

Old Content models

Content Models: Getting Started

The first problem when thinking of content models is probably trying to nail it down and come up with a definition. So, here is a first step in this direction: A content model can be seen as a combination of

- a structured definition of a type (e.g. article, book, image, ...)
- a set of constraints
- a pattern of datastreams (number and type)
- a pattern of datastreams and disseminators
- a subclass of a digital object
- a set of rules for creating a new object

Typically, a content model relates to one object type, not so much to a whole repository. Fedora is very flexible when it comes to content models. While this is one of its eminent features, at the same time it burdens the repository designers with the decision on how to set up a content model suitable for their needs. Object models can be viewed on four levels:

- Typing of objects (data formats)
- [Structural definition](#)
- Service definition (disseminators)
- Logical model (aggregation of all content models within a repository)

Fedora currently doesn't support explicit content models which are strongly typed. This allows setting up a repository without explicit content models. While this may look intriguing to an inexperienced user, it typically leads to a messy repository which may conceal the full value and richness of its digital artifacts caused by the lack of systematics.

Some kind of informal typing can be achieved by the CMODEL property, but this is rather based on a 'best-practice' approach than on prototypes and automated validation. Many projects have implemented some kind of validation mechanism (one of the most advanced is certainly the work of [Kostas Saidis and George Pyrounakis](#) of University of Athens with their Pergamos project).