

# 2014-03-20 - Special Topic - Content Model Validation

## Attendees

- A. Soroka
- Eric James
- Andrew Woods ★
- Benjamin Armintor
- Stefano Cossu
- Peter Gorman
- Nigel Banks
- Nick Ruest
- Mike Stroming
- David Wilcox
- Ed Fugikawa
- Michael Durbin
- Martin Dow

## General

- Time: 4 PM Eastern
- Indicates who took minutes - ★
- ReadyTalk call-in details:
  - U.S.A/Canada toll free: 866-740-1260, participant code: 2257295
  - International toll free: <http://www.readytalk.com/intl>
    - Use the above link and input 2257295 and the country you are calling from to get your country's toll-free dial-in number
    - Once on the call, enter participant code 2257295

## Agenda

1. Description of context and problem space
2. Discussion of use cases
  - a. inherited from Fedora 3
  - b. new for Fedora 4
3. Discussion of [proposal](#)
  - a. Does it meet the requirements?
  - b. Is there an impact on the use of CND?
  - c. ...
  - d. Next steps?

## Minutes

### Context setting

Fedora has history of content modeling (CM).  
Currently leveraging inherited type system offered by ModeShape.

- It has some limitations in the F4 context  
Art Inst. of Chicago then brought specific needs to the table
- Artic began pushing on Compact Node Definition (CND) capabilities and short-comings  
This is has raised the question of whether CND meets the needs
- What are the gaps?
- What can we do to fill the gaps

### Use cases

What use cases have not yet been recorded?

Q: Is F4 CM at the level of F3 ECM

a: In many ways F4 is more expressive

- Note, much of the underlying JCR type system is not being exposed

Q: What is most important to you about CM?

- Eric: Proposal, represent CM in RDF, not via CND
- Ben: Concern that proposal may be too broad in reach
  - Ideally, the additional functionality be optional
  - Current CND is more than adequate
- Stefano: CND covers 90% of needs
  - Drawn to starting point of restrictive types, and opening with mixins

- Is it possible to layer a fedora abstraction over CND?
  - Priority is placed on restriction, current F4 is permissive
- Peter: Invested in mixin content models
  - Less concerned about restriction
  - Interest in being able to tag objects with content models or properties
- Nigel: F3 use of CM is nominal
  - Drawn to the idea of new characteristics based on mixins
  - Notionally interested in validation of object structure
- Nick: same as Nigel
- Mike Stroming: needs internal CM discussions
- Martin: Interested in repository user roles
  - Interested in runtime flexibility, and optional validation
  - W3C languages offer powerful expression
- Ed: Resonates with Nigel/Nick/Ben/Martin
- Mike Durbin: Likes ability to enforce relationship rules
  - Validation can happen in the repo or in the upper-level application

Summary: People are interested in an easy-to-use, nominal type system.  
 Restriction and validation is less of a priority to most.

Nigel would like to ensure that if a content model includes a specific datastream, that that datastream exists with the correct mimetype.

Stefano: suggestion to associate access policies to CMs

Martin: suggestion for process moving forward: timeboxing requirement gathering, roles, proposal, etc

CM in F4 meets most of the community need  
 There is another conversation to be had regarding extension points for validation

## Actions

-