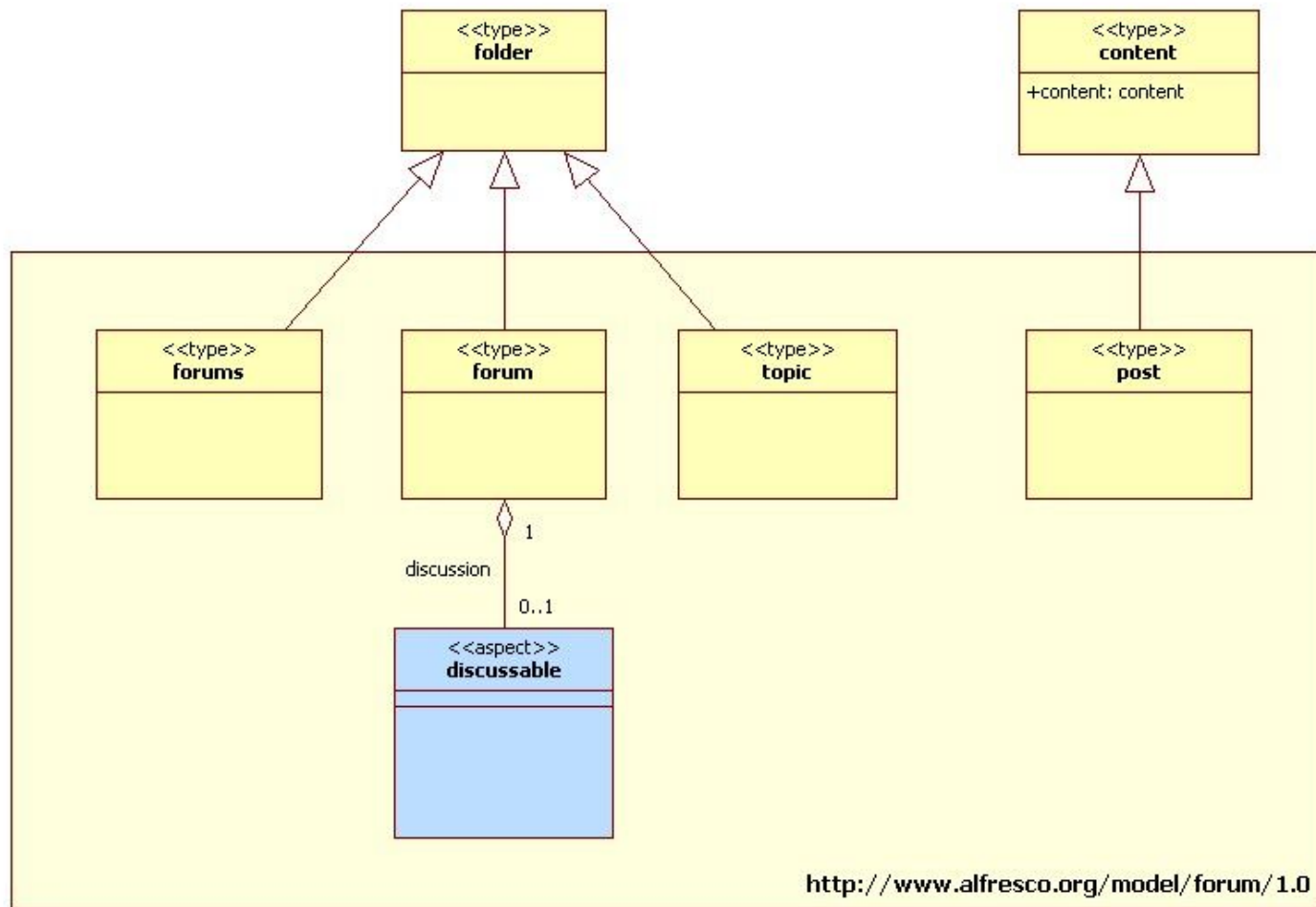
Out-of-the-box Content Models (Types): Workflow Model



Out-of-the-box Content Models (Types): Records Management Model



Out-of-the-box
Content Models
(Types):
**Forum
Model**



Out-of-the-box Content Models (Aspects)

- Some of the aspects available out of the box:
 - Transformable, Replaceable
 - Attachable, Referencing
 - Titled, Author, Auditable
 - Effectivity, Versionable
 - Workingcopy, Classifiable, Countable
 - Emailed, Published
 - EXIF, Geographic

When to Use Type and when to use Aspect?

- So when should we use a type and when should we use an aspect when defining a new content model for our domain?
- A rule of thumb is to listen to the business analysts when they talk about what they want and write down all nouns, adjectives, and verbs. They would then be converted to types or aspects as follows:
 - **Noun** – a type
 - **Adjective** – a subtype
 - **Verb** – an aspect (a.k.a. secondary type)

When to Use Type and when to use Aspect continued?

- Take the following sentence for example:
 - *“Acme Documents in this folder are finance whitepapers that should be versioned and that should be allowed to be published on the web, there should be a marker for if the document was emailed into us or not”*
- From this sentence we can extract the following types and aspects (secondary types):
 - **ACME Document** – type (custom)
 - **Folder** - type (out-of-the-box)
 - **Whitepaper** – type (could potentially also be an aspect...) (custom)
 - **Finance** (whitepaper) – subtype of Whitepaper (custom)
 - **Versioned** – aspect (out-of-the-box)
 - **Published** – aspect (out-of-the-box)
 - **Emailed** – aspect (out-of-the-box)