A Deep Validation Process for Open Document Repositories

Wolfram Horstmann[1], Maurice Vanderfeesten[2], Elena Nicolaki[3], Natalia Manola[3]

[1] Bielefeld University, P.O. Box 10 01 31, 33501 Bielefeld, Germany;
e-mail: wolfram.horstmann@uni-bielefeld.de
[2] SURF Foundation, P.O. Box 2290, 3500 GG UTRECHT, The Netherlands;
e-mail: Vanderfeesten@surf.nl
[3] National & Kapodistrian University of Athens, Panepistimioupolis-Ilissia, Athens 15784, Greece;
e-mail: {natalia; enikol}@di.uoa.gr

Abstract

Institutional document repositories show a systematic growth as well as a sustainable deployment. Therefore, they represent the current backbone of a distributed repository infrastructure. Many developments for electronic publishing through digital repositories are heading in the direction of innovative value-added services such as citation analysis or annotation systems. A rich service-layer based on machine-to-machine communication between value-added services and document repositories requires a reliable operation and data management within each repository. Aggregating search services such as OAISTER and BASE provide good results. But in order to provide good quality they also have to overcome heterogeneity by normalizing many of the data they receive and build specific profiles for sometimes even one individual repository. Since much of the normalization is done at the side of the service provider, it often remains unclear — maybe sometimes also to the manager of a local repository — exactly which data and services are exposed to the public. Here, an exploratory validation method for testing specification compliance in repositories is presented.

Keywords: OAI-PMH; Validation; Harvesting; Institutional Repository; Open Access

1. Introduction

Many developments for electronic publishing through digital repositories are heading in the direction of innovative value-added services (e.g. citation analysis or annotation systems) and abstract data representation of all types of resources (e.g. primary data or educational material). Still, conventional document repositories show the most systematic growth [1] as well as high-quality, sustainable deployment. Therewith, they represent the current backbone of a distributed repository infrastructure. A rich service-layer based on machine-to-machine communication between value-added services and document repositories requires a reliable operation and data management within each repository. Aggregating search services such as OAISTER [2] and BASE [3] provide good results. But in order to provide good quality they also have to normalize many of the data they receive and build specific profiles and processes for even individual repositories. Since much of the normalization is done at the side of the service provider, it often remains unclear — maybe sometimes also to the manager of a repository — exactly which data and services are exposed to the public by a local data provider. Existing validation techniques [4, 5] are important but their current scope is only at the level of testing basic compliance with OAI-PMH and simple-DC. As a consequence, quantitative assessments of data quality, i.e. of what is specifically exposed by a repository and by the whole repository landscape are widely missing. Mature infrastructures should, however, provide reliable data resources, robust standards, corresponding validation mechanisms for them and a systematic change-request cycle. Here, an exploratory validation method for testing specification compliance in repositories is presented. It produces specific and quantitative data on the quality of technical and content-related behaviour of a repository and shall be further developed as a tool that can be applied

by individual repository managers for monitoring the quality of their repository.

2. Methods

Basis for the validation are the DRIVER [6] Guidelines for Content Provides: "Exposing textual resources with OAI-PMH" [7]. They assume the usage of OAI-PMH [8] and are strongly influenced by guidelines for using simple Dublin Core of Powell et al. [9] DC-Best-Practice [10], the DINI certificate for document and publication repositories [11] and experiences from DAREnet [12]. Software has been developed by the National Kapodistrian University of Athens that is designed for repository managers or 'curators' of repository networks. It runs automated tests for three aspects: (i) general compliance with OAI-PMH, (ii) compliance with DRIVER-specific recommendations for OAI-PMH implementation and (iii) the metadata compliance according to the DRIVER guidelines. Aspect (i) tests the validity of XML according to the OAI-PMH schema in a variety of use patterns to discover flaws in expected behaviour. Aspect (ii) tests several strategies to be within the boundaries of the DRIVER guidelines, such as deleting strategy, batch size or the expiration time of a resumption token. Aspect (iii) looks into the record and tests how the simple Dublin Core fields are used compared to the recommendations in the DRIVER guidelines.

Technically, the DRIVER validator is a web application (Servlet/JSP based). Its main characteristics are:

- it is a rule based system allowing end users to customize the type of validation they want to perform through the selection of specific predefined rules
- end users are able to validate multiple sets/repository or multiple repositories in a batch mode, also having the option of sampling for a "quick-look" operation
- it produces comprehensive, full, on-line reports of the results with direct links to the original repository records
- it provides registration means and persistent storage of the results so that end users may view the history of the changes performed in their repositories
- provides mechanisms for scheduled or periodic validation runs
- extendable so that the predefined types of rules may be configured to run on additional fields/attributes of the data

Moreover, the internal software architecture is built into components, both local and remote Web Services, so that the system (i) is fully distributed, in order to accept and process simultaneous requests, (ii) may be used -some of its core components at least- by other services (e.g. DRIVER Aggregator), and (iii) allows for extension to protocols beyond OAI-PMH.

It is built with the Apache Struts 1.3.5 web development framework and is deployed and tested in the Apache Tomcat 5.0.28 Web Application server. It is currently deployed as a beta version on a temporary test environment [13] .

3. Results

In a sample of 61 repositories from 5 countries (Belgium, France, Germany, Netherlands, UK) test validation was performed for a set of 300 records each. Results indicate that no repository fully complies with the guidelines but many are near. The overall picture is heterogeneous: (i) for general OAI-PMH validation results show that only few repositories in the sample are-non harvestable. On the other hand only a few repositories deliver 100% valid XML. For specific DRIVER characteristics with respect to OAI-PMH (ii) and simple Dublin Core (iii), some systematic behaviours resulting from national conventions or platform

conventions such as ePrints [14], DSpace [15] and OPUS [16] are observable but many variations relating to data entry of single records also contributed to the error distribution.

4. Discussion

The validation method is functional. Further improvements lay in removing points of confusion by (i) changing some of the functions of the validation rules in a way that better fits the guideline explanation. And (ii) change descriptions in the DRIVER Guidelines that allow multiple interpretation of the validation rules. Currently new DRIVER guidelines are under development.. Advanced test routines are currently missing, for example for providing critical performance indicators, such as the record throughput that measures the number of records that are delivered per second. Quantitative values shall be used to visualise performance and raise awareness for repository managers. Communication between DRIVER helpdesk and the Mentor service [17] with repository managers based on these reports will be analysed to design the next development phase: establish a change-request cycle, define the validity threshold and automate the validation process.

5. References

[1] http://www.opendoar.org/
[2] http://www.oaister.org/
[3] http://www.base-search.net/
[4] http://www.openarchives.org/data/registerasprovider.html
[5] http://roar.eprints.org/
[6] http://www.driver-community.eu
[7] http://www.driver-support.eu/en/guidelines.html
[8] http://www.openarchives.org/pmh/
[9] http://eprints-uk.rdn.ac.uk/project/docs/simpledc-guidelines/
[10] http://oai-best.comm.nsdl.org/cgi-bin/wiki.pl
[11] http://www.dini.de/
[12] http://www.darenet.nl/en
[13] http://validator.driver.research-infrastructures.eu
[14] http://www.eprints.org/
[15] http://www.dspace.org/
[16] http://opusdev.bsz-bw.de/trac
[17] http://www.driver-support.eu/mentor.html