Migration issues of CMDI 1.2

Twan Goosen & Jozef Misultka & Menzo Windhouwer

CLARIN ERIC & LINDAT-Clarin & The Language Archive – DANS

twan@clarin.eu & misutka@ufal.mff.cuni.cz & menzo.windhouwer@dans.knaw.nl

CMDI Taskforce Meeting
Utrecht
2014-02-21

Statements to (re)consider

- Existing metadata and tools should not break as a result of the appearance of the new version of CMDI
    - But 1.2 instances are not backwards compatible, so central infrastructure components need to deal with 1.2
- Migration is strongly recommended but should remain optional
    - The core infrastructure components will continue to support 1.1
- New tools should ideally support both CMDI 1.1 and CMDI 1.2 metadata
    - in most cases this will require some degree of version-awareness
- There is no need to support migration ('downgrading') of CMDI 1.2 based *instances* to CMDI 1.1
    - Locally if a center/service/tool is still based on 1.1, but also need to handle ‘external’ instances, which might use 1.2
    - In known cases specific profiles are used, which can stay 1.1 compatible, and downgrading can be handled per profile, i.e., no need for general downgrading support

Migration of components

- The ComponentRegistry will migrate to 1.2, but is able to still export 1.1 versions of components/profiles
  - For this the 1.1 -> 1.2 -> 1.1 roundtrip needs to be lossless, and analysis showed that's true
- It will be possible to use natively created 1.2 components/profiles, which use 1.2 specific features, for 1.1 instances
  - This can cause problems if these 1.1 instances later move on to become 1.2 instances
    - A mandatory attribute *notation* in a 1.2 profile
    - Is an optional attribute *notation* in the 1.1 version of the profile
    - So 1.1 instances of that profile can omit the *notation* attribute
    - If these instances are upgraded to 1.2 they would be invalid for the 1.2 profile
  - Keep track of the CMD version at time of creation of the component/profile, so we know we should use a dumbed down version
    - 1.2 -> 1.1 -> 1.2'
    - Mandatory attributes become optional
    - Lost (multilingual) documentation, but that's no problem for validation

Migration of instances

- There will be an XSLT that can convert 1.1 instances to 1.2
  - Determine the used 1.1 profile
    - MdProfile or xsi:schemaLocation
    - If unknown throw an error and exit
  - Refer to the 1.2 schema of the profile
    - Or 1.2' if the profile is natively 1.2
  - Do all the changes required to be compliant by CMDI 1.2
- Sometimes downgrading from 1.2 to 1.1 might be needed, but at the moment is seems not needed to provide a generic XSLT to do so
  - WebLicht might expect 1.1 instances for WebService descriptions, but centers might start offering 1.2 instances
    - Just one specific XSLT needed for a specific profile

Tool aspects

- Infrastructure
  - ComponentRegistry: mass migration of components/profiles to 1.2, but allows 1.1 and 1.2' export
  - VLO: should support both 1.1 and 1.2, but can upgrade 1.1 to 1.2 in the importer and deal internally only with 1.2
  - SMC Browser: can deal internally with 1.2, and accept 1.1 via an upgrade to 1.2, but will also have to deal with component/profile specification changes
- Editors
  - Arbil: support both 1.1 and 1.2, special care needs to be taken with the namespaces
    - Can be used for a mass migration of 1.1 instances to 1.2
  - Proforma: will support one version, either 1.1 or 1.2; a 1.2 version might support 1.1 conversion
- Web Services
  - WebLicht: chaining engine could be based on 1.1 instances still, but centers might start to offer 1.2 instances which can be downgraded using a simple XSLT
- Some more tools need to be assessed