Deterministic assignment of PIDs: why and how? The SLDR experience

Bernard Bel, Arnaud Gingold <contact@sldr.org>
Laboratoire Parole et Langage (LPL)
CNRS – Université Aix-Marseille

DASISH workshop on Persistent Identifiers - Services and Policies, Cologne (Germany), 8-9 December 2014

ORTOLANG is a beneficiary of French State support via its "Investissements d'avenir" programme (ANR-11-EQPX-0032).

Background

- A site called **Speech & Language Data Repository** (SLDR, http://sldr.org) has been developed by *Laboratoire Parole et Langage* (LPL), a speech research laboratory of the French *Centre National de la Recherche Scientifique* (CNRS).
- The aim of SLDR is to preserve (field or laboratory) data eligible for linguistic research and facilitate its **non-commercial** dissemination. This means producing scientific data easy to maintain and to find in the evolving context od Digital Humanities. SLDR is open to research projects worldwide.
- SLDR (for speech) and CNRTL (for text) are the resources centres from which a French sub-network of CLARIN centres is being built: the ORTOLANG project (Open Resources and TOols for LANGuage) which is also involved in DARIAH-EU.
- SLDR/ORTOLANG is a **generic service** able to deal with any type of item (*information package*). ORTOLANG will further offer tools and functionalities to process the data.
- Resource pooling is constructed on an interoperable system (OAIS, **Open Archival Information System**) currently involving major computing centres: INIST (Nancy) for work space/data dissemination, and CINES (Montpellier) for its long-term preservation.

PIDs on SLDR/ORTOLANG: basic points

- A PID (**Persistent IDentifier**) is a unique string of alphanumeric symbols characterizing a digital resource.
- Resources on SLDR/ORTOLANG include:
  - Single files
  - Items (files grouped in a tree structure)
- A file keeps the same PID if (1) its **name** remains unchanged and (2) its **content** remains unchanged as verified by its digital signature (MD5). The location of the file in the item hierarchy has no influence on its PID.
- The PID of an item is bound to its version:
  - Versions being '**snapshots**' of items, their contents are immovable.
  - The 'source' of an item, i.e. its **workspace**, has a variable content.
  - Repository designers may decide whether the 'plain' PID on an item should target its source or its latest published version. In SLDR, a plain PID targets the highest version, and version-labelled PIDs target earlier versions.
- A PID can be made actionable by 'URL-ifying' it (cf. next slide). The result is a **permanent link**. Default content negotiation of a permanent link is:
  - Download a file – alternatively open it in a web browser;
  - Display the descriptive page of an item.

Actionable PIDs = cool URLs

- A **permanent link** is the association of a PID and the address of a proper resolver.
- SLDR/ORTOLANG creates PIDs based on the **Handle system** http://handle.net. These PIDs are strings starting with “11041”, the **unique identifier of SLDR** in the Handle domain. Therefore, it’s **up to the repository manager** to make sure that the rest of the string is unique.
- PIDs on SLDR may be imbedded in permanent links using either a **local** or **global** resolver.
- Examples of **local-resolver** permanent links:
  (We also run an ARK local resolver for archived items.)
  - http://sldr.org/hdl:11041/swedia-000788 (item)
  - http://sldr.org/hdl:11041/sldr000033/M01.TextGrid (file)
  - http://sldr.org/hdl:4263537/5041 (some external resource)
- Examples of **global-resolver** permanent links:
  - http://hdl.handle.net/11041/swedia-000788 (item)
  - http://hdl.handle.net/11041/sldr000033/M01.TextGrid (file)
  - http://hdl.handle.net/4263537/5041 (some external resource)

Random or deterministic?

- Evidently, permanent links based on PIDs should **never depend on the path** to a document in its file-system hierarchy.
- To dissociate PIDs and the physical locations of target files, engineers often assign universally unique identifiers (UUIDs) to files. This makes PIDs look like **random strings** (i.e. 'opaque') for the sake of discarding any 'semantic' information about their targets, including file names.
- A random-looking string is one which the depositor of data is not able to anticipate. This may be problematic for 3 reasons:
    1. A UUID is **unpractical to quote** in a paper publication.
       Try this: https://docs.google.com/document/d/1eVH35sFHauSn1gbeHE19ffk8zqVw5XRJpbprT1ArrP8/edit#
       or even this: http://tinyurl.com/n29wc9j
    2. In some cases, producers need to know **in advance** the permanent links that will be assigned to resources in the item, so that they put them into **documentation** or **metadata**.
    3. Documentation (HTML pages, TEI documents…) need to be edited **before its submission to the repository** => Occurrences of **absolute links** to resources need to be replaced with permanent links. **Simple string replacement** should be possible, based on simple rewrite rules using anticipated PIDs.
- When these constraints are not taken into consideration, producers and users of archived material might **not quote PIDs** in their publications. Worse, links in item documentation might point at locations which have become **obsolete** after sending the archival version to the dissemination site.

PID assignment on SLDR/ORTOLANG

- PID assignment on SLDR/ORTOLANG is based on the concatenation of identifiers **all (but one) predictable**. The unpredictable identifier is the index of the item/project in the database of the repository.
- For instance, hdl:11041/alipe-000853 is the concatenation of:
  - "hdl" = the PID type
  - "11041" = identifier of the SLDR/ORTOLANG repository in the Handle domain
  - "alipe-" = a prefix chosen by the data producer
  - "000853" = the unique index of this item in the repository (**unpredictable**)
- Item versioning introduces one more identifier. For instance, hdl:11041/sldr000018_v2 points at version 2 of item sldr000018.
- Once the descriptive page of the item has been created, its index and prefix are known. Every PID targetting a file contained in this item becomes **predictable**. Therefore, the process of PID assignment is **deterministic**.
- A PID pointing at a file is constructed by concatenating the PID of its container and a **unique identifier** of this file **within its container** => no need for a universally unique identifier!

Deterministic construction of PID for a file

- The file PID string starts with the PID of the container of the file (item). It is followed by an identifier of the file within its container.
- Any **separator** (other than space) may be used between the two parts. We suggest '/' to indicate the container relation. (This should not be interpreted as a path!)
- The **file name** is a good candidate as an identifier when all files bearing the same name **in a given item** have identical digital signatures.
- SLDR/ORTOLANG offers an option for hiding some extensions.
- Two options are possible if several files with different digital signatures bear the same name in an item:
  1. Append **discriminative identifier(s)** before the file name;
  2. Insert a **self-incremented index** in the file name.
- These options should be set up before creating PIDs and **clear enough** for the producer to be able to **guess the PID of any given file**.
- When option 1 fails (identifier is not discriminative) the system automatically resorts to inserting/incrementing an index.

Discriminative identifiers

- Discriminative identifiers should be **chosen by producers** and assigned by an an **automatic process**.
- Producers generally wish to pick up words contained in the tree structure of the item, e.g. ‘video14’ for the 14th set of video files or ‘session3’ for recordings in the third session… Words may be discarded if repeated.
- Options depend on interface design. In SLDR the producer may chose the ‘n’ first or ‘n’ last words contained in the file path. (Preferably, n = 1)
- Example 1: http://sldr.org/sldr000002/toc. Files named '10681131.wav' are not identical because they contain distinct tracks of the same recording. Discriminative identifiers contain channels numbers:
  - hdl:11041/sldr000002/channel1_10681131.wav
  - hdl:11041/sldr000002/channel2_10681131.wav
- Exemple 2: http://sldr.org/alipe-000853/toc. The first word of the path is used:
  - hdl:11041/alipe-000853/ali-baptiste-global_contenuArchive => extension 'html' has been discarded.
  - hdl:11041/alipe-000853/ali-baptiste-global.xml => word 'ali-baptiste-global' is discarded to avoid redundancy, as it is already part of the file name.

Self-incremented index in file names

- Look at hdl:11041/ortolang-000900?urlappend=/toc. Files named “mix.mp3” have different contents as they belong to recordings done in sessions 1, 2, 3, 4.
- It makes sense to set-up the PID assignment algorithm to insert indexes which (due to alphabetic sorting) will match these session numbers. This yields:
  - hdl:11041/ortolang-000900/mix_1.mp3
  - hdl:11041/ortolang-000900/mix_2.mp3
  - hdl:11041/ortolang-000900/mix_3.mp3
  - hdl:11041/ortolang-000900/mix_4.mp3
- The algorithm has an option for discarding this index when its value is ’1’. This prevents it from indexing all files by default.

Batch-processing files targetted by PIDs

- External software devices performing batch processes on the content of a repository may need to **guess the PID of a file** using local path/name information contained in an index used for the batch.
- For instance, hdl:11041/sldr000773/Pilote-carambouille_settings.txt is the PID on an index containing the names of files 'TRACK0_0.wav', … 'TRACK3_0.wav' belonging to the same directory:

```
file           speaker    volume      panoramic
TRACK0_0.wav   Laurent    100         R100
TRACK1_0.wav   Marion     100         L33
TRACK2_0.wav   Thomas     100         R33
TRACK3_0.wav   Julie      100         L100
```

- A deterministic assignment of PIDs makes it possible to **infer PIDs targetting these files**, thereby allowing their batch processing:
  - hdl:11041/sldr000773/Pilote-carambouille_TRACK0_0.wav
  - …
  - hdl:11041/sldr000773/Pilote-carambouille_TRACK3_0.wav

Replacing URLs with permanent links

- The following is a basic principle for **easily and safely** inserting permanent links or PIDs in documentation and metadata.
- Every occurrence of **absolute link to a resource** needs to be replaced with a **permanent link**. In order to achieve this, **simple string replacement** should be possible, based on syntactic rules using **anticipated PIDs**.
- For instance, in the ALIPE corpus, former link
  - http://lrl-diffusion.univ-bpclermont.fr/corpusAlipe/alipe-res/ali-prune-071122.mp3
- has been replaced with permanent link
  - http://hdl.handle.net/11041/alipe-000853/ali-prune-071122.mp3
- applying the **unique rewrite rule**:
  - lrl-diffusion.univ-bpclermont.fr/corpusAlipe/alipe-res
    --> hdl.handle.net/11041/alipe-000853

More on content negotiation

- Instructions or additional links may be appended after a permanent link to **negotiate the content** of the item in its target. By default, clicking the permanent link of an item displays its descriptive page on SLDR/ORTOLANG. Example: http://hdl.handle.net/11041/swedia-000788
- Retrieving structured **descriptive metadata**:
  - http://hdl.handle.net/11041/swedia-000788/oai_dc.xml
  - http://hdl.handle.net/11041/swedia-000788/olac.xml
  - http://hdl.handle.net/11041/swedia-000788/rdf.html
  - http://hdl.handle.net/11041/swedia-000788/cmdi.xml
- **Downloading** an item:
  - http://hdl.handle.net/11041/sldr000025?urlappend=/download
    (open-access downloading)
  - http://hdl.handle.net/11041/swedia-000788?urlappend=/download
    (authenticated downloading)
- Permanent links to **files** do not negotiate contents:
  - By default, a permanent link downloads the file;
  - If the file is located in the “preview” folder, clicking the link attempts to open it in a web browser.

Metadata in Handles

Metadata imbedded in Handle records are the 'DESC' fields written by the Handle creation script. For example, '10 DESC' contains the name of the item which the document belongs to. Calling:

http://hdl.handle.net/11041/sldr000033/M01.TextGrid?index=10

will yield: "Primary data (corpus): Aix-MARSEC database." whereas http://hdl.handle.net/11041/sldr000033/M01.TextGrid?noredirect will display all 'DESC' values in the Handle record created by the script:

```
100  HS_ADMIN 2014-12-01 14:44:40 handle=0.NA/11041; index=300; [create
hdl,delete hdl,read val,modify val,del val,add val,modify admin,del admin,add
admin]
7    EMAIL  2014-12-01 14:44:40Z   webmaster@sldr.org
8    URL    2014-12-01
14:44:40Z   http://sldr.org/sldr000033_v2/get/fileid:2153
9    DESC   2014-12-01 14:44:40Z   ORTOLANG www.ortolang.fr (via SLDR)
10   DESC   2014-12-01 14:44:40Z   Primary data (corpus): Aix-MARSEC database
11   DESC   2014-12-01 14:44:40Z   Données primaires (corpus) : Aix-MARSEC
base de données
12   DESC   2014-12-01 14:44:40Z   file §doc/TextGrids/M/M01.TextGrid
13   DESC   2014-12-01 14:44:40Z   Content-Type: text/plain; charset=utf-8
14   DESC   2014-12-01 14:44:40Z   Content-Length: 103838
```