Control de integridad y calidad en repositorios

DSPACE

Motivación. Condiciones actuales

1. El origen de los datos es diverso (autoarchivo, cosechas OAI-PMH, depósito vía SWORD, etc) y por tanto varían sus características
    - Es necesario validar que los ítems archivados respeten la política de contenidos.
2. Con el tiempo cambian los requerimientos mínimos que debe cumplir un ítem en el repositorio
    - Surge la necesidad de verificar periódicamente estos requerimientos mínimos
3. A medida que aumenta el volumen de datos, cada vez es más complicado detectar anomalías
    - Se necesitan mecanismos de control/monitoreo de metadatos y archivos.

Herramientas para Control de datos en repositorios DSpace

- ChecksumChecker
  - Verifica la integridad de los archivos
- MediaFilter
  - Transforma un bitstream de un ítem o extrae uno nuevo derivado del anterior. Ejemplo: imágenes eb miniatura, fulltext, etc.
- EmbargoChecker
  - Verifica que todas las partes de un ítem embargado (ítem, bundle, bitstream) se mantengan ocultas
- *Curation Tasks* o Tareas de *Curation*
  - permite un control semi-automático de objetos del repositorio

Curation Tasks en DSpace

- Permite aplicar una *"pieza de software"* sobre uno o varios objetos del repositorio.
- Una *tarea*
  - se aplica sobre un objeto puntual (ítem, Collection, Community)
  - puede acceder y modificar cualquier aspecto de su contexto como metadatos, bitstreams, etc.
- Provee varias tareas predefinidas. Por ejemplo:
  - Format Profiling → análisis de formatos de archivos
  - Required metadata → metadatos requeridos
  - ClamScan → detección de virus en los archivos
  - Microsoft translator → traduccion automática
  - Link Checker → Validacion de links

Tareas para control de calidad y preservación. Propuesta

I. Control y preservación de datos a partir de tareas de curation que realicen:
   - chequeos de calidad,
   - control de integridad y
   - extracción/generación de nuevos metadatos

I. Extensiones al módulo de curation de Dspace para:
   - mejorar las estrategias de selección de recursos a procesar.
   - agregar nuevas estrategias de ejecución de tareas para reducir su costo de ejecución (en tiempo y recursos)

Tarea 1 Chequeo de links muertos

Casi todos los ítems de repositorio tienen al menos un link en sus metadatos. Por ejemplo:

- URL a la licencia de uso
- URL de acceso al recurso (para recursos externos)
- URL al origen del registro
- URL a otras versiones o trabajos relacionados

Con el tiempo los links suelen dejar de funcionar:

- temporalmente: error interno, servidor en proceso de actualización, etc
- permanentemente: Cambio en el dominio o ruta, servicio descontinuado, etc

Tarea 1 Chequeo de links muertos. Implementaciones DSpace

DSpace provee 2 implementaciones básicas para verificación de enlaces en los metadatos:

1. *BasicLinkChecker* : chequea y genera un reporte para cada metadato cuyo calificador es “uri” (i.e. dc.rights.uri).

1. *MetadataValueLinkChecker:* selecciona todos los metadatos de un ítem y analiza para cada uno si su contenido comienza con “http://” o “https://”. En caso afirmativo, chequea el enlace y reporta el resultado de la prueba.

Tarea 1 Chequeo de links muertos. Implementaciones DSpace (2)

Ambas tareas presentan los mismos inconvenientes. No permiten:

- configurar el metadato que debe validarse
  - → No queremos chequear dc.rights.uri
  - → Sí queremos chequear dc.relation.isPartOf
- Definir timeout de conexión
- Cache de respuestas
  - → El 50% de los ítems en SEDICI tiene una de las 6 licencias CC con las mismas URL. En promedio cada URL de licencia CC se chequea unas 2500 veces. Ej. http://creativecommons.org/licenses/by-nc/3.0/
- Redirecciones Web (i.e. STATUS 3xx)

Estos problemas hacen que la ejecución de la tarea sea extremadamente lenta, ineficiente y que el reporte sea inexacto, dado que pueden haber URL reportadas como fallidas que no lo están.

Tarea 1 Chequeo de links muertos. Propuesta

Tarea de Curation que permita

- indicar cuáles metadatos deben validarse
- manejar **redirecciones** 301 (permanentes), 302 (temporales) y 303 (*see other*, las que usa handle.net)
- permita indicar un *timeout* de conexión máximo
- mantenga un historial de URL chequeadas durante la ejecución actual (como una cache muy simple)

Tarea 2 Metadatos conectados con Autoridades. Modelo Dspace

- **DSpace** no brinda soporte para gestión de autoridades
  - Se asume que estos datos están disponibles pero que se gestionan por fuera de la aplicación.
- Es posible conectar/vincular metadatos y autoridades a través de extensiones denominadas *ChoiceAuthority*
  - recuperan datos desde servicios complementarios para luego guardarlos en un metadato.
- Los datos provistos pueden ser:
  - internos. Ej: Términos de un vocabulario interno (XML Controlled Vocabularies)
  - externos al software. Ej: Materias de un sistema de clasificación
  - externos al repositorio. Ej: Autores de un sistema institucional.

Tarea 2 Metadatos conectados con Autoridades. Representación

- En el entorno de **DSpace**, los autoridades sólo “existen” en los metadatos que las referencian.
  - Cada metadato puede mantener un vínculo con una autoridad a partir del guardado de su clave y texto representativo.

Por ejemplo

dc.contributor.author=(“156442”, “Tim Berners-Lee”)

  - el metadato “dc.contributor.author” referencia al autor Tim Berners-Lee existente en la base de Autores de la biblioteca.

Tarea 2 Metadatos conectados con Autoridades. Problema

El vínculo es débil ya que no cumple con los principios de integridad referencial. Si la autoridad es modificada en el sistema de gestión de autoridades externo, el vínculo no se actualiza ya que Dspace no se entera y los datos quedan:

1. *descoordinados*: se referencia a una autoridad que ha cambiado su nombre pero en el repositorio se tiene el nombre viejo, o aún peor,
2. *con referencias colgantes*: se apunta a un autoridad que no existe más

Tarea 2 Metadatos conectados con Autoridades. Propuesta

Se propone la creación de una curation task que

1. se ejecute periódicamente,
2. verifique la existencia de las autoridades apuntadas desde los metadatos en el *ChoiceAuthority*
    - permite detectar referencias colgantes a autoridades eliminadas
    - podría desconectar el metadato y dejar solo el texto
3. contraste el texto de los metadatos controlados con el valor retornado por el *ChoiceAuthority*
    - detecta modificaciones en los textos
    - podría corregir el dato local, copiando el nuevo
4. genere un reporte con las discrepancias y acciones tomadas

Tarea 3 Accesibilidad de objetos digitales. Ubicación

Según las políticas de contenidos, **algunos objetos digitales pueden estar alojados fuera del repositorio.**

- Los recursos externos suelen referenciarse a partir de una URL o a partir de identificador persistente
- Algunas razones comunes son:
  - falta de derechos sobre la obra,
  - incapacidad de gestionar los datos. (ej. por tamaño excesivo).

Cualquiera sea el caso, el repositorio siempre debe dar acceso al objeto digital, ya sea directa o indirectamente.

Tarea 3 Accesibilidad de objetos digitales. Propuesta

Corrobora para cada ítem que al menos un objeto digital sea accesible a partir de al menos:

- uno o más bitstreams públicamente accesibles, o
- un metadato (configurable) cuyo contenido sea
  - una **URL** al archivo alojado en un servidor externo. Por ejemplo en "*dc.identifier.uri*", o
  - un **identificador persistente** (diferente al del ítem actual). Por ejemplo: a partir de hdl: 1822/24377→ generamos http://hdl.handle.net/1822/24377

El resultado de la ejecución es un reporte con los ítems que no cumplen las restricciones definidas de archivo o de enlace.

Tarea 4 Metadatos obligatorios

La obligatoriedad de un metadato está definida en base a:

1. **requisitos globales**, para cualquier clase de ítem. Ej: fecha, título, autor

1. **requisitos según la tipología documental**. Ej: ISBN para libros, ISSN para revistas.

1. **otros criterios** institucionales como licencias de uso, fuente de financiación, reglamentación local, etc.

Tarea 4 Metadatos obligatorios en Dspace

- Configuración única: input-forms.xml
  - especifica el procedimiento de carga de documentos (submission process)
  - rige la tarea de curation *RequiredMetadata*
    - revisa los documentos y
    - genera un reporte con todos los metadatos obligatorios globales ausentes.

Tarea 4 Metadatos obligatorios.
Propuesta

- Desde la versión 3, es posible indicar campos opcionales y obligatorios según el tipo de documento
  - se considera solo para la carga

Solución simple

- Extender la curation task de **Dspace** para que:
  - valide los metadatos globales obligatorios
  - según el tipo del documento (dc.type) y de acuerdo a la configuración, valide los metadatos necesarios

Tarea 5 Validación del dominio de metadatos. Contexto

- Un metadato posee un dominio que determina
  - cuáles son los valores que puede tomar o
  - al menos qué sintaxis debe respetar según el tipo de dato.

Ejemplos

- dc.contributor.author debe ser un texto
- dc.date debe ser una fecha en formato ISO8601
- dc.language debe ser un ISO 3166 (en, es, pt, etc)

El control de dominio debe realizarse:

1. inicialmente en la carga
2. periódicamente sobre los recursos existentes

Tarea 5 Validación del dominio de metadatos. ¿Qué hace DSpace?

- No permite asociar un dominio a un metadato
  - no brinda un mecanismo para validar su contenido.
- Define algunos controles de carga que “ayudan” a que los datos sean correctos:
  - fecha, caja de texto, vocabulario controlado
- Problemas:
  - los controles de carga poco estrictos representan un riesgo para la calidad de los nuevos registros.
  - Si existen metadatos con valores inválidos el sistema no lo nota

Tarea 5 Validación del dominio de metadatos. Propuesta

- Tarea de curation corrobore que el valor de ciertos metadatos respeta el dominio correspondiente.
- Alternativas para definir las reglas:
  a. la tarea de curation,
  b. el registro de metadatos de DSpace,
  c. archivo de configuración de carga (input-forms.xml)
- La última opción es la elegida:
  a. es relativamente simple,
  b. no invasivo, dado que no modifica el modelo
  c. permite que se use desde el resto del sistema,
  d. provee mayor flexibilidad de configuración

Tarea 5 Validación de metadatos según su dominio. Casos a soportar

- Tipos básicos: boolean, fechas, números, textos de una línea, multilínea.
- Enumerativos (value-pairs): se controla que sea uno de los valores permitidos
- Vocabularios controlados: se puede reusar la tarea 2.
- Potencial para tipos de datos más avanzados: URL, HTML, geolocation, doi, hdl, LaTeX, etc.

Ejemplo

```
<field>
    <dc-schema>dc</dc-schema>
   <dc-element>identifier</dc-
element>
    <dc-qualifier>uri</dc-qualifier>
    <label>Alternate URI</label>
    <domain>URI</domain>
</field>
```

Tarea 6 Extracción de metadatos de preservación

Para asegurar la usabilidad de los recursos digitales en el tiempo, el repositorio debe:

1. garantizar su accesibilidad
    - es decir, que los archivos no se pierdan,
2. evitar su obsolescencia
    - es decir, verificar los formatos de los archivos y transformarlos en caso de ser necesario.

El análisis de formatos de recursos requiere:

- la información de los recursos ,
- los objetos digitales (los archivos) y
- el contexto necesario para acceder a cada objeto (software, versión, etc)

Tarea 6 Extracción de metadatos de preservación

- La información asequible a partir de los archivos dependerá en gran medida de los formatos, pero en general incluye:
  - plataforma,
  - datos del software con el que se generó el archivo (nombre, versiones, etc),
  - fechas.

Ejemplo: un archivo PDF mantiene al menos tamaño físico de las páginas, software, y versión con el que se creó el archivo fuente y el PDF, fecha, autor y datos de la PC usada.

Dspace registra de cada bitstream/archivo sólo su formato, representado por su tipo MIME

Tarea 6 Extracción de metadatos de preservación. Propuesta

- Desarrollo de una tarea de curation que permita

1) realizar la extracción de datos de los archivos
2) almacenar los metadatos de preservación asociados a cada bitstream.

- **Parte 1**: Extracción de datos de los archivos.
  - Hay muchas aplicaciones y librerías estables que lo hacen.
  - Ej: Apache Tika soporta
    - imágenes (JPEG, PNG, etc),
    - audio y video (MPEG, AVI, etc),
    - MS Office (DOC, XLS, etc),
    - y muchos más.

Tarea 6 Extracción de metadatos de preservación. Propuesta (2)

- **Parte 2:** Guardado de metadatos de preservación
  - **DSpace** no permite metadatos en bitstreams.
  - Opción 1: extender DSpace para que los soporte
    - Muy complejo
    - seguramente será agregado pronto por Dspace
  - Opción 2: crear un bitstream que almacene los datos en un Bundle de preservación oculto:
    - Es menos complejo
    - No es invasivo

Tarea 7 Generación de reportes en base a expresiones

- Los administradores precisan detectar y analizar casos complejos en los ítems y sus metadatos. Por ejemplo:
  - registros con pocos metadatos
  - registros con metadatos faltantes (no obligatorios)
  - registros con múltiples valores de un mismo metadato

- El módulo de búsqueda de **Dspace** sólo considera metadatos públicos, es decir, no es posible
  - buscar por metadatos administrativos u ocultos
  - acceder a información de los recursos que no está en los metadato. Ej: última fecha de modificación.

Tarea 7 Generación de reportes en base a expresiones. Propuesta

Tarea de curation para selección y reporte de objetos DSpace en base a expresiones lógicas simples.

- Las expresiones permiten analizar y comparar los datos asociados a cada objeto:
  - ítem → datos del ítem, bundles, bitstreams y metadatos
  - colección → datos de la colección y comunidad padre
  - comunidad → datos de la comunidad, comunidades padre e hijas, colecciones hijas.
- Procedimiento: para cada objeto analizado (un ítem, una Colección o una Comunidad):

1. se almacena el objeto en un espacio común “ValueStack”
2. se evalúa la expresión lógica sobre el ValueStack
3. segun el caso, se incluye el objeto en el reporte final o no.

Tarea 7 Generación de reportes en base a expresiones. Propuesta

- las expresiones usan notación puntual para acceder al ValueStack y deben evaluar a verdadero o falso
- el ValueStack brinda acceso a los parámetros de configuración y a los datos del objeto (según el tipo)
- Alternativas de implementación:
  - expresiones algebraicas tradicionales
  - Criteria
- **Ejemplos**
  - cantidad de bitstreams en el Bundle ORIGINAL de un ítem
    - ítem.bundle('ORIGINAL').bitstream().count() > 0
  - cantidad de metadatos dc.title de un ítem
    - ítem.metadata('dc.title').count() > 0
  - metadatos dependientes
    - if(ítem.metadata('metadato1').count() > 0, ítem.metadata('metadato2').count() > 0,true)

Estrategia de selección de ítems. DSpace

- Dspace permite aplicar una tarea sobre
  - un ítem,
  - una colección y su contenido,
  - una comunidad y su contenido, o
  - todo el contenido del repositorio.
- Para aplicar otros criterios hay que agregar la lógica de inclusión/exclusión de ítems a la tarea en sí.
    - se dificulta la programación de la tarea
    - se dificulta el reuso de tareas.

Estrategia de selección de ítems. DSpace. Propuesta

- Extender el módulo de curation de DSpace para
  - permitir la aplicación de una tarea sobre un subconjunto de ítems del repositorio.

- Los criterios de selección serían expresiones lógicas simples (como las planteadas para la tarea 7).
  - Ejemplo:
    - invocación sobre todos los artículos
      - “ítem.metadata(‘dc.type’) = ‘Article’”
    - otros criterios podrían ser
      - fecha de carga,
      - fecha de última modificación,
      - etc.

Características comunes de las tareas de control de datos

1. realizan controles de **sólo lectura,** o al menos idempotentes

2. su ejecución es **independiente de la ejecución de la misma tarea sobre otros ítems**
   a. **no dependen del orden de ejecución**
   b. pueden ser **paralelizadas**

3. deben estar optimizadas para reducir su impacto en el sistema
   a. dado que se aplican regularmente sobre todo el repositorio.

Estrategia de aplicación de tareas en DSpace

- el orden de aplicación de una tarea sobre los ítems considerados es secuencial y síncrono.
  - al crecer el volumen del repositorio, aumenta el tiempo de procesamiento de las tareas
  - una tarea lenta y frecuente, podría superponerse con la siguiente ejecución.

- el orden de ejecución entre tareas es secuencial y síncrono:
  - si varias tareas lanzadas en simultáneo, se completa la aplicación de cada tarea sobre todo el conjunto de ítems, antes de pasar a la siguiente.
  - Genera overhead: cada ítem se levanta por cada tarea

Estrategia de aplicación de tareas. Extensiones propuestas

- Ejecución de tareas por bloques de ítems,
  - para reducir el impacto en la performance general del sistema.
  - el consumo de recursos es por ráfagas

- Ejecución de tareas en paralelo
  - para reducir el tiempo total de ejecución de tareas lentas como Control de links muertos.

- Ejecución de tareas compuestas
  - para permitir controles complejos sobre los ítems.
  - Por ejemplo: tarea estadísticas que indique el “riesgo” de los recursos en base al % de registros incompletos o incorrectos, según la salida de varias tareas de control.

Trabajos Futuros

Las tareas y estrategias planteadas representan solo una pequeña muestra de las acciones de control/monitoreo de datos que puede desarrollar un repositorio para tratar de preservar sus recursos a largo plazo.

Otras tareas no consideradas

- detección de duplicados
- relación de ítems (obras similares, etc)
- diagnóstico de formatos internos en base a servicios externos, como PRONOM

¡Muchas gracias!

Marisa De Giusti

marisa.degiusti@sedici.unlp.edu.ar

Ariel Lira

alira@sedici.unlp.edu.ar

Nestor Oviedo

oviedonestor@gmail.com

Gonzalo Villarreal

gonzalo@sedici.unlp.edu.ar

Proyecto de Enlace de Bibliotecas
prebi.unlp.edu.ar

SEDICI
REPOSITORIO INSTITUCIONAL DE LA UNLP

UNIVERSIDAD NACIONAL DE LA PLATA
www.unlp.edu.ar